Coduri utilizate
în sistemele de transmisiuni cu
spectru împrăştiat

Referat nr.1

Conducător ştiinţific: Prof. dr. ing. Miranda Naforniță
Doctorand: as.ing. Horia Balta

Cuprins

1. Sisteme de transmisiuni cu spectru împrăştiat

1.1 Principiul împrăştierii spectrului. Modelul unui sistem de comunicaţie cu spectru împrăştiat

Împrăştierea spectrului este o tehnică de *modulaţie* a semnalului purtător de informaţie. Această tehnică reprezintă o soluţie calitativ superioară în multe aplicaţii ce implică comunicaţii prin medii cu acces multiplu, în special canalele radio.

Spre deosebire de tehnicile clasice de modulaţie, împrăştierea spectrului aduce câteva elemente noi. Primul este că semnalele cu spectru împrăştiat utilizate pentru transmiterea informaţiei digitale se disting prin aceea că lăţimea lor de bandă W [Hz] este mult mai mare decât rata informaţiei R [biţi per secundă]. Astfel factorul expansiunii de bandă [2]

$$\beta_e = W/R \qquad (1.1)$$

pentru un semnal cu spectru împrăştiat este mult mai mare decât unitatea. Redundanţa mare, inerentă în semnale cu spectru împrăştiat, este necesară pentru a putea realiza transmisii la nivelele mari de interferenţă ce sunt întâlnite în unele canale radio sau pe satelit. Codarea este un element important în proiectarea semnalelor cu spectru împrăştiat deoarece formele de undă codate sunt de asemenea caracterizate printr-un factor de expansiune a benzii ce este mult mai mare decât unitatea şi deoarece codarea este o metodă eficientă de introducere a redundanţei.

Al doilea element important implicat în proiectarea semnalelor cu spectru împrăştiat este pseudoaleatorul, care face ca semnalele să apară similar cu zgomotul aleator şi dificil de demodulat de receptoare, altele decât cel dedicat. Acest element este descris împreună cu aplicaţia acestor semnale.

Semnalele cu spectru împrăştiat sunt utilizate cu scopul:

(1) de a combate sau suprima efectele dăunătoare ale interferenţei datorate bruiajului, a interferenţei cu alţi utilizatori ai canalului şi a interferenţei proprii datorate propagării pe căi multiple;
(2) de a ascunde semnalul prin transmiterea sa la o putere joasă şi, astfel, de a face dificilă detectarea sa de către un ascultător neavizat în prezenţa zgomotului de fond;
(3) crearea de mesaje private în prezenţa altor ascultători.
(4) În alte aplicaţii decât comunicaţiile, semnalele cu spectru împrăştiat sunt folosite pentru a obţine un nivel de acurateţe (timp de întârziere) cum şi un nivel al ratei (vitezei) cum măsurătorilor în radar sau navigaţie.

În combaterea bruiajului, este important pentru utilizatori că bruiorul, care încearcă să perturbe comunicaţiile, nu cunoaşte dinainte caracteristicile semnalului, cu excepţia întregii benzi a canalului şi a tipului modulaţiei (PSK, FSK, etc) care sunt folosite. Dacă informaţia digitală este codată „clasic", un bruior sofisticat poate imita uşor semnalul emis de transmiţător şi, aşadar, să deruteze receptorul. Pentru a preveni această posibilitate,

transmiţătorul introduce un element de impredictibilitate sau aleator (pseudoaleator) în fiecare formă de undă a semnalului transmis codat care este cunoscut de receptorul avizat dar nu şi de bruior. Ca o consecinţă, bruiorul trebuie să sintetizeze şi să transmită un semnal de interferenţă fără a cunoaşte forma pseudoaleatoare.

Interferenţa cu alţi utilizatori apare în sistemele de comunicaţie cu acces multiplu în care un număr de utilizatori împart o bandă comună a canalului. La orice moment dat, un subset al acestor utilizatori transmit simultan informaţie prin canalul comun la receptorii corespunzători. Presupunând că toţi utilizatorii folosesc acelaşi cod pentru codare şi decodare pentru secvenţele lor de informaţie, semnalele transmise în acest spectru comun pot fi distinse unul de altul prin impunerea unei forme pseudo-aleatoare[1], de asemenea numită cod, în fiecare semnal transmis. Astfel un receptor particular poate descoperi informaţia transmisă şi adresată lui prin cunoaşterea formei pseudoaleatoare, altfel spus a cheii, utilizate de transmiţătorul corespondent. Acest tip de tehnică de comunicaţie, care permite utilizatorilor multipli să folosească simultan un canal comun pentru transmiterea informaţiei, se numeşte *acces multiplu cu diviziune în cod* (CDMA—Code Division Multiple Access).

Componentele reductibile multicale, rezultate din propagarea dispersivă în timp prin canal, pot fi văzute ca o formă de interferenţă proprie. Acest tip de interferenţă poate fi de asemenea suprimată prin introducerea formei pseudoaleatoare în semnalul transmis, aşa cum va fi descrisă ulterior.

Un mesaj poate fi ascuns în zgomotul de fond prin împrăştierea benzii sale prin codarea şi transmiterea semnalului rezultat la o putere mică. Datorită nivelului scăzut al puterii sale, semnalul transmis se spune că este ”ascuns” (“covert”). Are o probabilitate scăzută de a fi interceptat (detectat) de un ascultător ne-avizat, şi, de aceea, este numit semnal cu probabilitate mică de intercepţie (LPI).

În final, mesajul privat poate fi obţinut prin impunerea unei forme pseudo-aleatoare în mesajul transmis. Mesajul poate fi demodulat de către receptorii avizaţi, ce cunosc forma pseudo-aleatoare sau cheia folosită la transmiţător, dar nu şi de alţi receptori ce nu au cunoştinţă despre cheie.

Schema bloc prezentată în Figura 1.1 ilustrează elementele de bază ale unui sistem de comunicaţie digital cu spectru împrăştiat, cu o secvenţă de informaţie binară la intrarea transmiţătorului şi la ieşirea receptorului. Codorul şi decodorul de canal, cât şi modulatorul şi demodulatorul sunt elementele de bază ale sistemului. În completarea acestor elemente avem două generatoare identice de secvenţe (tipar) pseudoaleatoare, una ce conlucrează cu modulatorul la finele transmisiei iar cea de-a doua ce conlucrează cu demodulatorul la finele recepţiei. Generatoarele generează o secvenţă binară pseudoaleatoare sau pseudonoise (PN) ce este introdusă în semnalul transmis la modulator şi extrasă din semnalul recepţionat la demodulator.

Sincronizarea secvenţei PN generate la receptor cu secvenţa PN conţinută în semnalul recepţionat este necesară pentru demodularea semnalul recepţionat. Iniţial, înainte de transmiterea informaţiei, sincronizarea poate fi obţinută prin transmiterea unui tipar de bit pseudoaleator pe care receptorul îl va recunoaşte, şi în prezenţa interferenţei, cu mare probabilitate. După realizarea sincronizării generatoarelor, poate să înceapă transmisia informaţiei.

Interferenţa este introdusă în transmisie la trecerea semnalului de informaţie prin canal. Caracteristicile semnalului ce interferă depind în mare măsură de originea sa. Interferenţa poate fi clasificată ca fiind fie de bandă largă fie de bandă îngustă relativ la lăţimea de bandă a semnalului purtător de informaţie; de asemenea, semnalul de interferenţă poate fi continuu în timp sau discontinuu (pulsatoriu) în timp. De exemplu, un semnal de

[1] în literatură sunt întâlnite denumirile: cod zgomot sau pseudo-zgomot (PN—pseudo noise), semnătură sau cheie. În lucrarea de faţă vor fi utilizate, după caz, toate aceste denumiri.

bruiaj poate consta dintr-una sau mai multe sinusoide în banda de transmisie utilizată pentru transmiterea informaţiei. Frecvenţele sinusoidelor pot rămâne fixe sau se pot schimba cu timpul. Ca un al doilea exemplu, interferenţa generată în CDMA de alţi utilizatori ai canalului poate fi de asemenea de bandă largă sau îngustă, depinzând de tipul semnalului cu spectru împrăştiat utilizat la realizarea accesului multiplu. Dacă este de bandă largă, poate fi caracterizată ca un zgomot echivalent, gaussian, alb, aditiv.

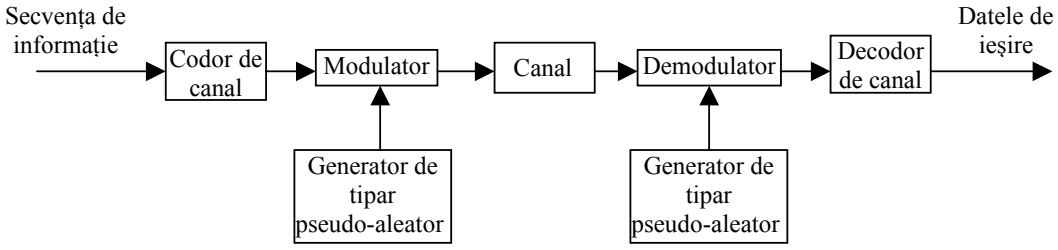


Figura 1.1 Model de sistem digital de comunicaţie cu spectru împrăştiat

Modulaţiile utilizate în sistemele de transmisiuni cu spectru împrăştiat pot fi de tip PSK sau FSK. Modulaţia PSK este adecvată aplicaţiilor unde coerenţa de fază dintre semnalul transmis şi semnalul recepţionat poate fi menţinută într-un interval de timp ce este relativ lung comparativ cu 1/W, unde W reprezintă banda semnalului transmis. Pe de altă parte, modulaţia FSK este potrivită aplicaţiilor în care o astfel de coerenţă de fază nu poate fi menţinută, datorită variaţiei în timp a parametrilor semnalelor din canalele de comunicaţii. Acesta poate fi cazul unei legături de comunicaţie dintre două avioane de mare viteză sau dintre un avion rapid şi un terminal de pe sol.

Secvenţa PN generată la modulator este utilizată împreună cu modulaţia PSK pentru a deplasa faza semnalului PSK pseudo-aleator. Semnalul modulat rezultat este numit semnal cu spectru împrăştiat cu *secvenţă directă* (DS) sau pseudonoise (PN). Utilizată împreună cu FSK binar sau M-ar, secvenţa pseudoaleatoare selectează frecvenţa semnalului transmis în mod pseudoaleator. Semnalul rezultat este denumit semnal cu spectru împrăştiat cu *salt de frecvenţă* (FH).

1.2 Avantajele comunicaţiilor cu spectru împrăştiat. Aplicaţii

În acest paragraf vor fi prezentate câteva dintre aplicaţiile ce utilizează semnale cu spectru împrăştiat. [8]

a)-*Estimarea întârzierii* semnalului reflectat în sistemele radar. Dispersia timpului de întârziere are expresia aproximativă:

$$\sigma_\tau^2 \approx \frac{1}{\xi^2 (2\pi W)^2} = \frac{\tau_c^2}{\xi^2 (2\pi)^2}, \qquad (1.2)$$

unde: ξ^2 = raportul semnal pe zgomot; W = banda semnalului; $\tau_c = 1/W$ = dublul timpul de creştere. Relaţia (1.2) arată că, pentru a avea o bună precizie în estimare, este necesar să se utilizeze un semnal de bandă largă. Dacă durata semnalului, T, este aleasă mică, atunci este necesar (în vederea unui raport semnal pe zgomot ξ^2 mare) un maxim de putere. O alternativă este alegerea lui $T>>\tau_c$, sau echivalent:

$$W\,T >> 1, \qquad (1.3)$$

relaţie satisfăcută prin utilizarea semnalului cu spectru împrăştiat. Raţionamentul şi concluzia sunt identice cu cele prezentate în cazul în care se doreşte o estimare a frecvenţei semnalului.

b)-Obţinerea de ***răspunsuri scurte şi cu RSZ mare*** la ieşirea filtrelor adaptate (FA) sau corelatoarelor [12]. Filtrul adaptat poate separa în timp semnalele de intrare, ce sunt întârziate şi suprapuse unele peste altele şi cu zgomotul de fond, dacă timpul de corelare, τ_c, este mai mic decât distanţa în timp dintre ele, Figura 1.2. Condiţia ca $\tau_c << \Delta T_i$ este realizată prin utilizarea de semnale cu spectru împrăştiat pentru care este valabilă relaţia (1.3).

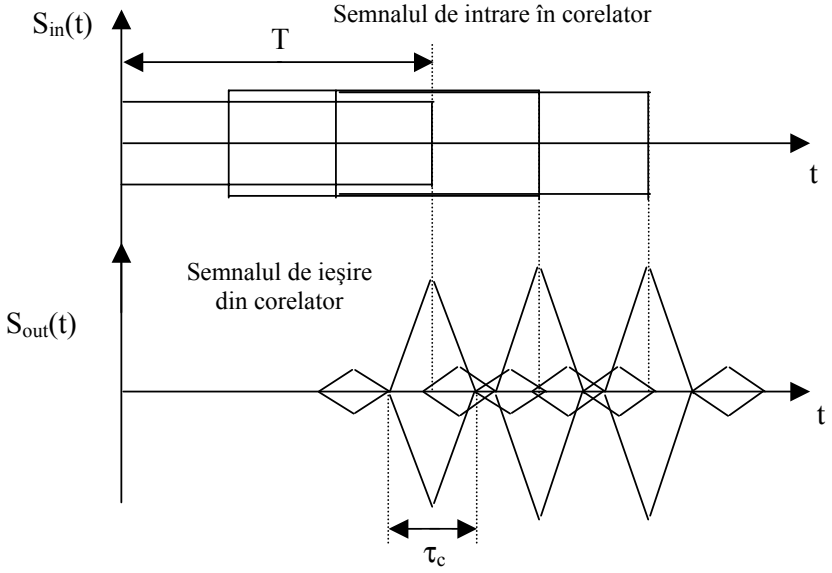


Figura 1.2 Separarea prin corelare a semnalelor cu spectru împrăştiat, suprapuse

c)-***Anti-bruiajul***. Presupunând că semnalul util, având banda de frecvenţă W şi puterea P_s, este perturbat de un semnal de bruiaj cu banda W_j şi puterea P_j (Figura 1.3), diminuarea efectelor de bruiaj se poate face prin:

-rejecţia benzii alterate, caz în care raportul semnal pe zgomot după rejecţie este

$$RSZ_r = RSZ_0\,(1-W_j / W), \qquad (1.4)$$

unde $RSZ_0 = 2P_s / N_0$ este raportul semnal pe zgomot în absenţa bruiajului;

-corelare (filtrare adaptată), caz în care raportul semnal pe zgomot (fără rejecţie) este

$$RSZ_{fa} = P_s / (P_n + P_j) = RSZ_0 / (1+ P_j / WN_0). \qquad (1.5)$$

Ambele relații, (1.4) şi (1.5), indică un câştig în raportul semnal pe zgomot la mărirea benzii, W, obiectiv satisfăcut prin împrăştierea spectrului.

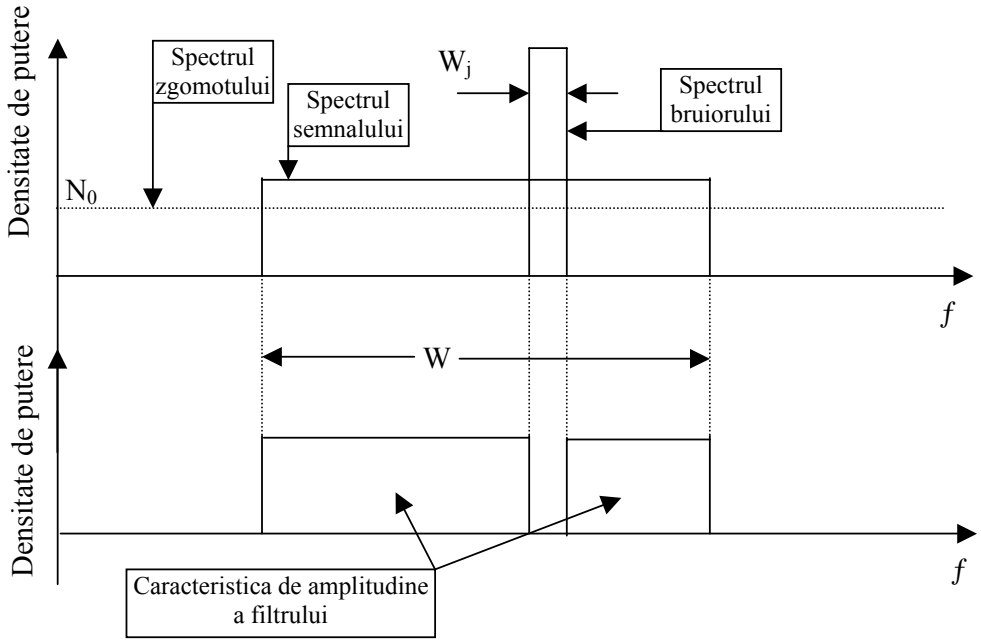


Figura 1.3 Semnal bruiat cu semnal de bruiaj de bandă îngustă

Dacă semnalul perturbator este de bandă largă, atunci singura posibilitate este filtrarea adaptată. Şi în acest caz rămâne valabilă relația (1.5), cu aceeaşi concluzie relativ la bandă şi la utilizarea spectrului împrăştiat.

d)-*Realizarea compatibilității* între diferite sisteme de transmisie. Dacă două sisteme de comunicație utilizează acelaşi mediu şi aceeaşi bandă de frecvență atunci fiecare dintre cele două este un perturbator pentru celălalt. Urmând acelaşi raționament ca şi în cazul bruiajului, cea mai bună soluție posibilă (exceptând evident separarea totală a benzilor) este ca cele două să utilizeze întreaga bandă posibilă.

e)-*„Ascunderea sau acoperirea”* semnalului. Păstrând constantă energia semnalului (pentru a nu afecta RSZ-ul) putem uşor „masca” semnalul sub zgomotul termic utilizând un produs WT mare (relația (1.3)). Astfel, cu notațiile din Figura 1.4, raportul semnal pe zgomot pentru un receptor ce ar intercepta transmisia este:

$$\xi_{\text{int}}^2 = \frac{\frac{P_s}{W}\cdot \delta f}{N_0 \cdot \delta f} = \frac{E}{W \cdot T \cdot N_0} = \frac{\xi_0^2/2}{W \cdot T}. \qquad (1.6)$$

Pentru a avea un raport semnal pe zgomot de „intercepție”, ξ_{int}^2, mult mai mic decât cel din transmisia în cauză, ξ_0^2, este necesar un produs W·T mare (relația (1.3)).

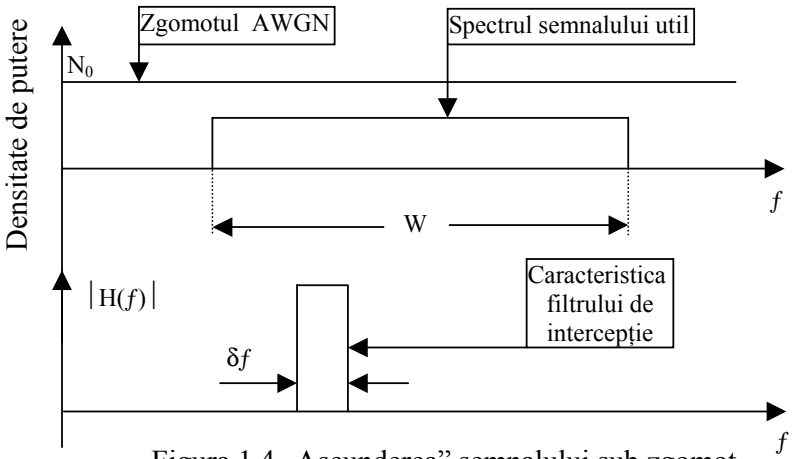


Figura 1.4 „Ascunderea" semnalului sub zgomot

f)-S*ecretizarea transmisiei*. Nivelul de secretizare este dat de numărul N_k de posibilităţi de codare, numite „chei". Dacă, spre exemplu, M mesaje sunt codate prin cuvinte cu n simboluri dintr-un alfabet de mărime A, rezultă, pentru numărul de chei, valoarea

$$N_k = A^n \cdot (A^n - 1)\cdot(A^n - 2)\cdot\ldots\cdot(A^n - (M-1)) \approx (A^M)^n. \qquad (1.7)$$

La o rată de transmisie, R, dată, T este fixat (T = 1/R). Având durata simbolului Δ (W≈1/Δ) lungimea cuvântului de cod este $n = T/\Delta \approx WT$, şi astfel

$$N_k \approx (A^M)^{WT}. \qquad (1.8)$$

Ultima relaţie indică că, pentru a avea o mică probabilitate de intercepţie (adică un N_k mare), este necesar un produs WT >> 1.

g)-C*ombaterea fading-ului*. Fading-ul (fluctuaţia semnalului recepţionat) este un efect al propagării semnalului pe mai multe căi, de la emiţător spre receptor, în cazul transmisiilor radio terestre. Propagarea diferitelor părţi din semnalul emis de antena emiţătoare spre antena receptoare se face, pe lângă „unda directă", prin reflexii multiple. Astfel, rezultatul este că la recepţie sosesc diferite semnale, copii întârziate şi/sau atenuate ale undei directe, suprapuse peste aceasta. În plus aceste întârzieri şi atenuări variază în timp. Urmarea este că, la aceeaşi putere a semnalului emis, probabilitatea de eroare la reconstrucţia semnalului în receptor creşte dramatic dacă componentele multicăi se suprapun în antifază, [6]. Combaterea fading-ului se realizează prin „*diversitate*". Principial, diversitatea presupune existenţa unui număr de L canale paralele, purtând aceleaşi date, pentru care fading-ul este statistic independent. Datorită independenţei probabilităţii, P_L, a suprimării complete a semnalelor din toate canalele faţă de a unuia, aceasta va descreşte cu L: $P_L = P_0^L$, unde P_0 este probabilitatea suprimării semnalului într-un canal. Modurile „clasice" de obţinere a diversităţii sunt:

–diversitate spaţială, prin dispunerea mai multor antene de recepţie, separate între ele prin câteva lungimi de undă. Fading-urile în ele sunt independente.

–diversitate în frecvenţă, prin transmisia paralelă a mai multor frecvenţe purtătoare cu diferite lungimi de undă.

Ambele metode fac uz de diferenţa de fază dintre diferitele căi:

$$\Delta\varphi = 2\cdot\pi\cdot \Delta D / \lambda, \qquad (1.9)$$

unde ΔD este diferența de lungime dintre căi iar λ este lungimea de undă. Astfel, considerând semnalele reprezentate prin vectori, este posibil ca o antenă de recepție (vorbind despre cazul diversității spațiale) să compună vectorii—semnal—recepționat ca şi în Figura 1.5.a, (cazul unei suprimări a semnalului), dar este mult mai probabil ca să existe măcar o antenă receptoare (atunci când sunt mai multe) care să compună vectorii—semnal—recepționat ca şi în Figura 1.5.b, caz în care avem o amplificare a semnalului şi deci o recepție posibilă. Asemenea, şi în cazul diversității în frecvență, pentru unele lungimi de undă este posibil să existe suprimare (Figura 1.5.a), dar este mult probabil ca măcar pentru o lungime de undă să existe amplificare.

a) suprimare; b) amplificare;

Figura 1.5 Diagrama fazorială în cazul recepției prin „diversitate”

O alternativă la metodele prezentate anterior o reprezintă *diversitatea temporală*. Aceasta se bazează pe separarea temporală a semnalelor diferitelor căi, la recepție, prin filtrare adaptată. Diferența de timp dintre semnalele aferente a două căi este

$$\Delta\tau = \Delta D / c, \tag{1.10}$$

unde c este viteza luminii. Astfel, dacă notăm cu $\Delta\tau_{min}$ valoarea minimă a lui $\Delta\tau$ peste toate perechile de căi, şi dacă $\Delta\tau_{min} > 2/W = 2\cdot\tau_c$, atunci conform Figurii 1.2, răspunsurile filtrului adaptat la semnalele aferente diferitelor căi pot fi separate în timp. După aceasta, ele pot fi procesate într-o manieră coerentă sau necoerentă. Dacă fiecare amplitudine complexă, A_i, poate fi măsurată suficient de precis, atunci cea mai bună soluție este să sumăm semnalele, ponderate cu A_i^* (procesarea coerentă):

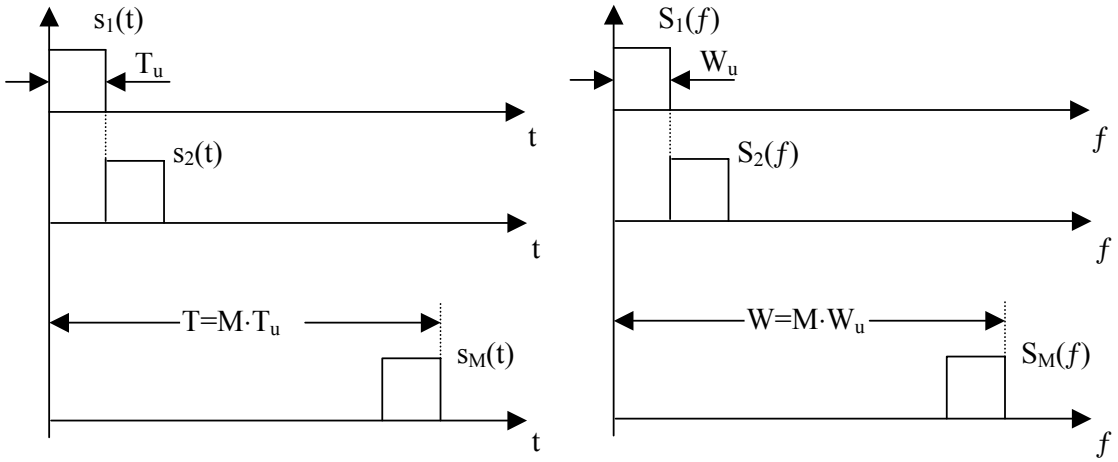
$$s_i(t) = \sum_{i=1}^{L} A_i^* \cdot s_i(t), \tag{1.11}$$

unde $s_i(t)$ este semnalul aferent căii i. În acest fel, în schimbul pierderilor, vom avea o creştere în RSZ de L^2 ori, iar utilizarea spectrului împrăştiat (cea care permite separarea căilor) face ca efectul propagării multicale să fie unul benefic.

Combaterea fading-ului se poate face combinând diversitatea cu alte metode, cum ar fi codarea, [36].

h)-*Accesul multiplu*. Problema accesului multiplu apare în situaţiile în care un număr relativ mare de utilizatori, poziţionaţi într-o *zonă geografică comună*, utilizează, (posibil) *simultan*, *aceeaşi alocaţie spectrală* în vederea comunicării separate[1] sau private.

În abordarea clasică, fiecare utilizator al sistemului cu acces multiplu este înzestrat cu resurse certe, ca frecvenţa (în tehnica numită FDMA –Frequency Division Multiple Access) sau sloturile de timp (în tehnica numită TDMA –Time Division Multiple Access), sau ambele, care sunt disjuncte cu cele ale oricărui alt utilizator. În acest fel canalul cu acces multiplu se reduce la o multititudine de canale punct la punct, asigurând o izolare perfectă a resurselor fiecărui utilizator de a celorlalţi.



TDMA–fiecare utilizator ocupă toată banda disponibilă şi doar o parte (slot) 1/M din timp;

FDMA –fiecare utilizator ocupă tot timpul disponibil şi doar o parte 1/M din banda disponibilă;

Figura 1.6 Sisteme de comunicaţie cu acces multiplu „tradiţionale”

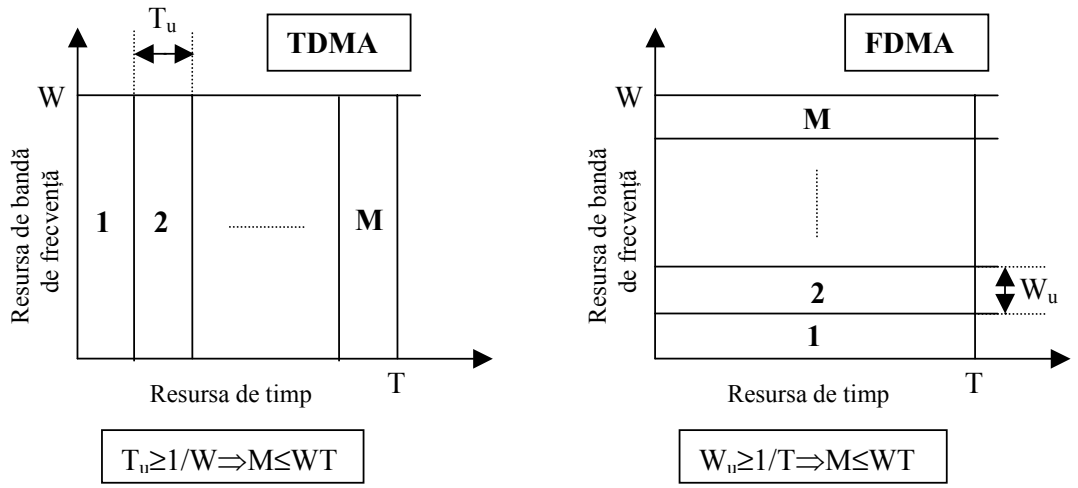


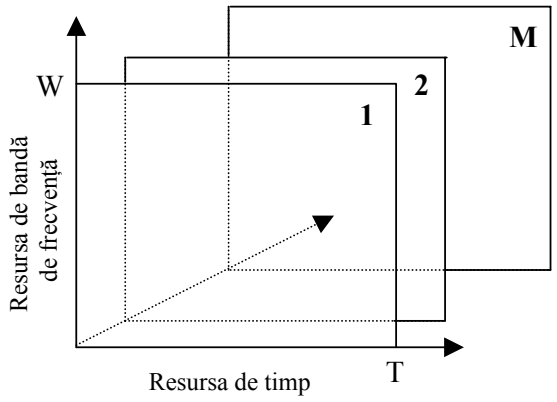
Figura 1.7 Alocarea resurselor şi numărul de utilizatori

[1] Apelativul „separate” subliniază faptul că nu este vorba de o comunicaţie prin „difuzare”, caz în care un emiţător transmite către mai mulţi utilizatori; comunicaţiile invocate în text reprezintă legături multiple între diverşi emiţători şi receptori.

Capacitatea fiecărui canal este limitată numai de lăţimea de bandă şi timpul alocat, de degradarea cauzată de zgomotul de fond, în principal de origine termică, şi de anomaliile de propagare, care produc efectele de fading multicăi (multipath fading) şi de umbrire (shadowing effect). Astfel, în aparenţă, capacitatea, compusă de componentele individuale ale canalelor cu acces multiplu, va fi egală cu capacitatea unui singur utilizator ocupând întregile resurse compuse ale canalelor individuale.

O alternativă a metodelor clasice, prezentate anterior, este CDMA—Code Division Multiple Access. Această metodă alocă (vezi Figura 1.8) toate resursele simultan tuturor utilizatorilor, iar puterea transmisă de fiecare este cea minim necesară pentru a menţine un raport semnal pe zgomot dat de nivelul de performanţă necesar. Fiecare utilizator întrebuinţează un semnal cu lăţime de bandă ca cea a zgomotului ocupând întreaga bandă alocată, pentru cât timp e necesar. În acest fel fiecare utilizator contribuie la zgomotul de fond ce afectează toţi utilizatorii, dar în modul cel mai puţin distructiv. Această interferenţă adiţională limitează capacitatea, dar pentru că resursele alocate de timp şi lărgime de bandă sunt nerestricţionate, capacitatea rezultantă este semnificativ mai mare decât la sistemele convenţionale. CDMA poate fi atât sincronă cât şi asincronă.

Figura 1.8 Alocarea Resurselor în CDMA



CDMA—sincronă. Acest mod este realizabil dacă toate cele M semnale ale utilizatorilor pot fi transmise sincron, de exemplu în legătura descendentă[1] a telefoniei mobile. Fiecare utilizator utilizează secvenţa sa unică (numită „semnătură" sau „cheie"). Fiecare semnătură ocupă toată resursa timp-frecvenţă disponibilă. Nu există o distribuire a resurselor de timp sau frecvenţă între utilizatori. Toate semnăturile sunt sincrone, adică staţia de bază le generează în acord cu un tact propriu. Într-o bandă W şi într-un timp T pot opera peste M=W·T utilizatori fără interferenţă mutuală, deoarece există M=W·T semnale ortogonale, ce joacă rolul de semnături. Exemplu: legătura descendentă pentru telefonia mobilă IS-95 dispune de 64 de funcţii Walsh cu rol de semnături pentru 64 de canale ale utilizatorilor.

Este posibilă creşterea numărului utilizatorilor peste un W·T dat, permiţând un anume nivel al interferenţei mutuale. Atunci, acest mod devine oarecum asemănător cu CDMA asincron. Exemplu: legătura descendentă pentru 3G cdma 2000.

CDMA—asincronă se utilizează atunci când este imposibilă sincronizarea tuturor semnăturilor, de exemplu în legătura ascendentă (de la MS către BS) din telefonia mobilă.

[1] Legătura descendentă, sau inversă (down link) este legătura dinspre o staţie de bază (BS) către o staţie mobilă (MS), sau canalul „unul către mai mulţi". Legătura ascendentă, sau directă (up link) este legătura dinspre o staţie mobilă către o staţie de bază, sau canalul „mai mulţi către unul".

Diferența de timp dintre semnalele utilizatorilor este necontrolabilă şi aleatoare, astfel că semnăturile (spre deosebire de CDMA sincronă) nu pot rămâne ortogonale şi interferența mutuală devine iminentă. Nivelul ei defineşte numărul de utilizatori. Semnăturile trebuie să fie alese în aşa fel încât, dacă un receptor este acordat pe o semnătură, toate celelalte apar ca şi zgomotul pentru acel receptor. În acest fel, ținând cont şi de interferența mutuală, RSZ al filtrului adaptat devine

$$\xi_r^2 = \frac{2 \cdot E}{N_0 + N_i} = \frac{\xi^2}{1 + (M-1) \cdot \xi^2 / (2 \cdot W \cdot T)}$$

$$\approx \frac{2 \cdot W \cdot T}{M} \qquad (1.12)$$

unde

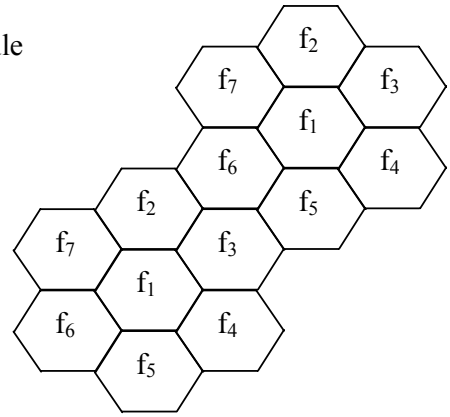
$$N_i = (M-1) \cdot P_s / W \qquad (1.13)$$

este densitatea spectrală de putere a interferenței totale de la alți utilizatori iar $\xi^2 = 2 \cdot E/N_0$ este RSZ în lipsa interferenței. Pentru un ξ^2 fixat, numărul maxim de utilizatori este:

$$M \approx \frac{2 \cdot W \cdot T}{\xi^2} \qquad (1.14)$$

Comparație între FDMA, TDMA şi CDMA. Pentru FDMA, TDMA şi CDMA sincronă $M = W \cdot T$, în timp ce pentru CDMA asincronă $M \approx 2 \cdot W \cdot T/\xi^2$. Dacă[1] $P_b < 10^{-2}$, este necesar un RSZ $\xi^2 > 5{,}4$ (7,3dB), ceea ce conduce la $M < 2 \cdot W \cdot T/5$, adică de 2,5 ori mai puțin decât în cazul FDMA şi TDMA. Exemplu: în legătura ascendentă IS-95 W=1,23MHz, T≈208μs. Cu FDMA, TDMA sau CDMA sincron rezultă M=256. Pentru CDMA asincron M=95 . Aparent aceasta înseamnă că CDMA este o soluție mai puțin performantă. În fapt nu este aşa.

Figura 1.9 Două grupuri de celule ce utilizează aceleaşi frecvențe



[1] este un exemplu de cerință de BER nepretențios. (BER = bit error rate, sau P_b = probabilitatea eronării unui bit).

Pentru telefonia celulară este imposibil de repetat aceeaşi frecvenţă (FDMA) sau slot de timp (TDMA) în celulele învecinate (sau chiar în celule separate printr-o altă celulă), deoarece interferenţa cu utilizatorii din alte celule poate fi dezastruoasă. Astfel pentru FDMA şi TDMA toată resursa W·T trebuie divizată între celulele unui grup (cluster). Grupurile tipice constau din 7 celule. (vezi Figura 1.9)

Astfel, pentru FDMA şi TDMA, numărul de utilizatori per celulă

$$M_c = M / 7 = W \cdot T / 7 \tag{1.15}$$

În exemplul în cauză, cu FDMA şi TDMA pot fi doar $M_c = 36$ de utilizatori per celulă.

Calculul numărului utilizatorilor pe celulă, M_c, în cazul CDMA ce admite interferenţa, trebuie refăcut ţinând cont de unele considerente practice, [1] [2]. Astfel, există următoarea relaţie:

$$P_r \approx k \cdot P_t \cdot D^{-m} \tag{1.16}$$

între P_r –puterea semnalului la recepţie şi P_t –puterea semnalului la emisie. D este distanţa între BS şi MS iar m este exponentul atenuării de propagare. Pentru arii urbane dense şi înălţimi tipice ale antenei emiţătoare (10÷20m) rezultă m ≈3,8. Datorită acestei căderi rapide, rezultă că puterea semnalului recepţionat de către o staţie de bază de la o staţie mobilă aflată într-o altă celulă este de câteva ori mai mică decât pentru o staţie mobilă din celula proprie. Practic, contribuţia totală, în interferenţă, a tuturor utilizatorilor din alte celule este jumătate din cea creată de utilizatorii propriei celule [1]. În acest fel, admiţând o creştere a interferenţei de 1,5 ori datorită utilizatorilor tuturor celorlalte celule, este posibil, pentru CDMA asincron, de a re-utiliza aceleaşi resurse frecvenţă-timp în celulele aceluiaşi grup.

O altă concluzie practică este faptul că vocea umană, într-o vorbire curentă, este activă doar β=40% (parametrul activităţii vocii) din timpul afectat ei. Dacă puterea de emisie a unei MS poate fi redusă (sau chiar anulată) atunci când vocea utilizatorului nu este activă, interferenţa totală se va reduce proporţional. Acest fapt conduce la o îmbunătăţire a utilizării resurselor. Astfel, recalculând interferenţa la valoarea $\alpha \cdot \beta \cdot N_i$, cu α=1,5 (factor responsabil de creşterea interferenţei datorită altor celule) şi β=0.4 rezultă, [1], pentru RSZ relaţia

$$\xi_r^2 = \frac{2 \cdot E}{N_0 + \alpha \cdot \beta \cdot N_i} = \frac{\xi^2}{1 + \alpha \cdot \beta \cdot (M-1) \cdot \xi^2 / W \cdot T}$$

$$\approx \frac{3.3 \cdot W \cdot T}{M} \tag{1.17}$$

iar numărul de utilizatori per celulă devine

$$M_c \approx \frac{3{,}3 \cdot W \cdot T}{\xi_r^2} \approx 0{,}6 \cdot W \cdot T \tag{1.18}$$

Pentru legătura înaltă din IS-95, ultima relaţie conduce la $M_c = 153$, care este mult mai mare decât pentru FDMA şi TDMA.

O ultimă remarcă practică ar fi legată de utilizarea de celule sectorizate. Dacă presupunem că populaţia de utilizatori este uniform distribuită pe suprafaţa unei singure celule izolate, întrebuinţarea antenei sectorizate reduce interferenţa [25] şi ca atare creşte

capacitatea prin factorul numit câştigul antenei, G_A. Definiția clasică pentru câştigul antenei (bidimensionale) este energia recepționată în direcția transmițătorului, divizată cu energia medie recepționată (mediere care se face pe tot cercul). Pentru o antenă sectorizată în trei, acest factor de câştig este mai mic de 3. Dacă pierderea față cazul ideal este 1 dB, $G_A \approx 2{,}4$.

1.3 Tehnici de împrăştiere a spectrului. Comparație

Există diferite tehnici de a împrăştia un semnal: –împrăştiere cu Secvență Directă (DS Direct Sequence), –împrăştiere cu salt de frecvență (FH Frequency-Hopping), –împrăştiere cu salt de timp (TH Time Hopping) şi Multipurtătoare CDMA. Sunt posibile, de asemenea, combinații ale lor.

1.3.1 Împrăştiere cu secvență directă

Secvența-directă este cea mai bună tehnică de împrăştiere a spectrului cunoscută. Semnalul de date este multiplicat cu un cod zgomot pseudo-aleator (*PN code*). Un cod PN este o secvență de semnale elementare (*chips*) de valori –1 şi 1 (polară) sau 0 şi 1 (nepolară) şi are proprietățile zgomotului. Aceste proprietăți de zgomot duc la valori mici ale intercorelației dintre coduri şi la dificultatea de a bruia sau de a detecta mesajul de date.

Există câteva familii de coduri PN binare, descrise în capitolul 2. O cale uzuală de a crea un cod PN este prin intermediul unui registru de întârziere. Dacă lungimea unui astfel de registru de întârziere este m, se poate afirma despre perioada N_{DS} a familiilor de cod menționate:

$$N_{DS} = 2^m - 1 \tag{1.19}$$

În sistemele cu secvență directă, lungimea codului este [22] aceeaşi cu factorul de împrăştiere adică:

$$\beta_e(DS) = N_{DS} \tag{1.20}$$

În Figura 1.10 se vede cum codul PN este aplicat cu semnalul de date, (în acest exemplu N_{DS}=7). Lărgimea de bandă a semnalului de date este multiplicată cu factorul N_{DS}. Puterea rămânând aceiaşi, rezultă că densitatea spectrală de putere a semnalului rezultat scade.

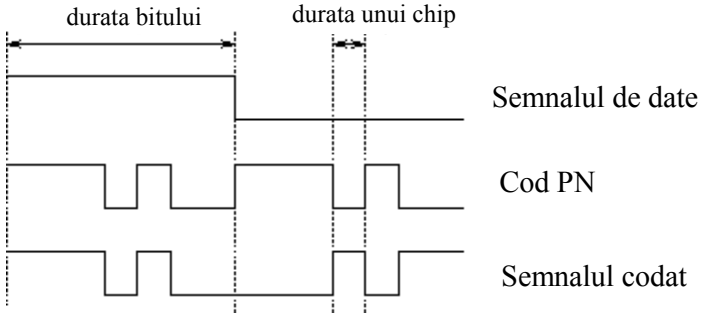


Figura 1.10 Conceptul împrăştierii cu DS

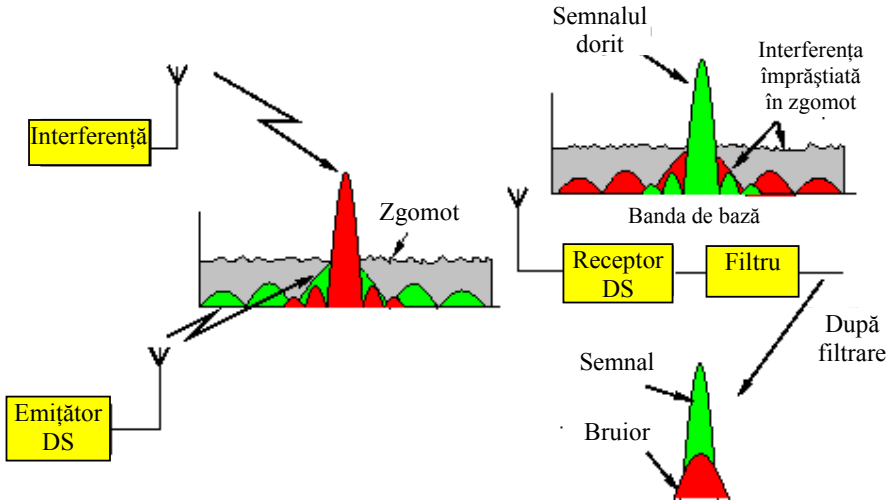


Figura 1.11 Concept DS, înainte şi după împrăştiere

Generarea codurilor PN este relativ uşoară, fiind necesare doar un număr de registre de întârziere. Din acest motiv este uşor să se introducă un câştig de procesare mare în sistemele cu secvență directă.

În receptor, semnalul recepționat este multiplicat din nou cu acelaşi cod-PN (sincronizat). Dacă semnalul recepționat şi codul-PN iau doar valorile +1 şi –1, operația de multiplicare elimină codul-PN din semnal şi reface semnalul de date original. Altă observație e că operația de deîmprăştiere este aceeaşi ca şi cea de împrăştiere. Consecința este că un posibil semnalul de bruiaj din canalul radio va fi împrăştiat înainte de detecția datelor. Astfel efectele de bruiaj sunt reduse.

Principala problemă când se aplică împrăştierea cu secvență directă apare atunci când emițătorul semnalul perturbator este mai aproape de receptor decât transmițătorul dorit, *(Near-Far effect)*, Figura 1.12 . Chiar dacă intercorelația dintre codurile A şi B este mică, corelația dintre semnalul recepționat de la transmițătorul-perturbator şi codul A poate fi mai mare decât corelația dintre semnalul recepționat de la transmițătorul dorit şi codul A. Rezultatul este că detecția datelor nu este posibilă.

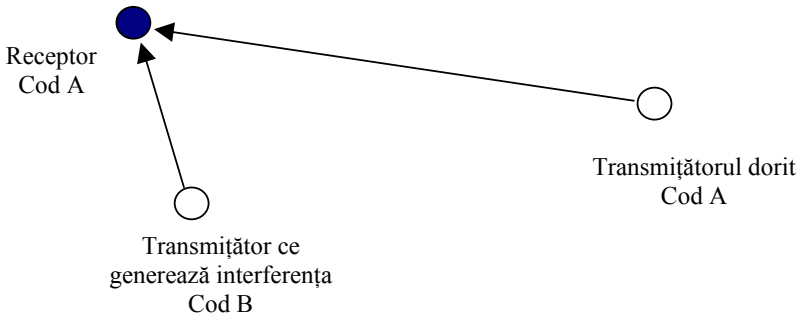


Figura 1.12 Ilustrarea efectului de perturbare a transmisiei de către terminalul apropiat

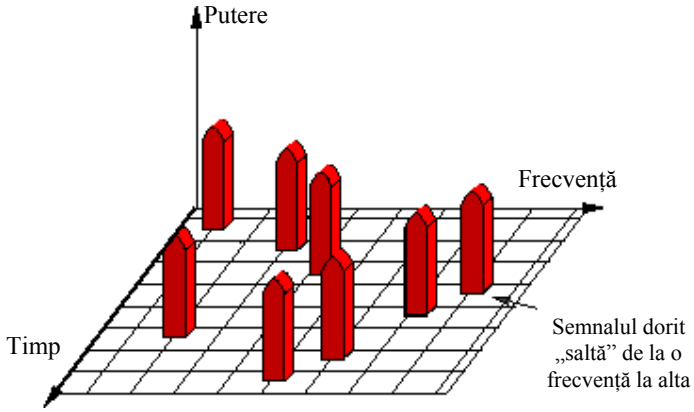


Figura 1.13 Ilustrarea conceptului Frequency-Hopping.

1.3.2 Împrăştiere cu salt de frecvență (Frequency Hopping FH)

O altă tehnică de împrăştiere a spectrului este prin salt de frecvență sau Frequency Hopping (FH), care prezintă un efect „near–far" mult mai scăzut. Frecvența purtătoare este modificată în acord cu o secvență unică (o secvență FH de lungime N_{FH}). În acest fel banda este mărită prin factorul N_{FN} (dacă canalele nu se suprapun):

$$\beta_e(FH) = N_{FH} \qquad (1.21)$$

Procesul împrăştierii cu FH e ilustrat în Figura 1.13. Dezavantajul împrăştierii cu FH, în comparație cu împrăştierea cu secvență directă, constă în dificultatea obținerii unui câştig de semnal mare. Creşterea câştigului de procesare presupune mărirea vitezei de salt.

Pe de altă parte, împrăştierea cu FH are un efect aproape-departe mai scăzut decât secvența directă. Fiecare secvență FH are doar un număr limitat de locații comune cu celelalte. Aceasta înseamnă că dacă există în apropiere un bruior, doar un număr de frecvențe de salt va fi blocat în loc să fie blocat tot semnalul. Pe baza frecvențelor care nu sunt blocate este posibil să se refacă mesajul de date original.

1.3.3 Împrăştiere cu salt de timp (Time Hopping TH)

DS şi FH sunt cele mai comune forme de semnale cu spectru împrăştiat utilizate în practică. Însă pot fi folosite şi alte metode pentru a introduce pseudoaleatorul în semnalul cu spectru împrăştiat. O metodă, similară cu FH, este saltul (comutarea) timpului. În TH, un interval de timp, care este ales mult mai mare decât 1/R, inversul ratei informației, este divizat într-un număr mare de sloturi de timp. Simbolurile informației codate sunt transmise, într-un slot de timp ales pseudoaleator, ca un bloc de unul sau mai multe cuvinte de cod. Pentru a transmite biții codați se poate utiliza modulația PSK.

Spre exemplu, presupunem că un interval de timp T este divizat în 1000 de sloturi de timp de lățime T/1000 fiecare. Cu o rată a biților de informație de R biți/s, numărul de biți transmişi în T secunde este R·T. Codarea creşte acest număr la $R \cdot T/R_c$ biți, unde R_c este rata de codare. În consecință, într-un interval de T/1000 secunde, trebuie transmişi $R \cdot T/R_c$ biți. Dacă se utilizează ca metodă de modulare PSK binară, rata biților este $1000R/R_c$, iar banda necesară este aproximativ $W = 1000 \cdot R/R_c$.

O schemă bloc a unui transmițător şi a unui receptor pentru un sistem cu spectru împrăştiat cu TH este prezentată în Figura 1.14. Datorită caracteristicilor de rafală ale semnalului emis, transmițătorul sistemului cu TH trebuie dotat cu buffer de stocare, ca în

Figura 1.14. De asemenea, şi la recepţie trebuie folosit un buffer pentru a asigura un flux de date uniform către utilizator.

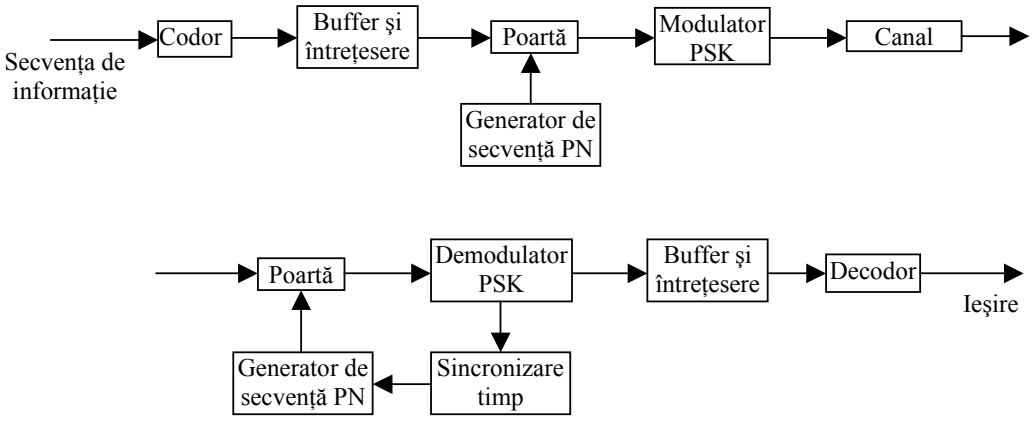


Figura 1.14 Diagrama bloc pentru un sistem cu spectru împrăştiat cu TH

Aşa cum interferenţa degradează doar o parte din banda sistemului necodat cu spectru împrăştiat FH, interferenţa parţială în timp (pulsată) are un efect similar în sistemul cu spectru împrăştiat TH. Pentru combaterea acestui tip de interferenţă se utilizează codarea şi întreţeserea, [2]. Dezavantajul major al sistemului TH este dat de dificultăţile mari de sincronizare în comparaţie nu doar cu FH ci şi cu DS.

1.3.4 Sisteme hibride

Alte tipuri de semnale cu spectru împrăştiat pot fi obţinute combinând DS, FH şi TH, [9]. Tehnica DS/FH este o combinaţie între DS şi FH. Un bit de date este divizat cu un număr egal cu numărul de canale de salt a frecvenţei (frecvenţele purtătoare). În fiecare canal de salt a frecvenţei un cod PN complet în lăţime e multiplicat cu semnalul dată (vezi Figura1.15).

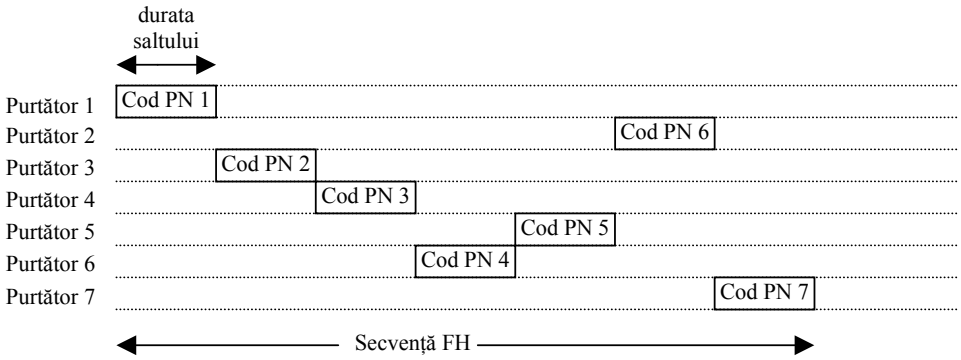


Figura 1.15 Diagrama de împrăştiere DS-FH

Atunci când sunt asociate secvența FH şi codul PN, adresa este o combinație a secvenței FH şi codurile PN. Pentru a limita şansa de suprapunere a locației (şansa ca doi utilizatori să folosească acelaşi canal în acelaşi timp), secvențele de salt a frecvenței sunt alese în aşa fel încât doi transmițători cu diferite secvențe FH să împartă cel mult 2 frecvențe în acelaşi timp (timpul de schimbare e aleator), [9].

Faptul că într-un salt detecția este coerentă reprezintă un avantaj obținut relativ la sistemul FH pur. Însă prețul plătit pentru câştigul în performanță este o creştere a complexității, un cost mai mare, şi necesități mai mari de sincronizare.

Un alt semnal hibrid posibil cu spectru împrăştiat este DS/TH. Acesta nu pare a fi la fel de practic ca DS/FH, în primul rând datorită creşterii în complexitate a sistemului cât şi faptului că apar probleme mult mai mari la sincronizare.

Detecția sau recepția semnalelor cu spectru împrăştiat este o problemă delicată şi de actualitate. În literatura actuală există numeroase articole pe această temă. Metodele de detecție propuse sunt dintre cele mai diverse: metode bazate pe rețele neuronale [13], bazate pe calculul funcțiilor de (auto)corelație [14], pentru semnale SS-FH, detecția prin corelare a semnalului chip în cazul SS-DS [17], [28], sau analiza timp-frecvență [41].

2. Secvențe pseudo-aleatoare (CN—coduri zgomot)

2.1 Cerințele codurilor zgomot

Semnalele transmise în sistemele cu spectru împrăştiat se doresc a fi aleatoare ca şi zgomotul [1]. Pentru a fi folosite în sistemele realizabile, aceste semnale trebuie să fie construite pe baza unui număr finit de parametri, selectați aleatoriu şi stocați. La fel de important, semnalele trebuie să fie generate şi la receptor şi trebuie să fie sincronizate pentru a coincide perfect cu tactul transmisiei recepționate, [1].

Deoarece doar un număr finit de parametri pot fi stocați memoria transmițătorului şi receptorului, urmărind ideile stabilite de teorema de eşantionare a lui Nyquist, valorile numerice ale formei de undă aleatoare trebuie doar să fie specificate ca eşantioane la intervale de timp proporționale cu inversul benzii ocupate de semnale. Trecând aceste eşantioane printr-un filtru liniar se generează întreaga formă de undă continuă în timp ca o interpolare a semnalelor de intrare. Deoarece semnalele sunt ca zgomotul gausian, fiecare eşantion poate fi aproximat ca şi o variabilă aleatoare gausiană. Însă, aceasta va necesita specificarea unui număr suficient de biți pe eşantion pentru a obține fidelitatea dorită. Chiar limitând complexitatea prin specificarea doar a unui bit pe eşantion, ceea ce corespunde unei secvențe binare, efectul utilizării unei astfel de forme de undă binare aleatoare este aproape la fel cu utilizarea unei forme de undă de tip zgomot gausian. Faptul că forma de undă aleatoare binară poate fi uşor şi flexibil modulată cu un semnal digital purtător de informație are consecințe practice importante.

O secvență aleatoare binară independentă este secvența Bernoulli (numită în literatura inginerească „coin-flipping”).Proprietățile de cheie „aleatoare” a secvenței Bernoulli sunt:

R.1 *Proprietatea de simetrie*:
Frecvențele relative pentru „0” şi „1” (ponderile zerourilor şi unu-urilor) sunt ½.

R.2 *Proprietatea de lungime de fugă*:
În secvența aleatoare binară apar grupuri de zerouri şi unu-uri, numite lungimi de fugă; jumătate din lungimile de fugă sunt de mărime (lungime) unitară; un sfert sunt de lungime doi; o optime sunt de lungime 3; o fracție $1/2^n$ sunt de lungime n pentru orice n finit.

R.3 *Proprietatea de deplasare şi adunare*:
Dacă secvența aleatoare este deplasată cu orice număr nenul de elemente, secvența rezultată va avea un număr egal de coincidențe şi necoincidențe cu secvența originală.

O secvență generată deterministic ce satisface (R.1)÷(R.3), cu discrepanțe extrem de mici, va fi numită *secvență pseudo-aleatoare*. O definiție mult mai precisă se poate da după considerarea procesului de generare (în paragraful următor).

Un cod PN utilizat pentru dispersia DS este compus din N_{DS} unități numite chipuri; aceste chipuri pot avea 2 valori: -1/1 (bipolar) sau 0/1 (unipolar). Dacă se combină fiecare simbol cu un cod-PN complet, câştigul de procesare la DS este egal cu lungimea codului,

relația (1.1). Pentru a fi utilizabil ca o secvență-directă, un cod PN trebuie să îndeplinească următoarele condiții:

- Secvențele trebuie să fie compuse din numere cu 2 niveluri.
- Autocorelația codurilor trebuie să aibă un maxim (de lățimea unui chip) pentru a permite sincronizarea de cod.
- Codurile trebuie să aibă o valoare mică a intercorelației. Cu cât este mai mică această intercorelație, cu atât mai mulți utilizatori vom putea avea în sistem. Această cerință este valabilă atât pentru corelare completă cât şi pentru corelare parțială. Se impune şi pentru corelarea de cod parțială deoarece în majoritatea situațiilor nu vom avea o corelare de-a lungul unei întregi perioade pentru două coduri, este mult mai probabil ca cele două coduri să fie corelate doar parțial (datorită modului de acces aleator).
- Codurile trebuie să fie „echilibrate": diferența dintre numărul de 1 şi de 0 în cod poate fi doar de 1 pentru ca densitatea spectrală de energie să fie distribuită uniform în bandă.

2.2 Secvențe pseudo-aleatoare. Proprietăți. Generare

2.2.1 Secvențe generate de Registru de Deplasare

Proprietățile de cheie aleatoare a secvenței Bernoulli pot fi obținute de o secvență periodică deterministă lungă care poate fi generată printr-o operație liniară simplă specificată printr-un număr moderat (zeci) de parametri binari (biți), [1]. Astfel, singura variabilă aleatoare este punctul de start al secvenței.

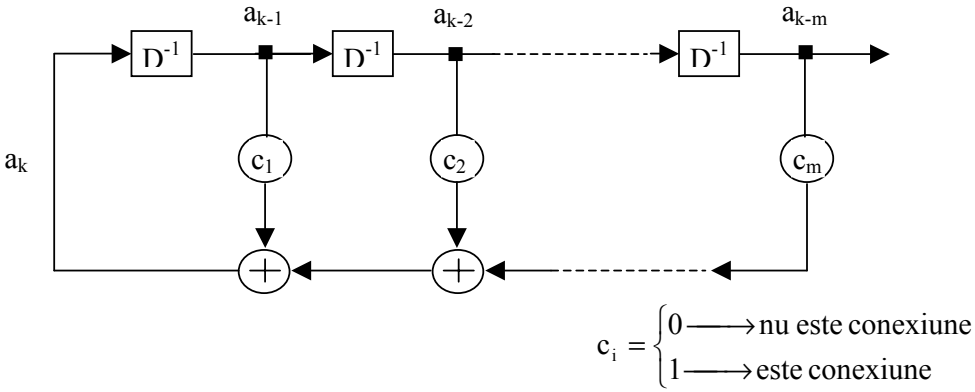


Figura 2.1 Registru de deplasare

Considerăm generatorul de secvență binară din Figura 2.1. La fiecare moment de tact registrul deplasează întregul conținut spre dreapta. Secvența $\{a_k\}$ se generează conform relației:

$$a_k = c_1 \cdot a_{k-1} + c_2 \cdot a_{k-2} + \ldots + c_m \cdot a_{k-m} = \sum_{i=1}^{m} c_i \cdot a_{k-i}, \quad m, k \in N \qquad (2.1)$$

Aici toţi termenii sunt binari, c_1 la c_m sunt variabilele conectori (1 indică prezenţa conexiunii, iar 0 absenţa ei), regulile de multiplicare sunt cele ordinare, dar adunarea este modulo 2. Cu aceste reguli toate operaţiile sunt liniare[1] şi se aplică legea de distributivitate. Secvenţa $\{a_k\}$ astfel generată poate fi dublu infinită. Însă, considerând doar termenii cu indice nenegativ, se defineşte funcţia G(D), generatoare a secvenţei, folosind operatorul D de întârziere:

$$G(D) = a_0 + a_1 \cdot D + a_2 \cdot D^2 + \ldots = \sum_{k=0}^{\infty} a_k \cdot D^k \qquad (2.2)$$

exponentul lui D din fiecare termen al acestui polinom corespunde cu numărul de unităţi (cicluri de tact) de întârziere pentru acel termen.

Din (2.2) şi (2.1) rezultă:

$$\begin{aligned} G(D) &= \sum_{k=0}^{\infty} a_k \cdot D^k = \sum_{k=0}^{\infty} \sum_{i=1}^{m} c_i \cdot a_{k-i} \cdot D^k \\ &= \sum_{i=1}^{m} c_i \cdot D^i \sum_{k=0}^{\infty} a_{k-i} \cdot D^{k-i} \\ &= \sum_{i=1}^{m} c_i \cdot D^i \left[a_{-i} \cdot D^{-i} + \ldots + a_{-1} \cdot D^{-1} + G(D) \right]. \end{aligned}$$

şi:

$$G(D) \cdot \left(1 + \sum_{i=1}^{m} c_i \cdot D^i \right) = \sum_{i=1}^{m} c_i \cdot D^i \cdot \left[a_{-i} \cdot D^{-i} + \ldots + a_{-1} \cdot D^{-1} \right],$$

Rezultă că G(D) poate fi exprimat ca raport de polinoame finite

$$G(D) = \frac{\sum_{i=1}^{m} c_i \cdot D^i \cdot (a_{-i} \cdot D^{-i} + \ldots + a_{-1} \cdot D^{-1})}{1 + \sum_{i=1}^{m} c_i \cdot D^i} = \frac{h_0(D)}{g(D)}, \qquad (2.3)$$

unde

$$g(D) = 1 + \sum_{i=1}^{m} c_i \cdot D^i \qquad (2.4)$$

este *polinomul caracteristic* al generatorului de secvenţă registru de deplasare şi depinde numai de vectorul conexiune $[c_1, \ldots, c_m]$. Polinomul $h_0(D)$ depinde de *vectorul condiţie iniţială* $[a_{-m}, a_{-m-1}, \ldots, a_{-1}]$, care reprezintă conţinutul registrului înainte ca termenul a_0 să fie generat.

Se poate scrie:

[1] Alternativ, în schimb de zerouri şi unu-uri, se pot pune ca simboluri binare numerele reale +1 şi –1, păstrând coeficienţii c_i ca şi anterior, dar înlocuind toate sumatoarele din căile de întoarcere prin multiplicatoare ordinare. Rezultatul este acelaşi, însă operaţiile nu mai apar ca şi liniare în sensul clasic.

$$h_0(D) = \sum_{i=1}^{m} c_i \cdot \left(a_{-i} + a_{-i+1} \cdot D + \ldots + a_{-1} \cdot D^{i-1}\right)$$
$$= c_1 \cdot a_{-1} + c_2 \cdot (a_{-2} + a_{-1} \cdot D) + c_3 \cdot (a_{-3} + a_{-2} \cdot D + a_{-1} \cdot D^2) + \ldots + c_m \cdot (a_{-m} + a_{-m+1} \cdot D + \ldots + a_{-1} \cdot D^{m-1}). \quad (2.5)$$

Notăm că dintre toate variabilele conectori, cel puțin $c_m = 1$, pentru ca registrul de deplasare să aibă m stări. În plus, dacă vom considera vectorul inițial:

$$a_{-m} = 1,\ a_{-m+1} = \ldots = a_{-2} = a_{-1} = 0,$$

(2.3) şi (2.5) se reduc la

$$h_0 = 1, \quad G(D) = \frac{1}{g(D)}. \quad (2.6)$$

Utilizând (2.3) şi (2.6), rezultă trei proprietăți de bază, [1], pentru secvențele generate de registru de deplasare liniar (LSR –Linear Shift Register),

P-1: Fiecare secvență LSR este periodică cu perioada

$$P \le 2^m - 1 \quad (2.7)$$

Aceasta ne permite să definim secvența generată de registrul de deplasare de lungime maximă (MLSR –Maximum Lengh *Linear* Shift Register), denumită în continuare ca şi secvență M, ca şi secvența LSR a cărei perioadă $P=2^m-1$ pentru toți vectorii inițiali nenuli. În plus avem

P-2: Pentru toate cazurile degenerate, perioada lui G(D) este cel mai mic întreg pozitiv P pentru care g(D) divide $1 + D^P$. Noțiunea de „degenerate" se referă la cazurile pentru care $h_0(D)$ şi g(D) au factori în comun.

Aceasta ne conduce spre a treia proprietate, care rezultă anulând caracteristica polinomului g(D) de a putea fi factorizat. Dacă g(D) are un factor, atunci va exista un vector inițial $h_0(D)$ ce corespunde la un factor pentru g(D). Aceasta cauzează degenerare ca cea definită anterior, care reduce m şi de aici perioada $P \le 2^m - 1$. Anume

P-3: O condiție necesară pentru G(D) de a genera o secvență M (cu $P = 2^m - 1$) este ca g(D) de grad m să fie ireductibil.

Din păcate, P-3 este o condiție *necesară*, ea nu este şi suficientă. Pentru un contraexemplu, luăm m = 4 cu $P = 2^4 - 1 = 15$ dorită. Considerăm $g(D) = 1 + D + D^2 + D^3 + D^4$, care este ireductibil. Însă, g(D) divide $1 + D^5$ şi de aceea are perioada 5, mai mică decât 15. Pentru a obține perioada 15, putem utiliza în schimb polinomul ireductibil de ordin patru $g(D) = 1 + D + D^4$, care divide $1 + D^{15}$ dar nici un polinom $1 + D^k$ pentru $k<15$.

Astfel de polinoame, ireductibile de grad m, ce generează secvențe M de perioadă $P = 2^m - 1$ sunt numite *primitive*. Există polinoame primitive pentru orice grad m>1. Numărul de polinoame primitive de grad m este dat prin formula:

$$N_P(m) = \frac{2^m - 1}{m} \cdot \prod_{i=1}^{J} \frac{P_i - 1}{P_i}. \qquad (2.8)$$

$\{P_i, i = 1, 2, \ldots, J\}$ este descompunerea în numere prime pentru $2^m - 1$, adică

$$2^m - 1 = \prod_{i=1}^{J} P_i^{e_i} ,$$

unde e_i este un întreg.

În Anexa A este prezentată o listă cu polinoamele primitive, până la gradul 13. Valori tipice de interes sunt pentru m între 10 şi 50.

2.2.2 Proprietăţile secvenţelor M

În continuare se va demonstra că secvenţele M satisfac îndeaproape proprietăţile aleatoare ale secvenţelor aleatoare binare după cum au fost anunţate în secţiunea 2.1. De remarcat faptul că parametrii procesului de generare [coeficienţii c_i ai polinomului generator g(D)] sunt determinişti. Singurii parametri aleatori sunt cei m termeni ai vectorului iniţial sau, echivalent, timpul de deplasare.

R.1: *Proprietatea de simetrie*

Examinăm prima celulă a registrului din Figura 2.1, atunci când întreaga secvenţă este deplasată prin el. Vom urmări acest ultim bit (cel mai din dreapta) al vectorului m-dimensional la fiecare ciclu de tact. Urmărind conţinutul registrului de deplasare atunci când el generează o secvenţă M, se va vedea că registrul trece prin toate cele 2^m-1 stările posibile, exceptând-o pe cea nulă. Dar dacă am include şi pe cea nulă, am avea cu siguranţă o secvenţă echilibrată: dintre cei 2^m vectori binari, jumătate sunt pari (au 0 pe ultima poziţie) şi jumătate sunt impari (au 1 pe ultima poziţie). Deoarece am inclus starea nulă, echilibrul diferă cu 1 faţă de 2^m. Aşadar, din 2^m-1 termeni ai secvenţei M, 2^{m-1} sunt unu-uri şi 2^{m-1}-1 sunt zerouri. Dacă presupunem vectorul iniţial aleatoriu, sau echivalent, presupunem un moment de start ales aleatoriu, probabilitatea ca la un anume tact ieşirea registrului să fie zero, respectiv unu, este:

$$P_m(0) = \frac{2^{m-1} - 1}{2^m - 1} = \frac{1}{2} \cdot \left(1 - \frac{1}{P}\right),$$

$$P_m(1) = \frac{2^{m-1}}{2^m - 1} = \frac{1}{2} \cdot \left(1 + \frac{1}{P}\right).$$

Astfel nesimetria este 1/P. Pentru m = 10, 30 şi 50, 1/P este aproximativ 10^{-3}, 10^{-9}, şi respectiv 10^{-15}.

R.2: *Proprietatea de lungime de fugă*

Considerăm toate conţinuturile de forma

$$x\ x\ \ldots x\ 0\ a_1\ a_2\ \ldots\ a_k\ 0 \qquad \text{şi} \qquad x\ x\ \ldots x\ 1\ b_1\ b_2\ \ldots\ b_k\ 1$$

unde „$a_1\ a_2 \ldots a_k$" este secvența de unu-uri, „$b_1\ b_2 \ldots b_k$" este secvența nulă iar prefixul „x x …x" poate să fie orice secvență binară de lungime m–2–k. Deoarece, aşa cum s-a demonstrat anterior, în decursul unei perioade de $2^m - 1$ tacte, registrul va conține toate secvențele posibile de lungime m (exceptând-o pe cea nulă), rezultă un număr de $2 \cdot 2^{m-2+k}$ secvențe având una dintre formele invocate. Acesta reprezintă de fapt numărul de lungimi de fugă de mărime k. În plus, numărul total de lungimi de fugă de mărime $k<m-1$ este: $N_{k<m-1} = \sum_{k=1}^{m-2} 2^{m-k-1} = \sum_{i=1}^{m-2} 2^i = 2^{m-1} - 2$.

Considerăm acum lungimea de fugă m-1. Pentru ca aceasta să existe, întregul conținut al registrului la un anumit punct trebuie să fie unul dintre

$$\underbrace{00 \ldots 0\ 1}_{m-1} \quad \text{sau} \quad \underbrace{11 \ldots 1\ 0}_{m-1}$$

În ambele cazuri, deoarece conținutele tuturor celor m celule sunt specificate, nu poate fi decât o singură ieşire. În primul caz, ea trebuie să fie unu. Altfel vom introduce starea toți pe zero, ceea ce este exclus. Şi în cel de-al doilea caz, ea trebuie să fie tot unu. Aceasta este singura cale ca să rezulte vectorul cu toți pe unu, care este unul din cei $2^m - 1$ vectori de stare care trebuie să existe în secvență. (Notăm că pentru acest ultim caz, următorul unu trebuie urmat de un zero; în caz contrar, starea cu toți pe unu re-apare indefinit.) Astfel, există doar o singură lungime de fugă m – 1 (cea cu zerouri) şi o singură posibilă lungime de fugă m (cea cu unu-uri). În consecință, numărul total de lungimi de fugă este: $N = 2^{m-1}$.

Concluzionăm astfel că frecvența relativă a lungimii de fugă k (cu zerouri sau unu-uri) este $1 / 2^k$ pentru orice $k \le m - 1$ şi $1 / 2^{(k-1)}$ pentru $k = m$, fără lungimi de fugă posibile pentru $k > m$.

R.3: *Proprietatea de deplasare şi adunare*

Considerăm pentru o secvență M oricare două perioade de tact deplasate. Dacă luăm întreaga secvență de lungime $P = 2^m - 1$ şi o deplasăm cu un număr arbitrar de perioade $\tau < P$, obținem secvența M pentru un vector inițial diferit. Acum, dacă adunăm modulo 2 secvențele originală şi deplasată, obținem o nouă secvență care este ea însăşi secvență M, cu alt vector de start, şi de aceea un alt punct inițial. Simbolic, utilizând notația polinomială din (2.2), numim secvența originală prin $G_0(D)$, cea deplasată prin $G_\tau(D)$, ambele de lungime $2^m - 1$, iar condițiile lor respective inițiale $h_0(D)$ şi $h_\tau(D)$ în acord cu (2.3),

$$G_0(D) = \frac{h_0(D)}{g(D)}, \qquad G_\tau(D) = \frac{h_\tau(D)}{g(D)},$$

unde g(D) este polinomul caracteristic (primitiv) al secvenței M. În plus, deoarece operațiile polinomiale sunt liniare, din legea de distributivitate rezultă că suma modulo 2 a două secvențe M are polinomul

$$G_0(D) + G_\tau(D) = \frac{h_0(D) + h_\tau(D)}{g(D)}.$$

Aceasta înseamnă că ea poate fi generată prin polinomul condiție inițială $h_0(D) + h_\tau(D)$ (a cărui componente sunt suma modulo 2 a respectivelor componente), dar acesta este

el însuşi un alt posibil vector iniţial. Aşadar, secvenţa astfel generată a cărei polinom generator este $G_0(D) + G_\tau(D)$, este ea însăşi o deplasare în timp a aceleiaşi secvenţe M. Aceasta este proprietatea de deplasare şi adunare.

Utilizând aceasta şi proprietatea R.1, rezultă că cele două secvenţe $G_0(D)$ şi $G_\tau(D)$, fiecare de lungime $2^m - 1$, diferă în 2^{m-1} poziţii şi coincid în $2^{m-1} - 1$. Aceasta deoarece suma lor modulo 2 are unu pentru fiecare deosebire şi zero pentru fiecare coincidenţă şi este ea însăşi o secvenţă M, care trebuie de asemenea să aibă proprietatea de cvasi-simetrie R.1.

R.1 şi R.3 pot să fie exprimate în termeni de medie şi corelaţie temporale dacă se asociază numerele reale +1şi –1 valorilor binare „0" şi „1". Astfel:

$$\textbf{(R.1)':}\ \frac{1}{P}\cdot\sum_{k=1}^{P}\alpha_k = -\frac{1}{P}, \qquad (2.9)$$

unde α_k este valoarea reală echivalentă celui de-al k-lea termen al secvenţei M, a_k. Relaţia (2.9) rezultă din proprietatea de cvasi-simetrie; ea ar fi zero pentru simetrie perfectă.

Similar, pentru $\tau \neq 0$,

$$\textbf{(R.3)':}\ \frac{1}{P}\cdot\sum_{k=1}^{P}\alpha_k\cdot\alpha_{kn+\tau} = -\frac{1}{P}, \qquad (2.10)$$

Relaţia (2.9) este media temporală iar (2.10) este corelaţia temporală. Însă, proprietăţile (R.1)' şi (R.3)' pot să fie luate mai degrabă ca şi medieri pe *ansamble* decât medieri de timp. Notăm că secvenţa (deterministă) M devine o secvenţă staţionar ergodică dacă tratăm vectorul iniţial (sau timpul de observaţie) ca şi un vector aleatoriu uniform distribuit (sau variabilă de timp).

Concluzie: cu mica nesimetrie 1/P (mai mică decât o parte per milion pentru m>20), o secvenţă M este indiscutabil o secvenţă binară Bernoulli sau „coin-flipping", cel puţin cu respectarea proprietăţilor (R.1) la (R.3), atâta timp cât vectorul iniţial sau timpul sunt alese arbitrar.

2.3 Coduri utilizate pentru împrăştierea spectrului

Codurile care se pot găsi în sistemele DS practice sunt: coduri Walsh-Hadamard, secvenţe M, coduri Gold şi coduri Kasami. Aceste seturi de coduri pot fi în mare împărţite în două clase: coduri ortogonale şi coduri neortogonale. Secvenţele Walsh fac parte din prima categorie, în timp ce celălalt grup conţine aşa numitele secvenţe generate cu registre de deplasare.

2.3.1 Codurile Walsh Hadamard

Secvenţele Walsh Hadamard au avantajul de a fi ortogonale, şi în acest fel se pot elimina interferenţele de multi-acces. Există totuşi şi câteva neajunsuri:

- Codurile nu au un singur maxim al autocorelaţiei.
- Împrăştierea nu este pe întreaga lăţime de bandă, energia fiind dispersată pe un număr de frecvenţe discrete (vezi Figura 2.2).

- Deşi intercorelarea de-a lungul întregii secvențe este zero, intercorelarea calculată pentru o porțiune din secvențe este nenulă. Consecința este că se pierde avantajul de a folosi coduri ortogonale.
- Ortogonalitatea este de asemenea afectată de efectele canalui asupra semnalului ca de exemplu apariția căilor multiple. În sistemele practice este utilizată egalizarea pentru a recupera semnalul inițial.

Aceste neajunsuri fac ca secvențele Walsh să nu poată fi utilizate în sistemele noncelulare. Sistemele în care sunt folosite secvențele Walsh sunt sistemele celulare CDMA. Spre exemplu, sistemul IS-95 foloseşte o combinație între o secvență Walsh şi o secvență M pentru a face posibilă sincronizarea.

Secvențele Walsh se obțin prin eşantionarea funcțiilor cu acelaşi nume. Funcțiile Walsh formează un set complet ortonormat de funcții rectangulare. Există trei grupuri de funcții Walsh care diferă între ele numai prin modul de ordonare:

– ordonare pe bază de secvență (Walsh);
– ordonare naturală (Hadamard);
– ordonare diadică (Paley).

Secvențele Walsh sunt linii ale matricii Hadamard, H_k–matrice pătrată de ordin k, ce se poate obține recursiv:

$$H_1 = [+1] \qquad H_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} \qquad H_3 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix}$$

$$H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix} \tag{2.11}$$

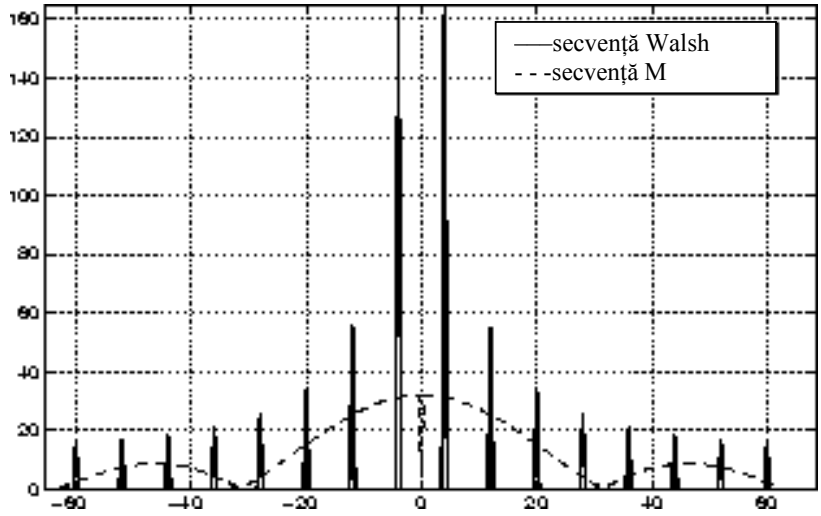


Figura 2.2 Comparație în domeniul frecvență între o secvență Walsh şi una M, [9]

Coduri ortogonale de lungime variabilă

Unul dintre obiectivele CDMA este îmbunătățirea eficienței comunicațiilor multimedia. Astfel, datorită varietății serviciilor, ratele de bit necesare variază de la valori mici la foarte mari. Deoarece banda semnalelor de împrăştiere este aceeaşi pentru toți utilizatorii, diversele rate de transmisie necesită factori de împrăştiere (sau factori expansiune de bandă, β_e) diferiți în canalele fizice. Considerăm că fiecare bit al celei mai mici rate de bit (R_{min}) este împrăştiat printr-un cod de lungime $N=2^n$. Deoarece durata bitului pentru rata $2 \cdot R_{min}$ este jumătate din durata bitului ratei minime, este necesar un cod de împrăştiere de lungime $N/2=2^{n-1}$. În general, pentru o rată $2^k \cdot R_{min}$ este necesar un cod de lungime 2^{n-k}. Domeniul lungimii codului depinde aşadar de ratele de bit, maximă şi minimă suportate de sistem, şi de lățimea de bandă. O metodă de obținere a codurilor ortogonale de lungime variabilă, ce păstrează ortogonalitatea între diferitele rate şi factori de împrăştiere, bazată pe transformata Hadamard modificată, este prezentată în [50].

Fie C_N o matrice de mărime NxN ce denotă setul de N coduri binare de împrăştiere de N chipuri lungime, $\{C_N(n)\}_{n=1,\ldots,N}$, unde $C_N(n)$ este vectorul linie de $N=2^n$ elemente; Această matrice este generată din $C_{N/2}$ astfel:

$$C_N = \begin{bmatrix} C_N(1) \\ C_N(2) \\ C_N(3) \\ \vdots \\ C_N(N-1) \\ C_N(N) \end{bmatrix} = \begin{bmatrix} C_{N/2}(1) \cdot C_{N/2}(1) \\ C_{N/2}(1) \cdot \overline{C_{N/2}(1)} \\ C_{N/2}(2) \cdot C_{N/2}(2) \\ \vdots \\ C_{N/2}(N/2) \cdot C_{N/2}(N/2) \\ C_{N/2}(N/2) \cdot \overline{C_{N/2}(N/2)} \end{bmatrix} \qquad (2.12)$$

Aşadar, aceste coduri ortogonale de lungime variabilă pot fi generate recursiv utilizând structura arborescentă prezentată în Figgura 2.3. Pornind de la $C_1(1) = 1$, se generează, la nivelul k, un set de 2^k coduri de împrăştiere de lungime 2^k chipuri.

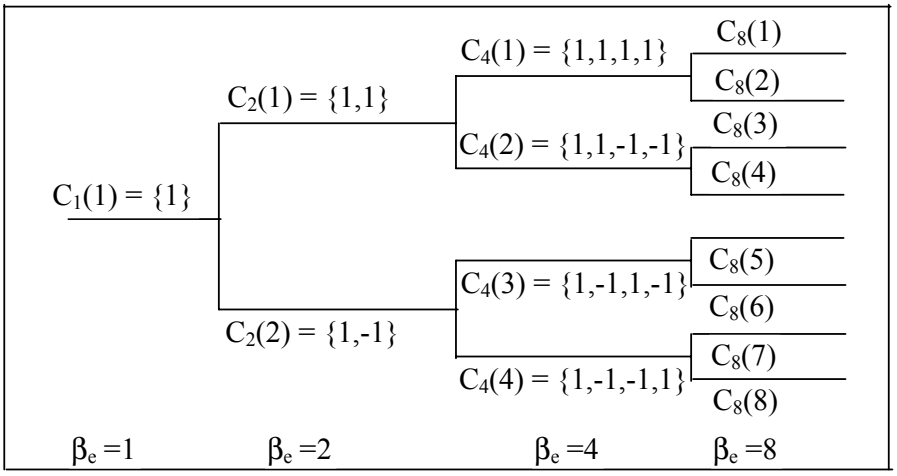


Figura 2.3 Arborele utilizat la generarea codurilor ortogonale de lungime variabilă

Codurile din acelaşi nivel constituie setul de funcţii Walsh şi, ca atare, sunt ortogonale. Mai mult, oricare două coduri ale diferitelor niveluri sunt de asemenea ortogonale, exceptând cazul în care unul dintre cele două coduri este părinte al celuilalt cod: de exemplu $C_{16}(2)$, $C_8(1)$, $C_4(1)$ şi $C_2(1)$ sunt părinţi pentru $C_{32}(3)$ şi, ca atare, nu sunt ortogonali cu acesta. Cu alte cuvinte, un cod poate fi utilizat într-un canal dacă şi numai dacă nu este utilizat nici un alt cod al căii codului specificat sau al altei căi ce conduce la aceeaşi rădăcină. Aceasta înseamnă că numărul codurilor disponibile nu este fix, ci depinde de ratele de bit şi factorii de împrăştiere ai canalului fizic. Această restricţie este impusă cu scopul de a menţine ortogonalitatea.

2.3.2 Secvenţe M

Secvenţele M, [11], sunt secvenţe generate prin registru de deplasare (secvenţe MLSR). Secvenţele M nu sunt ortogonale, dar au un maxim îngust al autocorelaţiei. Aşa cum s-a arătat deja, aceste coduri pot fi generate utilizând un registru de deplasare. Obţinerea de secvenţe de secvenţe M este condiţionată de utilizarea conexiunilor buclei de reacţie în acord cu coeficienţii unui polinom primitiv de grad m. În Anexa A sunt prezentate polinoamele primitive până la gradul m=13. Lungimea secvenţei M generate este egală cu 2^m-1. Numărul de coduri posibile este dat de numărul de polinoame primitive existente pentru respectivul m, de asemenea prezentat în Anexa A. Aşa cum s-a arătat în paragraful 2.2.2, secvenţele M au câteva proprietăţi speciale:

- Secvenţele M sunt cvasi-echilibrate: numărul de 1 este cu 1 mai mare decât numărul de 0.
- Spectrul unei secvenţe M are o anvelopă de forma „sinc^2”. În Figura 2.2 este prezentat, [9], spectrul unei secvenţe Walsh de lungime 64 şi al unei secvenţe M de lungime 63. Ambele secvenţe sunt (aproape) de aceeaşi putere. Figura arată că aplicarea unei secvenţe M distribuie mai bine puterea pe întregul domeniu de frecvenţe decât secvenţa Walsh.
- Proprietatea „deplasează-şi-adună” poate fi formulată după cum urmează:

$$T^k u = T^i u + T^j u. \qquad (2.13a)$$

Aici u este o secvenţă M. Combinând două deplasări ale acestei secvenţe (deplasări relative i şi j) obţinem secvenţa M, cu o altă deplasare relativă.

- Funcţia de autocorelare este bivalentă:

$$R_u(\tau) = \begin{cases} N & \tau = k \cdot N \\ -1 & \tau \neq k \cdot N \end{cases} \qquad (2.13b)$$

unde k este o valoare întreagă, iar τ este deplasarea relativă.

- Nu există o formulă generală pentru intercorelarea a două secvenţe M, putând fi formulate doar câteva reguli.
- O aşa numită „pereche preferată” este o combinaţie de secvenţe M pentru care intercorelarea duce la doar 3 valori diferite: -1, $-2^{[(m+1)/2]}$ şi $2^{[(m+1)/2]}$ –2. Nu există perechi preferate pentru secvenţele M cu o lungime de $4 \cdot k$ unde k este un întreg.

2.3.3 Secvenţe Gold şi Kasami

În unele aplicaţii proprietăţile intercorelaţiei secvenţelor PN sunt la fel de importante ca şi proprietăţile de autocorelaţie. Spre exemplu, în CDMA fiecărui utilizator îi este atribuită

o secvenţă PN particulară. Ideal, secvenţele PN dintre utilizatori ar fi mutual ortogonale însă, secvenţele PN utilizate în practică prezintă o oarecare corelaţie.

Funcţia de intercorelaţie periodică dintre orice perechi de secvenţe M de aceeaşi perioadă poate avea maximuri mari. Tabelul 2.4 prezintă valorile maximurilor Φ_{max} pentru intercorelaţia dintre perechi de secvenţe M pentru $3 \le m \le 12$. Este prezentat, de asemenea, în Tabelul 2.4 numărul de secvenţe M de lungime $n = 2^m-1$ pentru $3 \le m \le 12$. Aşa cum se poate vedea din acest tabel, numărul de secvenţe de lungime n creşte rapid cu m. De asemenea observăm că, pentru multe secvenţe, mărimea maximului Φ_{max} a intercorelaţiei este un procent mare din valoarea funcţiei de autocorelaţie, Φ_0.

Astfel de valori mari ale intercorelaţiei sunt inacceptabile în CDMA. Chiar dacă este posibil să se selecteze un mic subset de secvenţe M care au valori relativ mici ale maximului intercorelaţiei, numărul de secvenţe al subsetului este uzul prea mic pentru aplicaţii CDMA.

Secvenţe PN cu proprietăţi ale intercorelaţiei periodice superioare secvenţelor M au fost date de Gold (1967, 1968) şi Kasami (1966). Ele sunt derivate din secvenţele M aşa cum se va descrie în cele ce urmează.

Gold şi Kasami au dovedit că există perechi de secvenţe M, de lungime n, ce dau trei valori pentru funcţia de intercorelaţie {-1, -t(m), t(m)-2} unde

$$t(m) = \begin{cases} 2^{(m+1)/2} + 1 & m = \text{impar} \\ 2^{(m+2)/2} + 1 & m = \text{par} \end{cases} \qquad (2.14)$$

De exemplu, dacă m = 10, $t(10) = 2^6 + 1 = 65$ iar cele trei posibile valori pentru funcţia de intercorelaţie sunt {-1, -65, 63}. Astfel maximul intercorelaţiei pentru perechile de secvenţe M este 65 în vreme ce maximul familiei de 60 de secvenţe posibile generate printr-un registru de deplasare cu 10 stări, cu diferite conexiuni de reacţie, este $\Phi_{max} = 383$, de aproape 6 ori diferenţa în valoarea maximă.

Două secvenţe M, generate de $p_1(x)$ şi $p_2(x)$, polinoame primitive de grad m ce constituie polinoamele minimale pentru elementele α şi $\alpha^{t(m)}$ ale câmpului Galois $GF(2^m)$, a căror funcţie de intercorelaţie ia trei valori posibile {-1, -t(m), t(m)-2} se numesc *secvenţe preferate*. În Anexa B sunt prezentate secvenţele preferate posibile pentru $m \le 12$.

m	**$n=2^m-1$**	**Numărul de secvenţe M**	**Maximul intercorelaţiei Φ_{max}**	**Φ_{max}/Φ_0**	**t(m)**	**$T(m)/\Phi_0$**
3	7	2	5	0,71	5	0,71
4	15	2	9	0,6	9	0,6
5	31	6	11	0,35	9	0,29
6	63	6	23	0,36	17	0,27
7	127	18	41	0,32	17	0,13
8	255	16	95	0,37	33	0,13
9	511	48	113	0,22	33	0,06
10	1023	60	383	0,37	65	0,06
11	2047	176	287	0,14	65	0,03
12	4095	144	1407	0,34	129	0,03

Tabelul 2.4 Vârful intercorelaţiei secvenţelor M şi Gold

Pentru o pereche de secvenţe preferate, să spunem **a** = [a_1 a_2 … a_n] şi **b** = [b_1 b_2 … b_n], construim un set de secvenţe de lungime n prin sumarea modulo 2 a secvenţei **a** cu versiunea permutată ciclic a secvenţei **b**, sau viceversa. Atunci obţinem n noi secvenţe periodice[1] de perioadă $n=2^m-1$. Putem include, de asemenea, secvenţele originale **a** şi **b** şi, astfel, avem în total n+2 secvenţe. Cele n+2 secvenţe construite în această manieră se numesc ***secvenţe Gold***.

Cu excepţia secvenţelor **a** şi **b**, secvenţele Gold nu sunt secvenţe M (nu au perioadă maximă). Prin urmare funcţiile lor de autocorelaţie nu au doar 2 valori. Gold (1968) a arătat că funcţiile de autocorelaţie pentru orice pereche din setul de n + 2 secvenţe Gold are trei valori posibile {-1, -t(m), t(m)-2}, unde t(m) este dat prin (2.14). Similar, maximul secund al funcţiei de autocorelaţie pentru o secvenţă Gold ia valori din setul {-1, -t(m), t(m)-2}. Rezultă că valorile vârfului (secund) al funcţiei de autocorelaţie este mărginit superior de t(m).

Valorile maximului secund al autocorelaţiei şi maximului intercorelaţiei, adică t(m), pentru secvenţele Gold sunt listate în Tabelul 2.3. Sunt de asemenea listate valorile normalizate pentru $\Phi(0)$.

Este important de comparat valoarea maximului intercorelaţiei pentru secvenţele Gold cu marginea inferioară a intercorelaţiei dintre orice pereche de secvenţe binare de perioadă n. Marginea inferioară dedusă de Welch (1974) pentru Φ_{max} este:

$$\Phi_{max} \geq n \cdot \sqrt{\frac{M-1}{M \cdot n-1}} \qquad (2.15)$$

care, pentru valori mari ale lui n, este bine aproximată ca fiind $\sqrt{n}$. Pentru secvenţele Gold, $n= 2^m -1$ şi, de aceea, marginea inferioară este $\Phi_{max} \approx 2^{m/2}$. Marginea inferioară Welch este mai mică cu $\sqrt{2}$ pentru m impar şi cu 2 pentru m par decât $\Phi_{max} = t(m)$, pentru secvenţe Gold.

O procedură similară cu cea folosită pentru generarea secvenţelor Gold va genera setul de $N = 2^{m/2}$ secvenţe binare de perioadă $n = 2^m -1$, unde m este par. În această procedură, pornim cu o secvenţă M, **a**, şi formăm o secvenţă binară **b** luând tot al $2^{m/2}+1$ –lea bit din **a**. Adică secvenţa **b** este formată decimând **a** prin $2^{m/2}+1$. Se poate verifica că rezultatul, **b,** este o secvenţă periodică cu perioada $2^{m/2}-1$. De exemplu, dacă m=10, perioada lui **a** este n = 1023 iar perioada lui **b** este 31. Astfel, dacă observăm 1023 biţi pentru secvenţa **b**, vom vedea 33 de repetiţii de 31 de secvenţe de biţi. Luând acum $n = 2^m-1$ biţi pentru secvenţele **a** şi **b**, formăm un nou set de secvenţe prin adunarea, modulo 2, a lui **a** cu **b** şi tuturor celor $2^{m/2}-2$ secvenţe binare de lungime $n = 2^m-1$. Acestea sunt denumite ***secvenţe Kasami***. Funcţiile de autocorelaţie şi intercorelaţie pentru aceste secvenţe iau valori din tripletul {-1, $-(2^{m/2}+1)$, $2^{m/2}-1$ }. Astfel, valorile maximului intercorelaţiei pentru orice pereche de secvenţe ale setului este:

$$\Phi_{max} = 2^{m/2} + 1 \qquad (2.16)$$

Aceste valori pentru Φ_{max} satisfac limitarea inferioară Welch pentru un set de $2^{m/2}$ secvenţe de lungime $n = 2^m -1$. Din această cauză secvenţele Kasami sunt optimale.

2.3.4 Concluzii. Soluţii alternative

Alegerea codurilor (secvenţelor) PN depinde de aplicaţie. Fiecare dintre codurile PN prezentate oferă anumite avantaje. Secvenţele Walsh-Hadamard fiind ortogonale, vor fi utile

[1] O metodă de generare a celor n secvenţe este printr-un registru de deplasare de lungime 2·m cu conexiunile reacţiei specificate prin polinomul $h(p) = g_1(p) \cdot g_2(p)$, unde $g_1(p)$ şi $g_2(p)$ sunt polinoamele care specifică conexiunile reacţiei registrelor de deplasare cu m stări ce generează secvenţele M, **a** şi **b**.

în sistemele de transmisie cu spectru împrăştiat cu CDMA unde se poate realiza sincronizarea tuturor convorbirilor. Secvenţele M sunt uşor de generat, însă nu au proprietăţi bune de intercorelare. Nu vor putea fi folosite în sisteme multiacces. Astfel de proprietăţi de intercorelaţie au secvenţele Gold sau Kasami. Însă, în multe sisteme CDMA practice durata bitului nu acoperă decât o fracţie din secvenţa periodică. Într-un astfel de caz, este importantă intercorelaţia periodică parţială dintre două secvenţe. În [23] este prezentat modul de construcţie a unui set de secvenţe Gold optimal din punct de vedere al intercorelaţiei parţiale.

Secvenţe PN pot fi privite ca şi cuvinte de cod protector, ceea ce le face utile la recepţie. Astfel, codurile Hadamard, aşa cum se va prezenta în Cap.4, sunt cuvinte de cod Reed-Muller, iar secvenţele M sunt de fapt cuvinte de cod ciclic, fiind generate acelaşi procedeu (Cap.5). Acest fapt permite utilizarea tehnicilor de decodare ale codurilor protectoare şi pentru secvenţe PN, [20].

Selecţia secvenţelor preferate devine o problemă dificilă pentru m mare. În [38] este prezentată o metodă de aflare a unor astfel de perechi de secvenţe. Spre exemplu două secvenţe preferate pentru m =127 sunt:

$$x^{127} + x + 1,$$

şi

$$x^{127} + x^{113} + x^{99} + x^{98} + x^{88} + x^{85} + x^{84} + x^{80} + x^{71} + x^{69} + x^{59} + x^{57} + \\ + x^{55} + x^{51} + x^{44} + x^{43} + x^{42} + x^{41} + x^{40} + x^{37} + x^{35} + x^{32} + x^{29} + x^{28} + \\ + x^{26} + x^{25} + x^{24} + x^{23} + x^{20} + x^{19} + x^{11} + x^{10} + x^{5} + x + 1$$

În sistemele cu spectru împrăştiat sunt utilizate şi secvenţe neperiodice, numite haotice. Ele sunt generate de sisteme neliniare. Aceste secvenţe haotice oferă, în comparaţie cu primele, avantajul unor mai bune proprietăţi aleatoare (de zgomot) şi posibilitatea unei sincronizări mai precise la recepţie, [39].

O altă soluţie alternativă secvenţelor „clasice”, o reprezintă secvenţele nebinare. Acestea posedă proprietăţi superioare din punct de vedere al intercorelaţiei, [10].

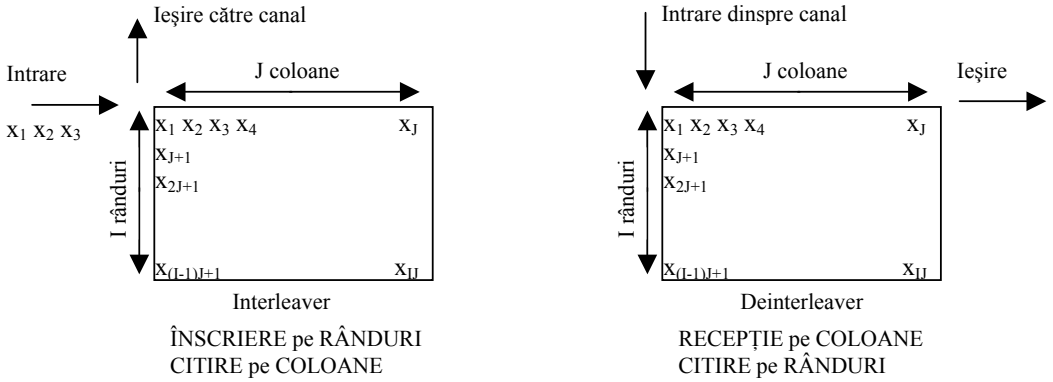
3. Tehnici adiționale de îmbunătățire a performanțelor sistemelor cu spectru împrăştiat

Prin natura sa, tehnica împrăştierii spectrului oferă o resursă timp-frecvență considerabil mai mare decât cea necesară transmisiei informației pentru oricare utilizator în parte. Aceasta se reflectă prin câştigul de procesare, G_p, mare, sau prin raportul W/R (lățime de bandă–rată de transmisie) mare. Acest exces de redundanță poate fi exploatat pentru a îmbunătăți performanța, fără a compromite alte avantaje ale câştigului mare de procesare [1]. În acest capitol sunt prezentate două tehnici de procesare ce aduc îmbunătățiri sistemului de transmisie: întrețeserea (interleavingul) şi codarea pentru corecția erorilor.

3.1 Întrețeserea

Avantajul imediat al excesului de redundanță este că determină independența ieşirilor canalului. Cu cât e mai mare numărul de componente de cale independente disponibile în prezența fadingului, cu atât e mai bună performanța. (vezi paragraful 1.2 punctul g)

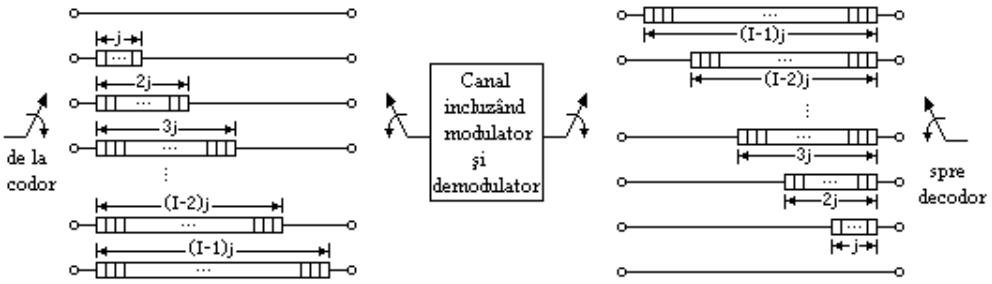
Dispunem de L căi şi de N chipuri per simbol. Problema e că chipurile succesive nu pot fi privite ca independente. Este posibil, însă, să reordonăm chipurile astfel încât cele N chipuri ce provin de la acelaşi simbol să nu mai fie transmise succesiv. Mai exact, ele sunt transmise la intervale suficient de largi astfel că fading-ul va conduce la amplitudine şi fază



Secvența întrețesută: $x_1, x_{J+1}, x_{2J+1}, \ldots, x_{(I-1)J+1}, x_2, x_{J+2}$.Oricare două simboluri aflate inițial la mai puțin de J poziții unul de celălalt vor fi depărtate la cel puțin I poziții.

Figura 3.1 Întrețesere/deîntrețesere bloc.

independente pentru fiecare dintre cele N chip-uri[1]. Procedeul de reordonare necesar pentru a dobândi diversitate temporală este numit întreţesere (*interleaving*) şi poate fi realizat în mai multe feluri. Două procedee sunt întreţeserea „bloc" şi întreţeserea „convoluţională" (Figura 3.1 şi 3.2). După recepţie, chipurile demodulate de la ieşire sunt din nou reordonate pentru a le pune la loc în ordinea originală prin procedeul invers numit deîntreţesere (*deinterleaving)*. Acest proces evident introduce întârziere între generarea datelor digitale şi livrarea lor la utilizatorul receptor. Întârzierea produsă prin întreţeserea bloc este de aproximativ 2·I·J·T (T este durata unui simbol). Aşa cum se poate vedea din cele două figuri, întârzierea produsă de întreţeserea convoluţională este aproximativ jumătate din întârzierea produsă de întreţeserea bloc: J·(I–1)·T, [5].



Ordinea la intrarea interliverului şi la ieşirea deinterliverului:

$\ldots\ x_i, x_{i+1}, x_{i+2}, x_{i+3}, \ldots$

Ordinea la ieşirea interliverului şi la intrarea deinterliverului:

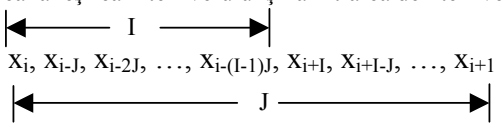


Figura 3.2 Întreţesere/deîntreţesere convoluţională

Aceasta este doar un alt exemplu al faptului cunoscut că, în canalele cu fading, diversitatea poate îmbunătăţi considerabil performanţa. Această diversitate poate fi realizată prin separare spaţială, cu antene multiple sau prin căi multiple produse natural, sau prin separare temporală cu ajutorul procesului de întreţesere descris anterior. Preţul ultimei metode este întârzierea. Cu cât procesul de fading este mai lent, cu atât este necesară o întreţesere pe o întindere mai mare şi de aceea o întârziere mai mare pentru realizarea independenţei prin diversitate temporală.

[1] Întreţeserea poate fi (şi uzual este) făcută nu pe chipuri individuale, ci pe sub-secvenţe de N chipuri, înainte de împrăştierea pseudo-aleatoare. Aceasta reduce necesarul de memorie, deoarece toate chipurile din fiecare sub-secvenţă, care au fost ţinute apropiate după întreţesere, au acelaşi semn. Mai mult, codând, sub-secvenţa constă în general dintr-un cuvânt de cod. Întreţeserea simbolurilor codate conferă mai mare avantaj fără memorie excesivă şi necesităţi de procesare pentru întreţeserea de chip.

Întreţeserea este utilizată şi în combinaţie cu concatenarea codurilor, aşa cum se va prezenta în paragraful 3.3.3. Performanţele codurilor respective depind în mod evident şi de dispozitivul de întreţesere utilizat. Din nefericire, performanţele întreţeserii depind de caracteristicile canalului încât nu există soluţii universal valabile în ceea ce priveşte dispozitivul de întreţesere, [34].

3.2 Codarea corectoare de erori

Metoda efectiv universală de exploatare a redundanţei este codarea corectoare de erori (FEC–forward error-correcting). FEC îmbunătăţeşte performanţa pentru canalele cu amplitudine şi fază fixe la fel de bine ca şi pentru canalele cu fading. În capitolele următoare vor fi prezentate câteva dintre cele mai utilizare coduri în sistemele cu spectru împrăştiat.

3.2.1 Decodare soft / hard

Indiferent de codul utilizat, structura unui receptor conţine un bloc de decizie. Acest bloc reprezintă un comparator ce compară doi termeni pentru a hotărâ valoarea fiecărui bit de informaţie din secvenţa recepţionată:

$$\Lambda_i \underset{D_0}{\overset{D_1}{\gtrless}} \mu \; ; \qquad (3.1)$$

Dacă primul termen, Λ_i, numit raport de plauzibilitate, este de valoare superioară celui de-al doilea, μ, numit pragul deciziei, atunci valoarea bitului în cauză este "1". În mod curent Λ_i depinde de semnalul recepţionat şi este un raport construit pentru fiecare simbol binar în parte, iar μ este constant pentru o aplicaţie dată.

Dacă decizia se face înainte de decodare, decodarea se numeşte **hard**. Dacă, vice versa, decizia se execută după decodare, decodarea se numeşte **soft**. În acest din urmă caz, blocul de decizie este parte din cecodor (se află la finele acestuia).

În cazul decodării hard intrarea decodorului este o secvenţă binară. Decodorul o analizează şi, eventual, constatând prezenţa erorilor, corectează inversând valoarea binară a biţilor presupuşi eronaţi.

În cazul decodării soft intrarea decodorului este un semnal numeric multinivel, rezultat prin eşantionarea (eventual şi cuantizarea valorilor eşantioanelor) semnalului recepţionat, şi/sau, funcţie de aplicaţie, prin demodulare. Analizând semnalul numeric de la intrarea sa, decodorul construieşte pentru fiecare bit termenul Λ_i şi-l compară cu pragul deciziei, μ (execută decizia). Utilizarea decodării soft creşte câştigul de codare.

Decodarea soft poate fi utilizată în combinaţie cu concatenarea codurilor.

3.2.2 Câştigul de codare

Câştigul de codare, pentru o aplicaţie dată, se defineşte ca şi raportul între puterea semnalui, P_s, necesară pentru a se obţine o anumită rată, impusă, a erorii, pentru transmisia necodată şi puterea semnalului, P_{sc}, necesară pentru a obţine aceeaşi rată a erorii în cazul transmisiei codate:

$$G_c = 10 \cdot \lg (P_s/P_{sc}) \quad [\text{dB}]. \qquad (3.2)$$

Presupunând acelaşi nivel de zgomot (în cazurile "codat" şi "necodat"), rezultă că G_c se poate defini şi ca diferența între rapoartele semnal per zgomot necesare obținerii ratei de eroare impuse, în cele două cazuri, necodat şi codat:

$$G_c = 10\cdot\lg\left(\frac{P_s}{N_0}\cdot\frac{N_0}{P_{sc}}\right) = 10\cdot\lg\left(\frac{P_s}{N_0}\right) - 10\cdot\lg\left(\frac{P_{sc}}{N_0}\right) = \xi_s - \xi_{sc} \qquad (3.3)$$

3.2.3 Criteriul MAP

Conform criteriul MAP (**M**aximum **Ap**osteriori), prin procesul de decodare se caută acea secvență emisibilă v_k, ce maximizează probabilitatea condiționată aposteriori $p(v_j/r)$, peste toate secvențele emisibile $\{v_j\}_{j=1\div N}$.

Indiferent de metoda de decodare propriu-zisă, algoritmul decodării urmăreşte, principial, aflarea acelei secvențe de cod (cuvânt de cod), v_k, din mulțimea secvențelor posibil a fi emise, $V = \{v_j\}_{j=1\div N}$, ce maximizează probabilitatea aposteriori:

$$P(k) = p(v_k/r) \qquad (3.4)$$

unde r este secvența recepționată curentă.

Cu alte cuvinte, criteriul de selecție MAP spune astfel: „dacă s-a recepționat r, alege drept cuvânt emis pe v_k, unde

$$p(v_k/r) > p(v_j/r) \qquad \forall\ v_j = \text{cuvânt emisibil } (\in V),\ j \neq k" \qquad (3.5)$$

Dacă admitem ipoteza independenței transmiterii biților (canal fără memorie), şi în plus $p_o < ½$ (probabilitatea ca un bit să fie eronat este mai mică decât probabilitatea ca el să nu fie eronat), atunci selecția, conform cu (3.5), este echivalentă cu selecția după distanța Hamming minimă, iar criteriul MAP devine: „dacă s-a recepționat r, alege drept cuvânt emis pe v_k, unde

$$d(v_k,r) < d(v_j,r) \qquad \forall\ v_j \in V",\ j \neq k" \qquad (3.6)$$

Un decodor ce face selecția cuvântului emisibil conform criteriului MAP se numeşte *optimal*.

Cu toate că, aparent, un decodor suboptimal este mai puțin performant decât unul optimal, totuşi cercetările ultimilor ani prezintă decodoare suboptimale mai performante (din punct de vedere al ratei erorii) decât cele optimale, [27].

Explicația este una simplă şi uşor de înțeles. Fie spre exemplu un cod oarecare C, alcătuit din cuvinte de lungime n biți, din care k sunt de informație şi m de control. Utilizând principiul concatenării codurilor, descris în paragraful următor, construim codul produs C^2 = CxC , în care secvența codată are o alcătuire conform Figurii 3.3a. Un decodor optimal al codului C pur trebuie să afle un v_k dintre 2^k cuvinte emisibile, v_j. Un decodor optimal al codului produs C^2 trebuie să afle o „matrice" de cuvinte emisibilă dintre $2^{k\cdot k}$, adică un volum de calcul de 2^k ori mai mare decât în primul caz. Datorită volumului imens de căutare, se poate renunța la decodarea optimală pentru codul C^2. Chiar dacă decodarea lui C^2 nu mai este optimală, este posibil de a obține performanțe superioare cu C^2 față de C pur. Adică, într-o situație concretă, din n cuvinte orizontale pentru C, este posibil, chiar cu decodarea optimală, datorită „concentrării" erorilor, să existe unul eronat, dar prin decodarea suboptimală a lui C^2, datorită „aportului" codării pe verticală să se găsească n cuvinte corecte.

Decodările iterative sunt, în general suboptimale.

3.3 Concatenarea

3.3.1 Concatenarea codurilor

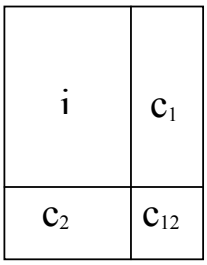
Concatenarea codurilor are drept scop utilizarea a două sau mai multe coduri simple în vederea obținerii unor performanțe superioare (referitor la câştigul de codare).

Codurile concatenate pot să lucreze separat, independent unul de celelalte, sau să coopereze la decodare, furnizându-şi reciproc informație despre secvența decodată. În acest de-al doilea caz, decodarea este soft. Fiecare decodor construieşte un set de rapoarte de plauzibilitate $\Lambda_j = [\Lambda_{ij}]_{i=1,N}$, cu j =1÷C, (unde C reprezintă numărul de decodoare). Aceste seturi de rapoarte de plauzibilitate reprezintă "opinia" fiecărui decodor despre secvența recepționată. Fiecare set Λ_j, al fiecărui decodor, este o sumă de doi termeni:

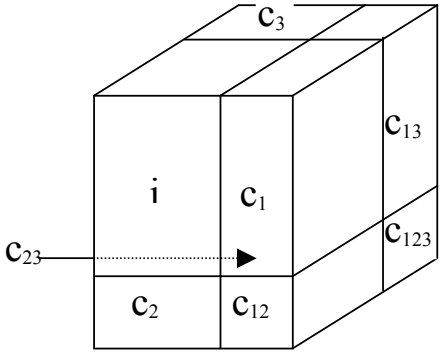
$$\Lambda_j = \Lambda_{jp} + \Lambda_{je}. \qquad (3.7)$$

unde Λ_{je} se numeşte informație extrinsecă şi este "opinia" decodorului construită pe baza unor observații independente de ale celorlalte decodoare. Această independență a observațiilor este posibilă dacă biții destinați separat fiecărui decodor în parte, sunt transmişi în perioade de timp diferite. Termenul Λ_{jp} este construit pe baza observațiilor făcute asupra biților cu destinație comună.

Secvența de informație furnizată codoarelor, în vederea obținerii unor „puncte de vedere" cât mai necorelate ale decodoarelor, este furnizată fiecărui codor în parte în altă ordine, prin întrețesere. Astfel, dacă într-o zonă a secvenței transmise s-au produs erori, aceasta alterează „opinia" doar a unui decodor, celelalte aflându-se în altă zonă a secvenței transmise.



a) două coduri;



b) trei coduri;

Figura 3.3 Structura secvenței codate pentru un produs de coduri

3.3.2 Coduri bloc produs

Două sau mai multe coduri bloc simple pot fi utilizate pentru a obține un cod produs. În Figura 3.3 este prezentată structura secvenței codate, pentru cazul a două, respectiv trei coduri concatenate.

Decodarea soft, iterativă, a codului produs de două coduri se poate face după următorul algoritm,[3]:

1. Se decodează liniile din $[r_1]$ –matricea "semnal recepționat"– utilizând decodarea soft. Fie v^+ cuvântul selectat pentru o linie particulară, unde simbolul $v_k = 1$.

2. Odată v^+ selectat, se determină coeficientul de plauzibilitate pentru fiecare simbol v_k din v^+. Aceasta constituie ieşirea soft pentru decodor ce poate fi utilizată pentru iterație. În scopul de a afla această informație, se caută cuvântul de cod v^- aflat la distanța Euclidiană minimă de vectorul linie recepționat astfel încât $v_k = -1$. Se construieşte raportul de plauzibilitate pentru v_k. Repetând procedura pentru toate simbolurile din toate liniile, se obține o matrice de estimatori normalizați, $[r_1']$.

3. Se generează ieşirea primului decodor astfel:

$$[w_2] = [r_1'] - [r_1] \tag{3.8}$$

Aici $[w_2]$ poate fi privită ca o informație extrinsecă adițională despre $[r_1]$.

4. Cel de-al doilea decodor execută aceleaşi operații pe coloane, utilizând intrarea:

$$[r_2] = [r_1] + \alpha_2 \cdot [w_2], \tag{3.9}$$

unde α_i este o constantă de ponderare ce reduce influența lui $[w_i]$ la fiecare iterație. Aici ia sfârşit un ciclu complet de decodare.

5. Se repetă paşii 1.÷ 4. un număr de iterații propus.

3.3.3 Coduri convoluționale concatenate (CCC)

Decodarea iterativă a codurilor convoluționale concatenate a fost introdusă de Berrou ş.a. în 1993, [47]. În respectiva lucrare se prezintă o structură de două CC-uri recursive, sistematice, concatenate paralel, de memorie 4, rezultând o schemă generală asemenea celei din Figura 3.4a.

Funcționarea decodorului urmăreşte conceptul prezentat în paragraful 3.3.1 şi poate fi sintetizată în următorul algoritm:

1. –se recepționează un bloc de $N = 3 \cdot k$ biți; dintre aceştia $2 \cdot k$ sunt utilizați de DEC_1 iar $2 \cdot k$ de DEC_2 (tot al treilea bit este destinat ambelor decodoare).
2. –DEC_1 calculează logaritmul raportului de plauzibilitate pentru fiecare din cei k biți de informație:

$$\mathbf{\Lambda}(u_i) = \mathbf{\Lambda}_s(u_i) + \mathbf{\Lambda}_e(u_i) \tag{3.10}$$

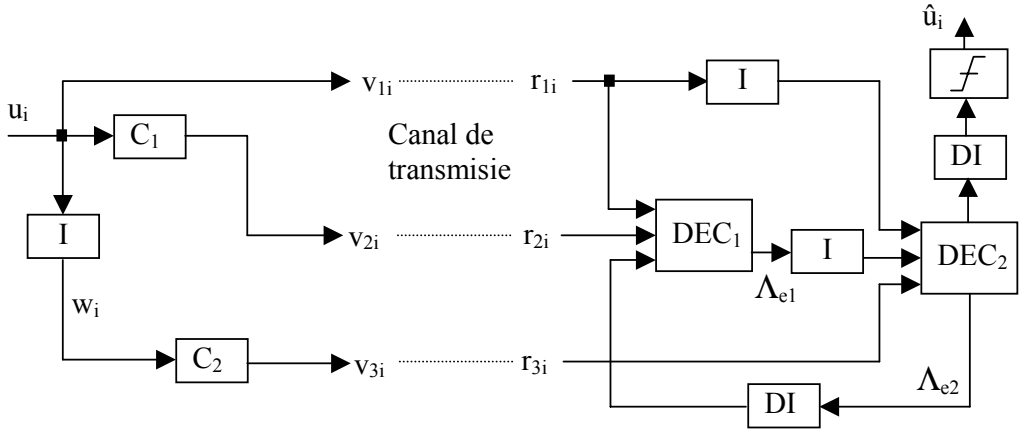
unde

$$\mathbf{\Lambda}_s(u_i) = \log\left[\frac{P(r_{1i} / u_i = 1)}{P(r_{1i} / u_i = 0)}\right] \tag{3.11}$$

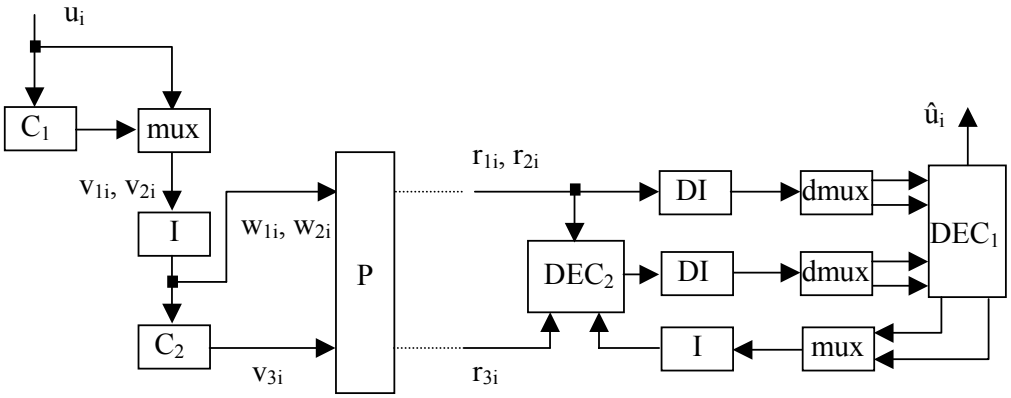
furnizează informație despre componenta sistematică a codului, iar:

$$\mathbf{\Lambda}_e(u_i) = \log\left[\frac{\sum_m \sum_n \sum_{l=0}^{1} \gamma_1(r_{2i}, n, m) \cdot \alpha_{i-1}^{l}(n) \cdot \beta_i(m)}{\sum_m \sum_n \sum_{l=0}^{1} \gamma_0(r_{2i}, n, m) \cdot \alpha_{i-1}^{l}(n) \cdot \beta_i(m)}\right] \tag{3.12}$$

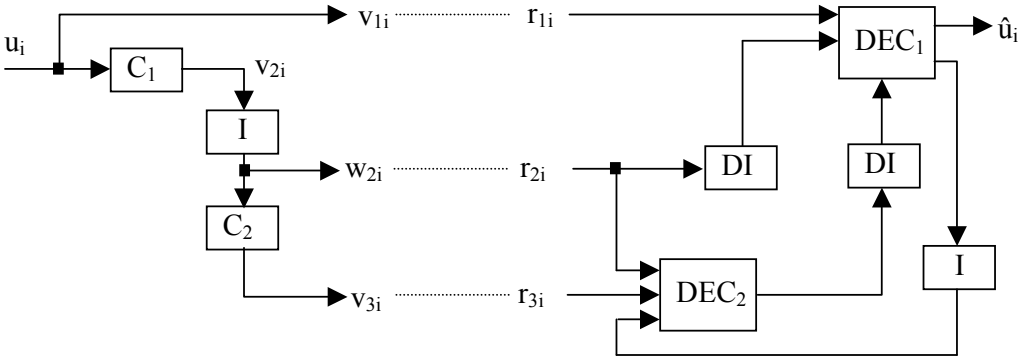
reprezintă informația extrinsecă sau adițională.



a) Cod convoluțional concatenat paralel;



b) Cod convolutional concatenat serial;



c) Cod convolutional concatenat hibrid;

Figura 3.4 Structuri „de bază” pentru concatenarea a două coduri convoluționale: a) structură paralelă (turbo-cod); b) structură serială; c) structură hibridă.

Semnificaţia notaţiilor din relaţia (3.12) este:

$\alpha^{l}_{i-1}(n)$= *coeficient la reacţia „înainte”* = probabilitatea ca la momentul i –1 starea codorului să fie n, bitul de intrare să fie de valoare l şi secvenţa codată până în acest moment să fie cea recepţionată, $\{r_{1j}\}_{j=1\div i}$;

$\beta_i(m)$ = *coeficient la reacţia „înapoi”* = probabilitatea ca secvenţa codată de la momentul i+1 la fine (momentul N) să fie cea recepţionată, $\{r_{1j}\}_{j=i+1\div N}$, dacă starea actuală este m;

$\gamma_k(r_{2i}, n, m)$ = probabilitatea ca bitul de intrare să fie k, starea actuală a codorului să fie m şi bitul codat să fie r_{2i}, dacă starea anterioară a fost n.

3. –DEC_2 calculează acelaşi raport de plauzibilitate utilizând în plus şi informaţia extrinsecă de la DEC_1, furnizându-i, la rândul său o informaţie extrinsecă, în eventalitatea unei noi iteraţii. Aici ia sfârşit un ciclu de iteraţie;

4. –ciclul se poate repeta, cu diferenţa că acum şi DEC_1 dispune de o informaţie extrinsecă provenită de la DEC_2;

5. –după execuţia tuturor iteraţiilor se procedează la construcţia secvenţei decodate pe baza deciziei asupra logaritmului raportului de plauzibilitate, furnizat de exemplu de DEC_2.

În Figura 3.4 sunt prezentate trei structuri de bază pentru concatenarea CC –lor cu decodare iterativă: a) paralel, PCCC –Parallel Concatenated Convolutional Code; b) serial, SCCC –Serial Concatenated Convolutional Code; c) hibrid, HCCC –Hybrid Concatenated Convolutional Code.

Dispozitivele de întreţesere utilizate pentru obţinerea „împrăştierii” secvenţei de informaţie pot fi de tip aleatoriu sau de tip „S”. Primul „amestecă” aleatoriu biţii în interiorul unui bloc de lungime N. Cel de-al doilea tip distanţează oricare doi biţi (vecini în blocul iniţial) la poziţii aflate la o distanţă mai mare de S biţi unul de celălalt. Simulările făcute arată o superioritate a dispozitivului de întreţesere de tip S.

Deşi în schemele de decodare individuale CC-urile nerecursive prezintă performanţe superioare celor recursive, în schemele ce folosesc CCC-uri se dovedeşte mai utilă folosirea codurilor convoluţionale recursive sistematice. Explicaţia este că CC-urile recursive împrăştie mai bine informaţia peste secvenţa codată, dar acest fapt este exploatat doar de CCC-uri.

Cu toate că subiectul utilizării decodării iterative în schemele cu CCC-uri este în plină dezvoltare, se poate afirma că la un RSZ mare SCCC sunt superioare turbocodurilor (PCCC), în schimb la RSZ mic ultimele au o rată a erorii mai bună, [3].

4. Coduri grup

4.1 Coduri grup

Codurile grup sunt denumite astfel deoarece mulțimea cuvintelor de cod formează un grup comutativ față de operația de adunare. Codarea se poate face în spațiul nul al matricii de control, H:

$$H \cdot v^T = 0 \tag{4.1}$$

sau în idealul generat de matricea generatoare G:

$$v = i \cdot G \tag{4.2}$$

Matricea de control H are dimensiunile m x n; m= numărul de biți de control din cuvântul de cod v; n= numărul total de biți; i este secvența de informație iar G are dimensiunile k x n. Liniile matricii G sunt cuvinte de cod, şi totodată reprezintă un set de vectori ce formează o bază a spațiului vectorial dat de mulțimea cuvintelor de cod.

Decodarea codurilor grup presupune calculul corectorului:

$$Z = H \cdot w^T \tag{4.3}$$

unde w = cuvântul recepționat.

În cazul unui cod grup detector de erori, analiza corectorului se rezumă la a verifica dacă Z este sau nu un vector nul. În caz afirmativ se hotărăşte că nu există erori în cuvântul recepționat (concluzie posibil falsă –cazul recepției unui alt cuvânt de cod, diferit de cel emis). În caz contrar, concluzia (100% adevărată –corectorul nu poate fi diferit de zero dacă nu au fost erori) este că în w există erori.

Pentru coduri corectoare algoritmul decodării continuă cu aflarea cuvântului eroare pe baza lui Z. În acest scop între mulțimea cuvintelor eroare corectabile şi mulțimea corectorilor trebuie să existe o corespondență biunivocă. Când codul trebuie să corecteze e erori atunci numărul cuvintelor corectabile este:

$$N_{ec} = C_n^1 + C_n^2 + \ldots + C_n^e . \tag{4.4}$$

Numărul de corectori ce se poate obține, m fiind dat, este:

$$N_c = 2^m - 1 \tag{4.5}$$

deoarece $Z \equiv 0$ nu poate fi utilizat pentru un cuvânt eroare, valoarea zero semnificând cuvânt corect. Trebuie ca:

$$N_c \geq N_{ec} \tag{4.6}$$

Dacă relaţia anterioară este o identitate, atunci codul se numeşte perfect. Singurele coduri perfecte sunt codul Hamming corector de o eroare (cu $n=2^m-1$) şi codul Golay, având n=23 şi m = 11.

Dintre codurile grup utilizate în sistemele cu spectru împrăştiat este codul Reed Muller [37]. Aceasta datorită proprietăţilor de *pseudo–noise* pentru cuvintele codului R-M, cuvinte ce se regăsesc printre secvenţele Walsh.

4.2 Coduri Reed Muller (R-M)

Codurile R-M sunt caracterizate de doi parametri, t (logaritmul numărului de coloane din matricea G) şi r (numărul de secţiuni ai matricii G) , funcţie de care se calculează toţi ceilalţi parametri:

$$\begin{aligned} &n = 2^t && \text{–numărul de biţi dintr-un cuvânt de cod;}\\ &d = 2^{t-r} && \text{–distanţa de cod;}\\ &e = 2^{t-r-1}-1 && \text{–numărul de erori corectabile;}\\ &k = \sum_{i=0}^{r} C_t^i && \text{–numărul de biţi de informaţie.} \end{aligned} \qquad (4.7)$$

Relaţia utilizată pentru codare este (4.2), unde G, pentru t=4 şi r=2, are forma:

$$G = \left[\begin{array}{cccc|l} 1\,1\,1\,1 & 1\,1\,1\,1 & 1\,1\,1\,1 & 1\,1\,1\,1 & L_0 \\ \hdashline 0\,0\,0\,0 & 0\,0\,0\,0 & 1\,1\,1\,1 & 1\,1\,1\,1 & L_4 \\ 0\,0\,0\,0 & 1\,1\,1\,1 & 0\,0\,0\,0 & 1\,1\,1\,1 & L_3 \\ 0\,0\,1\,1 & 0\,0\,1\,1 & 0\,0\,1\,1 & 0\,0\,1\,1 & L_2 \\ 0\,1\,0\,1 & 0\,1\,0\,1 & 0\,1\,0\,1 & 0\,1\,0\,1 & L_1 \\ \hdashline 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 & 1\,1\,1\,1 & L_{43} \\ 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,1\,1 & 0\,0\,1\,1 & L_{42} \\ 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,1\,0\,1 & 0\,1\,0\,1 & L_{41} \\ 0\,0\,0\,0 & 0\,0\,1\,1 & 0\,0\,0\,0 & 0\,0\,1\,1 & L_{32} \\ 0\,0\,0\,0 & 0\,1\,0\,1 & 0\,0\,0\,0 & 0\,1\,0\,1 & L_{31} \\ 0\,0\,0\,1 & 0\,0\,0\,1 & 0\,0\,0\,1 & 0\,0\,0\,1 & L_{21} \end{array}\right] \qquad (4.8)$$

Liniile matricii G formează r +1 subdiviziuni:
–diviziunea 0: o constituie prima linie, notată L_0, cea de pondere n;
–diviziunea 1: este formată din secvenţe Walsh, notate de la L_t la L_1, după o regulă uşor de remarcat din relaţia (4.8);
–diviziunea 2 (dacă r≥2): este constituită din C_t^2 linii, obţinute prin compunerea (operaţia „ŞI") liniilor din diviziunea 1.
În continuare, celelalte diviziuni se obţin aidoma diviziunii 2, cu diferenţa că pentru diviziunea i se compun i linii din diviziunea 1.

Relaţiile (4.7) rezultă din construcţia particulară a matricii G. Astfel $n = 2^t$, în mod evident. Codul R-M, fiind un cod grup, este şi liniar. Ca atare, distanţa de cod, d, este ponderea cuvântului de pondere minimă. Cuvintele din ultima diviziune sunt de pondere minimă, şi anume 2^{t-r}. Formula pentru e –numărul de erori corectabile– este o consecinţă a celei pentru d, iar cea pentru k este o consecinţă a dimensiunii matricii G.

După forma matricii G se observă că acest cod prezintă redundanță implicită, simbolurile de informație neregăsindu-se printre simbolurile cuvântului de cod corespunzător.

Decodarea codului R-M este o decodare cu logică majoritară. Pentru fiecare simbol de informație se pot scrie 2^{t-r} relații de calcul (cel puțin). În ecuațiile de calcul al fiecărui simbol de informație fiecare simbol din cuvântul de cod apare cel mult o dată. Astfel, dacă există cel mult $e=2^{t-r-1}-1$ erori în cuvântul recepționat, acestea vor afecta tot atâtea ecuații, rămânând nealterate de eroare cel puțin jumătate plus una ($2^{t-r}-2^{t-r-1}-1 = 2^{t-r-1}+1$). Ca urmare, valoarea bitului în cauză va fi aleasă cea majoritară peste cele 2^{t-r} ecuații.

Spre exemplu, fie codul R-M având matrice dată în relația (4.8). Definind cuvintele de informație, respectiv de cod prin relațiile:

$$\begin{aligned} i &= [i_0\ i_4\ i_3\ i_2\ i_1\ i_{43}\ i_{42}\ i_{41}\ i_{32}\ i_{31}\ i_{21}] \\ v &= [a_0\ a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ a_{10}\ a_{11}\ a_{12}\ a_{13}\ a_{14}\ a_{15}] \end{aligned} \qquad (4.9)$$

rezultă următoarele relații de codare:

$$\begin{aligned}
a_0 &= i_0 \\
a_1 &= i_0 + i_1 \\
a_2 &= i_0 + i_2 \\
a_3 &= i_0 + i_2 + i_1 + i_{21} \\
a_4 &= i_0 + i_3 \\
a_5 &= i_0 + i_3 + i_1 + i_{31} \\
a_6 &= i_0 + i_3 + i_2 + i_{32} \\
a_7 &= i_0 + i_3 + i_2 + i_1 + i_{32} + i_{31} + i_{21} \\
a_8 &= i_0 + i_4 \\
a_9 &= i_0 + i_4 + i_1 + i_{41} \\
a_{10} &= i_0 + i_4 + i_2 + i_{42} \\
a_{11} &= i_0 + i_4 + i_2 + i_1 + i_{42} + i_{41} + i_{21} \\
a_{12} &= i_0 + i_4 + i_3 + i_{43} \\
a_{13} &= i_0 + i_4 + i_3 + i_1 + i_{43} + i_{41} + i_{31} \\
a_{14} &= i_0 + i_4 + i_3 + i_2 + i_{43} + i_{42} + i_{32} \\
a_{15} &= i_0 + i_4 + i_3 + i_2 + i_1 + i_{43} + i_{42} + i_{41} + i_{32} + i_{31} + i_{21}
\end{aligned}$$

Pentru fiecare bit corespunzător celei de-a doua diviziuni se pot scrie 4 ecuații:

$$\begin{aligned}
i_{43} &= a_{12} + a_8 + a_4 + a_0 \\
i_{43} &= a_{13} + a_9 + a_5 + a_1 \\
i_{43} &= a_{14} + a_{10} + a_6 + a_2 \\
i_{43} &= a_{15} + a_{11} + a_7 + a_3
\end{aligned}
\qquad
\begin{aligned}
i_{42} &= a_{10} + a_8 + a_2 + a_0 \\
i_{42} &= a_{11} + a_9 + a_3 + a_1 \\
i_{42} &= a_{14} + a_{12} + a_6 + a_4 \\
i_{42} &= a_{15} + a_{13} + a_7 + a_5
\end{aligned}
\qquad
\begin{aligned}
i_{41} &= a_9 + a_8 + a_1 + a_0 \\
i_{41} &= a_{11} + a_{10} + a_3 + a_2 \\
i_{41} &= a_{13} + a_{12} + a_5 + a_4 \\
i_{41} &= a_{15} + a_{14} + a_7 + a_6
\end{aligned}$$

$$\begin{aligned}
i_{32} &= a_6 + a_4 + a_2 + a_0 \\
i_{32} &= a_7 + a_5 + a_3 + a_1 \\
i_{32} &= a_{14} + a_{12} + a_{10} + a_8 \\
i_{32} &= a_{15} + a_{13} + a_{11} + a_9
\end{aligned}
\qquad
\begin{aligned}
i_{31} &= a_5 + a_4 + a_1 + a_0 \\
i_{31} &= a_7 + a_6 + a_2 + a_2 \\
i_{31} &= a_{13} + a_{12} + a_9 + a_8 \\
i_{31} &= a_{15} + a_{14} + a_{11} + a_{10}
\end{aligned}
\qquad
\begin{aligned}
i_{21} &= a_3 + a_2 + a_1 + a_0 \\
i_{21} &= a_7 + a_6 + a_5 + a_4 \\
i_{21} &= a_{11} + a_{10} + a_9 + a_8 \\
i_{21} &= a_{15} + a_{14} + a_{13} + a_{12}
\end{aligned}$$

Pentru simbolurile din diviziunea 1 se pot scrie 8 ecuaţii:

$i_4 = a_8 + a_0$ $\quad$ $i_3 = a_4 + a_0$
$i_4 = a_9 + a_1 + i_{41}$ $\quad$ $i_3 = a_5 + a_1 + i_{31}$
$i_4 = a_{10} + a_2 + i_{42}$ $\quad$ $i_3 = a_6 + a_2 + i_{32}$
$i_4 = a_{11} + a_3 + i_{42} + i_{41}$ $\quad$ $i_3 = a_7 + a_3 + i_{32} + i_{31}$
$i_4 = a_{12} + a_4 + i_{43}$ $\quad$ $i_3 = a_{12} + a_8 + i_{43}$
$i_4 = a_{13} + a_5 + i_{43} + i_{41}$ $\quad$ $i_3 = a_{13} + a_9 + i_{43} + i_{31}$
$i_4 = a_{14} + a_6 + i_{43} + i_{42}$ $\quad$ $i_3 = a_{14} + a_{10} + i_{43} + i_{32}$
$i_4 = a_{15} + a_7 + i_{43} + i_{42} + i_{41}$ $\quad$ $i_3 = a_{15} + a_{11} + i_{43} + i_{32} + i_{31}$

$i_2 = a_2 + a_0$ $\quad$ $i_1 = a_4 + a_0$
$i_2 = a_3 + a_1 + i_{21}$ $\quad$ $i_1 = a_5 + a_1 + i_{21}$
$i_2 = a_6 + a_4 + i_{32}$ $\quad$ $i_1 = a_6 + a_2 + i_{31}$
$i_2 = a_7 + a_5 + i_{32} + i_{21}$ $\quad$ $i_1 = a_7 + a_3 + i_{31} + a_{21}$
$i_2 = a_{10} + a_8 + i_{42}$ $\quad$ $i_1 = a_{12} + a_8 + i_{41}$
$i_2 = a_{11} + a_9 + i_{42} + i_{21}$ $\quad$ $i_1 = a_{13} + a_9 + i_{41} + i_{21}$
$i_2 = a_{14} + a_{12} + i_{42} + i_{32}$ $\quad$ $i_1 = a_{14} + a_{10} + i_{41} + i_{31}$
$i_2 = a_{15} + a_{13} + i_{42} + i_{32} + i_{21}$ $\quad$ $i_1 = a_{15} + a_{11} + i_{41} + i_{31} + i_{31}$

În fine, pentru i_0 se pot scrie 16 ecuaţii:

$i_0 = a_0$
$i_0 = a_1 + i_1$
$i_0 = a_2 + i_2$
$i_0 = a_3 + i_2 + i_1 + i_{21}$
$i_0 = a_4 + i_3$
$i_0 = a_5 + i_3 + i_1 + i_{31}$
$i_0 = a_6 + i_3 + i_2 + i_{32}$
$i_0 = a_7 + i_3 + i_2 + i_1 + i_{32} + i_{31} + i_{21}$
$i_0 = a_8 + i_4$
$i_0 = a_9 + i_4 + i_1 + i_{41}$
$i_0 = a_{10} + i_4 + i_2 + i_{42}$
$i_0 = a_{11} + i_4 + i_2 + i_1 + i_{42} + i_{41} + i_{21}$
$i_0 = a_{12} + i_4 + i_3 + i_{43}$
$i_0 = a_{13} + i_4 + i_3 + i_1 + i_{43} + i_{41} + i_{31}$
$i_0 = a_{14} + i_4 + i_3 + i_2 + i_{43} + i_{42} + i_{32}$
$i_0 = a_{15} + i_4 + i_3 + i_2 + i_1 + i_{43} + i_{42} + i_{41} + i_{32} + i_{31} + i_{21}$

Decodarea decurge astfel:

–se calculează câte 3 valori pentru fiecare din biţii i_{43}, i_{42}, i_{41}, i_{32}, i_{31}, i_{21}, reţinând câte trei din cele patru relaţii scrise anterior pentru respectivii biţi. Valoarea fiecărui bit se stabileşte ca fiind cea care rezultă prin cel puţin două din cele trei relaţii.

–se calculează câte 7 valori pentru fiecare din biţii i_4, i_3, i_2, i_1, reţinând câte şapte din cele opt relaţii scrise anterior pentru respectivii biţi. Valoarea fiecărui bit se stabileşte ca fiind cea care rezultă prin cel puţin patru din cele trei relaţii.

–se calculează 15 valori pentru bitul i_0 reținând 15 din cele 16 relații scrise anterior pentru bitul i_0. Valoarea acestui bit se stabileşte ca fiind cea care rezultă prin cel puțin şapte din cele 15 relații.

5. Coduri ciclice

5.1 Coduri ciclice –descriere generală

Codurile ciclice sunt coduri bloc (toate cuvintele au aceeaşi lungime, codarea şi decodarea unui bloc este independentă de a celorlalte). Sunt numite coduri ciclice deoarece orice permutare ciclică a unui cuvânt de cod este, de asemenea, cuvânt de cod, [5]:
-dacă $u = u_{n-1}\ u_{n-2}\ \dots\ u_1\ u_0$ este un cuvânt de cod ciclic, atunci $u^* = u_{n-2}\ u_{n-3}\ \dots\ u_1\ u_0\ u_{n-1}$ este un cuvânt aparţinând aceluiaşi cod.

Cuvintele codului ciclic pot fi reprezentate sub formă de polinoame:

$$u(x) = u_{n-1}\cdot x^{n-1} + u_{n-2}\cdot x^{n-2} + \dots + u_2\cdot x^2 + u_1\cdot x + u_0 \qquad (5.1)$$

Structura cuvântului de cod ciclic cuprinde **n** biţi (coeficienţi binari), dintre care primii **k** biţi (pentru un cod sistematic) sunt biţii de informaţie: u_{n-1}, u_{n-2} , … , u_{n-k} iar ultimii **m** biţi sunt biţii de control: u_{m-1}, u_{m-2}, … , u_1 u_0. Puterile lui x indică tactele la care respectivii biţi sunt livraţi la ieşirea codorului, în ordine descrescătoare. Astfel:

$$n = k + m. \qquad (5.2)$$

Dacă codul este corector de o singură eroare (marginea Hamming coincide cu marginea Varşamov–Gilbert):

$$n = 2^m - 1. \qquad (5.3)$$

Cei k biţi de informaţie constituie polinomul de informaţie:

$$i(x) = u_{n-1}\cdot x^{k-1} + u_{n-2}\cdot x^{k-2} + \dots + u_{n-k+1}\cdot x + u_{n-k} \qquad (5.4)$$

Codurile ciclice corectoare de o eroare, având distanţa de cod (distanţa Hamming minimă între cuvintele codului) $\mathbf{d_{Hmin}} = 3$, sunt capabile să corecteze o eroare sau să detecteze două, [5].

Obs. *–Un cod este detector de e_d erori (corector de e_c erori) dacă detectează (corectează) orice combinaţie de e_d (e_c) erori, sau mai puţine. Codurile ciclice corectoare de o eroare sunt capabile să detecteze şi o parte dintre combinaţiile cu mai mult de două erori, dar nu orice combinaţie cu mai mult de două erori.*

-În practică sunt utilizate coduri ciclice detectoare de trei erori independente, sau de pachete de erori, dar aceste coduri ciclice nu satisfac (5.3), şi, în plus, au $d_{Hmin} \geq 4$.

-Codurile BCH şi Reed-Solomon sunt de asemenea coduri ciclice (ele au proprietatea de ciclicitate a cuvintelor de cod) şi prezintă acelaşi procedeu de codare ca şi codurile ciclice corectoare de o eroare, însă sunt corectoare de erori multiple şi, ca atare, diferă prin procedeele de decodare. Prezenta expunere referitoare la codurile ciclice include şi codurile

BCH şi Reed-Solomon, exceptând cazul în care se specifică că este vorba despre codurile ciclice corectoare de o eroare.

-În continuare, vom numi un cuvânt de cod (sau de informaţie) atât prin secvenţa u (sau i) cât şi prin polinomul ataşat u(x) (respectiv i(x)).

Polinoamele asociate cuvintelor de cod ciclic au proprietatea că sunt divizibile prin g(x): un polinom de grad m, numit ***polinom generator***. Această proprietate este utilizată la codare pentru calculul biţilor de control, iar la decodare pentru verificarea şi (eventual) corecţia cuvântului recepţionat.

Polinomul generator g(x) alături de parametrii n şi k definesc în totalitate codul.

5.2 Codarea codurilor ciclice

Codarea codurilor ciclice se poate face prin multiplicare sau prin împărţire. În continuare va fi descrisă ultima metodă, metodă care conduce la un cod sistematic. În această metodă, corespondenţa dintre i(x) şi u(x) este, [5]:

$$u(x) = i(x) \cdot x^m + \text{rest}\,\frac{i(x) \cdot x^m}{g(x)}, \qquad (5.5)$$

unde rest $(\cdot)/g(x)$ semnifică restul împărţirii polinomului $(\cdot)$ la g(x).

Obs. *Operaţia de adunare din ecuaţia (5.5) este sumă modulo 2, iar coeficienţii polinoamelor sunt din câmpul binar {0,1}.*

5.2.1 Registru de Deplasare cu Reacţie (RDR).

Implementarea ecuaţiei (5.5) se poate face în mai multe feluri. În cele ce urmează va fi descrisă în detaliu implementarea cu un *Registru de Deplasare cu Reacţie (RDR)*.

În Figura 5.1 este prezentat un RDR. Celulele C_1 la C_k sunt bistabile de tip „D" sincrone; blocurile notate cu „+" sunt sumatoare modulo 2, iar triunghiurile sunt amplificatoare cu amplificările $g_j \in \{0,1\}$, cu $j \in [1, m-1]$. g_j sunt coeficienţii polinomului g(x). Coeficientul $g_m = 1$ simbolizează reacţia.

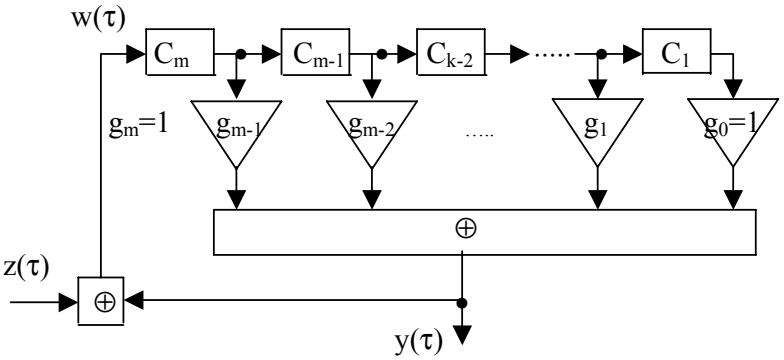


Figura 5.1. RDR

Pentru analiza funcționării circuitului RDR, vom numi ieşirea cu $y(\tau)$, cu $z(\tau)$ intrarea iar cu $w(\tau)$ ieşirea sumatorului ce adună $y(\tau)$ cu $z(\tau)$. Astfel, putem scrie:

$$w(\tau) = z(\tau) + y(\tau)$$
$$y(\tau) = w(\tau\text{-}1)\cdot g_{m-1} + \ldots + w(\tau\text{-}m)\cdot g_0 \qquad (5.6)$$

Utilizând transformata „D", obținem (aici tranformata „D" este transformata „Z" utilizată pentru semnalul binar):

$$W(D) = Z(D) + Y(D)$$
$$Y(D) = g_{m-1}\cdot D^{-1}\cdot W(D) + \ldots + g_0\cdot D^{-m}\cdot W(D) \qquad (5.7)$$

Transformata „D" a unui semnal $z(\tau)$ este definită prin:

$$D\{z(t)\} = Z(D) = \sum_{t=-\infty}^{\infty} z(t)\cdot D^{-t} \qquad (5.8)$$

iar:

$$D\{z(t\text{-}j)\} = \sum_{t=-\infty}^{\infty} z(t-j)\cdot D^{-t+j} D^{-j} = D^{-j}\cdot D\{z(t)\}. \qquad (5.9)$$

Astfel, sistemul de ecuații (5.7) se poate scrie:

$$W(D) = Z(D) + Y(D)$$
$$Y(D)\cdot D^m = W(D)\cdot(D^m + g(D)); \qquad (5.10)$$

unde:

$$g(D) = D^m + g_{m-1}\cdot D^{m-1} + \ldots + g_1\cdot D + g_0$$

Eliminând W(D) din (5.10), obținem: $Y(D)\cdot g(D) = Z(D)\cdot g(D) + D^m\cdot Z(D)$, adică:

$$Y(D) = Z(D) + D^m\cdot Z(D)/g(D) \qquad (5.11)$$

Însă, Z(D) şi Y(D) se scriu (conform ecuației (5.8)):

$$Z(D) = z_0 + z_1\cdot D^{-1} + z_2\cdot D^{-2} + \ldots$$
$$Y(D) = y_0 + y_1\cdot D^{-1} + y_2\cdot D^{-2} + \ldots \qquad (5.12)$$

Dar, ştiind că din punct de vedere al codării interesează doar primele n tacte, putem neglija ceilalți termeni, astfel că:

$$Z(D) = z_0 + z_1\cdot D^{-1} + z_2\cdot D^{-2} + \ldots + z_{n-1}\cdot D^{-n+1} =$$
$$= D^{-n+1}\cdot(z_0\cdot D^{n-1} + z_1\cdot D^{n-2} + \ldots z_{n-1}) \qquad (5.13)$$

z_0 este primul bit care intră în codor, în consecință este echivalent cu bitul u_{n-1} sau a_{n-1}. În acest caz, Z(D) se poate pune sub forma:

$$Z(D) = D^{-n+1} \cdot a(D), \quad (5.14)$$

unde:

$$a(D) = a_{n-1} \cdot D^{n-1} + \ldots + a_1 \cdot D^1 + a_0$$

Prin urmare, se poate scrie că:

$$Y(D) = D^{-n+1} \cdot b(D), \quad (5.15)$$

unde:

$$b(D) = b_{n-1} \cdot D^{n-1} + \ldots + b_1 \cdot D^1 + b_0$$

Ținând cont de ecuațiile (5.11), (5.14) şi (5.15) obținem:

$$b(D) = a(D) + D^m \cdot a(D)/g(D) \quad (5.16)$$

Fie acum a(D) = u(D) un cuvânt de cod ciclic, adică:

$$a(D) = u(D) = q(D) \cdot g(D) \quad (5.17)$$

unde q(D) este un polinom de grad cel mult k = n – m – 1:

$$q(D) = q_{k-1} \cdot D^{k-1} + \ldots + q_1 \cdot D + q_0 \quad (5.18)$$

Relația (5.5) se scrie, de asemenea:

$$u(D) = i(D) \cdot D^m + \text{rest}[i(D) \cdot D^m / g(D)] = i(D) \cdot D^m + r(D) \quad (5.19)$$

unde $r(D) = \text{rest}[i(D) \cdot D^m / g(D)]$, iar $i(D) = u_{n-1} \cdot D^{k-1} + u_{n-2} \cdot D^{k-2} + \ldots + u_{n-k+1} \cdot D + u_{n-k},$.
Înlocuind (5.17) şi (5.19) în (5.16) obținem:

$$\begin{aligned} b(D) &= i(D) \cdot D^m + r(D) + \cdot D^m \cdot q(D) \cdot g(D)/g(D) \\ &= [i(D) + q(D)] \cdot D^m + r(D) \end{aligned} \quad (5.20)$$

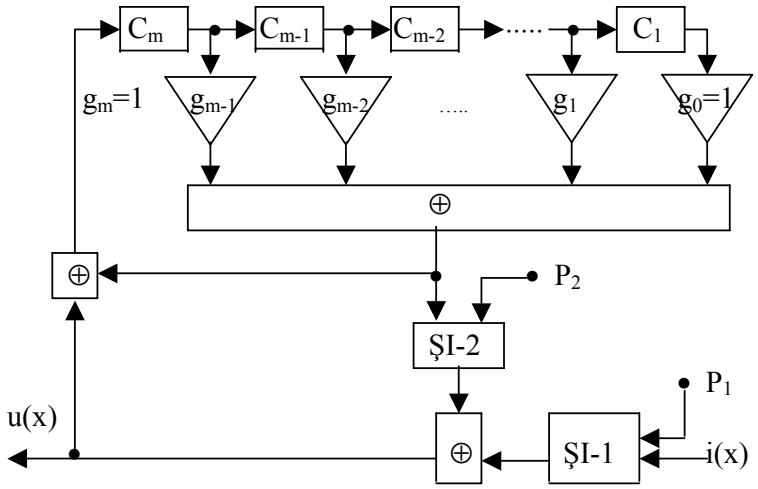
Cei doi termeni din ultima ecuație $[i(D) + q(D)] \cdot D^m$ şi $r(D)$ sunt complet separați în timp: $[i(D) + q(D)] \cdot D^m$ este un polinom cu termeni (posibil ne-nuli) de grad $\in [m, k+m-1]$ corespunzând tacturilor cuprinse între 0 şi k-1, iar r(D) este un polinom de grad cel mult m-1, corespunzător tacturilor cuprinse între k şi n-1.

În concluzie, dacă intrarea circuitului RDR, $z(\tau)$, este secvența de informație dată prin ecuația (5.4), după k tacte, RDR este capabil să genereze la ieşirea $y(\tau)$ secvența rest, r(D), cu condiția ca această secvență să fie conectată şi la intrarea z, pentru ultimele k tacturi.

5.2.2 Codor ciclic cu RDR şi sumatoare exterioare

În Figura 5.2 este prezentat un codor ciclic care implementează relația de codare (5.5). Secvența de informație, i(x), intră în codor în primele k tacte, primul bit fiind cel mai semnificativ şi, de asemenea, este conectată şi la ieşire. Pentru aceasta, întrerupătorul 1,

a) Codor ciclic cu RDR ;



b) Semnalele de validare a porţilor „ŞI”

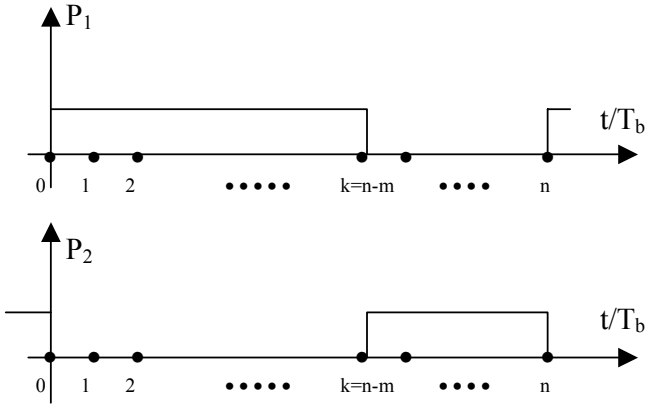


Figura 5.2 Codor ciclic cu RDR şi semnalele de comandă

format din poarta ŞI-1 este închis iar întrerupătorul 2, format din poarta ŞI-2 este deschis (vezi semnalele de comandă P_1 şi P_2).

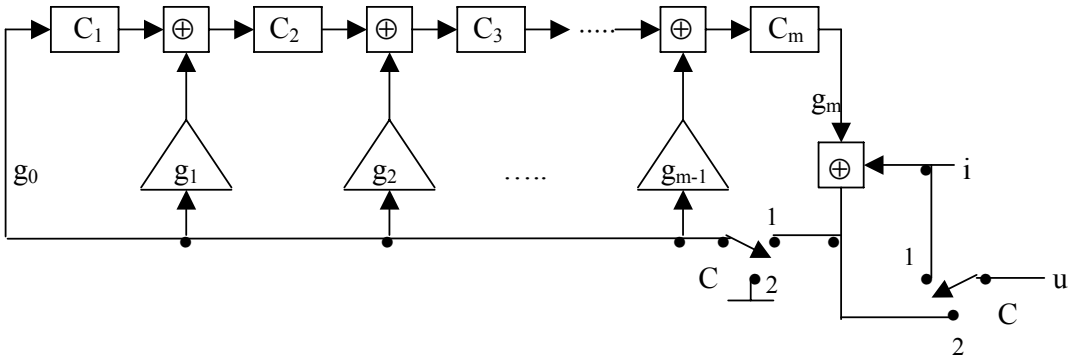
În următoarele m tacturi întrerupătorul 1 este deschis iar întrerupătorul 2 este închis, astfel că secvența r, generată de RDR, este livrată la ieşirea u.

5.2.3 Codor ciclic cu sumatoare interioare

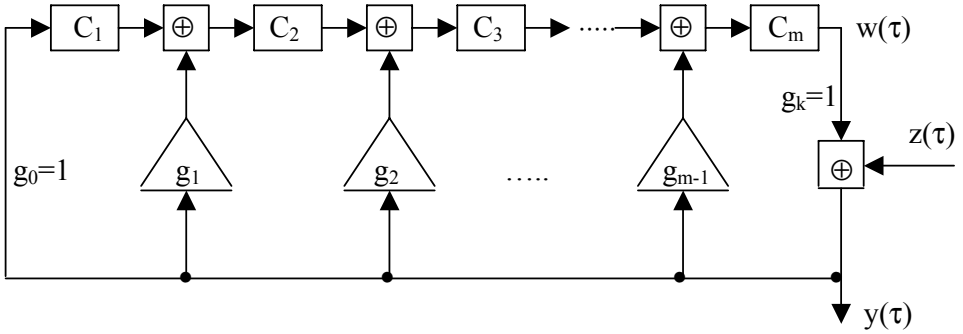
În Figura 5.3.a) este prezentată schema unui codor ciclic echivalent celui realizat cu sumatoare interioare. „C” este un comutator cu două poziții. Pe durata a k tacturi, comutatorul se află în poziția 1, respectiv 2 pentru următoarele m tacturi. Pentru analiza circuitului din figura 5.3.a), presupunem comutatorul în poziția 1, astfel încât obținem circuitul din figura 5.3.b).

Conform figurii 5.3.b, găsim următoarele relații:

$$y(\tau) = z(\tau) + w(\tau)$$
$$w(\tau) = g_0 \cdot y(\tau-m) + g_1 \cdot y(\tau-m+1) + g_2 \cdot y(\tau-m+2) + \ldots + g_{m-1} \cdot y(\tau-1) \quad (5.21)$$



a) Schema codorului ciclic



b)Circuitul de analiză

Figura 5.3 Codor ciclic cu sumatoare interioare

Aplicând transformata „D" şi efectuând aceleaşi operaţii ca şi pentru RDR, găsim:

$$Y(D) = Z(D) + W(D)$$
$$W(D) = D^{-m}\cdot(D^m + g(D))\cdot Y(D) \quad (5.22)$$

unde:

$$Y(D) = D^m\cdot Z(D)/g(D) \quad (5.23)$$
$$W(D) = Z(D) + D^m\cdot Z(D)/g(D) \quad (5.24)$$

Ecuaţia (5.24) este echivalentă relaţiei (5.11). Urmând acelaşi raţionament, dacă Z este un cuvânt de cod, adică:

$$Z(D) = q(D)\cdot g(D)\cdot D^{-n+1} \quad (5.25)$$

atunci rezultă:

$$Y(D) = q(D)\cdot D^m\cdot D^{-n+1} = q(D)\cdot D^{-k+1} \quad (5.26)$$
$$W(D) = [(i(D) + q(D))\cdot D^m + r(D)]\cdot D^{-n+1} \quad (5.27)$$

Altfel spus, în punctul w(τ), după k tacte, vom regăsi r(D), cu condiția ca Y să fie anulat după tactul k. Astfel ecuațiile (5.22) şi (5.10) sunt echivalente (schimbând W cu Y). Anularea lui Y după k tacte este realizată de către comutatorul „C". Echivalența ecuațiilor (5.22) şi (5.10) demonstrează faptul că cele două scheme (cu RDR şi sumatoare exterioare şi cu sumatoare interioare) implementează acelaşi procedeu de codare.

5.3 Decodarea codurilor ciclice

Decodarea presupune, în cazul detecției de erori, verificarea exactității transmisiei fiecărui cuvânt de cod recepționat şi semnalarea prezenței erorilor. La depistarea prezenței erorilor în cuvântul recepționat se cere retransmisia sa. În cazul corecției de erori, prin verificarea cuvântului recepționat, pe lângă depistarea prezenței erorilor, se precizează şi poziția lor (codurile ciclice în discuție sunt corectoare de o eroare, însă afirmația este valabilă şi pentru cazul general al corecției de erori multiple).

Obs. *–La recepția unui cuvânt, după verificarea sa, rezultă două concluzii alternative: cuvântul este eronat sau cuvântul este corect.*

Decizia în primul caz este 1OO% adevărată, în vreme ce, pentru cel de-al doilea caz este posibil ca decizia să fie una falsă. Este cazul recepției unui cuvânt emisibil, altul decât cel emis, rezultat prin eronarea celui emis. De remarcat că în astfel de cazuri se depăşeşte puterea de detecție/corecție a codului.

Proprietatea definitorie a cuvintelor de cod ciclic este dată prin ecuația (5.5), adică dacă u(x) este un cuvânt de cod ciclic (având polinomul generator g(x)), atunci u(x) este multiplul lui g(x):

$$u(x) = q(x)\cdot g(x), \tag{5.28}$$

unde q(x) este un polinom de grad cel mult m-1.

Presupunem acum că v(x) este un cuvânt recepționat, posibil eronat:

$$v(x) = u(x) + \varepsilon(x) \tag{5.29}$$

unde: -u(x) este cuvântul de cod transmis (care verifică relația (5.5));

-ε(x) = v(x) + u(x) este cuvântul eroare, o secvență binară de aceeaşi lungime ca şi u sau v, şi având unu-uri în pozițiile eronate din v, adică în pozițiile în care u şi v diferă.

Atunci, algoritmul decodării este:

1. – calculul sindromului:

$$s(x) = \text{rest } v(x)/g(x) \tag{5.30}$$

Obs. *Deoarece rest u(x)/g(x) = 0 rezultă că:*

$$s(x) = \text{rest } \varepsilon(x)/g(x) \tag{5.31}$$

2. –dacă s(x) = 0 se hotărăşte că nu există erori. Cum s-a remarcat anterior, este posibil ca această concluzie să fie eronată, însă nu există posibilitatea de a discerne între cazul

recepţiei cuvântului emis şi cazul unei combinaţii de erori ce transformă cuvântul emis într-un alt cuvânt emisibil.

–dacă $s(x) \neq 0$ atunci există erori (cu probabilitate 100%). În acest caz:

a) pentru detecţie, se ignoră cuvântul în cauză sau se cere retransmisia sa;

b) pentru corecţie (şi pentru codurile ciclice corectoare de o eroare), se caută poziţia erorii. De remarcat faptul că (pentru codurile ciclice corectoare de o eroare care au distanţa Hamming minimă între cuvintele codului trei) există totdeauna un cuvânt emisibil aflat la distanţă unu de cuvântul recepţionat. Corecţia erorii se va face prin inversarea valorii binare a bitului considerat (prin sumarea modulo 2 a unui unu bitului considerat eronat).

Rămâne de explicat procedeul prin care se află poziţia erorii în cazul corecţiei. Deoarece este o particularitate a decodorului, expunerea acestui procedeu se va face simultan cu descrierea decodorului.

5.3.1 Decodor cu RDR pentru un cod ciclic detector de erori

În Figura 5.5 este prezentată schema unui decodor cu RDR. Semnalul de ieşire, „Detecţie", este prezentat, pentru un caz particular, în Figura 5.4. Pentru cazul prezentat cuvintele v_1 şi v_3 sunt eronate iar v_0 şi v_2 sunt (presupuse) corecte.

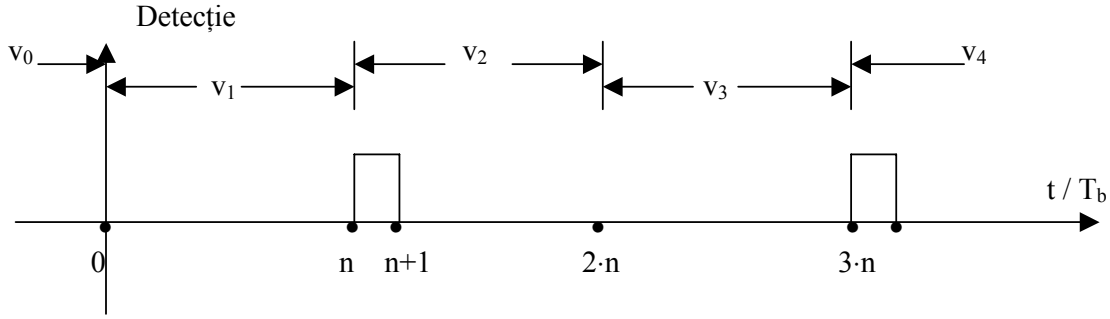


Figura 5.4 Exemplu de semnal „Detecţie"

Pentru explicarea funcţionării circuitului din Figura 5.5, pornind de la relaţia (5.11) şi de la prima ecuaţie din (5.7), (înlocuind Z cu $D^{-n+1}\cdot V$) găsim că:

$$W(D) = [V(D) + V(D) + D^m \cdot V(D)/g(D)]\cdot D^{-n+1}$$

adică:

$$W(D) = D^{-n+m+1}\cdot V(D)/g(D) \qquad (5.32)$$

Presupunem acum că:

$$V(D) = U(D) = q(D)\cdot g(D), \qquad (5.33)$$

în conformitate cu relaţia (5.24) şi prin analogie cu relaţiile (5.14) şi (5.15). Rezultă că:

$$W(D) = D^{-n+m+1}\cdot q(D) = \\ = D^{-k+1}\cdot(q_{k-1}\cdot D^{k-1} + q_{k-2}\cdot D^{k-2} + ... + q_1\cdot D + q_0) \qquad (5.34)$$

unde:

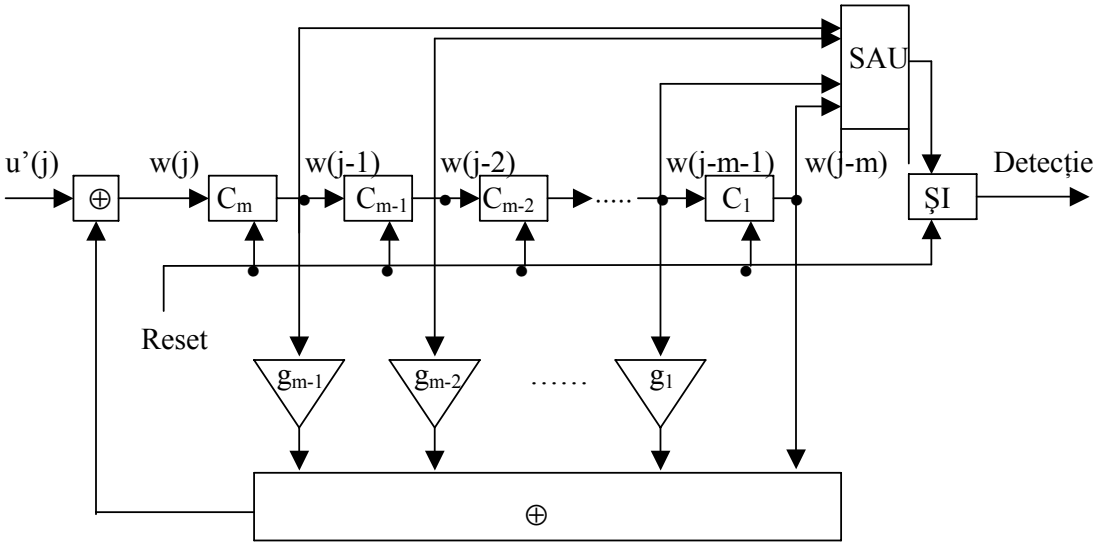
$$W(D) = q_{k-1} + q_{k-2}\cdot D^{-1} + ... + q_1\cdot D^{-k+2} + q_0\cdot D^{-k+1} \qquad (5.35)$$

Dar, prin definiție, W(D) se scrie:

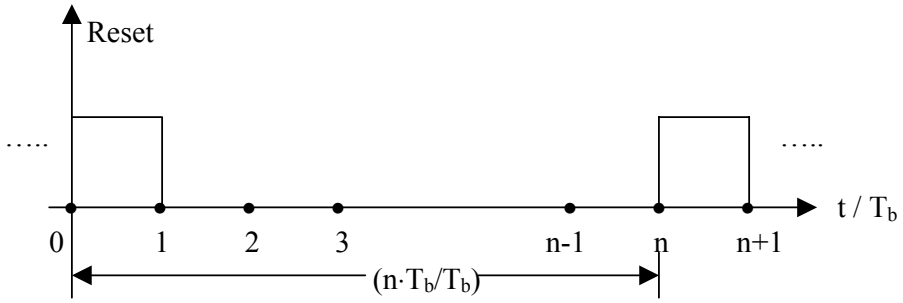
$$W(D) = w_0 + w_1\cdot D^{-1} + \ldots + w_{k-1}\cdot D^{-k+1} + w_{n-m}\cdot D^{-n+m} + \ldots + w_{n-1}\cdot D^{-n+1} \qquad (5.36)$$

q_{k-1} q_{k-2} ….. q_0 — k tacturi; 0 ….. 0 — m tacturi

Prin identificare cu relația (5.36) am arătat că, între tactele n-k şi n-1, prin punctul notat w(j) nu trec decât zerouri. Aceste zerouri se vor regăsi în tactul n în celulele registrului decodorului. Astfel, la tactul n, poarta „SAU" (cu m intrări), având doar zerouri pe intrări, generează un zero la ieşirea „Detecție". Am demonstrat, de asemenea, comportamentul circuitului la recepția unui cuvânt corect.



a) Schema decodorului detector de erori;



b) Semnalul Reset ;

Figura 5.5 Decodor cu RDR pentru detecția de erori

Presupunem acum că:

$$v(x) = u(x) + \varepsilon(x) \tag{5.37}$$

unde $\varepsilon(x)$ nu este cuvânt de cod, adică:

$$\text{rest } \varepsilon(x)/g(x) = \text{rest } v(x)/g(x) \neq 0. \tag{5.38}$$

Deoarece influenţa cuvântului de cod u(x), după tactul n, asupra semnalelor de la ieşirile celulelor este nulă, putem considera, fără a restrânge generalitatea că u(x)=0, astfel încât:

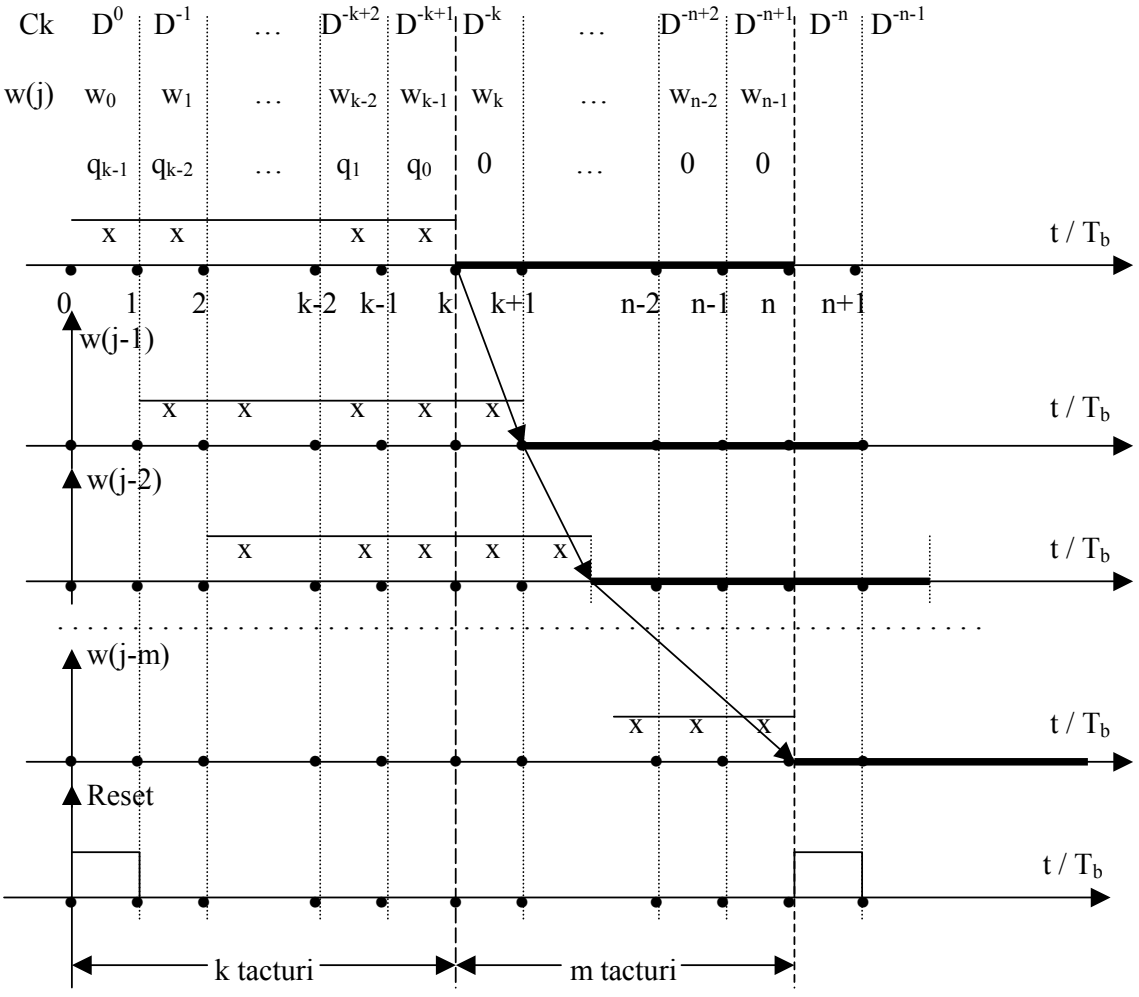


Figura 5.6 Diagrama de timp

$$W(D) = D^{-n+m+1} \cdot \varepsilon(D)/g(D) \tag{5.39}$$

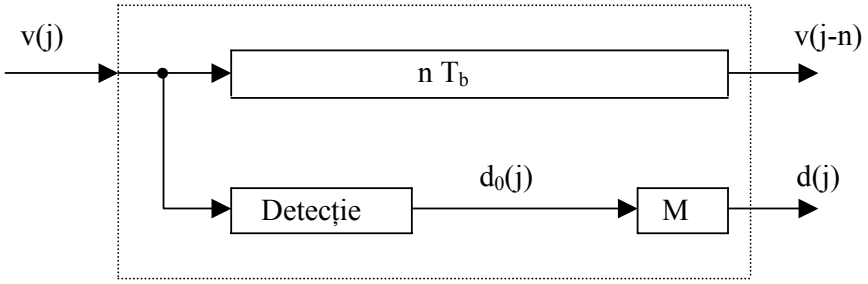
Poarta „SAU” din componenţa schemei decodorului va genera 1^L la tactul n (unu logic ce se va regăsi imediat şi la ieşirea porţii ŞI, la acelaşi tact) dacă cel puţin una din celulele $C_1, C_2, \ldots$ sau C_m are la ieşire unu logic.

Să presupunem prin absurd că nu este aşa. Dacă toate celulele C_j, j=1 la m, au zero logic la ieşire rezultă că la intrarea primei celule între tacturile n-m şi n-1 a fost 0 logic. Ori asta înseamnă că:

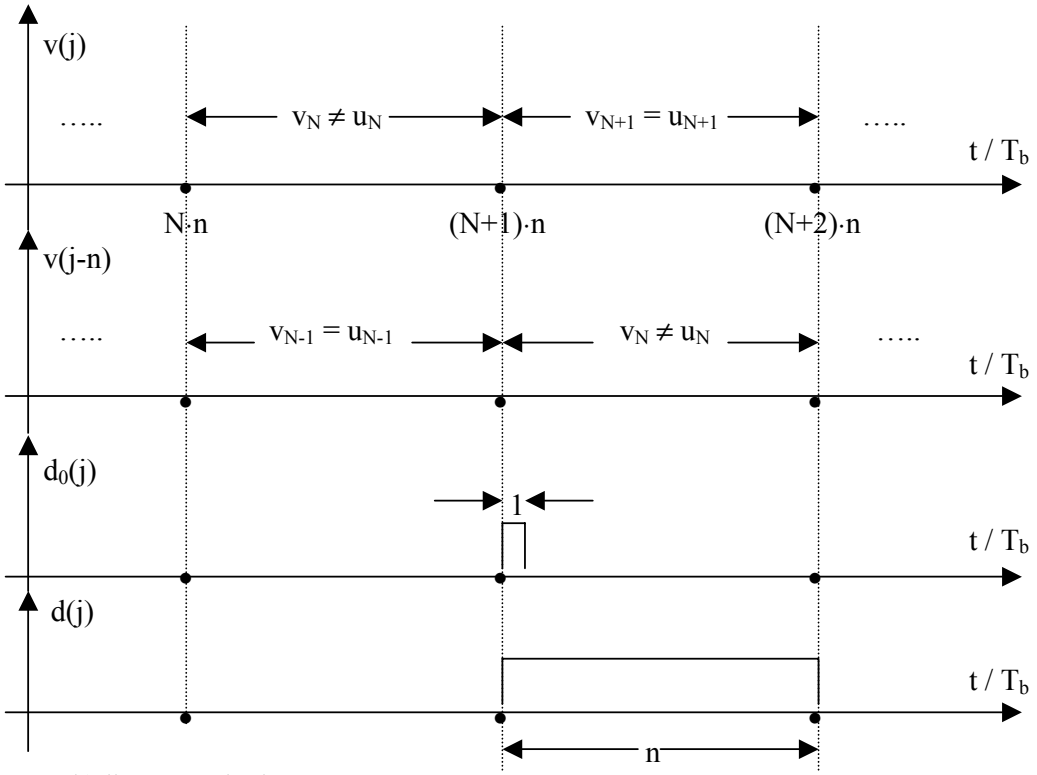
$$W(D) = \sum_{j=0}^{k-1} w_j \cdot D^{-j} = D^{-k+1} \cdot \sum_{j=0}^{k-1} w_j \cdot D^{k-j-1} \tag{5.40}$$

sau, notând:

$$q_j = w_{k-j-1} \tag{5.41}$$



a) schema decodorului ciclic pentru detecția de erori ;



b)diagrama de timp ;

Figura 5.7 Decodor ciclic pentru detecția de erori;

rezultă că:

$$W(D) = D^{-n+m+1} \cdot \sum_{j=0}^{k-1} q_{k-j-1} \cdot D^{k-j-1} = D^{-n+m+1} \cdot q(D) = D^{-n+m+1} \cdot \frac{q(D) \cdot g(D)}{g(D)} \quad (5.42)$$

Dar, conform relației (5.39):

$$W(D) = D^{-n+m+1} \cdot \varepsilon(D)/g(D) \quad \Rightarrow \quad \varepsilon(D) = q(D) \cdot g(D) \quad (5.43)$$

ceea ce înseamnă că ε(D) este un cuvânt de cod ⇒ contradicție! Aşadar, dacă rest ε(D)/g(D) ≠ 0 cel puțin o celulă din cele m are la ieşire 1^L la tactul n, astfel încât la ieşirea „Detecție" la acelaşi tact va fi 1^L.

Obs. *–De remarcat faptul că în demonstrația de mai sus nu s-a folosit necesitatea ca g(x) să fie primitiv, astfel încât schema funcționează la fel de bine atâta timp cât rest ε(D)/g(D) ≠ 0, aceasta fiind condiția necesară pentru selecția polinomului generator g(x). Este o condiție mai puțin restrictivă decât cea de polinom primitiv.*

În continuare este propusă o schemă de decodor ciclic ce face detecție de erori, obținută prin completarea schemei din Figura 5.5.

Blocul notat „nT_b" este un bloc de întârziere cu n tacturi, necesar pentru a alinia secvența ce indică zona cu erori (prin 1^L), d(j) şi secvența recepționată. Blocul „Detecție" este circuitul a cărui schemă a fost prezentată în Figura 5.5, iar „M" este un circuit tip „monostabil", care prelungeşte 1^L dat de blocul de „Detecție" pe un tact, la n tacturi, zonă semnalizată cu erori.

În concluzie, circuitul de detecție propus furnizează la ieşire secvența recepționată precum şi o secvență ce indică zona în care există erori (fără a le preciza pozițiile).

5.3.2 Decodor cu RDR pentru un cod ciclic corector de o eroare

Înainte de a prezenta o schemă de decodare trebuie făcute câteva precizări:

–polinomul utilizat în acest caz este un polinom primitiv. Rezultă că g(x) nu divide nici un polinom de forma unui monom sau a unui binom cu grad mai mic decât n-1;

–prin construcție, decodorul caută doar o eroare în cuvântul recepționat. Dacă există mai mult de o eroare în cuvântul recepționat, atunci acesta (decodorul) greşeşte, putând chiar corecta un bit ne-eronat;

–decodarea în acest caz, al corecției, necesită 2n tacturi –n pentru a verifica corectitudinea cuvântului recepționat şi încă n–e+1, pentru aflarea poziției bitului eronat (unde e este poziția erorii, presupusă doar una). Astfel schema va conține două decodoare cu RDR ce vor prelua, verifica şi (eventual) corecta, alternativ tot al doilea cuvânt din cele recepționate (un decodor preia cuvintele cu număr de ordine par iar celălalt cuvintele cu număr de ordine impar).

Se va face analiza decodării pentru un decodor, urmând ca apoi să se prezinte schema în ansamblul ei.

Presupunem aşadar:

$$\varepsilon(x) = x^e \qquad \text{cu} \quad e \in \{0,1,2,\ldots,n-1\} \quad (5.44)$$

Cu observațiile din paragraful precedent, presupunem de asemenea că u = 0, fapt ce nu restrânge generalitatea. Din relația (5.39) rezultă:

$$W(D) = D^{-n+m+1} \cdot \varepsilon(D)/g(D) = D^{-n+m+1} \cdot D^e/g(D) = D^{-n+m+e+1}/g(D) \quad (5.45)$$

Ştiind că pentru g(x) –primitiv:

$$x^n + 1 = g(x)\cdot h(x) \tag{5.46}$$

rezultă:

$$W(D) = D^n \cdot D^{-2n+m+e+1}/g(D) = [g(D)\cdot h(D) + 1]\cdot D^{-2n+m+e+1}/g(D) =$$

$$= h(D)\cdot D^{-2n+m+e+1} + D^{-2n+m+e+1}/g(D) =$$

$$= \sum_{j=0}^{k} h_j \cdot D^j \cdot D^{-2n+m+e+1} + \sum_{i=2n-e-1}^{\infty} \alpha_i \cdot D^{-i} \tag{5.47}$$

Unde am pus în evidență momentele de timp (tacturile), pe care le ocupă termenii ce compun pe W(D). Astfel se poate construi diagrama din Figura. 5.8, ținând cont şi de faptul că $h_m = h_0 = 1$.

Din analiza făcută reiese că la tactul 2n – e –1 (situat între n, pentru eroare pe poziția e=n –1 şi 2n –1 pentru eroare situată pe ultima poziție) starea RDR-ului este: $C_m = C_{m-1} = \ldots C_2 = 0$ iar $C_1 = 1$ indiferent de poziția erorii, e.

Pe de altă parte această stare nu mai poate fi întâlnită câtă vreme intrarea este blocată pe zero (între momentele de tact n şi 2n), deoarece inițializat şi lăsat liber (cu intrarea pe 0) RDR-ul, timp de n tacte parcurge toate cele $2^m - 1 = n$ stări posibile. (Este o proprietate a polinomului primitiv.)

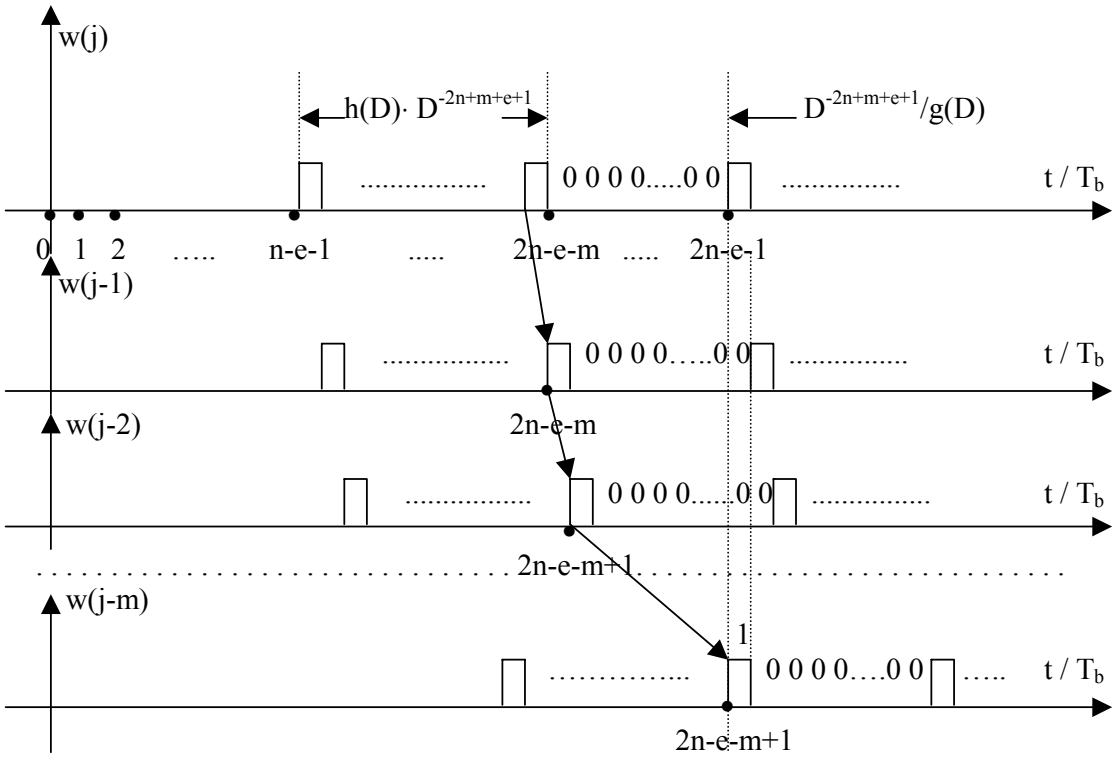


Figura 5.8 Diagrama de timp pentru decodorul ciclic corector de o eroare

Astfel putem identifica starea amintită a RDR-ului, iar în momentul identificării să facem corecția bitului eronat. Structura unui decodor ciclic corector de o eroare arată ca în Figura 5.9.

Blocul „nT_b" este un registru de întârziere cu n celule. Conform celor discutate anterior, momentul de timp în care se identifică starea „0 0 … 0 1" este momentul 2n-e-1 = n+n-e-1, adică este exact momentul în care eroarea (aflată pe poziția e la momentul de tact n) ajunge la ieşirea celulei n –1. Ca atare momentul 2n – e –1 este tactul în care se face corecția (se sumează bitului eronat u_e —aflat în celula n–1— un 1^L, generat de blocul IDENTIFICĂ).

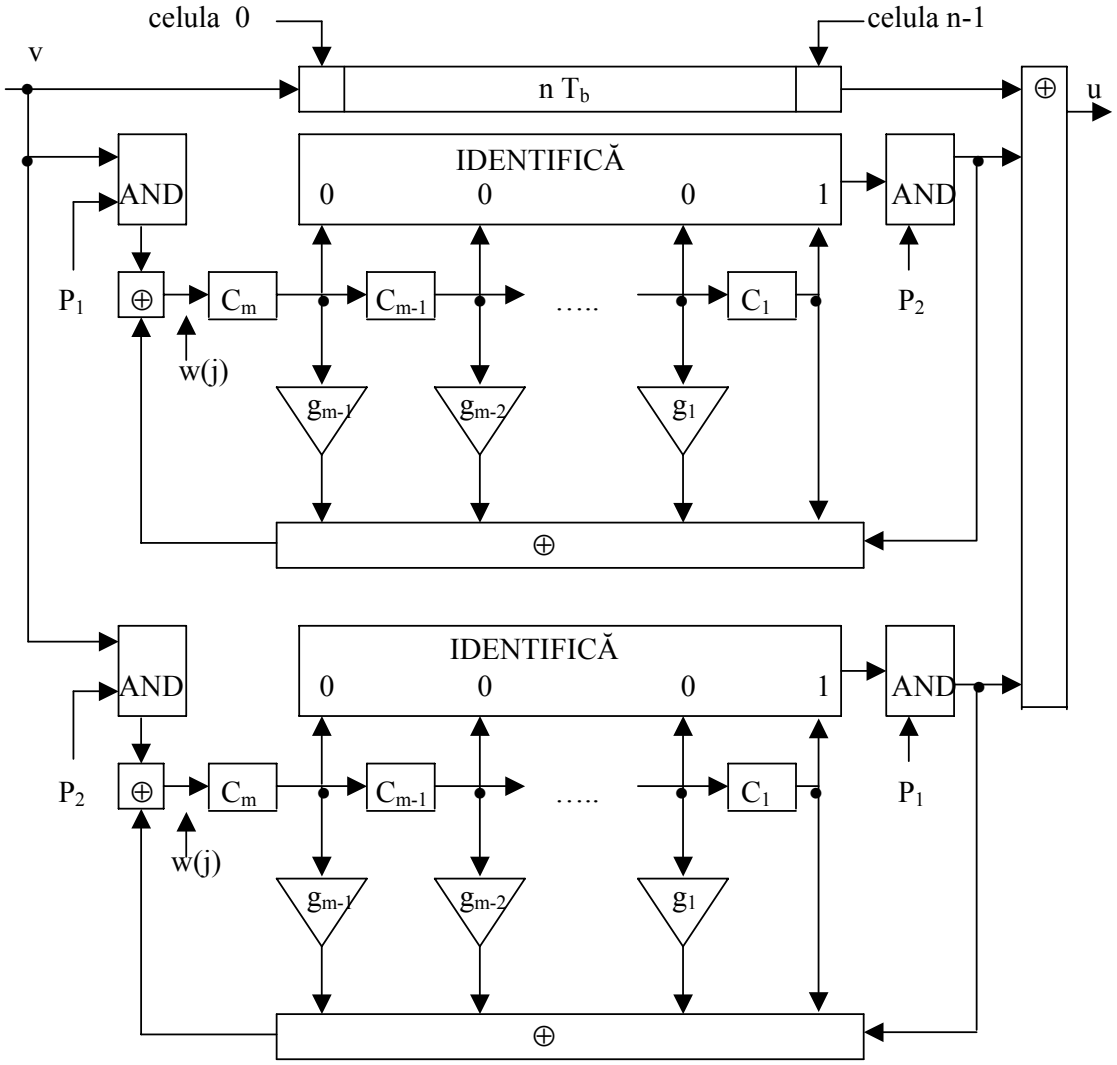
În Figura 5.10 se prezintă o diagramă de timp asociată decodorului ciclic ce prezintă semnalele (secvențele binare) corespunzătoare celulelor RDR-ului pentru cazurile de recepție a unui cuvânt: –corect;

–eronat în poziția n-1;

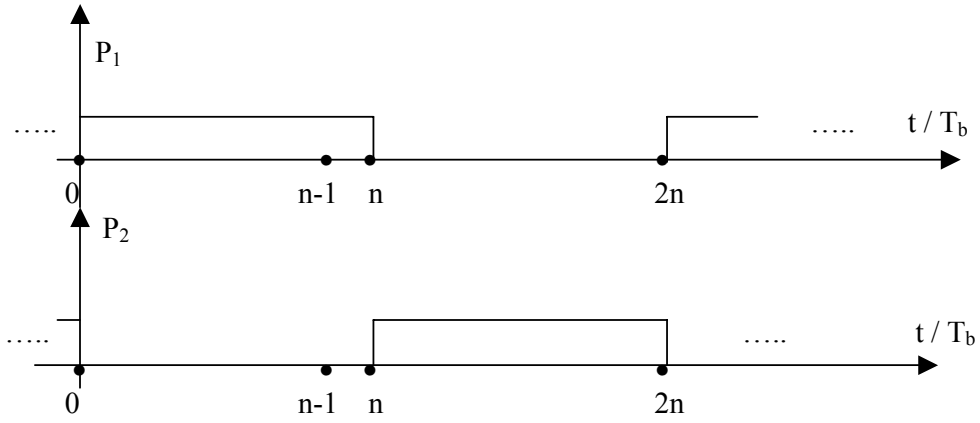
–eronat în poziția e, unde 0<e<n-1;

–eronat în poziția 0;

–eronat în două poziții, n-1 şi e.



a) schema ;



b)Semnalele de validare a porților AND ;

Figura 5.9 Decodor ciclic corector de o eroare

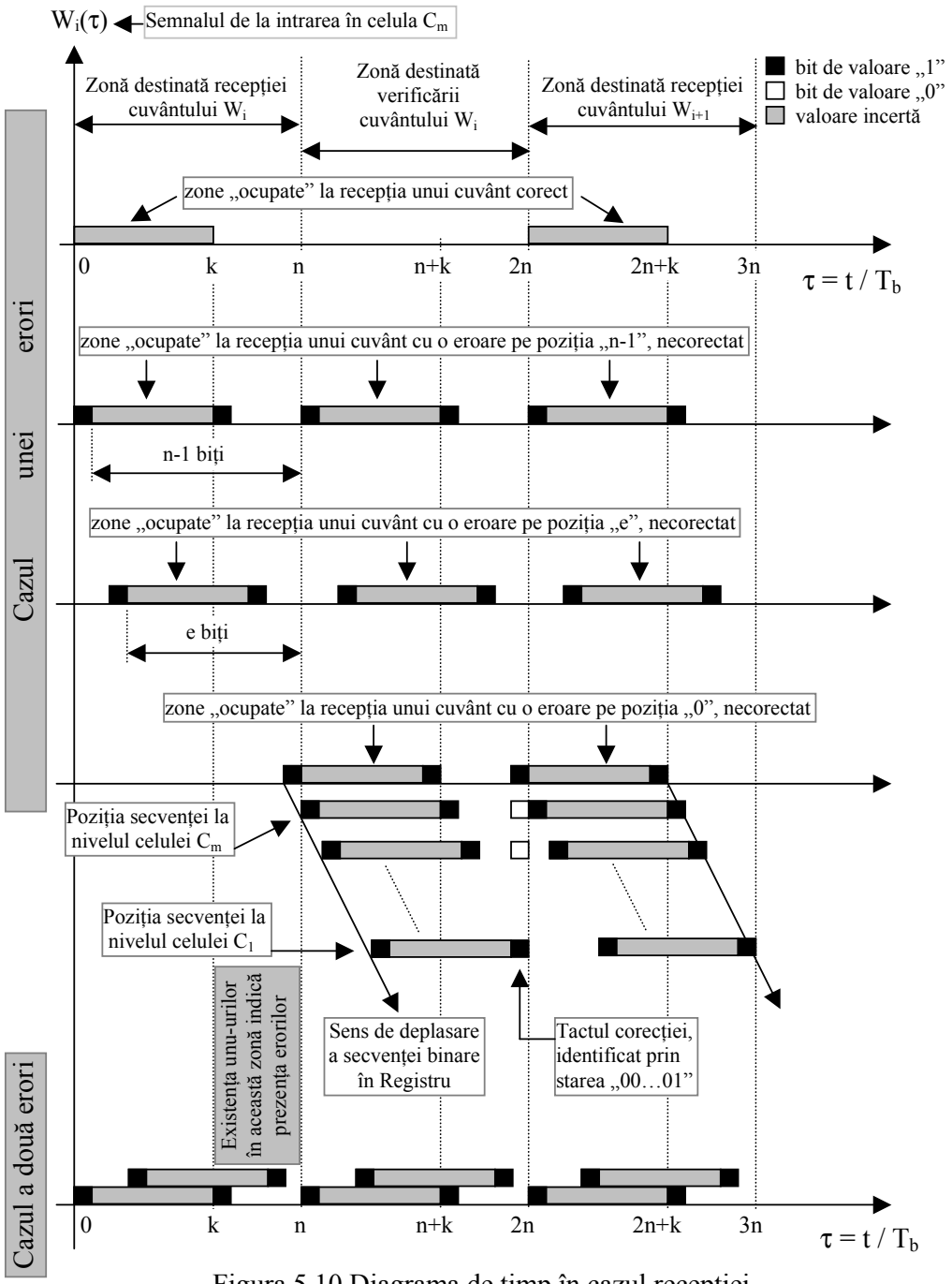


Figura 5.10 Diagrama de timp în cazul recepției.

- Dacă eroarea nu se corectează, atunci ea generează o secvență infinită periodică, de perioadă n (un cod PN);
- Tactul corecției este cel în care bitul eroare a reapărut la intrarea în prima celulă;
- Zona cu m –1 zerouri apare, în secvența PN generată de eroare/erori, o singură dată pe durata unei perioade. În cazul a două (sau mai multe) erori ea nu mai indică corect poziția vreunei erori.

5.4 Coduri BCH

5.4.1 Construcţia polinomului generator al codului

Codurile BCH sunt coduri ciclice corectoare de erori multiple [3], [5]. Structura cuvântului de cod BCH, relaţia de codare, algoritmul codării şi implementarea sa, sunt identice cu cele prezentate pentru codul ciclic corector de o eroare. Deosebirea constă în construcţia şi proprietăţile polinomului generator, g(x).

Pentru a se corecta t erori dintr-un cuvânt este necesar a se preciza poziţia fiecăreia. Dacă ne referim la un cuvânt de lungime n, unde:

$$n = 2^q - 1 \tag{5.48}$$

atunci informaţia necesară pentru a preciza un caracter eronat între cele n este:

$$i_{p1} = -\log_2(1/n) \tag{5.49}$$

Pentru a include şi varianta "cuvânt fără eroare", considerăm că în acest caz se "eronează" un al n+1–lea caracter, fictiv, astfel încât informaţia necesară pentru a preciza poziţia unui caracter eronat (dintre n+1 caractere) este:

$$i_{p1} = \log_2(n+1) = q \text{ biţi} \tag{5.50}$$

Ultima relaţie furnizează numărul de biţi de control, m = q, dintr-un cuvânt de cod necesar pentru a se putea face corecţia unei erori şi totodată reprezintă argumentarea relaţiei (5.3). Dacă se doreşte a se corecta mai mult de o eroare, atunci cantitatea de informaţie furnizată de biţii de control, de valoare m (biţi), necesară corecţiei a „t" erori, cu t >1,trebuie să fie peste valoarea q. Spre exemplu, dorind a construi un cod corector de t = 2 erori, cantitatea de informaţie necesară a fi furnizată de biţii de control, în număr de m, trebuie să fie:

$$m \geq i_{p1} + i_{p2} = \log_2 [1 + n + (n+1)\cdot n/2] \approx 2\cdot q - 1, \tag{5.51}$$

adică o informaţie aproape dublă (şi implicit un număr dublu de biţi de control) faţă de cazul corecţiei unei erori.

Deoarece precizarea un caracter din cele n+1 ale câmpului Galois GF(2^q) reprezintă o informaţie de q biţi, rezultă posibilitatea de a crea „legături" între biţii de informaţie şi cei de control pentru un cuvânt de cod BCH, construind polinomul generator, g(x), astfel încât să aibă rădăcini t elemente ale câmpului respectiv. În felul acesta, deoarece orice cuvânt de cod este multiplul lui g(x), cele t rădăcini ale lui g sunt rădăcini şi pentru orice v. Această proprietate a cuvântului de cod furnizează informaţia necesară şi suficientă pentru ca decodorul să afle poziţia a t erori.

Obs. *Cele t rădăcini invocate în construcţia polinomului g(x) nu pot fi alese oricare dintre cele n+1. Asta deoarece g(x) trebuie să fie un polinom cu coeficienţi binari. g(x) este de fapt un produs de polinoame minimale, a căror rădăcini sunt elementele respective. Polinomul minimal $m_k(x)$ ce are rădăcină pe t_k, este prin definiţie polinomul cu coeficienţi binari, de gradul cel mai mic posibil, ce are ca rădăcină pe t_k. (pentru mai multe detalii vezi [5])*

5.4.2 Decodarea codurilor BCH

Aşa cum s-a arătat în paragraful precedent decodarea codurilor BCH nu este posibilă prin acelaşi procedeu ca şi la codurile ciclice corectoare de o eroare. Decodarea codurilor BCH este prezentată algoritmic, în cele ce urmează:

Pasul 1

Se verifică dacă există erori în cuvântul recepționat. În acest scop se calculează coeficienții sindrom:

$$S_{2j-1} = w(\alpha^{2j-1}) \qquad \text{cu } j = 1 \div t \tag{5.52}$$

unde w(x) este polinomul ataşat cuvântului recepționat, iar α^{2j-1}, cu j = 1÷t, sunt cele t rădăcini ale lui g(x).

Dacă $S_{2j-1} = 0$ pentru orice j, se concluzionează că s-a recepționat un cuvânt corect şi se trece la Pasul 5. Dacă există $S_{2j-1} \neq 0$, atunci:

Pasul 2

Se calculează coeficienții polinomului:

$$\begin{aligned}\sigma(x) &= (x + X_1)\cdot(x + X_2) \ldots (x + X_t) = \\ &= x^t + \sigma_1\cdot x^{t-1} + \ldots + \sigma_{t-1}\cdot x + \sigma_t\end{aligned} \tag{5.53}$$

unde $X_j = \alpha^{i_j}$ sunt locatorii erorilor, i_j precizând poziția erorii. Coeficienții σ_1 la σ_t se calculează funcție de coeficienții sindrom după cum urmează:

–ținând cont de faptul că w = v + ε şi că $v(\alpha^{2j-1}) = 0$, ∀j = 1÷t, rezultă

$$\begin{aligned}S_{2j-1} &= w(\alpha^{2j-1}) = \varepsilon(\alpha^{2j-1}) = \\ &= \sum_{i=1}^{t}\left(\alpha^{i_j}\right)^{2j-1} = \sum_{i=1}^{t} X_i^{2j-1} \qquad \text{cu } j = 1 \div t\end{aligned} \tag{5.54}$$

–pe de altă parte, înlocuind x cu X_j în (5.53) se poate forma sistemul:

$$\sum_{i=1}^{t}\sigma_j \cdot X_j^{t-i} = X_j^t \qquad \text{cu } j = 1 \div t \tag{5.55}$$

–înmulțind pe rând ecuațiile cu X_j^p, p = 1÷t, şi sumându-le obținem sistemul

$$\begin{cases} \sigma_1\cdot S_t + \sigma_2\cdot S_{t-1} + \ldots + \sigma_t\cdot S_1 = S_{t+1} \\ \sigma_1\cdot S_{t+1} + \sigma_2\cdot S_t + \ldots + \sigma_t\cdot S_2 = S_{t+2} \\ \quad\ldots\ldots\ldots \\ \sigma_1\cdot S_{2t-1} + \sigma_2\cdot S_{2t-2} + \ldots + \sigma_t\cdot S_t = S_{2t} \end{cases} \tag{5.56}$$

Dintre cei 2·t coeficienți sindrom, S_j, cei pari se află cu relația

$$S_{2t} = (S_t)^2 \tag{5.57}$$

care este o consecință a faptului că se operează în câmpul binar.

Pasul 3

Cunoscând coeficienţii σ_j, cu j = 1÷t, se pot afla locatorii erorilor, X_j din ecuaţia:

$$\sum_{i=1}^{t} \sigma_j \cdot X_j^{-i} = 1 \qquad (5.58)$$

căutând soluţii de forma:

$$X_j = \alpha^{i_j} \qquad (5.59)$$

Ecuaţia (5.58) se poate modifica, ţinând cont de identitatea $\alpha^{n \cdot j} \equiv 1$:

$$\sum_{i=1}^{t} \sigma_j \cdot \alpha^{-i \cdot k} = \sum_{i=1}^{t} \sigma_j \cdot \alpha^{n \cdot i} \cdot \alpha^{-i \cdot k} = \sum_{i=1}^{t} \sigma_j \cdot \alpha^{(n-i) \cdot k} = \sum_{i=1}^{t} \sigma_j \cdot \alpha^{j \cdot i} = 1 \quad ? \qquad (5.60)$$

Pasul 4

Se inversează biţii corespunzători poziţiilor j pentru care este satisfăcută relaţia (5.60).

Pasul 5

Se selectează biţii de informaţie, primii k biţi din cuvântul recepţionat, verificat şi eventual corectat.

5.4.3 Codul Golay

Codul Golay împreună cu codul Hamming corector de o eroare sunt singurele coduri corectoare de erori perfecte. Codul Golay are parametrii: n = 23, k = 12 şi m = 11. Este un cod corector de 3 erori.

Este un cod perfect fiindcă numărul cuvintelor eroare corectabile este egal cu cel al corectorilor:

$$C_{23}^1 + C_{23}^2 + C_{23}^3 = 2^{11} - 1 \qquad (5.61)$$

Polinomul generator poate fi ales dintre:

$$g_1(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$
$$g_1(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$$

Decodarea se poate face pe baza corespondenţei dintre corectori şi cuvinte eroare corectabile, prezentată în Anexa C. Spre exemplu fie cuvântul recepţionat: w = 31321305_8. Decodarea sa presupune:

1. –calculul restului r, presupunând g_1 ca şi polinom generator:

```
1 1 0 0 1 0 1 1 0 1 0 0 0 1 0 1 1 0 0 0 1 0 1 | 1 1 0 0 0 1 1 1 0 1 0 1
1 1 0 0 0 1 1 1 0 1 0 1                       | 1 0 0 0 1 0 0 0 0 1 0 0
      1 1 0 0 0 0 0 1 0 1 0 1                 |
      1 1 0 0 0 1 1 1 0 1 0 1                 |
            1 1 0 0 0 0 0 1 0 0 0 1           |
            1 1 0 0 0 1 1 1 0 1 0 1           |
                  0 0 1 1 0 0 1 0 0 0 1 = 0 6 2 1 = r
```

2. –găsirea cuvântului eroare corespunzător. În acest scop se caută în tabel linia corespunzătoare valorii „062” şi în această linie termenul al doilea (ce corespunde cifrei „1”). Rezultatul căutării fiind „160602”, rezultă că w are erori pe pozițiile 16, 6 şi 2.
3. –se corectează pozițiile găsite eronate, rezultând v = 3 1 2 2 1 3 4 7_8, care se verifică a fi cuvânt de cod.

5.5 Codurile Reed-Solomon

Codurile Reed-Solomon (RS) fac parte din categoria codurilor ciclice, însă sunt coduri nebinare [3], [5]. Spre deosebire de celelalte coduri ciclice, alfabetul codului RS nu este câmpul binar {0, 1}, ci un câmp finit de ordin superior, numit câmp Galois şi care va fi descris în paragraful următor. În acest fel, cuvintele codului RS nu sunt secvențe (succesiuni) de biți, ci de caractere. Aceste caractere pot fi reprezentate, la rândul lor, prin secvențe binare, însă sunt indivizibile din punct de vedere al codării şi decodării Reed-Solomon.

Structural, cuvintele de cod RS au aceeaşi alcătuire ca şi cele de cod ciclic:

$$u = u_{n-1}\, u_{n-2} \ldots u_1\, u_0 \qquad u_j \in GF(2^q, p(x)) \quad j = 0 \div n-1 \tag{5.62}$$

unde: u – cuvântul de cod, format din n caractere;
$u_{n-1}\, u_{n-2} \ldots u_k$ – caracterele de informație, în număr de m;
$u_{k-1}\, u_{k-2} \ldots u_0$ – caracterele de control, în număr de k;
q – ordinul câmpului;
p(x) – polinomul generator al câmpului GF.
Relația de codare are aceeaşi formă ca şi la codurile ciclice:

$$u(x) = i(x)\cdot x^k + \text{rest}\,(i(x)\cdot x^k / g(x)) \tag{5.63}$$

unde g(x) este polinomul generator al codului, al cărui construcție este prezentată în paragraful următor, iar

$$i(x) = u_{n-1}\cdot x^{m-1} + u_{n-2}\cdot x^{m-2} + \ldots + u_{k+1}\cdot x + u_k \tag{5.64}$$

este polinomul de informație.

Prin relația de codare (5.63) se obține polinomul ataşat cuvântului de cod, polinom ai cărui coeficienți sunt tocmai caracterele ce alcătuiesc cuvântul de cod dat de (5.62). Relația (5.63) indică, deasemenea, că u(x) este un multiplu al lui g(x).

Codul RS, având parametrii n, k şi m, construit după relația (5.63), este capabil să corecteze un număr e_c de caractere eronate, unde:

$$2\cdot e_c = k = n-m \tag{5.65}$$

La decodare, spre deosebire de codurile ciclice, într-un cuvânt de cod RS recepționat, în vederea corecției, este necesară atât localizarea erorii, cât şi stabilirea valorii ei.

5.5.1 Câmpul Galois GF(2^q)

Câmpul Galois este generat de un polinom primitiv p(x) de grad q şi reprezintă mulțimea claselor de resturi rezultate prin împărțirea polinoamelor cu coeficienți binari la p(x). În figura 5.11 este prezentat câmpul Galois GF(2^3, $p(x)=x^3+x+1$).

z	α^j	M	r(α)
0	0	000	0
1	1	001	1
2	α	010	α
3	α^2	100	α^2
4	α^3	011	α+1
5	α^4	110	α^2+α
6	α^5	111	α^2+α+1
7	α^6	101	α^2+1

Figura 5.11 Câmpul Galois GF(2^3)

În tabel se disting patru reprezentări distincte ale elementelor câmpului:
- zecimal sau octal (coloana "**z**");
- monom (coloana "$\boldsymbol{\alpha^j}$");
- matricial (coloana "**M**");
- polinom-clase de resturi modulo p(x) (coloana "**r(α)**").

Pe mulțimea claselor de resturi p(x) sunt definite operațiile de adunare (sumă modulo doi) şi înmulțire. Vom exemplifica modul de operare al acestor doi operatori.

Fie două elemente (clase rest):

$$a = a_{q-1}\, a_{q-2} \ldots a_1\, a_0 = \alpha^x$$
$$b = b_{q-1}\, b_{q-2} \ldots b_1\, b_0 = \alpha^y \qquad (5.66)$$

Elementul sumă este:

$$c = a+b = a_{q-1}+b_{q-1}\;\; a_{q-2}+b_{q-2} \ldots a_1+b_1\;\; a_0+b_0 = c^z \qquad (5.67)$$

iar elementul produs:

$$d = a\cdot b = d_{q-1}\, d_{q-2} \ldots d_1\, d_0 = \alpha^{x+y} \qquad (5.68)$$

Trebuie menționat că:

$$\alpha^7 = \alpha^n = \alpha^0 = 1 \qquad (5.69)$$

Exemplu: fie cele două elemente a şi b din câmpul Galois prezentat în Figura 5.11:

$a = 3 = 100 = \alpha^2$

$b = 5 = 110 = \alpha^4$

Rezultă că $c = 010 = 2 = \alpha$ *şi*

$d = \alpha^6 = 101 = 7$

Pentru a putea urmări înmulțirea utilizând doar reprezentarea zecimală, fără a face apel la coloana α^j, se procedează astfel:

$$d = a\cdot b = a +_z b -_z 1 \quad \text{dacă} \quad a +_z b \le 8$$
$$= a +_z b -_z 8 \quad \text{dacă} \quad a +_z b > 8 \qquad (5.70)$$

unde:

"·" - operatorul înmulțire în GF(2^q);

"$+_z$" şi "$-_z$" - operaţiile de sumare şi scădere din N-mulţimea numerelor naturale.

În concluzie, operaţia de înmulţire în GF(2^q) corespunde unei operaţii de sumare în zecimal. De asemenea, pentru implementarea fizică, poate fi înţeleasă ca o operaţie de numărare de la elementul "a" până la elementul "a $+_z$ b $-_z$ 1", unde "a" şi "b" au reprezentare zecimală (prima coloană în Figura 5.11). În contraparte, operaţia de divizare a lui "a" la "b" va corespunde unei diferenţe "a $-_z$ (b $-_z$ 1)" sau "a $-_z$ (b $-_z$ 8)" şi totodată unei numărări în sens descrescător de la "a_z" până la "a $-_z$ (b $-_z$ 1)" sau "a $-_z$ (b $-_z$ 8)".

5.5.2 Polinomul generator, g(x), al codului

Pentru a se corecta t erori dintr-un cuvânt este necesar a se preciza poziţia fiecăreia precum şi valoarea ei. Dacă ne referim la un cuvânt de lungime n, unde:

$$n = 2^q - 1 \tag{5.71}$$

atunci informaţia necesară pentru a preciza un caracter eronat între cele n este conţinută de un caracter din GF(2^q), relaţia (5.50). De asemenea şi valoarea erorii, ε, poate să fie orice caracter din GF(2^q):

$$w = v + \varepsilon \tag{5.72}$$

unde: w –caracterul recepţionat $\in$ GF(2^q);
v –caracterul emis $\in$ GF(2^q);
ε -valoarea erorii $\in$ GF(2^q)\{0}.

Incluzând şi cazul "eronare a caracterului fictiv", rezultă că ε poate lua orice valoare din GF(2^q), adică 2^q valori posibile. Informaţia necesară pentru a preciza valoarea ei este identică cu cea dată de (5.50).

În concluzie, pentru fiecare eroare ce se doreşte a fi corectată este necesară o informaţie egală cu 2q biţi, adică două caractere din GF(2^q). La t erori sunt necesare 2t caractere (cantitate de informaţie).

Obs. *În fapt condiţia anterioară este una suficientă, cea necesară implică mai puţină informaţie deoarece nu se poate erona un caracter de două ori.*

Cele 2t caractere de informaţie necesare soluţionării problemei corecţiei se află din 2t ecuaţii, care înseamnă tot atâtea legături (proprietăţi) pentru cuvântul recepţionat. Aceste 2t proprietăţi pentru cuvintele de cod RS sunt generate prin relaţia de codare (5.63). Prin această relaţie u(x) devine multiplul lui g(x), ceea ce înseamnă că rădăcinile lui g vor fi şi rădăcini pentru u. Rezultă necesitatea ca g să aibă 2t rădăcini. Aceste rădăcini pot fi oricare dintre cele 2^q elemente ale câmpului GF(2^q). Spre exemplu, se pot alege rădăcinile α, α^2, α^3, ... α^{2t}, datorită simplităţii şi simetriei:

$$\begin{aligned} g(x) &= (x + \alpha)(x + \alpha^2) \ldots (x + \alpha^{2t}) \\ &= x^k + g_{k-1}x^{k-1} + \ldots + g_1 x + g_0 \end{aligned} \tag{5.73}$$

Aşadar cuvântul de cod RS, u, rezultat prin codarea cu ajutorul relaţiei (5.63), în care g este dat de (5.73), are proprietatea că elementele α, α^2, α^3, ... α^{2t} sunt rădăcini pentru polinomul ataşat, u(x).

5.5.3 Decodarea codurilor Reed-Solomon

Decodarea codurilor RS este în mare parte asemănătoare decodării codurilor BCH, deosebirea constând în necesitatea de a afla pentru fiecare eroare pe lângă poziție şi valoarea ei. De remarcat că un cuvânt de cod RS este format din n caractere, adică q·n biți. Astfel o eroare de caracter poate însemna până la q erori de bit. În cele ce urmează este descrisă decodarea codurilor RS prin paşi algoritmici:

Pasul I

Conform relației (5.73) g(x) este un polinom de grad k=2t > 2. Prin construcție, cuvintele de cod RS au proprietatea că polinoamele ataşate lor sunt divizibile cu g(x), adică elementele câmpului GF (2^2) $\alpha,\alpha^2,\ldots,\alpha^{2t}$ sunt rădăcini atât pentru g(x) cât şi pentru orice cuvânt de cod u(x):

$$\begin{cases} g(\alpha^j) = 0 \\ u(\alpha^j) = 0 \end{cases} \qquad j = 1 \div 2t \tag{5.74}$$

Aceste proprietăți constituie şi punctul de plecare în decodare. Presupunând că V este un cuvânt recepționat:

$$v=u+\varepsilon \quad \text{sau} \quad v(x) = u(x) + \varepsilon(x) \tag{5.75}$$

vom calcula 2t coeficienți, numiți coeficienți sindrom, Sj, în forma:

$$S_j = v(\alpha^j) = u(\alpha^j) + \varepsilon(\alpha^j) = \varepsilon(\alpha^j) \qquad j=1 \div 2t \tag{5.76}$$

Dacă nu există erori $S_j = 0$. În acest caz se trece la pasul VI. Evident concluzia poate fi eronată. Un exemplu în argumentarea acestei afirmații este situația: ε = cuvânt de cod. Dar în acest caz numărul erorilor depăşeşte puterea de corecție de t erori.

Pasul II

Dacă există erori în limitele corectabile (numărul erorilor este mai mic sau egal cu t) atunci există coeficienți sindrom diferiți de zero. Fie cuvântul eroare în forma:

$$\varepsilon(x) = \sum_{i=1}^{t} \alpha^{r_i} \cdot x^{k_i} \tag{5.77}$$

unde:

$$Y_i = \alpha^{r_i} \tag{5.78}$$

reprezintă valoarea erorii $r_i \in \{0, 1, 2, \ldots n-1\}$, iar:

$$X_i = \alpha^{k_i} \tag{5.79}$$

reprezintă locatorul erorii $k_i \in \{0, 1, 2, \ldots, n-1\}$. Cu aceste notații coeficienții sindrom au expresiile:

$$S_j = \sum_{i=1}^{t} Y_i \cdot \left(\alpha^j\right)^{k_i} = \sum_{i=1}^{t} Y_i \cdot X_i^j \qquad , \qquad j = 1 \div 2t \tag{5.80}$$

Ecuaţiile (5.80) reprezintă un sistem de 2t ecuaţii cu 2t necunoscute: t locatori ai erorilor X_i şi t valori pentru respectivele erori Y_i. Rezolvarea acestui sistem de ecuaţii se va face în mai multe etape. La pasul prezent se vor calcula coeficienţii polinomului $\sigma(x)$ ai cărui rădăcini sunt locatorii erorilor:

$$\sigma(x) = \prod_{i=1}^{t}(x + X_i) = x^t + \sigma_1 \cdot x^{t-1} + ... + \sigma_{t-1} \cdot x + \sigma_t \tag{5.81}$$

Pentru că X_i , $1 \le i \le t$ este o rădăcină a lui $\sigma(x)$ putem scrie:

$$X_i^t + \sigma_1 \cdot X_i^{t-1} + ... + \sigma_{t-1} \cdot X_i + \sigma_t = 0 , \quad i = 1 \div t \tag{5.82}$$

Înmulţind ecuaţiile (5.82) pe rând cu $X_i^k Y_i$ şi sumându-le obţinem ecuaţia:

$$\begin{aligned} &\sum_{i=1}^{t} Y_i \cdot X_i^{t+k} + \sigma_1 \cdot \sum_{i=1}^{t} Y_i X_i^{t+k-1} + ... + \sigma_{t-1} \cdot \sum_{i=1}^{t} Y_i \cdot X_i^{k+1} + \\ &+ \sigma_t \cdot \sum_{i=1}^{t} Y_i \cdot X_i^k = 0 \end{aligned} \tag{5.83}$$

sau, ţinând cont de (5.80) pentru k luând valorile 1,2,..,t:

$$S_{t+k} + \sigma_1 \cdot S_{t+k-1} + \ldots + \sigma_{t-1} \cdot S_{k-1} + \sigma_t \cdot S_k = 0 , \quad k = 1 \div t \tag{5.84}$$

Ecuaţiile (5.84) reprezintă un sistem de t ecuaţii cu t necunoscute (coeficienţii σ_i) a cărui rezolvare constituie obiectivul acestui pas algoritmic. Ecuaţiile (5.84) pot fi puse sub forma compactă:

$$A_s \cdot \sigma = B_s \tag{5.85}$$

unde:

$$A_s = \begin{bmatrix} S_t & S_{t-1} & \ldots & S_1 \\ S_{t+1} & S_t & \ldots & S_2 \\ \ldots & \ldots & \ldots & \ldots \\ S_{2t-1} & S_{2t-2} & \ldots & S_t \end{bmatrix}; \quad \sigma = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \ldots \\ \sigma_t \end{bmatrix}; \quad B_s = \begin{bmatrix} S_{t+1} \\ S_{t+2} \\ \ldots \\ S_{2t} \end{bmatrix} \tag{5.86}$$

Calculând inversa matricii A_s găsim soluţia sistemului (5.85) în forma:

$$\sigma = A_s^{-1} \cdot B_s \tag{5.87}$$

Obs: *Toate calculele trebuiesc făcute în câmpul GF(2²), atât coeficienţii sindrom, S_j, cât şi coeficienţii σ fiind elemente ale respectivului câmp.*

În rezolvarea ecuaţiei (5.85) pot apărea trei situaţii:

1° rangul matricii A_s este $e<t$ şi este egal cu al matricii $[A_s B_s]$. În acest caz numărul de erori este e şi din rezolvarea ecuaţiei (5.85) rezultă un număr e de coeficienţi σ_I nenuli

Rezolvarea ecuației (5.85) presupune restrângerea sistemului (5.84) la un număr e<t de ecuații cu e necunoscute, rezolvabil.

2° rangul matricii A_s este t. În acest caz există A_s^{-1} iar ecuația (5.8) are soluție dată prin relația (5.86).Se vor găsi t erori în acest caz.

3° rangul matricii A_s este e <t şi este mai mic decât al matricii $[A_sB_s]$. O astfel de situație este posibil să apară dacă numărul erorilor depăşeşte t. În acest caz se semnalează prezența erorilor în număr necorectabil. Funcție de aplicație se va abandona cuvântul în cauză sau se va cere retransmisia sa.

Pasul III

Ecuația (5.82) se poate re-scrie în forma:

$$\sum_{j=1}^{t} \sigma_j \cdot X_i^{-j} = 1 \qquad (5.88)$$

Ştiind că X_i este de forma $X_i = \alpha^{k_i}$ unde k_i indică rangul pe care îl ocupă eroarea (ex k_i=n-1 este prima poziție) se vor putea afla locatorii erorilor printr-o operație de căutare:

$$\sum_{j=1}^{t} \sigma_j \cdot \alpha^{k \cdot j} = 1 \quad ? \qquad k = 1, 2, \ldots, n \qquad (5.89)$$

Acei k pentru care (5.89) este o identitate, indică prezența erorii pe poziția:

$$r = n-k \qquad (5.90)$$

Obs: *Înlocuind pe:*

$$X_r = \alpha^r \qquad (5.91)$$

în (5.88) şi utilizând identitatea $\alpha^n = 1$ *obținem:*

$$\sum_{j=1}^{t} \sigma_j \cdot \alpha^{-j \cdot r} = \sum_{j=1}^{t} \sigma_j \cdot \alpha^{n \cdot j} \alpha^{-j \cdot r} = \sum_{j=1}^{t} \sigma_j \cdot \alpha^{(n-j) \cdot r} = \sum_{j=1}^{t} \sigma_j \cdot \alpha^{k \cdot jr} = 1$$

Pasul IV

Dispunând de pozițiile erorilor dispunem implicit de numărul lor. Reținem că problema are soluție doar dacă e<t. Cunoscând aşadar e locatori ai erorilor în forma $X_i = \alpha^{k_i}$, i = 1÷e, din sistemul de ecuații (5.80) se reții e ecuații în vederea aflării valorilor Y_i pentru cele e erori. Acest sistem este compatibil unic determinat. Rezolvarea sa conduce la aflarea celor e valori necesare Y_i.

PasulV

Cunoscând atât pozițiile erorilor $X_i = \alpha^i$, i = 1÷e, cât şi valorile lor $Y_i = \alpha^{r_i}$, i=1÷e putem face corecția caracterelor eronate:

$$u_{k_i} = v_{k_i} + Y_i, \qquad i = 1 \div e \qquad (5.92)$$

Pasul VI

Se face selecția caracterelor de informație şi livrarea lor la ieşire.

O altă metodă de decodare a codurilor R-S o reprezintă algoritmul Berlekamp-Massey. În Anexa F se prezintă un exemplu de decodare ce utilizează acest algoritm. Spre deosebire de metoda prezentată anterior (cunoscută sub denumirea de algoritmul Peterson), algoritmul Berlekamp-Massey, este o decodare „în frecvență” şi ca atare o metodă mai rapidă pentru dimensiuni mari ale codului R-S.

Alături de codurile convoluționale, codurile R-S constituie soluții atractive pentru protecția informației în sistemele cu spectru împrăştiat. Codurile R-S sunt cu precădere utilizate în combinație cu FH, [15] şi [16].

6. Coduri convoluționale

6.1 Coduri convoluționale –descriere generală

6.1.1 Codor convoluțional

Un codor convoluțional poate fi văzut ca un nivel adițional al filtrării digitale liniare (peste câmpul binar) care introduce redundanță în secvența de date originală. Această redundantă este deja prezentă în sistemele cu spectru împrăştiat şi disponibilă pentru exploatare. Pentru uşurința descrierii codurilor convoluționale, aceasta va fi făcută în continuare printr-un exemplu. Figura 6.1 prezintă unul dintre codoarele convoluționale simple dar non-triviale.

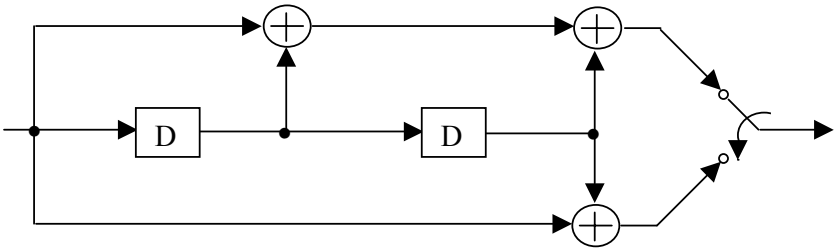
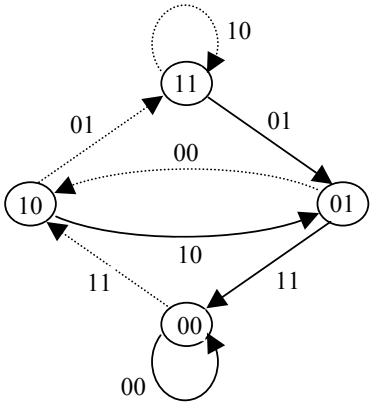


Figura 6.1 Codor convoluțional cu K=3, r=1/2

Acest codor este o „maşină" cu stări finite ce poate fi uşor descrisă în termenii diagramei proprii de stare.



Notă: Starea corespunde ultimelor două intrări, *cel mai din stânga* este cel mai recent. În registrul de deplasare, cea mai recentă intrare rezidă în *cea mai din stânga stare.*

Figura 6.2 Diagrama de stare pentru codorul din Figura 6.1

Aceasta se vede în Figura 6.2, unde nodurile sau stările se referă la conţinutul registrului. Intrările „0” şi „1” sunt indicate pe ramuri, care reprezintă tranziţiile între noduri: fiecare ramură este o linie plină pentru intrare „0” sau o linie întreruptă este pentru „1”. Ieşirile codorului sunt constituite din două simboluri per bit de intrare la tranziţie. Astfel, pentru condiţia sau starea iniţială 00, secvenţa de intrare 11010 produce secvenţa de ieşire 1101010010. Aceasta se poate vedea fie prin codorul cu registru de deplasare din Figura 6.1 fie din reprezentarea diagramei de stare, Figura 6.2.

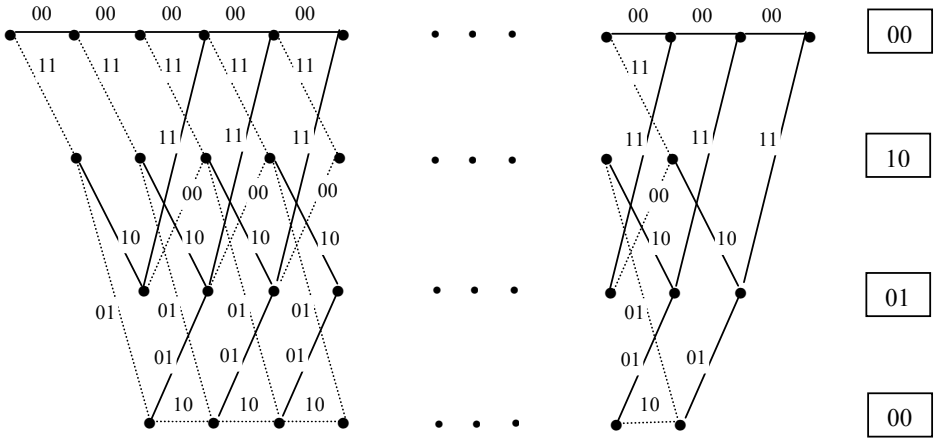


Figura 6.3 Reprezentarea trellisului pentru codorul din Figura 6.1

6.1.2 Diagrama trellis

Cu toate că nu este neapărat necesar pentru analizarea caracteristicilor codului sau performanţei decodorului optimal, este folositor pentru înţelegerea ambelor, de a expune codul în diagrama trellis[1] (Figura 6.3). Diagrama trellis este o replică infinită a diagramei de stare. Nodurile (stările) de la un anume nivel din trellis se obţin din nodurile (stările) nivelului anterior prin tranziţia dată de o ramură, corespunzătoare unui bit de intrare, aşa cum s-a determinat prin diagrama de stare. Orice cuvânt de cod al unui cod convoluţional corespunde simbolurilor de-a lungul căii (constând în ramuri succesive) în diagrama trellis. Performanţa sistemului codat depinde de *distanţa relativă Hamming* dintre cuvintele de cod: numărul de simboluri prin care diferă. *Distanţa „liberă”* se defineşte ca şi distanţa Hamming minimă dintre oricare două căi peste toată porţiunea de ne-suprapunere.

Este simplu să se determine distanţa pentru toate căile faţă de calea nulă peste porţiunea pe care ele sunt diferite faţă de ultima. Din fericire, acest set al tuturor distanţelor căilor relativ la calea nulă este de asemenea setul tuturor distanţelor faţă de orice altă cale. Această proprietate de simetrie rezultă din faptul că un codor liniar transformă setul tuturor secvenţelor binare de intrare posibile într-un set închis de cuvinte de cod sub adunarea modulo 2. Astfel, dacă codorul transformă secvenţa de biţi de intrare u_i în secvenţa x_i şi u_j în x_j, atunci secvenţa $x_i \oplus x_j$, este de asemenea un cuvânt de cod, ce a fost generat din secvenţa de intrare $u_i \oplus u_j$. (Sumele oricăror două secvenţe de intrare posibile sunt ele însele secvenţe

[1] Trellis este un termen, formulat de Forney [45], care descrie un arbore în care ramurile nu doar se bifurcă în două sau mai multe ramuri ci de asemenea în care două sau mai multe ramuri se pot uni într-una.

de intrare posibile, deoarece orice secvenţă binară poate fi o secvenţă de intrare). Este la fel de evident că distanţa dintre oricare două secvenţe cuvinte de cod x_i şi x_j este exact ponderea (numărul de „unu-uri") sumei lor modulo 2, $w(x_i \oplus x_j)$.

Considerăm acum setul tuturor distanţelor de la secvenţa nulă. Este evident

$$\{w(0 \oplus x_i) \text{ pentru orice } i\} = \{w(x_i) \text{ pentru orice } i\} \tag{6.1}$$

deoarece orice secvenţă la care se adună numai zero rămâne neschimbată. Similar, setul tuturor distanţelor de la orice altă secvenţă de ieşire specificată x_m este

$$\{w(x_m \oplus x_i) \text{ pentru orice } i\} = \{w(x_i) \text{ pentru orice } i\} \tag{6.2}$$

Aceasta rezultă din faptul că suma modulo 2 pentru oricare două cuvinte de cod este un alt cuvânt de cod (set închis), şi astfel $x_m \oplus x_j \neq x_m \oplus x_k$ doar dacă $x_j = x_k$, şi de aceea numărul componentelor setului este acelaşi cu numărul cuvintelor din setul original. Atunci, concluzionăm din (6.1) şi (6.2) că setul distanţelor de la 0 este acelaşi cu setul distanţelor de la (către) oricare alt cuvânt de cod.

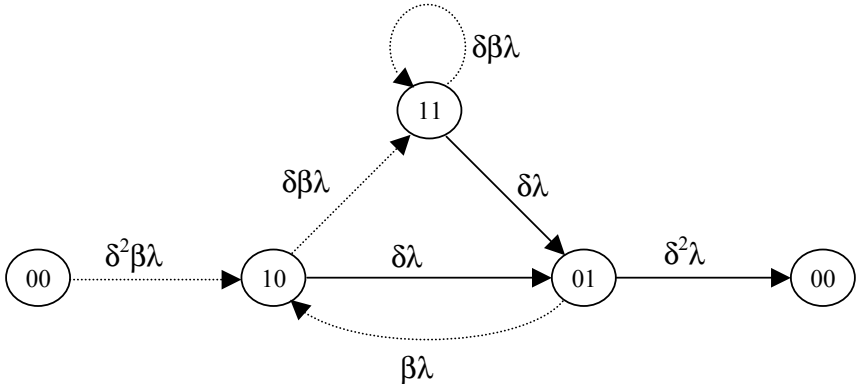


Figura 6.4 Diagrama de stare etichetată cu distanţă, lungime şi număr de unu-uri în intrare

6.1.3 Distanţa de cod

Este uşor de stabilit din Figura 6.3 prin urmărirea căilor de-a lungul trellisului că, pentru calea care diverge de la cea nulă (calea de referinţă) la primul (sau oricare alt) nod, revine la cea nulă, cel mai devreme, după trei ramuri, cu o distanţă Hamming acumulată de 5. Două căi revin cu o distanţă acumulată 6, una după patru ramuri (generată prin intrare 1100) iar una după cinci ramuri (generată prin intrarea 10100); şi aşa mai departe. Aceeaşi concluzie poate fi obţinută prin urmărirea căilor pe diagrama de stare din Figura 6.2. Figura 6.4 arată diagrama de stare etichetată cu diferite monoame de literele δ, β şi λ. În figură, nodul 00 este despicat în două şi denotă nodurile *iniţial* şi *final*. Aceasta reflectă faptul că se consideră toate căile ce diverg de la calea nulă şi revin la ea după câteva tranziţii. Monoamele ce etichetează ramurile indică: –ponderea ramurii (numărul de simboluri de ieşire ce sunt unu-uri) ca şi exponent al lui δ, –ponderea bitului de intrare (1 pentru „1" şi 0 pentru „0") ca şi exponent al lui β, şi –durata ramurii, în biţi de intrare, ca şi exponent al lui λ (unitar în toate cazurile). Multiplicând monoamele ramurilor pe orice cale ce trece de la nodul iniţial la cel final rezultă

monom cu exponenţi de δ, β şi λ. Acestea sunt, respectiv, distanţa totală a secvenţei cuvânt de cod de la calea nulă peste porţiunea de ne-suprapunere; numărul de intrări prin care ea diferă de intrarea nulă; şi lungimea porţiunii de ne-suprapunere. Astfel, de exemplu, produsele monom pentru cele două căi în cauză sunt (Anexa G):

$$\delta^5\beta\lambda^3 \text{ şi } \delta^6\beta^2\lambda^4.$$

Putem obţine funcţia generatoare pentru toate căile ce diverg de la starea 0 la un nod particular şi revin la un moment viitor (sau prin simetrie, revin la 0 la un nod particular şi diverg la orice moment anterior –atribuind procesul de start la momentul de timp negativ infinit). Aceasta este funcţia de transfer a fluxului grafului din Figura 6.4, ce este uşor calculată[1] a fi

$$T(\delta,\beta,\lambda) = \frac{\delta^5 \cdot \beta \cdot \lambda^3}{1 - \delta\beta\lambda \cdot (1+\lambda)} \tag{6.3}$$

Dacă doar distanţele de cod şi numărul de intrări pe unu contează, putem ignora lungimea căii setând λ=1 şi obţinând

$$T(\delta,\beta) = T(\delta,\beta,\lambda)\big|_{\lambda=1} = \frac{\delta^5 \cdot \beta}{1 - 2\delta\beta} \tag{6.4}$$

Dacă doar distanţele de cod interesează, putem de asemenea seta β = 1 şi obţinem

$$T(\delta) = T(\delta,\beta)\big|_{\beta=1} = \frac{\delta^5}{1 - 2\delta} \tag{6.5}$$

Dezvoltând această ultimă funcţie generatoare (prin împărţirea polinoamelor), obţinem

$$T(\delta) = \delta^5 + 2\cdot\delta^6 + 4\cdot\delta^7 + \ldots + 2^k\cdot\delta^{5+k} + \ldots \tag{6.6}$$

Aceasta înseamnă că o cale cu retur la zero acumulează distanţă 5 (diferă de cea nulă prin cinci simboluri), două au distanţă 6, şi în general 2^k căi au distanţa 5 + k. Similar divizând polinoamele din (6.4) găsim

$$T(\delta,\beta) = \delta^5\cdot\beta + 2\cdot\delta^6\cdot\beta^2 + 4\cdot\delta^7\cdot\beta^3 + \ldots + 2^k\cdot\delta^{5+k}\cdot\beta^{1+k} + \ldots \tag{6.7}$$

Aceasta însemnă că în general 2^k căi, cu retur la zero, cu distanţă acumulată 5 + k, corespund unor secvenţe de intrare ce conţin 1 + k unu-uri. Această ultimă funcţie generatoare se va dovedi utilă în determinarea probabilităţilor de eroare de bit. De asemenea rezultă că exponentul lui δ din primul termen din dezvoltare este distanţa liberă a codului, 5 în acest caz.

[1] fie rezolvând un set de ecuaţii liniare, fie prin regula lui Mason [Mason, 1956]

6.2 Decodor maximum plauzibil –algoritmul Viterbi

6.2.1 Decodarea MAP

Am stabilit structura codurilor convoluționale prin intermediul diagramei de stare şi diagramei trellis. Aceleaşi concepte pot să conducă către decodorul optim al codului convoluțional în canalul fără memorie –unul în care impactul aleatoriu al canalului asupra simbolurilor de cod este independent de la simbol la simbol, –independență posibil a fi dobândită prin intermediul întrețeserii. Fie secvența simbolurilor de intrare x şi secvența corespunzătoare de la ieşirea din canal y. Atunci, câtă vreme canalul este fără memorie, probabilitatea condiționată, sau funcția de plauzibilitate, este dată prin

$$p(y/x) = \prod_{\text{toti } k} p(y_k/x_k) \tag{6.8}$$

Deoarece timpul este totdeauna mărginit, domeniul indexului poate fi luat ca un set finit de întregi. Dacă căutăm cea mai plauzibilă cale, şi astfel căutăm să minimizăm probabilitatea erorii secvenței peste toate secvențele posibile (sau peste căile prin trellis) când toate sunt a priori echiprobabile, atunci trebuie să maximizăm (6.8) peste toate secvențele de intrare x. Atunci maximul corespunde celei mai plauzibile secvențe de cod de intrare în canal, iar secvența corespunzătoare de intrare în codor este cea mai plauzibilă secvență de informație. Echivalent, trebuie să căutăm maximul logaritmului lui p(y/x) din (6.8), care este o funcție aditivă de a logaritmului plauzibilității de simbol

$$\Lambda(x,y) = \ln p(y/x) = \sum_{\text{toti } k} \ln p(y_k/x_k). \tag{6.9}$$

6.2.2 Metrica ramurii

Pentru a putea implementa algoritmul decodării este mult mai convenabil să se organizeze (6.9) ca o sumă de sume parțiale peste ramuri. Se defineşte *metrica* ramurii j ca

$$\mu_j = \mu(x_j,y_j) = \ln p(y_j/x_j), \tag{6.10}$$

unde vectorii sunt n dimensionali cu n fiind numărul de simboluri pe ramură. Atunci,

$$\mu_j = \ln p(y_j/x_j) = \sum_{k=1}^{n} \ln p(y_{jk}/x_{jk}). \tag{6.11}$$

(În exemplul prezent, Figurile 6.1 la 6.4, n = 2, dar se poate generaliza uşor la valori întregi arbitrare.) Atunci pentru calea cu secvența de intrare în canal x şi ieşirea sa y, logaritmul plauzibilității din (6.9) poate fi scris ca

$$\Lambda(x,y) = \sum_{\text{j=toate ramurile}} \mu(y_j/x_j). \tag{6.12}$$

Atunci decodorul maxim plauzibil alege calea pentru care suma metricilor ramurilor, aşa cum s-a definit prin (6.10) şi (6.11), este maximă. Figurile 6.5a şi b ilustrează două exemple de canal şi decodarea corespunzătoare pentru codul considerat în Figura 6.1. În fiecare caz, secvența de ieşire y se vede ca şi referința figurii. În (a), canalul este un canal binar simetric (BSC), unde z este o secvență binară ce diferă de secvența de cod de la intrarea în canal în

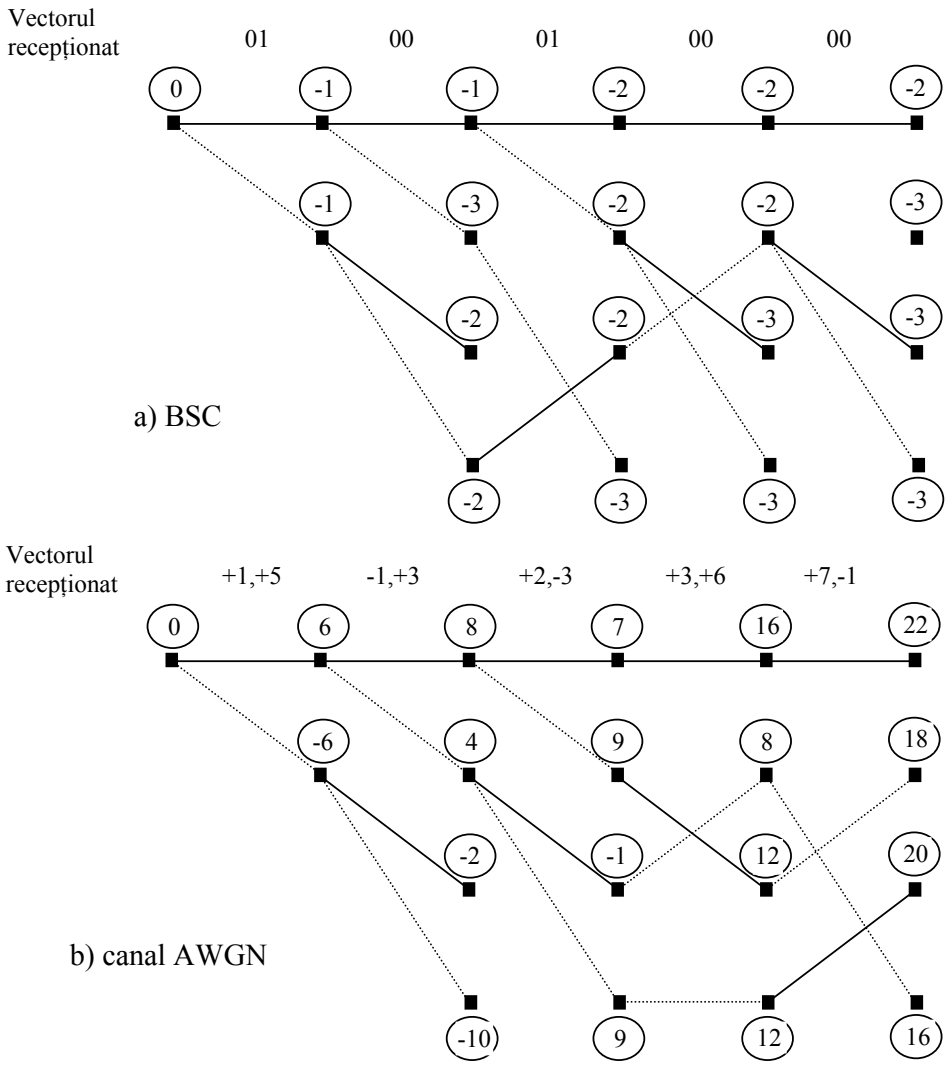


Figura 6.5 Exemplu de decodare pentru codorul din Figura 6.1

poziţiile de simbol în care au apărut erori. În acest caz, luând probabilitatea erorii de simbol ca $p<½$, metrica ramurii

$$\mu_j = \mu(x_j,y_j) = \ln\left[p^{d_j}\cdot(1-p)^{n-d_j}\right]$$
$$= d_j\cdot\ln\left(\frac{p}{1-p}\right) + n\cdot\ln(1-p), \qquad (6.13)$$

unde $d_j = d(x_j,y_j)$ este distanţa Hamming dintre x_j şi y_j pe ramura j. Dar scopul este să maximizăm suma metricilor ramurilor peste întreaga cale. De aceea, putem să adunăm o constantă arbitrară la fiecare ramură şi să scalăm rezultatul printr-un număr arbitrar pozitiv, fără să schimbăm liniile rândurilor relative. Notând că $\ln[p/(1-p)]<0$ pentru $p<½$, putem înlocui metrica prin metrica scalată

$$\hat{\mu} = -d_j = -d(x_j, y_j) \qquad \text{(BSC)} \qquad (6.14)$$

fără să schimbăm ordinea relativă pentru toate căile. În exemplul Figurii 6.5a, fiecare ramură are metrica 0,-1 sau –2, depinzând de care simboluri de cod diferă de simbolurile recepționate în pozițiile 0, 1 sau 2. (vezi Anexa G)

Pentru exemplul din Figura 6.5b, secvența de simboluri de ieşire y este ieşirea pentru un canal cu zgomot aditiv, alb, gaussian (AWGN) pentru care secvența transmisă normalizată este +1 sau –1, atunci când simbolul de cod a fost „0" sau „1", respectiv. Aşadar, deoarece $x_{jk}^2 = 1$,

$$\mu_j = \ln p(y_j/x_j) = \ln\left\{\exp\left[-\sum_{k=1}^{n}\left(y_{jk} - \sqrt{E_S}\cdot x_{jk}\right)^2 / I_0\right] / (\pi \cdot I_0)^{n/2}\right\}$$
$$= \frac{1}{I_0}\cdot\left(-\sum_{k=1}^{n} y_{jk}^2 - n\cdot E_S + 2\cdot\sqrt{E_S}\cdot\sum_{k=1}^{n} x_{jk}\cdot y_{jk}\right) - n\cdot\ln(\pi\cdot I_0)/2\,, \qquad (6.15)$$

unde E_S este energia per simbol. Prima sumă este evident aceeaşi pentru ramura j pe toate căile. Atunci, regrupând din nou termenii comuni şi scalând rezultatul prin I_0, metrica ramurii poate fi înlocuită prin

$$\hat{\mu}_j = \sum_{k=1}^{n} x_{jk}\cdot y_{jk} \qquad \text{(AWGN).} \qquad (6.16)$$

Atunci fiecare ramură din exemplul din Figura 6.5b este produsul interior al simbolurilor sale de cod (luate ca şi +1 sau –1) cu ieşirile canalului recepționate (vizibile în partea de sus a figurii)[1]. Aceste exemple se pot generaliza în mod evident la orice canal fără memorie, aşa cum s-a definit prin (6.8) la (6.12).

6.2.3 Algoritmul Viterbi

Găsirea celei mai plauzibile căi aşadar constă în găsirea căii pentru care suma metricilor ramurilor sale (sau modificărilor sale) este cea mai mare peste toate căile. Pentru L biți succesivi, există L noduri cu două ramuri generate de la fiecare, astfel că prin această tehnică, „forță brută", avem nevoie să examinăm 2^L căi –o sarcină disperată pentru fiecare secvență de biți de intrare de lungime moderată. Exemplele precedente sugerează o metodă simplă, totuşi optimală, prin care efortul computațional poate creşte doar liniar cu L. Aceasta rezultă din observația că pentru orice două căi ce converg într-un nod dat, putem exclude calea cu cea mai mică sumă de metrică. Se defineşte metrica de stare la orice nod i de la momentul k, $M_i(k)$, ca şi suma metricilor ramurilor peste acea stare (nod) care este cea mai mare dintre sumele metricilor ramurilor pentru toate căile acelui nod. Atunci, avem relația de recurență

$$M_i(k+1) = \text{Max}\,[M_{i'}(k) + m_{i'i},\, M_{i''}(k) + m_{i''i},]. \qquad (6.17)$$

i' şi i'' sunt nodurile care au tranziții posibile către nodul i într-o ramură (durată de bit), iar $m_{i'i}$ şi $m_{i''i}$ sunt metricile ramurilor μ peste ramurile de tranziție în chestiune.

[1] Pentru notație compactă, toate metricile ramurilor sunt luate a fi întregi, pe când în general ele vor fi numere reale. Pe de altă parte, metricile sunt numere întregi pentru AWGN cuantizat.

Operaţia din (6.17) este ilustrată în exemplele Figurii 6.5a şi 6.5b, cu $M_i(k)$ privite ca valori circulare la fiecare nod şi fiecare nivel din trellis. Aceasta este operaţia fundamentală ce reduce căutarea pentru calea maxim plauzibilă dintre 2^L operaţii (calcule) pentru o cale cu L ramuri la numai 4L, în acest caz. La fiecare stare şi fiecare nivel din trellis ea implică o operaţie *adună—compară—selectează (ACS)*, unde

> *Adună* se referă la adunarea fiecărei metrici de stare de la nivelul precedent la cele două metricii ale ramurilor pentru tranziţiile permise.
>
> *Compară* se referă la compararea perechilor pentru astfel de sume de metrici pentru căile ce intră într-o stare (nod) de la un nivel dat.
>
> *Selectează* se referă la selecţionarea celui mai mare dintre cele două şi îndepărtarea celeilalte. Astfel, numai ramura câştigătoare este păstrată la fiecare nod, împreună cu metrica de stare a nodului. Dacă cele *două cantităţi* fiind comparate sunt *egale, oricare ramură* poate fi selectată, pentru că probabilitatea de selecţie eronată va fi aceeaşi în fiecare caz.

Ecuaţia (6.17), în toată simplitatea ei, este cunoscută ca şi algoritmul Viterbi [1967a]. Pentru implementarea decodorului corespunzător acestui algoritm sunt de asemenea necesare memorarea a două seturi de date. Primul este, bineînţeles, starea căii $M_i(k)$ recalculată pentru fiecare nivel succesiv k, dar necesitând doar patru registre pentru exemplul urmărit aici. Aceasta se numeşte *memoria metricii*. Dacă această metrică este un număr real, ca şi în cazul AWGN, trebuie cuantizată cu acurateţe rezonabilă. Însă, în toate cazurile practice virtuale observaţiile sunt cuantizate înainte de generarea metricii, făcând şi observaţiile şi variabilele metrici discrete[1].

Al doilea set de date ce este memorat sunt selecţiile de la fiecare nod sau stare. Numită *memoria căii*, care de asemenea necesită patru registre de lungimi arbitrare, ea memorează selecţiile de la fiecare nivel înainte ca decizia finală să poată fi făcută. Calea de a obţine această decizie finală peste toţi biţii este aproape evidentă dacă forţăm codorul să revină la starea iniţială nulă. Aceasta se poate face numai prin inserarea a două zerouri (în exemplul nostru) la sfârşitul şuvoiului de biţi de durată arbitrară, B. Această procedură, numită „tailing off" („închiderea cozii"), face codul convoluţional să fie un cod bloc de lungime B biţi, cu B + 2 ramuri şi 2(B + 2) simboluri. Decizia finală este luată atunci printr-o procedură numită „chaining back" („înlănţuire înapoi"): pornind cu ultimul nod, trasăm calea deciziei înapoi de la ultima decizie către prima.

Însă nu este necesar să închidem coada codului convoluţional într-un cod bloc (inserând zerouri „sterile") cu scopul de a face o decizie. Cu o foarte mică degradare a performanţei datorită sub-optimalităţii, după fiecare set de selecţii de nod, facem lanţul înapoi de la starea cu cea mai mare metrică[2] (dintre cele patru la acel nivel de nod) pentru o lungime suficientă (să spunem 10 ramuri). Facem decizia finală a acestui bit corespunzătoare celei mai plauzibile dintre cele două ramuri (biţi) ce au plecat de la nodul la care avem lanţul înapoi. Dăm la ieşire bitul pe care avem lanţul înapoi şi-l ştergem din memoria căii. Acest proces, numit *trunchierea memoriei*, conduce la aproape aceeaşi performanţă ca şi pentru codul cu coadă închisă, dacă lanţul înapoi este suficient de lung. Aceasta deoarece probabilitatea este

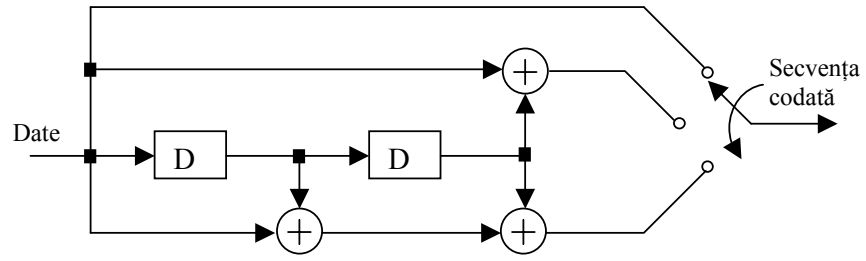
[1] Pentru a păstra metricile tuturor sensurilor în sus şi în jos la valori ce sunt rangul memoriei, este necesar să normalizăm ocazional prin adunarea sau scăderea aceleiaşi cantităţi la toate registrele. Este uşor de văzut că diferenţa dintre metrici de la acelaşi nivel de nod este totdeauna limitată [Viterbi şi Omura, 1979].

[2] Putem alege alternativ de a înlănţui înapoi de la o stare arbitrară, cu ceva mai multă degradare. Aceasta poate fi recuperată prin extinderea înlănţuirii înapoi la o lungime mai mare de ramuri.

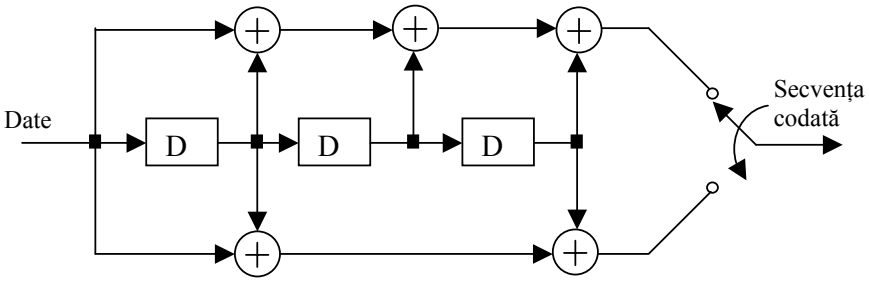
foarte mică pentru calea incorectă, ce nu se suprapune peste cea corectă pe o porțiune mare, menținând o metrică mai mare decât calea corectă.

6.3 Generalizarea exemplului precedent

Două generalizări ale exemplului de cod convoluțional considerate până acum vor fi evidente: lungimea registrului de deplasare a codorului şi numărul de surse (robinete, guri)



a)



b)

Figura 6.6 Codoare convoluționale pentru: a) K= 3, R = ½; b) K= 4, R = ½;

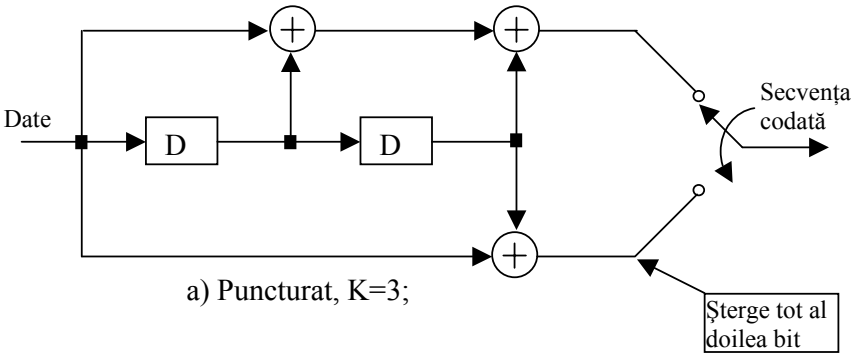
de ieşire. Numărul de stări (celule) din codor, care este egal cu numărul de elemente de întârziere plus unu, se numeşte *lungimea de contrângere*, K. Astfel, exemplul în curs din Figura 6.1 are K = 3, pe când exemplele din Figurile 6.6a şi b au K = 3 şi, respectiv, K = 4. De asemenea, exemplul în curs are două surse de ieşire, pe când cel din Figura 6.6a are trei. Numărul de surse determină *rata codului* convoluțional, definită ca şi numărul de biți de intrare per simbol de ieşire. Astfel, rata codului pentru exemplul în curs este R = ½, pe când cea a exemplului din Figura 6.6a este R = 1/ 3. Atunci va fi evident că toate procedurile descrise până acum pot fi aplicate mult mai general. Singurele modificări sunt că diagrama de stare şi trellisul au 2^{K-1} stări sau noduri (care sunt egale cu 4 pentru K = 3 şi 8 pentru K =4), iar acum ramurile conțin 1/R simboluri (2 pentru R = ½; 3 pentru R = 1/ 3).

O generalizare mai puțin evidentă este pentru un cod de rată mai mare decât ½. Aceasta poate fi dobândită pe două căi. Cea mai directă este prin procedeul numit *puncturare*. De exemplu, având un cod de rată ½, biții alternativi ai celui de-al doilea robinet (cea de jos) pot fi şterşi (sau puncturați) –vezi Figura 6.7a. Astfel, un număr oarecare de ieşiri sunt luate de la robinetul de sus, dar numai cât jumătate din aceştia sunt luați de la sursa de jos. Rezultatul este un cod de rată R = 2/3, deoarece pentru fiecare doi biți, generăm doar trei simboluri de ieşire. Această procedură poate fi atunci generalizată la ştergerea a n –1 ieşiri din 2n simboluri de ieşire pentru un codor de rată ½ pentru a genera un cod cu rata R = n / (n+1),

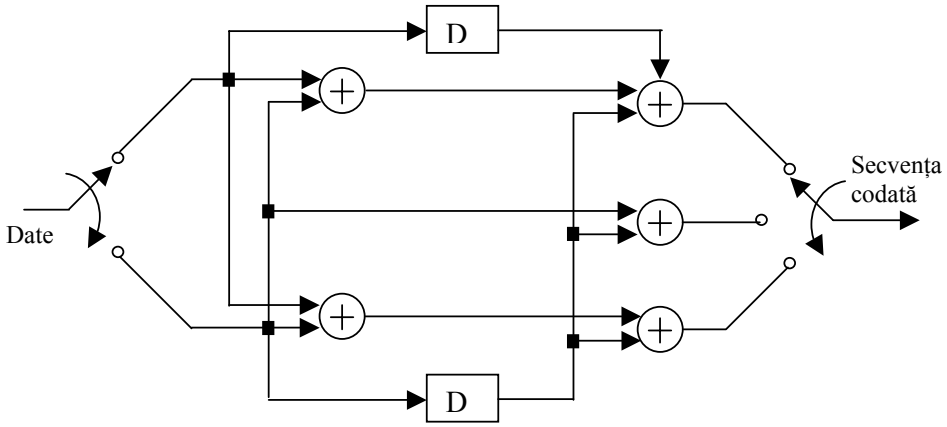
n≥2. Decodorul pentru un cod convoluțional puncturat este acelaşi cu cel original exceptând că în acele poziții unde simbolul a fost puncturat, şi astfel nu a fost transmis, nu se calculează nici o metrică μ. Aşadar, unele ramuri au metricile bazate pe două simboluri, iar altele (cu simboluri puncturate) au metricile bazate doar pe un simbol.

O mult mai generală clasă de coduri convoluționale cu o rată rațională arbitrară b/n poate fi generată utilizând b registre paralele, fiecare potențial alimentând unul dintre cele n robinete de ieşire la fiecare stare. Un exemplu cu b = 2 şi n = 3 este prezentat în Figura 6.7b. Diagrama de stare pentru o lungime de constrângere (a registrului de deplasare) K va avea $2^{b(K-1)}$ stări, şi fiecare nod are acum 2^b ramuri ce pleacă şi intră în el. Decodorul operează ca şi înainte exceptând faptul că, calea selectată, este una cu cea mai mare metrică dintre 2^b căi de intrare în nod, şi nu doar două ca şi pentru codurile de rată 1/n.

Utilizat împreună cu decodarea soft, algoritmul Viterbi este întâlnit în literatură sub denumirea de SOVA (Soft Output Viterbi Algorithm) [46], [48].



a) Puncturat, K=3;



a) General, b = 2, K= 2;

Figura 6.7 Codor convoluțional cu patru stări

Bibliografie

1. A. J. Viterbi, „CDMA –Principles of Spread Spectrum Communication”, Ed. Addison-Wesley Publishing Company, 1995

2. J. G. Proakis, „Digital Communications”, Ed. McGraw-Hill, 1989

3. G. Wade, „Coding Techniques –An Introduction to Compression and Error Control”, Creative Print and Design, 2000

4. R. Johannesson, K. Sh. Zigangirov, „Fundamentals of Convolutional Coding”, IEEE Press, 1999

5. M. E. Borda, „Teoria Transmiterii Informaţiei”, Ed. Dacia, Cluj–Napoca, 1999

6. I. Marghescu, Şt. Nicolaescu, N. Coţanis, „Comunicaţii mobile terestre”, Ed. Tehnică, Bucureşti, 1997

7. C.E. Shannon, „A Mathematical Theory of Communications”, The Bell System Technical Journal, Vol.27, pp. 379-423, 623-656. October, 1948

8. V. P. Ipatov, „Spread Spectrum Signals and Systems & CDMA”, 2001, www.physics.utu.fi

9. J. Glas, „The princilpes of Spread Spectrum communications”, teza de doctorat, 1996, cas.et.tudelft.nl

10. Jong-Seon No, „p-ary Unified Sequences: p-ary Extended d-Form Sequences With the Ideal Autocorelation Property, IEEE-Trans. on Information Theory,2002

11. ***, „Linear Feedback Shift Registers”, New Wave Instrumenrs, www.newwaveinstruments.com

12. R. Roberts, „All About Correlators”, „Spread Spectrum Scene” magazine, www.sss-mag.com

13. C. Bouder, G. Burel, „Spread Spectrum Codes Identification by Neural Networks”, „Systems and Control: Theory and Aplications”, published by World Scientific Press, 2000,pp.257-262

14. O Berder, C. Bouder, G. Burel, „Identification of Frequency Hopping Communications", published by World Scientific Press, 2000, pp. 259-264

15. Sang W. Kim, Wayne Stark, „Optimum Rate Reed-Solomon Codes for Frequency-Hopped Spred-Spectrum Multiple-Access Communication Systems", IEEE –Transaction on Communications, 1989

16. A.Bolstad, „Reed-Solomon Codes in Slow Frequency Hop Spread Spectrum Systems", SURE Program, 2002

17. M.L.Schiff, "Detecting Pseudo-Noise (PN) Spread Spectrum Signals", www.sss-mag.com

18. R.Roberts, "The ABCs of Spread Spectrum – A Tutorial", www.sss-mag.com

19. C.R.Netherton, "Data Randomizing with Pseudo-Noise Coding Techniques", www.sss-mag.com

20. Li Ping, S.Chan,"Concatenated Hadamard codes for spread spectrum systems", Electronics Letters 20th November 1997 vol 33 No 24 pp2032-2033 www.ee.cityu.edu.hk

21. R.Chauhan, "Principles of Spread Spectrum Communication", www.geocites.com

22. J.Fakatselis, "Processing Gain for Direct Sequence pread Spectrum Communication Systems and PRISM", www.intersil.com

23. Chi-Chung Chen, Kung Yao, E.Biglieri, "Optimal Spread Spectrum Sequences – Construnted From Gold Codes", 2000, www.ee.ucla.edu

24. J.P.Diogo Pinto, A. Rodrigues, F. Cercas, "Multi-*h* CPM Spread Spectrum in Satellite Mobile Communications", www.estec.esa.nl

25. L.B.Michael, M. Nakagawa, "Spread Spectrum Inter-Vehicle Communication Using Sector Antennas", www.hamradio-online.com

26. R.Kohno, R.Meidan, L.B.Milstein, "Spread Spectrum Access Methods for Wireless Communications", swig.Stanford.edu

27. J.Hagenauer, E. Offer, L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes", IEEE Transactions on Information Theory, Vol 42 No 2, March 1996 pp 429-445

28. G.Burel, "Detection of Spread Spectrum Transmissions Using Fluctutions of Correlation Estimators", www.intel-research.net

29. J.R. Seberry, B.J. Wysocki, T.A. Wysocki, "On a use of Golay sequences for asynchronous DS CDMA Applications", www.elec.uow.edu.au

30. Sang Jae Bae, Ho Soon Lee, Dong Won Lee, Eon Kyeong Joo, "Performance Comparison and Analysis of Serial Concatenated Convolutional Codes", IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

31. H.J. Zepernick, M. Caldera, "Retransmission Termination in Soft-combining Algorithms for Product Codes", IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

32. V. Bota, A. Vlaicu, „Some Aspects Regarding the Efficiencies of Hybrid ARQ Schemes Employing Concatenated Codes", IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

33. Wael Adi, Faisal Kriaa, "High Speed Decoding for Variable Redundancy Burst Error-correcting Code", IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

34. Byung Gil Lee, Sang Jae Bae, Chang Ki Jeong, Eon Kyeong Joo, "Performance Analysis of Swap Interleaver for Turbo Codes", IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

35. Dong-Feng Yuan, Peng Zhang, Qian Wang, "A Novel Concatenation Scheme (MLC-STBC) Combining MLC and STBC over Rayleigh Fading Channels," IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

36. G.A. da Silva, F.M. de Assis, „Transmission Schemes Combining Coding, Space Diversity and Fading-resistant Multidimensional Constellations", IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

37. Kun-Wah Yip, "Reed-Muller-Coded Multicode Spread-Spectrum Communications", IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

38. V. Greu, Al. Şerbănescu, M. Luca, „Selected Pseudo-noise Sequences for Multiple Access in Spread Spectrum Communications Systems", IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

39. Al. Şerbănescu, V. Greu, B. Cristea, „Chaotical Sequences for Multiple Access in Spread Spectrum Communications", IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

40. G.N. Karystinos, D.A. Pados, "New Bounds on the Total-Squared-Correlation and Optimum Design of DS-CDMA Binary Signature Sets", IEEE International Conference on Telecommunications Romania, Bucureşti, 2001

41. G. Burel, A. Quinquis, S. Azou, „Interception and Furtivity of Digital Transmissions", IEEE International Conference on Telecommunications Romania, Bucureşti, 2002

42. S.V. Halunga, O. Fratu, D.N. Vizireanu, "Error Probability Performances of Mixed Variable Length RLL –Reed Solomon Codes", IEEE International Conference on Telecommunications Romania, Bucureşti, 2002

43. D. Andrei, B. Luca, „Chaotic CDMA System: Simulation and Results", International Conference on Telecommunications Romania, Bucureşti, 2002

44. V. Greu, B. Vasilescu, „Analysis of Structural Performances for a Hybrid Spread-Spectrum System Confrunted with Interference", IEEE International Conference on Telecommunications Romania, Bucureşti, 2002

45. G.D. Forney Jr, "Convolutional Codes I: Algebraic Structure", IEEE Transaction on Information Theory", nov.1970, pp.720-738

46. A. Svensson, C.E. Sundberg, T. Aulin, „A Class of Reduced-Complexity Viterbi Detectors for Partial Response Continuous Phase Modulation", IEEE Transaction on Communications, oct.1984, pp.1079-1087

47. C. Berrou, A. Glavieux, P. Thitimajshima, „Near Shannon Limit Error – Correcting Coding and Decoding: Turbo –Codes", Proc. of ICC, Geneve, may 1993, pp. 1064-1070

48. M. Benchrifa, M. Belkasmi, A. Benouna, „Amélioration des Performances des Codes Convolutifs Concaténés en Parallèle avec un Turbo Décodage Utilisant APRI-SOVA",ICSIP'2001, Agadir, Morocco, 2001

49. Li Ping, Kwan L. Yeung, „Symbol-by-Symbol APP Decoding of Golay Code and Iterative Decoding of Concatenated Golay Codes", IEEE Transactions on Information Theory", Vol 45 No 7 November 1999

50. Esmael H.Dinan, B. Jabbari, „Spreading Codes for Direct Sequence CDMA and Wideband CDMA Cellular Networks", IEEE Communications Magazine, September 1998

Anexa A
Lista polinoamelor primitive de grad m ≤ 13

- Polinoamele sunt scrise în octal. Spre exemplu polinomul primitiv 51 indicat în lista pentru m = 5 este: g = [1 0 1 0 0 1] sau $g(x) = x^5 + x^3 + 1$.
- Programul MATLAB utilizat pentru aflarea prezentei liste este: **primitiv.m**.
- În listele următoare **m** reprezintă gradul polinomului generator, iar **N** numărul de polinoame existente pentru fiecare m.

m=1 N=1
3

m=2 N=1
7

m=3 N=2
13 15

m=4 N=2
23 31

m=5 N=6
45 51 57 67 73 75

m=6 N=6
103 133 141 147 155 163

m=7 N=18
203 211 217 221 235 247 253 271 277 301 313 323 325 345 357 361 367 375

m=8 N=16
435 453 455 515 537 543 545 551 561 607 615 651 703 717 747 765

m=9 N=48

1021	1033	1041	1055	1063	1131	1137	1151	1157	1167
1175	1207	1225	1243	1245	1257	1267	1275	1317	1321
1333	1365	1371	1423	1425	1437	1443	1461	1473	1517
1533	1541	1553	1555	1563	1577	1605	1617	1665	1671
1707	1713	1715	1725	1731	1743	1751	1773		

m= 10 N=60

2011	2033	2047	2055	2145	2157	2201	2213	2305	2327
2347	2363	2377	2415	2431	2443	2461	2475	2503	2527
2553	2605	2617	2627	2641	2707	2745	2767	2773	3023
3025	3045	3067	3103	3117	3133	3171	3177	3211	3265
3301	3323	3337	3375	3427	3435	3441	3471	3507	3515
3525	3531	3543	3575	3615	3623	3661	3733	3763	3771

m=11 N=176

4005	4027	4053	4055	4107	4143	4145	4161	4173	4215
4225	4237	4251	4261	4317	4321	4341	4347	4353	4365
4415	4423	4445	4451	4473	4475	4505	4511	4521	4533
4563	4565	4577	4603	4617	4653	4655	4671	4707	4731
4745	4767	5001	5007	5023	5025	5051	5111	5141	5155
5171	5177	5205	5221	5235	5247	5253	5263	5265	5325
5337	5351	5357	5361	5373	5403	5411	5421	5463	5477
5501	5513	5531	5537	5545	5557	5575	5607	5613	5623
5625	5657	5667	5675	5711	5733	5735	5747	5755	6013
6015	6031	6037	6127	6141	6153	6163	6205	6211	6227

6233	6235	6263	6277	6307	6315	6323	6325	6343	6351
6367	6403	6417	6435	6447	6455	6501	6507	6525	6531
6543	6557	6561	6623	6637	6651	6673	6675	6711	6727
6733	6741	6747	6765	7005	7035	7041	7047	7053	7063
7071	7107	7113	7125	7137	7161	7173	7175	7201	7223
7237	7243	7273	7317	7335	7363	7371	7413	7431	7461
7467	7535	7553	7555	7565	7603	7621	7627	7633	7647
7655	7665	7715	7723	7745	7751				

m=12 N=144

10123	10151	10173	10175	10231	10321	10353	10407	10437	10443
10473	10517	10527	10541	10553	10605	10663	10731	10737	11015
11067	11075	11147	11163	11177	11271	11301	11313	11417	11435
11441	11471	11477	11515	11561	11631	11643	11651	12007	12061
12067	12117	12135	12147	12165	12247	12255	12323	12417	12435
12515	12623	12705	12727	12735	12753	13011	13107	13125	13131
13245	13275	13425	13431	13503	13505	13565	13611	13655	13663
13677	13701	14127	14135	14221	14227	14271	14357	14433	14465
14501	14545	14573	14613	14661	14675	14711	14717	14747	15033
15053	15063	15151	15213	15321	15341	15365	15413	15423	15437
15527	15621	15647	15677	15701	15723	16005	16021	16027	16047
16115	16207	16237	16245	16273	16305	16311	16317	16363	16407
16443	16503	16521	16533	16565	16605	16611	17025	17031	17057
17105	17121	17147	17163	17217	17343	17421	17433	17447	17561
17631	17673	17675	17711						

m=13 N=630

20033	20047	20065	20123	20145	20157	20213	20215	20237	20245
20257	20273	20275	20303	20311	20341	20363	20415	20425	20451
20457	20473	20503	20547	20553	20571	20611	20627	20635	20677
20701	20707	20715	20737	20743	20761	20773	21031	21045	21067
21075	21103	21133	21135	21171	21177	21211	21227	21233	21263
21277	21315	21357	21367	21373	21405	21447	21453	21507	21525
21531	21557	21561	21575	21607	21615	21625	21643	21651	21661
21667	21673	21741	21755	21771	22013	22023	22037	22045	22051
22075	22121	22127	22141	22155	22177	22203	22233	22235	22265
22277	22301	22307	22313	22343	22411	22427	22435	22441	22455
22471	22523	22525	22543	22561	22567	22607	22613	22625	22631
22637	22657	22675	22705	22717	22727	22753	23003	23005	23021
23055	23077	23113	23123	23131	23151	23167	23173	23207	23223
23231	23261	23267	23275	23303	23353	23365	23423	23451	23473
23517	23527	23535	23553	23563	23571	23603	23621	23641	23671
23707	23713	23737	23757	23761	24007	24031	24037	24043	24061
24073	24075	24105	24147	24165	24205	24253	24255	24277	24315
24325	24337	24343	24351	24373	24411	24417	24421	24433	24453
24465	24477	24501	24513	24525	24567	24575	24601	24623	24637
24657	24667	24675	24703	24727	24763	24765	25003	25017	25035

25041	25063	25065	25115	25151	25157	25161	25173	25175	25245
25251	25261	25305	25327	25333	25353	25363	25401	25425	25443
25445	25457	25467	25503	25511	25555	25577	25605	25627	25633
25655	25663	25731	25745	25775	26017	26041	26053	26055	26077
26101	26115	26161	26213	26215	26225	26243	26257	26275	26305
26321	26327	26341	26347	26353	26415	26431	26451	26457	26467
26473	26505	26533	26547	26565	26611	26617	26647	26653	26665
26743	26761	26775	27007	27023	27025	27051	27111	27117	27133
27135	27141	27153	27217	27221	27227	27235	27253	27263	27271
27337	27373	27375	27405	27411	27421	27427	27477	27501	27513
27515	27531	27537	27545	27551	27625	27645	27657	27661	27717
27735	27747	27755	27765	27777	30007	30025	30031	30057	30111
30117	30147	30171	30177	30221	30241	30265	30277	30301	30323
30331	30345	30357	30405	30417	30465	30507	30515	30537	30543
30561	30573	30643	30651	30667	30705	30711	30733	30741	30753
30755	30763	30777	31011	31017	31035	31047	31071	31113	31123
31131	31145	31201	31223	31231	31237	31251	31267	31273	31303
31327	31333	31347	31407	31425	31457	31521	31535	31565	31627
31633	31653	31671	31701	31707	31725	31743	31745	31767	31773
32011	32033	32047	32101	32115	32137	32151	32167	32173	32207
32223	32231	32245	32275	32311	32333	32347	32371	32415	32437
32445	32461	32467	32505	32517	32535	32555	32563	32577	32635
32641	32671	32715	32725	32731	32743	32751	32757	33001	33013
33037	33045	33057	33073	33111	33121	33133	33163	33165	33221
33233	33235	33255	33313	33323	33325	33343	33357	33405	33417
33433	33441	33455	33471	33501	33507	33523	33561	33567	33613
33625	33631	33643	33705	33717	33721	33727	33735	33741	33763
34003	34005	34027	34035	34047	34063	34113	34131	34151	34161
34243	34261	34273	34311	34317	34341	34363	34371	34401	34407
34413	34423	34461	34517	34547	34555	34603	34605	34627	34641
34647	34655	34713	34715	34723	34757	34767	34775	35007	35051
35057	35075	35121	35135	35141	35147	35163	35165	35211	35271
35277	35315	35323	35325	35337	35345	35351	35373	35421	35453
35455	35465	35477	35523	35531	35543	35545	35557	35561	35567
35613	35631	35645	35651	35667	35673	35721	35747	35763	35777
36015	36023	36025	36037	36043	36045	36073	36117	36135	36155
36203	36217	36235	36247	36253	36271	36307	36351	36373	36375
36403	36427	36433	36441	36455	36463	36465	36501	36515	36545
36551	36575	36601	36625	36661	36667	36703	36721	36733	36747
36753	36771	37005	37011	37017	37033	37053	37077	37101	37123
37145	37151	37213	37243	37275	37305	37327	37335	37341	37371
37415	37431	37437	37445	37467	37475	37503	37505	37511	37521
37527	37541	37603	37611	37621	37653	37665	37743	37767	37775

Anexa B
Perechi de polinoame ce generează secvențe preferate

- Polinoamele sunt indicate prin numerele lor de ordine din listele corespunzătoare din Anexa A.
- Programul MATLAB cu care s-au aflat aceste perechi de polinoame este **scvpref.m**
- În listele următoare **m** reprezintă gradul polinomului generator, iar **ns** numărul de perechi de secvențe (respectiv polinoame ce le generează) existente pentru fiecare m.

m=3 ns=1

1—2

m=4 ns = 0

m=5 ns=9

1-4 1-5 1-6 2-3 2-4 2-5 3-5 4-6 5-6

m=6 ns = 6

1-2 1-4 2-6 3-5 3-6 4-5

m=7 ns = 47

1- 3	1- 7	1- 8	1- 9	1-14	2- 3	2- 5	2- 9	2-15	3- 6
3- 7	3- 9	3-12	3-15	4- 8	4-10	4-16	4-17	4-18	5- 6
5-10	5-11	5-15	6-12	6-13	7- 8	7- 9	7-12	7-17	8-12
8-14	8-16	8-17	9-12	10-13	10-16	10-18	11-13	11-15	11-16
12-17	13-15	13-18	14-16	16-17	16-18	17-18			

m=8 ns = 0

Anexa C

Corespondența resturi-cuvinte eroare corectabile

$g(x) = x^8 + x^7 + x^6 + x^4 + 1$ n = 15

	0	1	2	3	4	5	6	7
00	0	1	2	201	3	301	302	1410
01	4	401	402	0	403	1205	1511	0
02	5	501	502	0	503	1204	0	0
03	504	1203	1306	0	1201	12	0	1202
04	6	601	602	0	603	0	0	1312
05	604	0	1305	0	0	0	0	0
06	605	0	1304	1007	1407	0	0	1109
07	1302	1509	13	1301	0	1206	1303	0
10	7	701	702	0	703	0	0	0
11	704	1310	0	0	0	0	1413	0
12	705	908	0	1006	1406	0	0	0
13	0	0	0	0	0	1207	0	0
14	706	0	0	1005	1405	0	1108	0
15	1508	1412	0	0	0	0	1210	0
16	1403	1002	1001	10	14	1401	1402	1003
17	0	0	1307	1004	1404	0	0	0
20	8	801	802	0	803	0	0	0
21	804	0	0	0	0	0	0	0
22	805	907	1411	0	0	1110	0	0
23	0	0	0	1510	1514	1208	0	0
24	806	0	1009	0	0	1409	1107	0
25	1507	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0
27	0	0	1308	0	0	0	0	0
30	807	905	0	0	0	0	1106	0
31	1506	0	0	0	1209	0	0	0
32	901	9	1513	902	0	903	0	0
33	0	904	0	0	1311	0	0	0
34	1504	0	1103	0	1102	0	11	1101
35	15	1501	1502	1309	1503	0	1104	0
36	0	906	0	1008	1408	1512	1105	0
37	1505	0	0	1211	0	0	0	0

Notă : –restul se citeşte: primele două cifre octale pe linie iar a treia pe coloană.

–cuvântul eroare se găseşte citind pozițiile erorilor la intersecția linie-coloană. Dacă nu este nici o eroare (doar primul 0), sau sunt mai mult de două atunci poziția este 0. Dacă este una singură atunci poziția ei este indicată printr-un număr mai mic decât 15. Dacă sunt două erori atunci pozițiile lor sunt indicate printr-un număr cu 3 sau 4 cifre. Prima cifră (sau primele două) indică poziția primei erori iar ultimele două cifre indică poziția celei de-a doua erori. În cuvântul eroare poziția 1 este LSB.

$g(x) = x^8 + x^4 + x^2 + x + 1$ n = 15

	0	1	2	3	4	5	6	7
00	0	1	2	201	3	301	302	905
01	4	401	402	1508	403	0	1006	0
02	5	501	502	903	503	902	901	9
03	504	0	0	0	1107	0	0	904
04	6	601	602	0	603	0	1004	1307
05	604	0	1003	0	1002	0	10	1001
06	605	0	0	1512	0	0	0	906
07	1208	1009	0	1311	0	0	1005	0
10	7	701	702	0	703	1514	0	1306
11	704	1310	0	1109	1105	0	1408	0
12	705	0	0	0	1104	0	0	907
13	1103	0	0	0	11	1101	1102	0
14	706	0	0	1303	0	1302	1301	13
15	0	0	0	0	0	0	1007	1304
16	1309	0	1110	0	0	0	1412	1305
17	0	0	0	0	1106	0	0	0
20	8	801	802	1504	803	0	0	0
21	804	1502	1501	15	0	0	1407	1503
22	805	0	1411	0	0	0	1210	908
23	1206	0	0	1505	1509	0	0	0
24	806	0	0	0	0	1510	0	0
25	1205	1413	0	1506	0	0	1008	1209
26	1204	0	0	0	0	0	0	0
27	12	1201	1202	0	1203	0	0	0
30	807	0	0	0	0	0	1404	0
31	0	0	1403	1507	1402	0	14	1401
32	0	0	0	0	0	0	0	1511
33	0	1409	0	0	1108	0	1405	1312
34	1410	0	0	0	1211	0	0	1308
35	0	0	0	0	1513	0	1406	0
36	0	0	0	0	0	0	0	0
37	1207	0	0	0	0	0	0	0

	0	1	2	3	4	5	6	7
000	0	1	2	201	3	301	302	30201
001	4	401	402	40201	403	40301	40302	181506
002	5	501	502	50201	503	50301	50302	231714
003	504	50401	50402	212012	50403	221009	191607	131108
004	6	601	602	60201	603	60301	60302	181504
005	604	60401	60402	181503	60403	181502	181501	1815
006	605	60501	60502	160908	60503	121107	222113	201910
007	60504	191413	231110	221707	201708	232116	141209	181505
010	7	701	702	70201	703	70301	70302	222008
011	704	70401	70402	141109	70403	211713	191605	231210
012	705	70501	70502	181310	70503	121106	191604	211509
013	70504	231508	191603	221706	191602	201814	1916	191601
014	706	70601	70602	232119	70603	121105	171009	161413
015	70604	201610	131208	221705	232214	190908	212011	181507
016	70605	121103	201514	221704	121101	1211	231808	121102
017	211809	221702	221701	2217	151310	121104	191606	221703
020	8	801	802	80201	803	80301	80302	222007
021	804	80401	80402	191710	80403	161412	232109	131105
022	805	80501	80502	160906	80503	211918	151210	131104
023	80504	231507	221814	131103	201706	131102	131101	1311
024	806	80601	80602	160905	80603	231310	191411	211712
025	80604	222111	131207	232014	201705	190907	221610	181508
026	80605	160902	160901	1609	201704	221514	231807	160903
027	201703	181210	211915	160904	2017	201701	201702	131106
030	807	80701	80702	222003	80703	222002	222001	2220
031	80704	231505	131206	211816	181110	190906	171514	222004
032	80705	231504	211711	191412	141309	171610	231806	222005
033	231501	2315	201009	231502	222112	231503	191608	131107
034	80706	181714	131204	151110	211615	190904	231805	222006
035	131202	190903	1312	131201	190901	1909	131203	190902
036	221910	212013	231803	160907	231802	121108	2318	231801
037	161411	231506	131205	221708	201707	190905	231804	211410
040	9	901	902	90201	903	90301	90302	191312
041	904	90401	90402	141107	90403	221005	232108	201716
042	905	90501	90502	160806	90503	221004	201811	211507
043	90504	221003	171513	231918	221001	2210	141206	221002
044	906	90601	90602	160805	90603	212014	171007	232211
045	90604	231712	222019	211310	161311	190807	141205	181509
046	90605	160802	160801	1608	231915	181713	141204	160803
047	211807	201511	141203	160804	141202	221006	1412	141201
050	907	90701	90702	141104	90703	231816	171006	211505
051	90704	141102	141101	1411	201512	190806	221813	141103
052	90705	201917	232212	211503	141308	211502	211501	2115
053	211806	161312	201008	141105	231711	221007	191609	211504
054	90706	221513	171003	201812	171002	190804	1710	171001
055	211805	190803	231615	141106	190801	1908	171004	190802
056	211804	231410	191311	160807	222016	121109	171005	211506
057	2118	211801	211802	221709	211803	190805	141207	232013
060	908	90801	90802	160605	90803	171511	232104	181410
061	90804	201813	232103	221512	232102	190706	2321	232101
062	90805	160602	160601	1606	141307	232012	221917	160603
063	191211	211714	201007	160604	181615	221008	232105	131109
064	90806	160502	160501	1605	221812	190704	201513	160503
065	151410	190703	181711	160504	190701	1907	232106	190702
066	160201	1602	1601	16	211110	160302	160301	1603
067	232213	160402	160401	1604	201709	190705	141208	160403
070	90807	211210	191815	231713	141305	190604	161211	222009
071	221716	190603	201005	141108	190601	1906	232107	190602
072	141303	221811	201004	160706	1413	141301	141302	211508
073	201002	231509	2010	201001	141304	190605	201003	181712
074	232011	190403	222114	160705	190401	1904	171008	190402
075	190301	1903	131209	190302	1901	19	190201	1902
076	171512	160702	160701	1607	141306	190504	231809	160703
077	211808	190503	201006	160704	190501	1905	221511	190502
	0	1	2	3	4	5	6	7

Codul Golay

$$g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

	0	1	2	3	4	5	6	7
100	10	1001	1002	100201	1003	100301	100302	211611
101	1004	100401	100402	191708	100403	220905	201413	231207
102	1005	100501	100502	181307	100503	220904	151208	201906
103	100504	220903	231106	161514	220901	2209	211817	220902
104	1006	100601	100602	221412	100603	231308	170907	201905
105	100604	201607	231105	211309	211912	171411	221608	181510
106	100605	211715	231104	201903	181614	201902	201901	2019
107	231102	181208	2311	231101	151307	220906	231103	201904
110	1007	100701	100702	181305	100703	191514	170906	231204
111	100704	201606	222115	231203	181108	231202	231201	2312
112	100705	181302	181301	1813	232120	171608	221411	181303
113	171412	211911	200908	181304	151306	220907	191610	231205
114	100706	201604	170903	151108	170902	222118	1709	170901
115	201601	2016	191814	201602	151305	201603	170904	231206
116	221908	231409	211612	181306	151304	121110	170905	201907
117	151303	201605	231107	221710	1513	151301	151302	211408
120	1008	100801	100802	191704	100803	231306	151205	181409
121	100804	191702	191701	1917	181107	212015	221606	191703
122	100805	201411	151203	232221	151202	171607	1512	151201
123	211613	181206	200907	191705	231914	220908	151204	131110
124	100806	231303	212018	151107	231301	2313	221604	231302
125	151409	181205	221603	191706	221602	231304	2216	221601
126	221907	181204	171413	161009	211109	231305	151206	201908
127	181201	1812	231108	181202	201710	181203	221605	211407
130	100807	211209	231614	151106	181104	171605	211913	222010
131	181103	221413	200905	191707	1811	181101	181102	231208
132	221906	171603	200904	181308	171601	1716	151207	171602
133	200902	231510	2009	200901	181105	171604	200903	211406
134	221905	151102	151101	1511	201412	231307	170908	151103
135	232117	201608	131210	151104	181106	191009	221607	211405
136	2219	221901	221902	151105	221903	171606	231810	211404
137	221904	181207	200906	211403	151308	211402	211401	2114
140	1009	100901	100902	232015	100903	220504	170706	181408
141	100904	220503	181612	211306	220501	2205	191511	220502
142	100905	220403	211914	171211	220401	2204	231613	220402
143	220301	2203	200807	220302	2201	22	220201	2202
144	100906	191811	170703	211304	170702	161512	1707	170701
145	151408	211302	211301	2113	232018	220605	170704	211303
146	201312	231407	221815	161008	211108	220604	170705	201909
147	191716	220603	231109	211305	220601	2206	141210	220602
150	100907	211208	170603	221916	170602	201311	1706	170601
151	231913	181715	200805	141110	211614	220705	170604	231209
152	161511	231406	200804	181309	191812	220704	170605	211510
153	200802	220703	2008	200801	220701	2207	200803	220702
154	170302	231405	1703	170301	1702	170201	17	1701
155	221211	201609	170403	211307	170402	191008	1704	170401
156	231401	2314	170503	231402	170502	231403	1705	170501
157	211810	231404	200806	191512	151309	220706	170504	181611
160	100908	211207	221311	181403	201916	181402	181401	1814
161	151406	231611	200705	191709	171312	220805	232110	181404
162	231817	191513	200704	161006	211106	220804	151209	181405
163	200702	220803	2007	200701	220801	2208	200703	220802
164	151404	222017	231912	161005	211105	231309	170807	181406
165	1514	151401	151402	211308	151403	191007	221609	201211
166	211103	161002	161001	1610	2111	211101	211102	161003
167	151405	181209	200706	161004	211104	220806	191813	231715
170	211201	2112	200504	211202	232215	211203	170806	181407
171	200502	211204	2005	200501	181109	191006	200503	161513
172	200402	211205	2004	200401	141310	171609	200403	231911
173	2002	200201	20	2001	200302	220807	2003	200301
174	181613	211206	170803	151109	170802	191004	1708	170801
175	151407	191003	200605	232218	191001	1910	170804	191002
176	221909	231408	200604	161007	211107	201815	170805	221312
177	200602	171311	2006	200601	231612	191005	200603	211409
	0	1	2	3	4	5	6	7

Codul Golay

$$g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

	0	1	2	3	4	5	6	7
200	11	1101	1102	110201	1103	110301	110302	211610
201	1104	110401	110402	140907	110403	232019	221712	130805
202	1105	110501	110502	221915	110503	120706	201809	130804
203	110504	181716	231006	130803	211514	130802	130801	1308
204	1106	110601	110602	201713	110603	120705	191408	232209
205	110604	222108	231005	191612	161309	171410	212007	181511
206	110605	120703	231004	211814	120701	1207	171615	120702
207	231002	201509	2310	231001	221918	120704	231003	130806
210	1107	110701	110702	140904	110703	120605	231513	191817
211	110704	140902	140901	1409	181008	221615	212006	140903
212	110705	120603	211708	232016	120601	1206	221410	120602
213	222013	211910	181512	140905	231709	120604	191611	130807
214	110706	120503	221816	151008	120501	1205	212004	120502
215	191715	231813	212003	140906	212002	120504	2120	212001
216	120301	1203	191309	120302	1201	12	120201	1202
217	161408	120403	231007	221711	120401	1204	212005	120402
220	1108	110801	110802	231812	110803	171509	191406	130504
221	110804	222106	201615	130503	181007	130502	130501	1305
222	110805	201410	211707	130403	232216	130402	130401	1304
223	191209	130302	130301	1303	130201	1302	1301	13
224	110806	222104	191403	151007	191402	201816	1914	191401
225	222101	2221	181709	222102	231512	222103	191404	130605
226	181513	231917	222012	161109	211009	120807	191405	130604
227	161407	222105	231008	130603	201711	130602	130601	1306
230	110807	191613	211705	151006	181004	232114	161209	222011
231	181003	201712	232219	140908	1810	181001	181002	130705
232	211702	221809	2117	211701	201915	120806	211703	130704
233	161406	231511	211704	130703	181005	130702	130701	1307
234	232009	151002	151001	1510	221713	120805	191407	151003
235	161405	222107	131211	151004	181006	191109	212008	231716
236	161404	120803	211706	151005	120801	1208	231811	120802
237	1614	161401	161402	201918	161403	120804	221509	130706
240	1109	110901	110902	140704	110903	171508	201805	232206
241	110904	140702	140701	1407	161306	211812	191510	140703
242	110905	232113	201803	171210	201802	191614	2018	201801
243	191208	201506	222116	140705	231707	221110	201804	130908
244	110906	191810	211512	232203	161304	232202	232201	2322
245	161303	201505	181708	140706	1613	161301	161302	232204
246	221714	201504	191307	161108	211008	120907	201806	232205
247	201501	2015	231009	201502	161305	201503	141211	211917
250	110907	140402	140401	1404	222119	201310	161208	140403
251	140201	1402	1401	14	231705	140302	140301	1403
252	161510	221808	191306	140504	231704	120906	201807	211511
253	231703	140502	140501	1405	2317	231701	231702	140503
254	232008	211716	191305	140604	181514	120905	171110	232207
255	221210	140602	140601	1406	161307	191108	212009	140603
256	191302	120903	1913	191301	120901	1209	191303	120902
257	211811	201507	191304	140605	231706	120904	221508	181610
260	110908	171503	221310	212019	171501	1715	161207	171502
261	191205	231610	181706	140807	222014	171504	232111	130905
262	191204	221807	231514	161106	211006	171505	201808	130904
263	1912	191201	191202	130903	191203	130902	130901	1309
264	232007	141312	181704	161105	211005	171506	191409	232208
265	181702	222109	1817	181701	161308	191107	181703	201210
266	211003	161102	161101	1611	2110	211001	211002	161103
267	191206	201508	181705	161104	211004	231814	221507	130906
270	232006	221805	161203	140804	161202	171507	1612	161201
271	211513	140802	140801	1408	181009	191106	161204	140803
272	221801	2218	211709	221802	141311	221803	161205	231910
273	191207	221804	201110	140805	231708	212016	221506	130907
274	2320	232001	232002	151009	232003	191104	161206	211813
275	232004	191103	181707	140806	191101	1911	221505	191102
276	232005	221806	191308	161107	211007	120908	221504	201714
277	161409	171310	221503	232112	221502	191105	2215	221501
	0	1	2	3	4	5	6	7

Codul Golay

$$g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

	0	1	2	3	4	5	6	7
300	1110	111001	111002	211603	111003	211602	211601	2116
301	111004	151312	230605	222018	180807	171406	191509	211604
302	111005	201408	230604	171209	191713	231815	221407	211605
303	230602	211907	2306	230601	201612	221109	230603	131008
304	111006	191809	230504	150807	222015	171404	181312	211606
305	230502	171403	2305	230501	171401	1714	230503	171402
306	230402	221613	2304	230401	210908	121007	230403	201911
307	2302	230201	23	2301	230302	171405	2303	230301
310	111007	232217	201912	150806	180804	201309	221405	211607
311	180803	211905	171613	141009	1808	180801	180802	231211
312	161509	211904	221403	181311	221402	121006	2214	221401
313	211901	2119	230706	211902	180805	211903	221404	201715
314	211413	150802	150801	1508	231916	121005	171109	150803
315	221209	201611	230705	150804	180806	171407	212010	221913
316	201817	121003	230704	150805	121001	1210	221406	121002
317	230702	211906	2307	230701	151311	121004	230703	181609
320	111008	201405	221309	150706	180704	221912	232017	211608
321	180703	231609	211412	191711	1807	180701	180702	131005
322	201401	2014	191816	201402	210906	201403	151211	131004
323	221715	201404	230806	131003	180705	131002	131001	1310
324	171612	150702	150701	1507	210905	231311	191410	150703
325	201913	222110	230805	150704	180706	171408	221611	201209
326	210903	201406	230804	150705	2109	210901	210902	221817
327	230802	181211	2308	230801	210904	191615	230803	131006
330	180403	150602	150601	1506	1804	180401	180402	150603
331	1803	180301	180302	150604	18	1801	1802	180201
332	231312	201407	211710	150605	180504	171611	221408	231909
333	180503	211908	201109	221612	1805	180501	180502	131007
334	150201	1502	1501	15	180604	150302	150301	1503
335	180603	150402	150401	1504	1806	180601	180602	150403
336	221911	150502	150501	1505	210907	121008	201613	150503
337	161410	171309	230807	150504	180605	232220	191712	211411
340	111009	191806	221308	171205	231412	201307	191504	211609
341	212017	231608	191503	141007	191502	221105	1915	191501
342	161507	171202	171201	1712	210806	221104	201810	171203
343	181413	221103	230906	171204	221101	2211	191505	221102
344	191801	1918	201614	191802	210805	191803	171107	232210
345	221207	191804	230905	211311	161310	171409	191506	201208
346	210803	191805	230904	171206	2108	210801	210802	151413
347	230902	201510	2309	230901	210804	221106	230903	181607
350	161505	201303	232118	141004	201301	2013	171106	201302
351	221206	141002	141001	1410	180908	201304	191507	141003
352	1615	161501	161502	171207	161503	201305	221409	231908
353	161504	211909	201108	141005	231710	221107	211312	181606
354	221204	191807	171103	150908	171102	201306	1711	171101
355	2212	221201	221202	141006	221203	232115	171104	181605
356	161506	231411	191310	222120	210807	121009	171105	181604
357	221205	171308	230907	181603	201914	181602	181601	1816
360	221302	231604	2213	221301	210605	171510	221303	181411
361	231601	2316	221304	231602	180907	231603	191508	201206
362	210603	201409	221305	171208	2106	210601	210602	231907
363	191210	231605	201107	211815	210604	221108	171614	131009
364	210503	191808	221306	150907	2105	210501	210502	201204
365	151411	231606	181710	201203	210504	201202	201201	2012
366	2103	210301	210302	161110	21	2101	2102	210201
367	210403	171307	230908	221914	2104	210401	210402	201205
370	191714	211211	221307	150906	180904	201308	161210	231905
371	180903	231607	201105	141008	1809	180901	180902	222117
372	161508	221810	201104	231903	210706	231902	231901	2319
373	201102	171306	2011	201101	180905	151412	201103	231904
374	232010	150902	150901	1509	210705	221614	171108	150903
375	221208	171305	211916	150904	180906	191110	231413	201207
376	210703	171304	181412	150905	2107	210701	210702	231906
377	171301	1713	201106	171302	210704	171303	221510	181608
	0	1	2	3	4	5	6	7

Codul Golay

$$g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

Anexa D

Lista polinoamelor ireductibile în exprimare zecimală

m=0
1

m=1
2 3

m=2
7

m=3
11 13

m=4
19 25 31

m=5 N=6
37 41 47 55 59 61

m=6 N=9
67 73 87 91 97 103 109 115 117

m=7 N=18
131 137 143 145 157 167 171 185 191 193
203 211 213 229 239 241 247 253

m=8 N= 30
283 285 299 301 313 319 333 351 355 357
361 369 375 379 391 395 397 415 419 425
433 445 451 463 471 477 487 499 501 505

m=9 N=56
515 529 535 539 545 557 563 587 601
607
613 617 623 631 637 647 661 665 675
677
687 695 701 719 721 731 757 761 769
787
789 799 803 817 827 841 847 859 865
875
877 883 895 901 911 929 949 953 967
971
973 981 985 995 1001 1019

m=10 N=99
1033 1039 1051 1053 1063 1069 1077 1095 1107 1123
1125 1135 1153 1163 1177 1193 1199 1221 1225 1239
1255 1261 1267 1279 1291 1293 1305 1311 1315 1329
1341 1347 1367 1377 1383 1387 1413 1423 1431 1435
1441 1451 1465 1473 1479 1509 1527 1531 1555 1557
1571 1573 1585 1591 1603 1615 1617 1627 1657 1663
1669 1673 1703 1709 1717 1727 1729 1741 1747 1759
1783 1789 1807 1809 1815 1821 1825 1835 1845 1849
1863 1869 1877 1881 1891 1915 1917 1921 1927 1933
1939 1961 1969 1989 2011 2027 2035 2041 2047

m=11 N=186
2053 2071 2091 2093 2119 2147 2149 2161 2171 2189
2197 2207 2217 2225 2243 2255 2257 2273 2279 2283
2293 2317 2323 2341 2345 2359 2363 2365 2373 2377

2385 2395 2419 2421 2431 2435 2447 2475 2477 2489
2503 2521 2533 2543 2551 2561 2567 2579 2581 2601
2633 2657 2669 2681 2687 2693 2705 2717 2727 2731
2739 2741 2773 2783 2787 2793 2799 2801 2811 2819
2825 2833 2867 2879 2881 2891 2905 2911 2917 2927
2941 2951 2955 2963 2965 2991 2999 3005 3017 3035
3037 3047 3053 3083 3085 3097 3103 3121 3159 3169

3179 3187 3189 3205 3209 3223 3227 3229 3251 3263
3271 3277 3283 3285 3299 3305 3319 3331 3343 3357
3367 3373 3393 3399 3413 3417 3427 3439 3441 3475
3487 3497 3515 3517 3529 3543 3547 3553 3559 3573
3583 3589 3613 3617 3623 3627 3635 3641 3655 3659
3669 3679 3697 3707 3709 3713 3731 3743 3747 3771
3785 3791 3805 3827 3833 3851 3865 3889 3895 3933
3947 3949 3957 3961 3971 3985 3991 3995 4007 4013
4021 4045 4051 4069 4073 4091

m=12 N= 335

4105 4119 4129 4147 4149 4159 4173 4179 4201
4215
4219 4221 4225 4235 4249 4259 4261 4303 4305
4331
4333 4351 4359 4383 4387 4401 4407 4411 4431
4439
4449 4459 4461 4473 4483 4485 4497 4523 4531
4569
4575 4579 4591 4593 4609 4621 4627 4633 4645
4663
4667 4669 4675 4677 4711 4717 4723 4735 4789
4793
4801 4811 4873 4879 4891 4893 4897 4915 4921
4927
4941 4965 4977 5017 5023 5027 5033 5039 5051
5059
5073 5079 5085 5107 5109 5127 5139 5169 5175
5193
5199 5211 5213 5223 5227 5237 5247 5257 5281
5287

5293 5301 5325 5331 5337 5343 5349 5391 5405
5451
5453 5505 5523 5541 5545 5563 5573 5591 5597
5611
5625 5635 5641 5659 5695 5697 5703 5707 5717
5721
5731 5733 5743 5779 5797 5821 5827 5833 5841
5857

5863 5875 5887 5899 5909 5913 5949 5955 5957
5967
5975 5981 6003 6005 6009 6025 6031 6039 6045
6061
6067 6079 6081 6111 6139 6151 6157 6175 6179
6193
6199 6217 6231 6237 6253 6265 6271 6275 6289
6295
6305 6329 6347 6349 6383 6385 6395 6405 6409
6427
6445 6453 6465 6475 6495 6501 6511 6523 6529
6539

6553 6577 6583 6589 6601 6607 6621 6631 6637
6649
6683 6685 6689 6699 6707 6733 6739 6741 6751
6755
6761 6779 6795 6833 6853 6865 6881 6887 6891
6901
6923 6925 6931 6937 6943 6959 6981 6999 7049
7055
7057 7079 7093 7097 7103 7105 7115 7123 7139
7165
7171 7173 7183 7185 7191 7207 7245 7263 7303
7327
7333 7351 7355 7365 7369 7375 7383 7403 7405
7411
7425 7431 7459 7471 7485 7491 7505 7515 7517
7527
7541 7545 7555 7557 7561 7569 7591 7603 7617
7663
7687 7701 7705 7727 7739 7741 7749 7761 7773
7777

7783 7795 7823 7831 7835 7865 7871 7885 7891
7907
7921 7927 7939 7953 7963 7975 7993 8007 8011
8019
8037 8049 8061 8065 8077 8089 8111 8123 8125
8131
8133 8137 8161 8173 8191

m=13 N=630

8219 8231 8245 8275 8293 8303 8331 8333 8351 8357
8367 8379 8381 8387 8393 8417 8435 8461 8469 8489
8495 8507 8515 8551 8555 8569 8585 8599 8605 8639
8641 8647 8653 8671 8675 8689 8699 8729 8741 8759
8765 8771 8795 8797 8825 8831 8841 8855 8859 8883
8895 8909 8943 8951 8955 8965 8999 9003 9031 9045
9049 9071 9073 9085 9095 9101 9109 9123 9129 9137
9143 9147 9185 9197 9209 9227 9235 9247 9253 9257
9277 9297 9303 9313 9325 9343 9347 9371 9373 9397
9407 9409 9415 9419 9443 9481 9495 9501 9505 9517

9529 9555 9557 9571 9585 9591 9607 9611 9621 9625
9631 9647 9661 9669 9679 9687 9707 9731 9733 9745
9773 9791 9803 9811 9817 9833 9847 9851 9863 9875
9881 9905 9911 9917 9923 9963 9973 10003 10025 10043
10063 10071 10077 10091 10099 10105 10115 10129 10145 10169
10183 10187 10207 10223 10225 10247 10265 10271 10275 10289
10299 10301 10309 10343 10357 10373 10411 10413 10431 10445
10453 10463 10467 10473 10491 10505 10511 10513 10523 10539
10549 10559 10561 10571 10581 10615 10621 10625 10643 10655
10671 10679 10685 10691 10711 10739 10741 10755 10767 10781

10785 10803 10805 10829 10857 10863 10865 10875 10877 10917
10921 10929 10949 10967 10971 10987 10995 11009 11029 11043
11045 11055 11063 11075 11081 11117 11135 11141 11159 11163
11181 11187 11225 11237 11261 11279 11297 11307 11309 11327
11329 11341 11377 11403 11405 11413 11427 11439 11453 11461
11473 11479 11489 11495 11499 11533 11545 11561 11567 11575
11579 11589 11611 11623 11637 11657 11663 11687 11691 11701
11747 11761 11773 11783 11795 11797 11817 11849 11855 11867
11869 11873 11883 11919 11921 11927 11933 11947 11955 11961
11999 12027 12029 12037 12041 12049 12055 12095 12097 12107

12109 12121 12127 12133 12137 12181 12197 12207 12209 12239
12253 12263 12269 12277 12287 12295 12309 12313 12335 12361
12367 12391 12409 12415 12433 12449 12469 12479 12481 12499
12505 12517 12527 12549 12559 12597 12615 12621 12639 12643
12657 12667 12707 12713 12727 12741 12745 12763 12769 12779
12781 12787 12799 12809 12815 12829 12839 12857 12875 12883
12889 12901 12929 12947 12953 12959 12969 12983 12987 12995
13015 13019 13031 13063 13077 13103 13137 13149 13173 13207
13211 13227 13241 13249 13255 13269 13283 13285 13303 13307
13321 13339 13351 13377 13389 13407 13417 13431 13435 13447

13459 13465 13477 13501 13513 13531 13543 13561 13581 13599
13605 13617 13623 13637 13647 13661 13677 13683 13695 13725
13729 13753 13773 13781 13785 13795 13801 13807 13825 13835
13855 13861 13871 13883 13897 13905 13915 13939 13941 13969
13979 13981 13997 14027 14035 14037 14051 14063 14085 14095
14107 14113 14125 14137 14145 14151 14163 14193 14199 14219
14229 14233 14243 14277 14287 14289 14295 14301 14305 14323
14339 14341 14359 14365 14375 14387 14411 14425 14441 14449
14499 14513 14523 14537 14543 14561 14579 14585 14593 14599

14603	14611	14641	14671	14695	14701	14723	14725	14743	14753
14759	14765	14795	14797	14803	14831	14839	14845	14855	14889
14895	14909	14929	14941	14945	14951	14963	14965	14985	15033
15039	15053	15059	15061	15071	15077	15081	15099	15121	15147
15149	15157	15167	15187	15193	15203	15205	15215	15217	15223
15243	15257	15269	15273	15287	15291	15313	15335	15347	15359
15373	15379	15381	15391	15395	15397	15419	15439	15453	15469
15491	15503	15517	15527	15531	15545	15559	15593	15611	15613
15619	15639	15643	15649	15661	15667	15669	15681	15693	15717
15721	15741	15745	15765	15793	15799	15811	15825	15835	15847
15851	15865	15877	15881	15887	15899	15915	15935	15937	15955
15973	15977	16011	16035	16061	16069	16087	16093	16097	16121
16141	16153	16159	16165	16183	16189	16195	16197	16201	16209
16215	16225	16259	16265	16273	16299	16309	16355	16375	16381

m=14 N=1161

16417	16427	16435	16441	16447	16467	16479	16485	16507	16519
16553	16559	16571	16573	16591	16599	16619	16627	16633	16651
16653	16659	16699	16707	16713	16727	16743	16749	16785	16795
16797	16807	16811	16813	16821	16853	16857	16881	16897	16909
16965	16969	16983	16993	17011	17017	17023	17027	17029	17053
17057	17095	17099	17101	17123	17129	17135	17155	17161	17179
17185	17191	17215	17257	17275	17277	17287	17301	17327	17353
17373	17387	17389	17407	17419	17421	17475	17501	17523	17545
17601	17619	17621	17631	17635	17649	17659	17667	17673	17679
17707	17721	17753	17775	17783	17789	17805	17817	17823	17829
17847	17861	17865	17873	17879	17895	17907	17919	17935	17949
17959	17973	17991	18009	18019	18033	18043	18061	18067	18069
18083	18085	18117	18127	18139	18155	18175	18213	18225	18243
18255	18303	18313	18321	18331	18343	18357	18369	18387	18393
18405	18409	18415	18429	18451	18457	18463	18491	18499	18513
18523	18529	18535	18559	18563	18577	18623	18631	18659	18673
18679	18685	18717	18721	18733	18745	18753	18771	18783	18789
18793	18807	18823	18827	18857	18895	18897	18909	18913	18919
18967	18997	19033	19045	19067	19073	19079	19083	19091	19107
19119	19133	19145	19165	19181	19193	19231	19255	19273	19291
19297	19307	19309	19315	19321	19333	19343	19351	19361	19371
19379	19385	19403	19405	19413	19423	19441	19451	19465	19483
19485	19495	19499	19519	19527	19531	19539	19541	19557	19581
19597	19621	19645	19653	19665	19671	19693	19711	19733	19743
19753	19761	19781	19791	19793	19829	19845	19855	19885	19891
19905	19923	19953	19963	19969	19989	20003	20023	20035	20041
20049	20075	20077	20099	20123	20147	20179	20197	20201	20207
20253	20257	20299	20309	20319	20329	20335	20353	20365	20383
20389	20393	20407	20411	20439	20459	20461	20473	20487	20511
20517	20571	20573	20641	20683	20693	20697	20707	20713	20719
20731	20763	20769	20781	20799	20819	20825	20831	20847	20861
20875	20889	20901	20913	20919	20943	20945	20955	20971	20973
20981	20991	20997	21007	21037	21093	21105	21131	21145	21155
21169	21181	21187	21189	21199	21201	21223	21227	21241	21249

21273	21285	21289	21303	21321	21339	21351	21365	21403	21405
21415	21433	21439	21447	21459	21477	21489	21501	21507	21519
21527	21557	21561	21575	21593	21599	21627	21645	21651	21653
21663	21681	21687	21691	21725	21729	21739	21779	21785	21807
21815	21863	21867	21877	21881	21887	21891	21893	21905	21911
21933	21953	21971	21983	21993	22007	22023	22029	22037	22051
22057	22063	22065	22103	22109	22171	22187	22189	22195	22209
22215	22221	22257	22263	22267	22315	22317	22335	22347	22357
22361	22371	22373	22397	22419	22447	22461	22467	22469	22487
22503	22515	22531	22545	22561	22573	22579	22581	22591	22593
22653	22663	22667	22677	22681	22691	22703	22705	22737	22749
22759	22763	22777	22783	22803	22819	22843	22863	22911	22927
22935	22941	22945	22951	22955	22965	22987	23007	23017	23037
23053	23059	23071	23077	23099	23101	23107	23109	23113	23157
23183	23207	23221	23233	23251	23253	23257	23287	23311	23319
23325	23339	23347	23353	23361	23395	23401	23415	23449	23459
23465	23491	23493	23521	23531	23545	23559	23563	23577	23601
23607	23625	23645	23661	23673	23683	23713	23743	23745	23755
23757	23781	23813	23825	23837	23859	23861	23879	23919	23943
23949	23957	23967	23971	23977	23995	24009	24015	24027	24033
24067	24079	24091	24109	24135	24139	24163	24189	24193	24217
24229	24233	24279	24283	24295	24309	24327	24333	24345	24351
24355	24381	24387	24389	24401	24417	24427	24437	24457	24471
24491	24525	24543	24547	24549	24561	24587	24589	24597	24623
24637	24655	24657	24673	24679	24683	24713	24727	24733	24737
24747	24755	24761	24787	24789	24823	24841	24849	24877	24889
24897	24915	24945	24957	24991	24997	25007	25019	25051	25069
25077	25087	25131	25139	25141	25145	25159	25165	25187	25199
25213	25229	25247	25253	25257	25265	25271	25303	25307	25309
25323	25325	25331	25343	25379	25393	25399	25405	25435	25453
25461	25477	25481	25489	25505	25535	25583	25597	25609	25623
25645	25665	25671	25677	25685	25739	25749	25759	25769	25777
25831	25845	25857	25867	25881	25911	25915	25923	25925	25929
25947	25987	26001	26023	26029	26041	26047	26067	26069	26073
26085	26095	26097	26103	26113	26119	26125	26147	26171	26191
26205	26219	26221	26227	26243	26255	26263	26279	26283	26293
26297	26329	26335	26345	26385	26395	26401	26419	26443	26463
26473	26487	26497	26531	26543	26551	26577	26599	26603	26613
26627	26641	26651	26653	26667	26689	26707	26735	26743	26763
26765	26771	26783	26789	26793	26821	26825	26879	26887	26905
26927	26941	26967	26987	26995	26997	27001	27013	27023	27035
27037	27041	27051	27079	27085	27113	27137	27143	27147	27161
27171	27183	27217	27227	27239	27243	27245	27253	27267	27287
27315	27317	27327	27329	27339	27341	27369	27375	27387	27389
27395	27415	27435	27443	27449	27463	27467	27477	27497	27517
27521	27533	27541	27551	27555	27557	27569	27575	27589	27607
27617	27629	27635	27641	27659	27673	27695	27709	27717	27735
27745	27763	27829	27833	27839	27841	27847	27851	27877	27889
27909	27913	27919	27927	27947	27987	28003	28005	28009	28027
28067	28081	28091	28093	28099	28101	28125	28169	28199	28205
28211	28225	28237	28243	28271	28283	28289	28295	28309	28335

28343 28355 28379 28381 28409 28417 28437 28457 28465 28475
28495 28503 28507 28513 28549 28561 28567 28587 28597 28615
28633 28639 28649 28677 28701 28715 28723 28725 28747 28797
28801 28813 28841 28855 28859 28873 28879 28893 28897 28947
28949 28953 28963 28977 28983 28989 29021 29035 29065 29079

29083 29089 29109 29119 29131 29151 29157 29175 29179 29209
29215 29231 29233 29243 29263 29281 29287 29327 29357 29363
29377 29389 29395 29407 29413 29425 29431 29443 29449 29479
29483 29505 29525 29541 29551 29581 29587 29605 29629 29641
29649 29671 29683 29685 29695 29715 29717 29737 29775 29783
29787 29803 29805 29827 29867 29875 29895 29901 29909 29919
29929 29947 29949 29975 29979 29985 30005 30017 30027 30071
30075 30081 30105 30115 30141 30159 30161 30187 30197 30201
30207 30237 30265 30279 30291 30293 30303 30307 30309 30313
30343 30357 30367 30371 30383 30395 30405 30417 30443 30451

30457 30475 30511 30537 30545 30551 30573 30579 30595 30601
30631 30637 30645 30663 30675 30677 30703 30741 30757 30769
30781 30799 30801 30811 30829 30887 30893 30899 30911 30923
30925 30937 30943 30953 30959 30979 30991 30999 31015 31027
31053 31065 31087 31089 31099 31105 31111 31141 31153 31173
31177 31191 31197 31235 31259 31271 31275 31285 31295 31307
31317 31351 31361 31373 31401 31415 31419 31427 31457 31475
31477 31499 31523 31547 31557 31567 31569 31581 31591 31609
31621 31631 31649 31659 31673 31699 31715 31729 31735 31749
31753 31783 31789 31833 31849 31869 31883 31891 31893 31907

31927 31939 31953 31965 31979 31993 31999 32001 32021 32055
32069 32073 32115 32121 32143 32145 32151 32167 32179 32199
32205 32213 32233 32251 32253 32257 32269 32281 32303 32325
32353 32373 32383 32393 32399 32411 32413 32427 32447 32455
32467 32483 32485 32521 32545 32575 32589 32597 32625 32651
32653 32665 32671 32675 32689 32707 32709 32721 32727 32737
32743

Anexa E

Program de simulare a unei scheme de HCCC

```
clear
clc
% HCCC Simulare

%Parametrii codului convolutional
  k=3; % lungimea de constrangere
  R=1/3; % rata de codare
  % nepuncturat; G=[1, (1+D^2)/(1+D+D^2)]

%Parametrii de intrare
  N=1000; %marimea interliverului; Sa fie multiplu de 10
  DB=.6; %  Raport Semnal-Zgomot (in dB)
  M=4; % numarul de stari
  n=10;  %  numarul de blocuri transmise
  tu=1:n*N; % timpul la informatie
  tv=1:n*N/R; % timpul in canal
  iter=12; %numarul de iteratii
  ELIM=0.00001; %prag pt. prob minima

%Datele / Informatia
   for i=1:n*N
       u(i)=rand(1);
      if u(i)>0.5
         u(i)=1;
      else
         u(i)=0;
      end
   end

%Interliverul
ina=1:N;
for i=1:N
   y=(N-i+1)*rand(1);
   y=floor(y)+1;
   inb(i)=ina(y);
   if i<N
      ina=[ina(1:y-1) ina(y+1:N-i+1)];
   end
end

for i=1:N
   rnum(inb(i))=i;
end
```

```
%Codorul 1
   for j=1:n
      s0=0; s1=0; % starea initiala
      for i=1:N
         if i>N-3 & s0==s1 % readucere la zero
            u(i+j*N-N)=0;
         elseif i>N-3
            u(i+j*N-N)=1;
         end
         v(1,i+j*N-N)=u(i+j*N-N); % primul bit din v
         af1=u(i+j*N-N)+s0+s1; % reactia codorului 1
         af2=floor(af1/2);
         af1=af1-2*af2;
         if af1==s1 % iesirea codorului 1
            af2=0;
         else
            af2=1;
         end
         s1=s0;
         s0=af1;
         v(2,i+j*N-N)=af2; % bitul livrat (via I) codorului 2
      end
   end

%Codorul 2
   for j=1:n
      for i=1:N
         w(rnum(i)+j*N-N)=v(2,j*N-N+i); % interlivare
      end
   end
   for j=1:n
   s0=0; s1=0; % starea initiala
      for i=1:N
         v(2,i+j*N-N)=w(i+j*N-N); % bitul al doilea din v
         af1=w(i+j*N-N)+s0+s1; % reactia codorului 2
         af2=floor(af1/2);
         af1=af1-2*af2;
         if af1==s1 % iesirea codorului 2
            af2=0;
         else
            af2=1;
         end
         s1=s0;
         s0=af1;
         v(3,i+j*N-N)=af2; % bitul al treilea din v
```

```
      end
   end

%Zgomotul
  B=10^(0.1*DB);
  sigma=1/(sqrt(2*R*B)); % dispersia zgomotului
  sig2=2*sigma*sigma; % dublul puterii zg.

%Semnalul receptionat
   for i=1:n*N
      for j=1:3
         if v(j,i)==0
            v(j,i)=-1; % modularea lui "0" prin "-1"
         end
         y=sigma*randn(1); % esantion de zgomot cu dispersie=sig2
         r(j,i)=v(j,i)+y;
      end
   end

% Erori inainte de decodare
ercan=zeros(1,n);
for j=1:n
   for i=1:N
      if r(1,j*N-N+i)<0 & v(1,j*N-N+i)==1
         ercan(j)=ercan(j)+1;
      elseif r(1,j*N-N+i)>0 & v(1,j*N-N+i)==-1
         ercan(j)=ercan(j)+1;
      end
   end
end
ercan

%DECODAREA
for j=1:n  % contorul blocului in decodare
   % probabilitatea starilor initiale pentru dec1 si dec2
       P(1,1)=1;
       P(2,1)=0;
       P(3,1)=0;
       P(4,1)=0;
   %Valoarea initiala a informatiei extrinseci
      for i=1:N
         Pw20(i)=0.5;
      end
%j
   % Initializarea buclei iteratiilor pt blocul j
      for t=1:iter
```

```
%t
%Pw20
      % Decodorul 2
         % Recurenta inainte (dec.2)
         for i=1:N
           pw20=exp(-(r(2,i+j*N-N)+1)*(r(2,i+j*N-N)+1)/sig2);
           pw21=exp(-(r(2,i+j*N-N)-1)*(r(2,i+j*N-N)-1)/sig2);
           pv30=exp(-(r(3,i+j*N-N)+1)*(r(3,i+j*N-N)+1)/sig2);
           pv31=exp(-(r(3,i+j*N-N)-1)*(r(3,i+j*N-N)-1)/sig2);
             pw20=pw20/(pw20+pw21);
             pw21=pw21/(pw20+pw21);
             pv30=pv30/(pv30+pv31);
             pv31=pv31/(pv30+pv31);
           Px20(i)=pw20;

           P(1,i+1)=Pw20(i)*pw20*pv30*P(1,i)+(1-Pw20(i))*pw21*pv31*P(2,i);
           P(2,i+1)=(1-Pw20(i))*pw21*pv30*P(3,i)+Pw20(i)*pw20*pv31*P(4,i);
           P(3,i+1)=Pw20(i)*pw20*pv30*P(2,i)+(1-Pw20(i))*pw21*pv31*P(1,i);
           P(4,i+1)=(1-Pw20(i))*pw21*pv30*P(4,i)+Pw20(i)*pw20*pv31*P(3,i);
              sum=P(1,i+1)+P(2,i+1)+P(3,i+1)+P(4,i+1);
           P(1,i+1)=P(1,i+1)/sum;
           P(2,i+1)=P(2,i+1)/sum;
           P(3,i+1)=P(3,i+1)/sum;
           P(4,i+1)=P(4,i+1)/sum;
         end
%P
         % Recurenta inapoi (dec.2)
         Pb(1,N+1)=P(1,N+1);
         Pb(2,N+1)=P(2,N+1);
         Pb(3,N+1)=P(3,N+1);
         Pb(4,N+1)=P(4,N+1);

         for i=N:-1:1
           pw20=exp(-(r(2,i+j*N-N)+1)*(r(2,i+j*N-N)+1)/sig2);
           pw21=exp(-(r(2,i+j*N-N)-1)*(r(2,i+j*N-N)-1)/sig2);
           pv30=exp(-(r(3,i+j*N-N)+1)*(r(3,i+j*N-N)+1)/sig2);
           pv31=exp(-(r(3,i+j*N-N)-1)*(r(3,i+j*N-N)-1)/sig2);
             pw20=pw20/(pw20+pw21);
             pw21=pw21/(pw20+pw21);
             pv30=pv30/(pv30+pv31);
             pv31=pv31/(pv30+pv31);

           Pb(1,i)=Pw20(i)*pw20*pv30*Pb(1,i+1)+(1-Pw20(i))*pw21*pv31*Pb(3,i+1);
           Pb(2,i)=Pw20(i)*pw20*pv30*Pb(3,i+1)+(1-Pw20(i))*pw21*pv31*Pb(1,i+1);
           Pb(3,i)=Pw20(i)*pw20*pv31*Pb(4,i+1)+(1-Pw20(i))*pw21*pv30*Pb(2,i+1);
           Pb(4,i)=Pw20(i)*pw20*pv31*Pb(2,i+1)+(1-Pw20(i))*pw21*pv30*Pb(4,i+1);
```

```
            sum=Pb(1,i)+Pb(2,i)+Pb(3,i)+Pb(4,i);
          Pb(1,i)=Pb(1,i)/sum;
          Pb(2,i)=Pb(2,i)/sum;
          Pb(3,i)=Pb(3,i)/sum;
          Pb(4,i)=Pb(4,i)/sum;

          Pw20(i)=P(1,i)*Pb(1,i+1)*pv30+P(2,i)*Pb(3,i+1)*pv30+P(3,i)*Pb(4,i+1)*(1-
pv30)+P(4,i)*Pb(2,i+1)*(1-pv30);
          Pw21(i)=P(1,i)*Pb(3,i+1)*(1-pv30)+P(2,i)*Pb(1,i+1)*(1-
pv30)+P(3,i)*Pb(2,i+1)*pv30+P(4,i)*Pb(4,i+1)*pv30;
          Pw20(i)=Pw20(i)/(Pw20(i)+Pw21(i));
          if Pw20(i)>1-ELIM
            Pw20(i)=1-ELIM;
          elseif Pw20(i)<ELIM
            Pw20(i)=ELIM;
          end
        end

        % Inf. extinsec dec.2 deinterlivata
        for i=1:N
          Pv20(i)=Px20(rnum(i)); % lui dec1 in loc de bit receptionat (r2) se livreaza
P(r2)
        end
        for i=1:N
          Px20(i)=Pw20(rnum(i)); % inf. extrinseca de la dec2 catre dec1
        end
%Pb
%Pv20
%Px20
      % Decodorul 1
        % Recurenta inainte (dec.1)
        for i=1:N
          g0=exp(-(r(1,i+j*N-N)+1)*(r(1,i+j*N-N)+1)/sig2);
          g1=exp(-(r(1,i+j*N-N)-1)*(r(1,i+j*N-N)-1)/sig2);
           P(1,i+1)=g0*Px20(i)*Pv20(i)*P(1,i)+g1*(1-Px20(i))*(1-Pv20(i))*P(2,i);
           P(2,i+1)=g0*(1-Px20(i))*(1-Pv20(i))*P(4,i)+g1*Px20(i)*Pv20(i)*P(3,i);
           P(3,i+1)=g0*Px20(i)*Pv20(i)*P(2,i)+g1*(1-Px20(i))*(1-Pv20(i))*P(1,i);
           P(4,i+1)=g0*(1-Px20(i))*(1-Pv20(i))*P(3,i)+g1*Px20(i)*Pv20(i)*P(4,i);
          sum=P(1,i+1)+P(2,i+1)+P(3,i+1)+P(4,i+1);
           P(1,i+1)=P(1,i+1)/sum;
           P(2,i+1)=P(2,i+1)/sum;
           P(3,i+1)=P(3,i+1)/sum;
           P(4,i+1)=P(4,i+1)/sum;
        end
%P
        % Recurenta inapoi (dec.1)
```

```
        Pb(1,N+1)=1;
        Pb(2,N+1)=0;
        Pb(3,N+1)=0;
        Pb(4,N+1)=0;
        for i=N:-1:1
           g0=exp(-(r(1,i+j*N-N)+1)*(r(1,i+j*N-N)+1)/sig2);
           g1=exp(-(r(1,i+j*N-N)-1)*(r(1,i+j*N-N)-1)/sig2);
             Pb(1,i)=g0*Px20(i)*Pv20(i)*Pb(1,i+1)+g1*(1-Px20(i))*(1-
Pv20(i))*Pb(3,i+1);
             Pb(2,i)=g0*Px20(i)*Pv20(i)*Pb(3,i+1)+g1*(1-Px20(i))*(1-
Pv20(i))*Pb(1,i+1);
             Pb(3,i)=g0*(1-Px20(i))*(1-
Pv20(i))*Pb(4,i+1)+g1*Px20(i)*Pv20(i)*Pb(2,i+1);
             Pb(4,i)=g0*(1-Px20(i))*(1-
Pv20(i))*Pb(2,i+1)+g1*Px20(i)*Pv20(i)*Pb(4,i+1);
           sum=Pb(1,i)+Pb(2,i)+Pb(3,i)+Pb(4,i);
             Pb(1,i)=Pb(1,i)/sum;
             Pb(2,i)=Pb(2,i)/sum;
             Pb(3,i)=Pb(3,i)/sum;
             Pb(4,i)=Pb(4,i)/sum;

        % OUTPUT dec.1
           Pu0(i)=g0*(P(1,i)*Pb(1,i+1)+P(2,i)*Pb(3,i+1))*Px20(i)*Pv20(i);
           Pu0(i)=Pu0(i)+g0*(P(3,i)*Pb(4,i+1)+P(4,i)*Pb(2,i+1))*(1-Px20(i))*(1-
Pv20(i));
              Pu1=g1*(P(1,i)*Pb(3,i+1)+P(2,i)*Pb(1,i+1))*(1-Px20(i))*(1-Pv20(i));
              Pu1=Pu1+g1*(P(3,i)*Pb(2,i+1)+P(4,i)*Pb(4,i+1))*Px20(i)*Pv20(i);
           Pu0(i)=Pu0(i)/(Pu0(i)+Pu1);
        % Decizia / refacerea semnalului de iesire
           uh(i+j*N-N)=-1;
           if Pu0(i)<0.5
              uh(i+j*N-N)=1;
           end

        % Inf. extinsec dec.1

Pv20(i)=P(1,i)*Pb(1,i+1)*g0+P(2,i)*Pb(3,i+1)*g0+P(3,i)*Pb(2,i+1)*g1+P(4,i)*Pb(4,i+1
)*g1;

Pv21(i)=P(1,i)*Pb(3,i+1)*g1+P(2,i)*Pb(1,i+1)*g1+P(3,i)*Pb(4,i+1)*g0+P(4,i)*Pb(2,i+1
)*g0;
             Pv20(i)=Pv20(i)/(Pv20(i)+Pv21(i));
           if Pv20(i)>1-ELIM
              Pv20(i)=1-ELIM;
           elseif Pv20(i)<ELIM
              Pv20(i)=ELIM;
```

```
            end
         end
%Pb
%Pu0
%uh
         % Inf. extinsec dec.1 interlivata
         for i=1:N
            Pw20(rnum(i))=Pv20(i); % inf. de la dec1 catre dec2 interlivata
         end

      % Erori la prezentul pas de iterare
         nrer(t,j)=0;
         for i=1:N
            if v(1,i+j*N-N)~=uh(i+j*N-N)
               nrer(t,j)=nrer(t,j)+1;
            end
         end
nrer
      % Pasul iterativ urmator
      end

% Blocul urmator
end

%Date finale

%Index utilixat
   % i, j  uzuali
   % t     contorul iteratiilor
%Variabile utilizate
   % af1   parametru temporar in codoare
   % af2   parametru temporar in codoare
   % DB    RSZ in dB
   % ELIM  prag pt. prob minima
   % iter  numarul de iteratii
   % k     lungimea de constrangere
   % M     numarul de stari posibile
   % n     numarul de blocuri de informatie
   % N     lungimea unui bloc de informatie
   % R     rata de codare
   % sigma dispersia zgomotului
   % sig2  dublul puterii zg.
   % sum   suma de probabilitati
   % s0    starea primei celulei (pt. ambele codoare)
   % s1    starea celulei a doua (pt. ambele codoare)
%Vectori utilizati
```

```
% rnum(N)   functia interliver
% tu(n*N)   timpul la informatie
% tv(n*N/R) timpul in canal
% u(n*N)    secventa de date (informatia)
% v(3,n*N)  secventa codata / iesirea codorului 1
% w(n*N)    iesirea codorului 1 interlivata
```

Anexa F

Decodarea în frecvență a codurilor Reed-Solomon. Algoritmul Berlekamp-Massey

Descrierea teoretică a algoritmului decodării

Ipoteze: - câmpul $GF(2^3)$ generat prin $p(x) = x^3 + x + 1$

(1)

- codul RS (7, 3, 2)
- polinomul generator dintre:

A. $g_A(x) = (x + 1)(x + \alpha)(x + \alpha^2)(x + \alpha^3) =$
$= x^4 + \alpha^2x^3 + \alpha^5x^2 + \alpha^5x + \alpha^6 = (1\ 3\ 6\ 6\ 7)$

(2)

B. $g_B(x) = (x + \alpha)(x + \alpha^2)(x + \alpha^3)(x + \alpha^4) =$
$= x^4 + \alpha^3x^3 + x^2 + \alpha x + \alpha^3 = (1\ 4\ 1\ 2\ 4)$

(3)

- cuvântul emis $v(x) = 0$

(4)

- cuvântul eroare $\varepsilon(x) = \alpha^4x^3 + \alpha^2x =$

$$= \sum_{i=0}^{n-1} \varepsilon_i x^i = (0\ 0\ 0\ 5\ 0\ 3\ 0)$$

(5)

- cuvântul recepționat $w(x) = v(x) + \varepsilon(x) = \varepsilon(x)$

(6)

Considerații teoretice: - Transformata Fourier Discretă $\varepsilon_i \xleftrightarrow{\text{TFD}} E_k$

$$E_k = \sum_{i=0}^{n-1} \varepsilon_i \alpha^{ik} = \varepsilon(\alpha^k) \qquad k = \overline{0, n-1}$$

(7)

$$\varepsilon_i = \frac{1}{n} \sum_{k=0}^{n-1} E_k \alpha^{-ik} = E(\alpha^{-i}) \qquad i = \overline{0, n-1}$$

(8)

- polinomul erorii $\Lambda(x) = (xX_{i1} + 1)(xX_{i2} + 1) =$
$= 1 + \Lambda_1 x + \Lambda_2 x^2$

(9)

- locatorii erorilor X_{i1}, X_{i2}:

$\Lambda(X_{i1}^{-1}) = 0 = \Lambda(X_{i2}^{-1})$ $X_{i1} = \alpha^{i1}$, $X_{i2} = \alpha^{i2}$

(10)

- ecuația erorilor: $\Lambda(x)E(x) = \Gamma(x)(x^n + 1)$

(11)

datorită gradelor polinoamelor: $\Gamma(x) = ax + b$

(12)

Obs: *(11) se datorează: α^{ij}, cu $j = i_1, i_2$, e rădăcină a lui $\Lambda(x)$, ec. (10)*

α^{i_j}, cu $j \neq i_1, i_2$, e rădăcină a lui E(x), ec. (8)($e_j \neq 0$ doar pentru $j = i_1$ şi i_2)

Soluționarea problemei:

- se identifică coeficienții ecuației (11):

$$\begin{array}{ll} x^0 & E_0 = b \\ x^1 & E_1 + \Lambda_1 E_0 = a \\ x^2 & E_2 + \Lambda_1 E_1 + \Lambda_2 E_0 = 0 \\ \hline x^3 & E_3 + \Lambda_1 E_2 + \Lambda_2 E_1 = 0 \\ x^4 & E_4 + \Lambda_1 E_3 + \Lambda_2 E_2 = 0 \\ x^5 & E_5 + \Lambda_1 E_4 + \Lambda_2 E_3 = 0 \\ x^6 & E_6 + \Lambda_1 E_5 + \Lambda_2 E_4 = 0 \\ x^7 & \Lambda_1 E_6 + \Lambda_2 E_5 = b \\ x^8 & \Lambda_2 E_6 = a \end{array} \qquad (13)$$

- se află coeficienții sindromului:

$$A\begin{cases} S_0 = w(\alpha^0) = \alpha^4 + \alpha^2 = \alpha \\ S_1 = w(\alpha^1) = \alpha^7 + \alpha^3 = \alpha \\ S_2 = w(\alpha^2) = \alpha^3 + \alpha^4 = \alpha^6 \\ S_3 = w(\alpha^3) = \alpha^6 + \alpha^5 = \alpha \end{cases} B. \Rightarrow$$

$$S_4 = w(\alpha^4) = \alpha^2 + \alpha^6 = 1$$

$$A\begin{cases} E_0 = S_0 = \alpha \\ E_1 = S_1 = \alpha = E_1 \\ E_2 = S_2 = \alpha^6 = E_2 \\ E_3 = S_3 = \alpha = E_3 \\ S_4 = 1 = E_4 \end{cases} B. \Rightarrow \qquad (14)$$

- se află coeficienții Λ_1 şi Λ_2 din ec. (13)

$$A\left\{\begin{array}{ll} x^2 & E_2 + \Lambda_1 E_1 + \Lambda_2 E_0 = 0 \\ x^3 & E_3 + \Lambda_1 E_2 + \Lambda_2 E_1 = 0 \\ x^4 & E_4 + \Lambda_1 E_3 + \Lambda_2 E_2 = 0 \end{array}\right\} B$$

$$A.\begin{cases} \Lambda_1 = \dfrac{E_1E_2 + E_0E_3}{E_1^2 + E_0E_2} = 1 \\ \Lambda_2 = \dfrac{E_1E_3 + E_2^2}{E_1^2 + E_0E_2} = \alpha^4 \end{cases} \qquad B.\begin{cases} \Lambda_1 = \dfrac{E_2E_3 + E_1E_4}{E_2^2 + E_1E_3} = 1 \\ \Lambda_2 = \dfrac{E_2E_4 + E_3^2}{E_2^2 + E_1E_3} = \alpha^4 \end{cases} \qquad (15)$$

- se află pozițiile eronate:

$$\Lambda(x) = \alpha^4 x^2 + x + 1$$

(16)

cu metoda descompunerii în factori a lui $\Lambda(x)$:

$$\Lambda(x) = \alpha^4 x^2 + (\alpha^3 + \alpha)x + 1 = \alpha^3(\alpha x + 1)x + \alpha x + 1 =$$
$$= (\alpha^3 x + 1)(\alpha x + 1)$$

(17)

$$\Rightarrow x_{i1}^{-1} = \alpha^4 \Rightarrow x_{i1} = \alpha^3 \Rightarrow i_1 = 3$$
$$x_{i2}^{-1} = \alpha^6 \Rightarrow x_{i2} = \alpha \Rightarrow i_2 = 1$$

(18)

cu metoda căutării:

$\Lambda(\alpha^0) = \alpha^4 + 1 + 1 = \alpha^4 \neq 0$
$\Lambda(\alpha) = \alpha^6 + \alpha + 1 = \alpha^4 \neq 0$
$\Lambda(\alpha^2) = \alpha + \alpha^2 + 1 = \alpha^5 \neq 0$
$\Lambda(\alpha^3) = \alpha^3 + \alpha^3 + 1 = 1 \neq 0$
$\Lambda(\alpha^4) = \alpha^5 + \alpha^4 + 1 = 0 \Rightarrow$ poziție eronată $i_1 = 7 - 4 = 3$
$\Lambda(\alpha^5) = 1 + \alpha^5 + 1 = \alpha^5 \neq 0$
$\Lambda(\alpha^6) = \alpha^2 + \alpha^6 + 1 = 0 \Rightarrow$ poziție eronată $i_2 = 7 - 6 = 1$

(19)

- se află valoarea caracterelor eronate v_3 şi v_1.

Se calculează polinomul $\Gamma(x)$:

A. $b = E_0 = \alpha$
$a = E_1 + \Lambda_1 E_0 = \alpha + \alpha = 0 \quad \Rightarrow \Gamma(x) = \alpha$

(20)

B. $b = E_0 = 1/\Lambda_2(E_2 + \Lambda_1 E_1) = (\alpha^6 + \alpha)/\alpha^4 = \alpha$
$a = E_1 + \Lambda_1 E_0 = E_1 + \Lambda_1 b = 0 \Rightarrow \Gamma(x) = \alpha$

(21)

Se derivează formal polinomul $\Lambda(x) = (\alpha^3 x + 1)(\alpha x + 1)$:

$\Lambda'(x) = \alpha^3(\alpha x + 1) + \alpha(\alpha^3 x + 1) = \alpha^3 + \alpha = 1$

(22)

Se calculează coeficienții cuvântului eroare cu formula:

$$\varepsilon_i = \frac{\alpha^i \Gamma(\alpha^{-i})}{\Lambda'(\alpha^{-i})} \text{ pentru } i = i_1 \text{ şi } i = i_2$$

(23)

Obs: *formula (23) se află derivând formal (11) şi ținând cont de (8) şi de faptul că*
$\Lambda(\alpha^{-i1}) = 0 = \Lambda(\alpha^{-i2})$:
$\Lambda'(x)E(x) + \Lambda(x)E'(x) = \Gamma'(x)(x^n + 1) + nx^{n-1}\Gamma(x)$
cu $x = \alpha^{-i}$, $i = i_1$ *sau* $i = i_2 \Rightarrow \Lambda(\alpha^{-i}) = 0 = (\alpha^{-i})^n + 1$
iar (24)devine:
$\Lambda'(\alpha^{-i})E(\alpha^{-i}) = n\alpha^{-ni}\alpha^{i}\Gamma(\alpha^{-i})$
de unde rezultă (23).

$\varepsilon_3 = \alpha^3\alpha /1 = \alpha^4 \quad \Rightarrow v_3 = w_3 + \varepsilon_3 = \alpha^4 + \alpha^4 = 0$
$\varepsilon_1 = \alpha\alpha /1 = \alpha^2 \quad v_1 = w_1 + \varepsilon_1 = \alpha^2 + \alpha^2 = 0$

(24)

Algoritmul Berlekamp-Massey –aplicație

Cazul A : $g(x) = g_A(x)$

- se calculează coeficienții sindrom:

$E_0 = S_0 = \alpha$
$E_1 = S_1 = \alpha$
$E_2 = S_2 = \alpha^6$
$E_3 = S_3 = \alpha$

- se află $\Lambda(x)$ şi $\Gamma(x)$ prin aplicarea algoritmului B-M:

iniţializare r = 0 $\Lambda^{(0)}(x) = 1 \quad \Gamma^{(0)}(x) = 0 \quad L_0 = 0$
$B^{(0)}(x) = 1 \quad A^{(0)}(x) = x^{-1}$

pasul r = 1

- se calculează discrepanța:

$E(x)$: $E_0 = \alpha \quad E_1 = \alpha \quad E_2 = \alpha^6 \quad E_3 = \alpha$
$\Lambda^{(0)}(x) = 1$: $\Lambda_2{}^0 = 0 \quad \Lambda_1{}^0 = 0 \quad \Lambda_0{}^0 = 1$
$\Delta = 1 \cdot E_0 = \alpha \neq 0$

- se calculează coef. $\delta_1 \begin{cases} 1 \text{ dacă } \Delta_1 \neq 0 \text{ şi } 2L_0 \leq 0 \ = r - 1 \\ 0 \text{ în rest} \end{cases}$

$\Rightarrow \delta_1 = 1$

-se calculează matricea D_1:

$$D_1 = \begin{bmatrix} 1 & \Delta_1 x \\ \delta_1 \Delta^{-1} & (1-\delta_1)x \end{bmatrix} = \begin{bmatrix} 1 & E_0 x \\ E_0^{-1} & 0 \end{bmatrix} = \begin{bmatrix} 1 & \alpha x \\ \alpha^6 & 0 \end{bmatrix}$$

-se calculează $\Lambda'(x)$ şi $B'(x)$:

$$\begin{bmatrix} \Lambda'(x) \\ B'(x) \end{bmatrix} = D_1 \begin{bmatrix} \Lambda^0(x) \\ B^0(x) \end{bmatrix} = \begin{bmatrix} 1 + E_0 x \\ E_0^{-1} \end{bmatrix} = \begin{bmatrix} 1 + \alpha x \\ \alpha^6 \end{bmatrix}$$

-se calculează $\Gamma'(x)$ şi $A'(x)$:

$$\begin{bmatrix} \Gamma'(x) \\ A'(x) \end{bmatrix} = D_1 \begin{bmatrix} \Gamma^0(x) \\ A^0(x) \end{bmatrix} = \begin{bmatrix} E_0 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$$

-se calculează coeficientul L_1:

$L_1 = \max(L_0, r - L_0) = \max(0, 1) = 1$

pasul r = 2

- se calculează discrepanța:

$E(x)$: $E_0 = \alpha \quad E_1 = \alpha \quad E_2 = \alpha^6 \quad E_3 = \alpha$

$\Lambda'(x) = 1 + \alpha x = 1 + E_0 x$: $\Lambda_2{}' = 0 \quad \Lambda_1{}' = \alpha \quad \Lambda_0{}' = 1$
$\Delta_2 = \Lambda_0{}' E_1 + \Lambda_1{}' E_0 = E_1 + E_0{}^2 = \alpha + \alpha^2 = \alpha^4 \neq 0$

- se calculează coef. $\begin{cases} 1 \text{ dacă } \Delta_2 \neq 0 \text{ şi } 2L_1 \leq 1 \end{cases}$

$$\delta_2 \begin{cases} \\ \lfloor\ 0 \text{ în rest} \end{cases}$$

$\Rightarrow \delta_2 = 0$

-se calculează matricea D_2:

$$D_2 = \begin{bmatrix} 1 & \Delta_2 x \\ \delta_2 \Delta_2^{-1} & (1-\delta_2)x \end{bmatrix} = \begin{bmatrix} 1 & (E_1 + E_0^2)x \\ 0 & x \end{bmatrix} = \begin{bmatrix} 1 & \alpha^4 x \\ 0 & x \end{bmatrix}$$

-se calculează $\Lambda^{(2)}(x)$ şi $B^{(2)}(x)$:

$$\begin{bmatrix} \Lambda^{(2)}(x) \\ B^{(2)}(x) \end{bmatrix} = D_2 \begin{bmatrix} \Lambda^{(1)}(x) \\ B^{(1)}(x) \end{bmatrix} = \begin{bmatrix} 1 + E_0^{-1}E_1 x \\ E_0^{-1} x \end{bmatrix} = \begin{bmatrix} 1+x \\ \alpha^6 x \end{bmatrix}$$

-se calculează $\Gamma^{(2)}(x)$ şi $A^{(2)}(x)$:

$$\begin{bmatrix} \Gamma^{(2)}(x) \\ A^{(2)}(x) \end{bmatrix} = D_2 \begin{bmatrix} \Gamma^{(1)}(x) \\ B^{(1)}(x) \end{bmatrix} = \begin{bmatrix} E_0 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$$

-se calculează coeficientul L_2:

$L_2 = \max(L_1, r - L_1) = \max(1, 1) = 1$

pasul r = 3

- se calculează discrepanța:

$E(x)$: $E_0 = \alpha$ $E_1 = \alpha$ $E_2 = \alpha^6$ $E_3 = \alpha$

$\Lambda^{(2)}(x) = 1 + x = 1 + E_0^{-1}E_1 x$:

$\Lambda_2^{(2)} = 0$ $\Lambda_1^{(2)} = E_0^{-1}E_1 = 1$ $\Lambda_0^{(2)} = 1$

$\Delta_3 = \Lambda_2^{(2)}E_0 + \Lambda_1^{(2)}E_1 + \Lambda_0^{(2)}E_2 = E_2 + E_0^{-1}E_1^2 = \alpha^6 + \alpha = \alpha^5 \neq 0$

- se calculează coef. $\delta_3 \begin{cases} 1 \text{ dacă } \Delta_3 \neq 0 \text{ şi } 2L_2 = 2 \leq r - 1 = 2 \\ \lfloor\ 0 \text{ în rest} \end{cases}$

$\Rightarrow \delta_3 = 1$

-se calculează matricea D_3:

$$D_3 = \begin{bmatrix} 1 & \Delta_3 x \\ \delta_3 \Delta_3^{-1} & (1-\delta_3)x \end{bmatrix} = \begin{bmatrix} 1 & (E_2 + E_0^{-1}E_1^2)x \\ (E_2 + E_0^{-1}E_1^2)^{-1} & 0 \end{bmatrix} = \begin{bmatrix} 1 & \alpha^5 x \\ \alpha^2 & 0 \end{bmatrix}$$

-se calculează $\Lambda^{(3)}(x)$ şi $B^{(3)}(x)$:

$$\begin{bmatrix} \Lambda^{(3)}(x) \\ B^{(3)}(x) \end{bmatrix} = D_3 \begin{bmatrix} 1 + E_0^{-1}E_1 x \\ E_0^{-1} x \end{bmatrix} = \begin{bmatrix} 1 + E_0^{-1}E_1 x + E_0^{-1}(E_2 + E_0^{-1}E_1^2)x^2 \\ (1 + E_0^{-1}E_1 x)(E_2 + E_0^{-1}E_1^2)^{-1} \end{bmatrix} =$$

$$= \begin{bmatrix} 1 + x + \alpha^4 x^2 \\ \alpha^2 + \alpha^2 x \end{bmatrix}$$

-se calculează $\Gamma^{(3)}(x)$ şi $A^{(3)}(x)$:

$$\begin{bmatrix} \Gamma^{(3)}(x) \\ B^{(3)}(x) \end{bmatrix} = D_3 \begin{bmatrix} E_0 \\ 0 \end{bmatrix} = \begin{bmatrix} E_0 \\ E_0(E_2 + E_0^{-1}E_1^2)^{-1} \end{bmatrix} = \begin{bmatrix} \alpha \\ \alpha^3 \end{bmatrix}$$

-se calculează coeficientul L_3:

$L_3 = \max(L_2, 3 - L_2) = \max(1, 2) = 2$

pasul r = 4

- se calculează discrepanța:

$E(x)$: $E_0 = \alpha$ $E_1 = \alpha$ $E_2 = \alpha^6$ $E_3 = \alpha$

$\Lambda^{(3)}(x) = 1 + E_0^{-1}E_1x + E_0^{-2}(E_0E_2 + E_1^2)x^2 = 1 + x + \alpha^4x^2$:

$\Lambda_2^{(3)} = E_0^{-2}(E_0E_2 + E_1^2)$ $\Lambda_1^{(3)} = E_0^{-1}E_1$ $\Lambda_0^{(3)} = 1$

$\Delta_4 = E_3 + E_0^{-1}E_1E_2 + E_0^{-1}E_1E_2 + E_0^{-2}E_1^3 = E_3 + E_0^{-2}E_1^3 = 0$

- se calculează coef. $\delta_4 = 0$
- se calculează matricea D_4:

$$D_4 = \begin{bmatrix} 1 & \left(E_3 + E_0^{-2}E_1^3\right)x \\ 0 & x \end{bmatrix}$$

-se calculează $\Lambda^{(4)}(x)$ şi $B^{(4)}(x)$:

$$\begin{bmatrix} \Lambda^{(4)}(x) \\ B^{(4)}(x) \end{bmatrix} = D_4 \begin{bmatrix} 1 + E_0^{-1}E_1x + E_0^{-2}\left(E_0E_2 + E_1^2\right)x^2 \\ E_0\left(E_0E_2 + E_1^2\right)^{-1}\left(1 + E_0^{-1}E_1x\right) \end{bmatrix} =$$

$$= \begin{bmatrix} 1 + \dfrac{E_1E_2 + E_0E_3}{E_0E_2 + E_1^2}x + \dfrac{E_2^2 + E_1E_3}{E_0E_2 + E_1^2}x^2 \\ \dfrac{E_0}{E_0E_2 + E_1^2}x + \dfrac{E_1}{E_0E_2 + E_1^2}x^2 \end{bmatrix} = \begin{bmatrix} 1 + x + \alpha^4x^2 \\ \alpha^2x + \alpha^2x^2 \end{bmatrix}$$

-se calculează $\Gamma^{(4)}(x)$ şi $A^{(4)}(x)$:

$$\begin{bmatrix} \Gamma^{(4)}(x) \\ A^{(4)}(x) \end{bmatrix} = D\begin{bmatrix} E_0 \\ \dfrac{E_0^2}{E_0E_2 + E_1^2} \end{bmatrix} = \begin{bmatrix} E_0 + \dfrac{E_0^2E_3 + E_1^3}{E_0E_2 + E_1^2}x \\ \dfrac{E_0^2}{E_0E_2 + E_1^2}x \end{bmatrix} = \begin{bmatrix} \alpha \\ \alpha^3 \end{bmatrix}$$

Polinoamele $\Lambda(x)$ şi $\Gamma(x)$ sunt:

$$\Lambda(x) = \Lambda^4(x) = 1 + \frac{E_1E_2 + E_0E_3}{E_0E_2 + E_1^2}x + \frac{E_2^2 + E_1E_3}{E_0E_2 + E_1^3}x^2 = 1 + x + \alpha^4x^2$$

$$\Gamma(x) = \Gamma^4(x) = E_0 + \frac{E_0^2E_3 + E_1^3}{E_0E_2 + E_1^3}x = \alpha$$

- se află poziţiile eronate i_j prin metoda de căutare:

$\Lambda(\alpha^{-i}) = 0$ $i = 0, n-1$

rezultă (vezi ec. (19)): $i_1 = 3$ şi $i_2 = 1$

- se află valorile caracterelor eronate: v_3 şi v_1.
  - derivata formală a polinomului $\Lambda(x)$ este (vezi (22)):

    $\Lambda'(x) = 1$
  - coef. cuvântului eroare sunt (vezi (23) şi (24)):

    $\varepsilon_3 = \alpha^4$ $\varepsilon_1 = \alpha^2$
  - valorile caracterelor eronate sunt:

    $v_3 = w_3 + \varepsilon_3 = 0$

    $v_1 = w_1 + \varepsilon_1 = 0$
- se selectează caracterele de informaţie:
  - cuvântul de cod este v = 0 0 0 0 0 0 0
  - cuvântul de informaţie este i = 0 0 0

Cazul B $\mathbf{g(x) = g_B(x)}$

- se calculează coeficienţii sindrom:

$E_1 = S_1 = \alpha$

$E_2 = S_2 = \alpha^6$

$E_3 = S_3 = \alpha$

$E_4 = S_4 = 1$

- se află $\Lambda(x)$ şi $\Gamma(x)$ prin aplicarea algoritmului B-M:

iniţializare r = 0 $\Lambda^{(0)}(x) = 1 \quad \Gamma^{(0)}(x) = 0 \quad L_0 = 0$

$B^{(0)}(x) = 1 \quad A^{(0)}(x) = 1$

pasul r = 1

- se calculează discrepanţa:

E(x): $E_1 = \alpha \quad E_2 = \alpha^6 \quad E_3 = \alpha \quad E_4 = 1$

$\Lambda^{(0)}(x) = 1$: $\Lambda_2^{(0)} = 0 \quad \Lambda_1^{(0)} = 0 \quad \Lambda_0^{(0)} = 1$

$\Delta_1 = E_1\Lambda_0^{(0)} = \alpha \neq 0$

- se calculează coef. $\delta_1 \begin{cases} 1 \text{ dacă } \Delta_1 \neq 0 \text{ şi } 2L_0 \leq 0 = r - 1 \\ 0 \text{ în rest} \end{cases}$

$\Rightarrow \delta_1 = 1$

-se calculează matricea D_1:

$$D_1 = \begin{bmatrix} 1 & \Delta x \\ \delta_1\Delta^{-1} & (1-\delta_1)x \end{bmatrix} = \begin{bmatrix} 1 & E_1 x \\ E_1^{-1} & 0 \end{bmatrix} = \begin{bmatrix} 1 & \alpha x \\ \alpha^6 & 0 \end{bmatrix}$$

-se calculează $\Lambda^{(1)}(x)$ şi $B^{(1)}(x)$:

$$\begin{bmatrix} \Lambda^{(1)}(x) \\ B^{(1)}(x) \end{bmatrix} = D_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + E_1 x \\ E_1^{-1} \end{bmatrix} = \begin{bmatrix} 1 + \alpha x \\ \alpha^6 \end{bmatrix}$$

-se calculează $\Gamma^{(1)}(x)$ şi $A^{(1)}(x)$:

$$\begin{bmatrix} \Gamma^{(1)}(x) \\ A^{(1)}(x) \end{bmatrix} = D_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} E_1 x \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha x \\ 0 \end{bmatrix}$$

-se calculează coeficientul L_1:

$L_1 = \max(L_0, r - L_0) = \max(0, 1) = 1$

pasul r = 2

- se calculează discrepanţa Δ_2:

E(x): $E_1 = \alpha \quad E_2 = \alpha^6 \quad E_3 = \alpha \quad E_4 = 1$

$\Lambda^{(1)}(x) = 1 + E_1 x : \Lambda_2^{(1)} = 0 \quad \Lambda_1^{(1)} = E_1 \quad \Lambda_0^{(1)} = 1$

$\Delta_2 = E_1\Lambda_1^{(2)} + E_2\Lambda_0^{(2)} = E_1^2 + E_2 = 1 \neq 0$

- se calculează coef. $\delta_2 \begin{cases} 1 \text{ dacă } \Delta_2 \neq 0 \text{ şi } 2L_1 \leq r - 1 = 1 \\ 0 \text{ în rest} \end{cases}$

$\Rightarrow \delta_2 = 0$

-se calculează matricea D_2:

$$D_2 = \begin{bmatrix} 1 & \Delta_2 x \\ \delta_2 \Delta_2^{-1} & (1-\delta_2)x \end{bmatrix} = \begin{bmatrix} 1 & (E_1^2 + E_2)x \\ 0 & x \end{bmatrix} = \begin{bmatrix} 1 & x \\ 0 & x \end{bmatrix}$$

-se calculează $\Lambda^{(2)}(x)$ şi $B^{(2)}(x)$:

$$\begin{bmatrix} \Lambda^{(2)}(x) \\ B^{(2)}(x) \end{bmatrix} = D_2 \begin{bmatrix} 1 + E_1 x \\ E_1^{-1} \end{bmatrix} = \begin{bmatrix} 1 + E_1^{-1} E_2 x \\ E_1^{-1} x \end{bmatrix} = \begin{bmatrix} 1 + \alpha^5 x \\ \alpha^6 x \end{bmatrix}$$

-se calculează $\Gamma^{(2)}(x)$ şi $A^{(2)}(x)$:

$$\begin{bmatrix} \Gamma^{(2)}(x) \\ A^{(2)}(x) \end{bmatrix} = D_2 \begin{bmatrix} E_1 x \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha x \\ 0 \end{bmatrix}$$

-se calculează coeficientul L_2:

$L_2 = \max(L_1, r - L_1) = \max(1, 1) = 1$

pasul r = 3

- se calculează discrepanța Δ_3:

E(x): $E_1 = \alpha$ $E_2 = \alpha^6$ $E3 = \alpha$ $E_4 = 1$

$\Lambda^{(2)}(x) = 1 + E_1^{-1}E_2 x$: $\Lambda_2^{(2)} = 0$ $\Lambda_1^{(2)} = E_1^{-1E}{}_2 = 1$ $\Lambda_0^{(2)} = 1$

$\Delta_3 = E_3 + E_1^{-1}E_2^2 = \alpha^2 \neq 0$

- se calculează coef. $\delta_3 \begin{cases} 1 \text{ dacă } \Delta_3 \neq 0 \text{ şi } 2L_2 \leq r - 1 = 2 \\ 0 \text{ în rest} \end{cases}$

$\Rightarrow \delta_3 = 1$

-se calculează matricea D_3:

$$D_3 = \begin{bmatrix} 1 & \Delta_3 x \\ \delta_3 \Delta_3^{-1} & (1-\delta_3)x \end{bmatrix} = \begin{bmatrix} 1 & (E_3 + E_1^{-1}E_2^2)x \\ (E_3 + E_1^{-1}E_2^2)^{-1} & 0 \end{bmatrix} = \begin{bmatrix} 1 & \alpha^2 x \\ \alpha^5 & 0 \end{bmatrix}$$

-se calculează $\Lambda^{(3)}(x)$ şi $B^{(3)}(x)$:

$$\begin{bmatrix} \Lambda^{(3)}(x) \\ B^{(3)}(x) \end{bmatrix} = D_3 \begin{bmatrix} 1 + E_1^{-1}E_2 x \\ E_1^{-1} x \end{bmatrix} = \begin{bmatrix} 1 + E_1^{-1}E_2 x + E_1^{-1}(E_3 + E_1^{-1}E_2^2)x^2 \\ (E_3 + E_1^{-1}E_2^2)^{-1} + E_1^{-1}E_2(E_3 + E_1^{-1}E_2^2)^{-1} x \end{bmatrix} =$$

$$= \begin{bmatrix} 1 + \alpha^5 x + \alpha x^2 \\ \alpha^5 + \alpha^3 x \end{bmatrix}$$

-se calculează $\Gamma^{(3)}(x)$ şi $A^{(3)}(x)$:

$$\begin{bmatrix} \Gamma^{(3)}(x) \\ A^{(3)}(x) \end{bmatrix} = D_3 \begin{bmatrix} E_1 x \\ 0 \end{bmatrix} = \begin{bmatrix} E_1 x \\ \dfrac{E_1^2 x}{E_3 E_1 + E_2^2} \end{bmatrix} = \begin{bmatrix} \alpha x \\ \alpha^6 x \end{bmatrix}$$

-se calculează coeficientul L_3:

$L_3 = \max(L_2, r - L_2) = \max(1, 2) = 2$

pasul r = 4

- se calculează discrepanța Δ_4:

$E(x)$: $E_1 = \alpha$ $E_2 = \alpha^6$ $E_3 = \alpha$ $E_4 = 1$

$\Lambda^{(3)}(x) = 1 + E_1^{-1}E_2x + E_1^{-1}(E_3 + E_1^{-1}E_2^2)x^2$:

$\Lambda_2^{(3)} = E_1^{-1}(E_3 + E_1^{-1}E_2^2)$ $\Lambda_1^{(3)} = E_1^{-1}E_2$ $\Lambda_0^{(3)} = 1$

$\Delta_4 = E_4 + E_1^{-1}E_2E_3 + E_1^{-1}E_2(E_3 + E_1^{-1}E_2^2) = \alpha^6 \neq 0$

- se calculează coef. . $\delta_4 \begin{cases} 1 \text{ dacă } \Delta_4 \neq 0 \text{ şi } 2L_3 = 4 \leq r - 1 = 3 \\ 0 \text{ în rest} \end{cases}$

$\Rightarrow \delta_4 = 0$

- se calculează matricea D_4:

$$D_4 = \begin{bmatrix} 1 & \Delta_4 x \\ \delta_4\Delta_4^{-1} & (1-\delta_4)x \end{bmatrix} = \begin{bmatrix} 1 & \left(E_4 + E_1^{-2}E_2^3\right)x \\ 0 & x \end{bmatrix} = \begin{bmatrix} 1 & \alpha^6 x \\ 0 & x \end{bmatrix}$$

-se calculează $\Lambda^{(4)}(x)$ şi $B^{(4)}(x)$:

$$\begin{bmatrix} \Lambda^{(4)}(x) \\ B^{(4)}(x) \end{bmatrix} = D_4 \begin{bmatrix} 1 + E_1^{-1}E_2x + E_1^{-1}\left(E_3 + E_1^{-1}E_2^2\right)x^2 \\ \left(E_3 + E_1^{-1}E_2^2\right)^{-1} + E_1^{-1}E_2\left(E_3 + E_1^{-1}E_2^2\right)^{-1}x \end{bmatrix} =$$

$$= \begin{bmatrix} 1 + \dfrac{E_1E_4 + E_2E_3}{E_1E_3 + E_2^2}x + \dfrac{E_3^2 + E_2E_4}{E_1E_3 + E_2^2}x^2 \\ \left(E_3 + E_1^{-1}E_2^2\right)^{-1}x + E_1^{-1}E_2\left(E_3 + E_1^{-1}E_2^2\right)^{-1}x^2 \end{bmatrix} = \begin{bmatrix} 1 + x + \alpha^4x^2 \\ \alpha^5x + \alpha^3x^2 \end{bmatrix}$$

-se calculează $\Gamma^{(4)}(x)$ şi $A^{(4)}(x)$:

$$\begin{bmatrix} \Gamma^{(4)}(x) \\ A^{(4)}(x) \end{bmatrix} = D_4 \begin{bmatrix} E_1x \\ \dfrac{E_1^2x}{E_1E_3 + E_2^2} \end{bmatrix} = \begin{bmatrix} E_1 + \dfrac{E_1^2E_4 + E_2^3}{E_1E_3 + E_2^2}x \\ \dfrac{E_1^2}{E_1E_3 + E_2^2}x \end{bmatrix} = \begin{bmatrix} \alpha x + \alpha^5x^2 \\ \alpha^6x \end{bmatrix}$$

Polinoamele $\Lambda(x)$ şi $\Gamma(x)$ sunt:

$$\Lambda(x) = \Lambda^4(x) = 1 + x + \alpha^4x^2$$

$$\Gamma(x) = \Gamma^4(x) = \alpha x + \alpha^5x^2$$

- se află poziţiile eronate i_j prin metoda căutare:

$\Lambda(\alpha^{-i}) = 0$ $i = 0, n-1$

rezultă (vezi ec. (19)): $i_1 = 3$ şi $i_2 = 1$

- se află valorile caracterelor eronate: v_3 şi v_1.

- derivata formală a polinomului $\Lambda(x)$ este (vezi (22)):

$\Lambda'(x) = 1$

- coef. cuvântului eroare sunt (vezi (23) şi (24)):

$\varepsilon_3 = \alpha^3\Gamma(\alpha^{-3}) / \Lambda'(\alpha^{-3}) = \alpha^3(\alpha^{-2} + \alpha^{-1}) = \alpha^3 \cdot \alpha = \alpha^4$

$\varepsilon_1 = \alpha\Gamma(\alpha^{-1}) / \Lambda'(\alpha^{-1}) = \alpha(1 + \alpha^3) = \alpha \cdot \alpha = \alpha^2$

- valorile caracterelor eronate sunt:

$v_3 = w_3 + \varepsilon_3 = 0$

$$v_1 = w_1 + \varepsilon_1 = 0$$

- se selectează caracterele de informație:
  - cuvântul de cod este v = 0 0 0 0 0 0 0
  - cuvântul de informație este i = 0 0 0

Anexa G

1. Calculul funcţiei de transfer T(δ,β,λ), relaţia (6.3).

Utilizând notaţiile din Figura G.1, se pot scrie ecuaţiile:

$$\begin{cases} w = \delta^2 \cdot \beta \cdot \lambda \cdot x + \beta \cdot \lambda \cdot v \\ u = \delta \cdot \beta \cdot \lambda \cdot w + \delta \cdot \beta \cdot \lambda \cdot u \\ v = \delta \cdot \lambda \cdot w + \delta \cdot \lambda \cdot u \end{cases} \qquad \text{(G.1)}$$

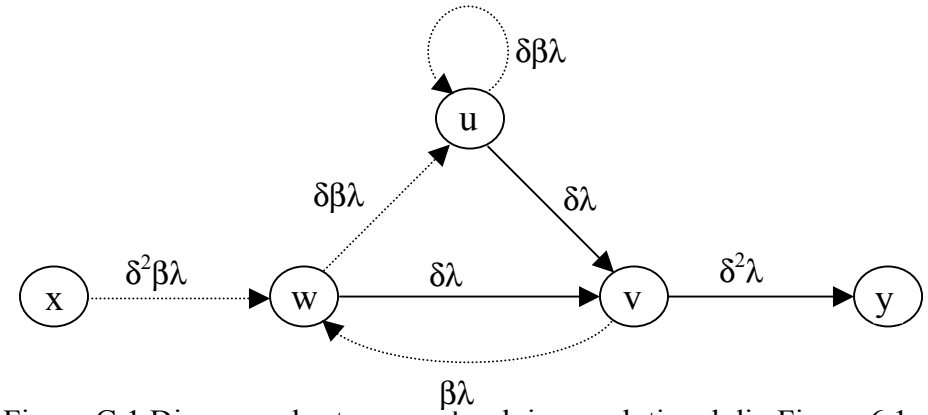


Figura G.1 Diagrama de stare a codorului convoluţional din Figura 6.1

Ultimele două ecuaţii din (G.1) se pot scrie:

$$u = \beta \cdot v$$
$$\delta \cdot \lambda \cdot w = (1 + \delta \cdot \beta \cdot \lambda) \cdot v$$

Eliminând pe w între ultima ecuaţie şi prima din (G.1) rezultă:

$$v \cdot (1 + \delta \cdot \beta \cdot \lambda + \delta \cdot \beta \cdot \lambda^2) = \delta^3 \cdot \beta \cdot \lambda^2 \cdot x$$

Din Figura G.1 rezultă că $y = \delta^2 \cdot \lambda \cdot v$, ca atare putem calcula funcţia de transfer:

$$T(\delta, \beta, \lambda) = \frac{y}{x} = \frac{\delta^5 \cdot \beta \cdot \lambda^3}{1 + \delta\beta\lambda \cdot (1 + \lambda)} \qquad \text{(G.2)}$$

Ecuaţia (G.2) se poate obţine şi prin regula lui Mason:

$$T = \frac{\sum_i C_i \cdot (1 - \sum B_{ij})}{1 - \sum_j B_j} \qquad (G.3)$$

unde: $-C_i$ = transmitanța căii „i" între nodurile final şi inițial;
$-B_j$ = „transmitanța" buclei „j", dintre cele ne-adiacente, peste tot graful;
$-B_{ij}$ = „transmitanța" buclei „j", dintre cele ne-adiacente căii „i";
Astfel, în graful din Figura G.1 există două bucle ne-adiacente, una aferentă nodului „u", de transmitanță $B_1 = \delta\cdot\beta\cdot\lambda$, iar cea de-a doua între nodurile „w" şi „v", de transmitanță $B_2 = \delta\cdot\beta\cdot\lambda^2$. De asemenea există două căi între nodurile „x" şi „y", şi anume: x-w-u-v-y şi x-w-v-y, având transmitanțele $C_1 = \delta^6\cdot\beta^2\cdot\lambda^4$ şi $C_2 = \delta^5\cdot\beta\cdot\lambda^3$. Cu acestea, funcția de transfer se scrie:

$$T(\delta,\beta,\lambda) = \frac{C_1 + C_2\cdot(1+B_2)}{1+B_1+B_2} = \frac{\delta^5\cdot\beta\cdot\lambda^3\cdot(1+\delta\cdot\beta\cdot\lambda)+\delta^6\cdot\beta^2\cdot\lambda^4}{1+\delta\cdot\beta\cdot\lambda+\delta\cdot\beta\cdot\lambda^2}$$

$$= \frac{\delta^5\cdot\beta\cdot\lambda^3}{1+\delta\beta\lambda\cdot(1+\lambda)}$$

Trebuie menționat că, în cazul sumei modulo-doi, operațiile de adunare şi scădere sunt identice.

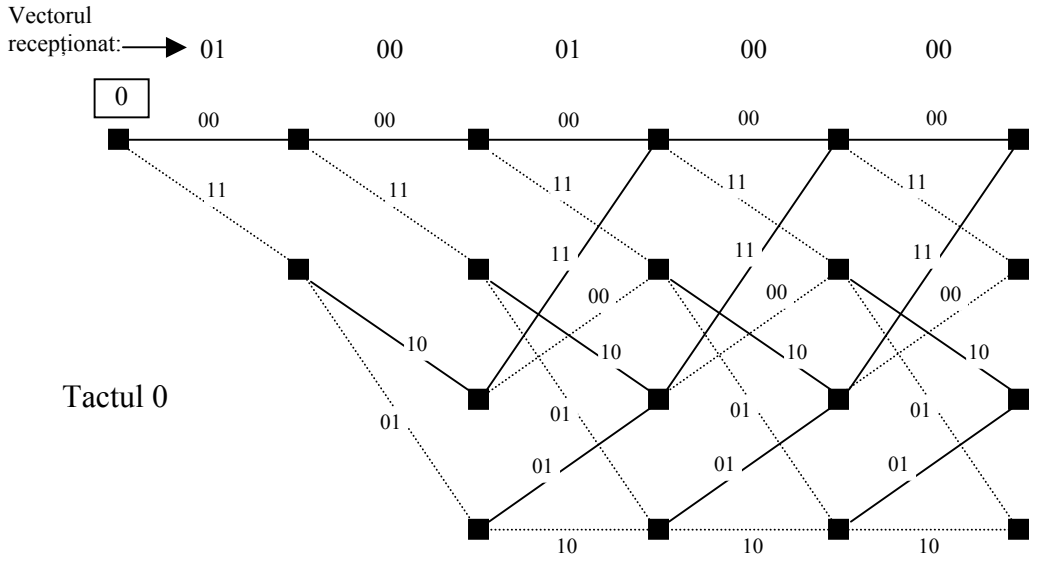


Figura G.2 Diagrama trellis şi semnalul recepționat în cazul canalului binar simetric

2.Exemplificarea algoritmului Viterbi

În cazul canalului CBS, diagrama trellis şi semnalul recepționat, arată ca în Figura G.2. Algoritmul recepției presupune, la fiecare tact, calculul metricii (relația (6.14)) şi selecția căii (ramurii) celei mai plauzibile. Deoarece în primele două tacturi, diagrama trellis nu conține noduri cu două intrări, selecția se face începând cu tactul trei, Figura G.3a.

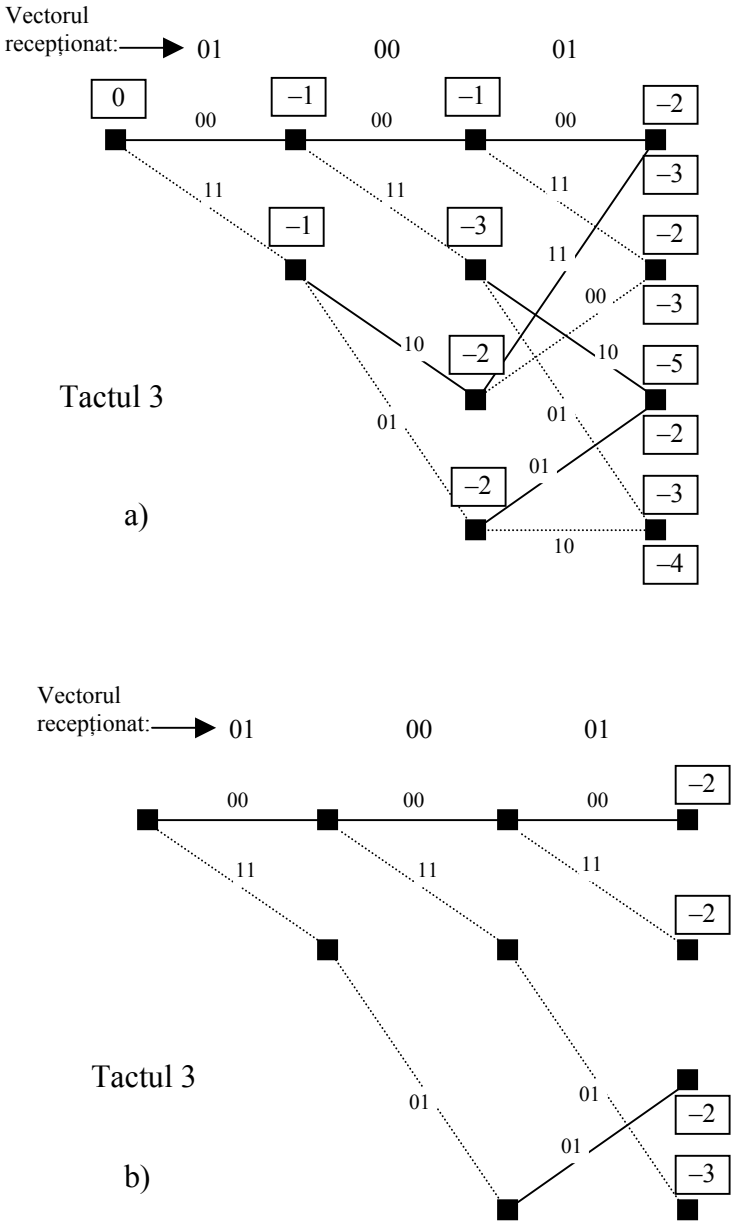


Figura G.3 Diagrama trellis până la tactul 3, a) –înainte şi b) –după selecția aferentă tactului trei;

După selecția de la tactul 3, au rămas doar patru drumuri posibile, având distanța față de secvența recepționată –2, (pentru primele trei), şi –3 (pentru ultimul). Aceste

distanţe constituie „zestrea” fiecărei căi şi, totodată, datele pentru un nou pas algoritmic, cel aferent tactului 4. Figurile G.4 şi G.5 prezintă operaţiile aferente tacturilor 4 şi 5.

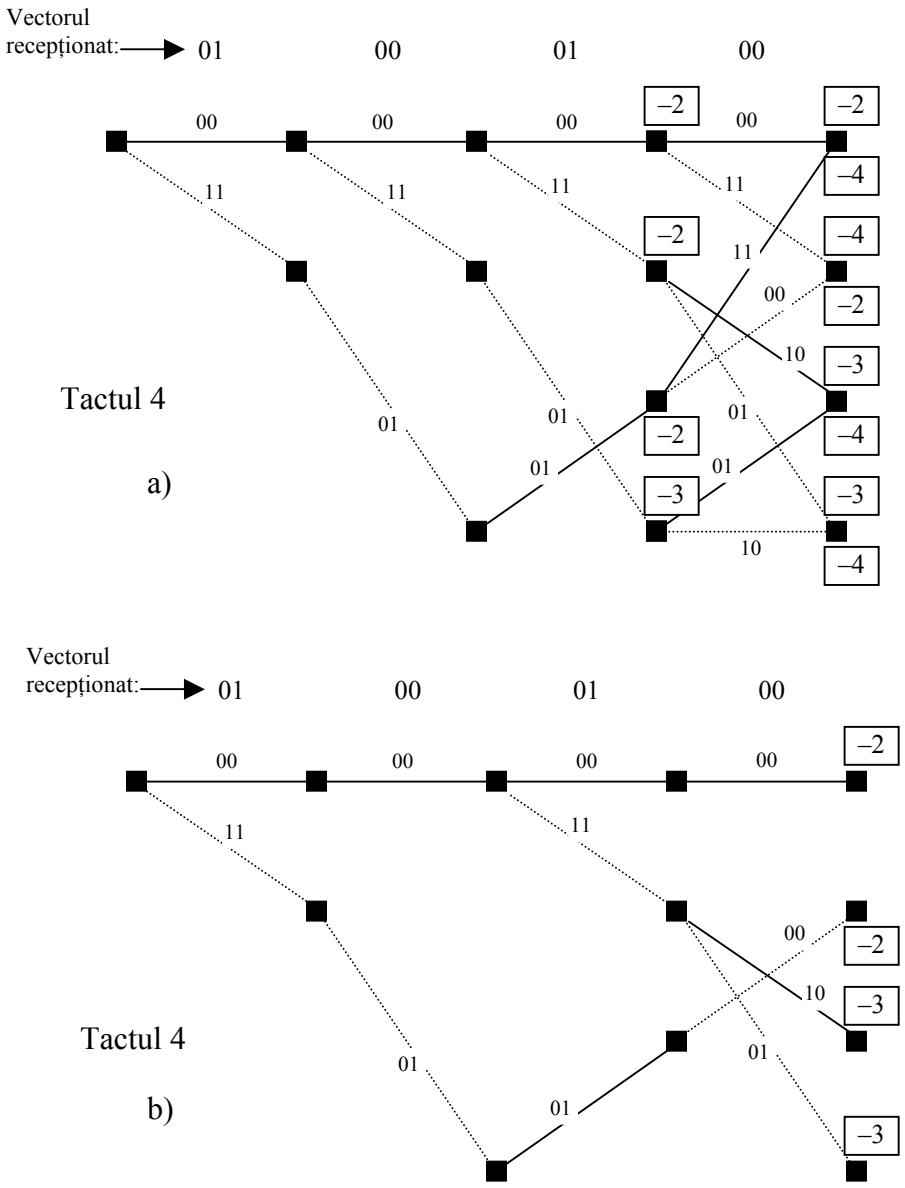
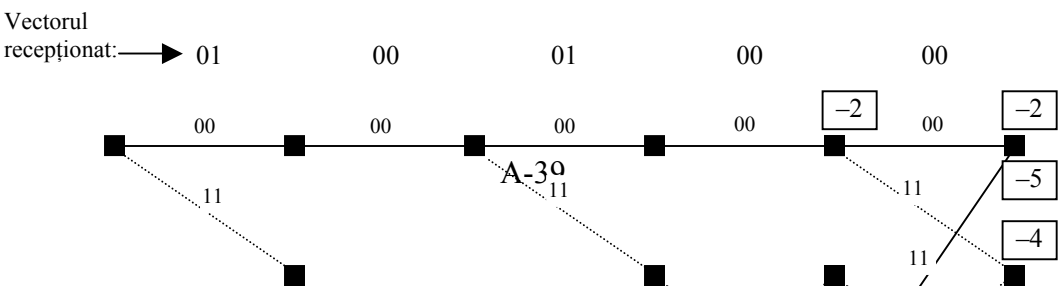


Figura G.4 Diagrama trellis la tactul 4: a) –înainte şi b) –după selecţie;



a)

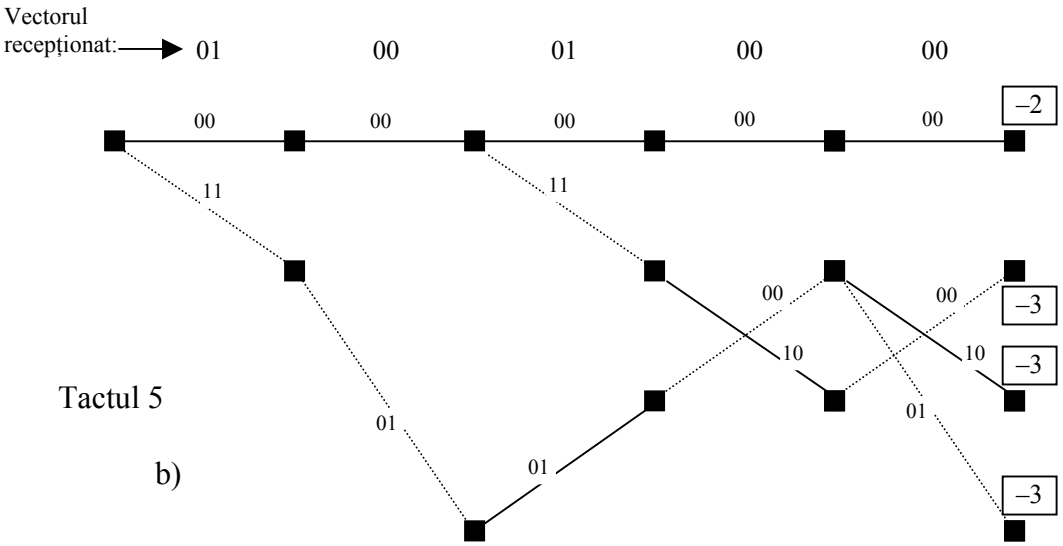
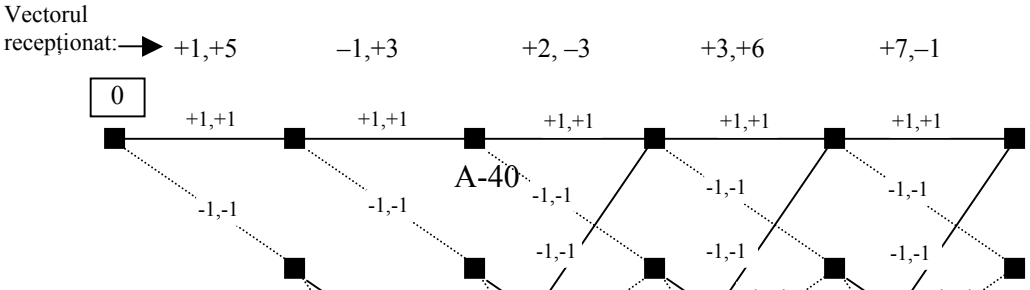


Figura G.5 Diagrama trellis la tactul 5: a) –înainte şi b) –după selecție;

Acelaşi algoritm, exemplificat pentru canalul AWGN, este prezentat prin figurile următoare. Metrica ramurii se calculează în acest caz după relația (6.16), iar ramura selectată este cea cu metrică mai mare (dintre cele două ce converg în acelaşi nod).



a)

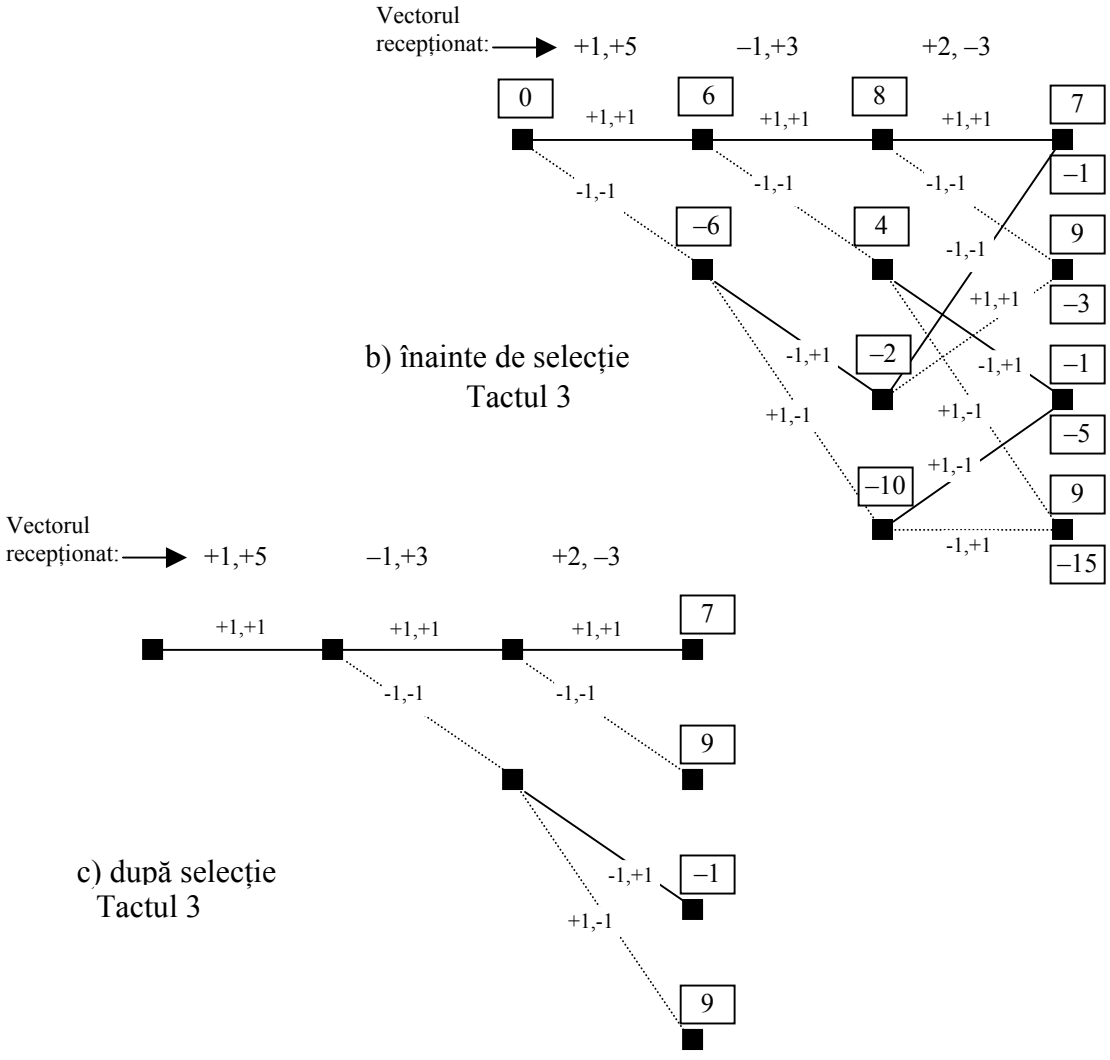
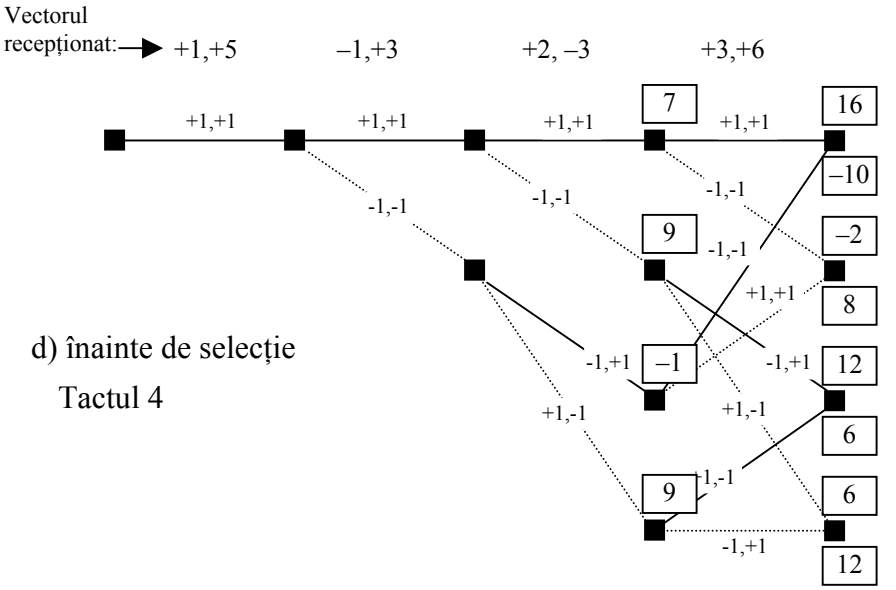


Figura G.6 Diagrama trellis în cazul canalului AWGN



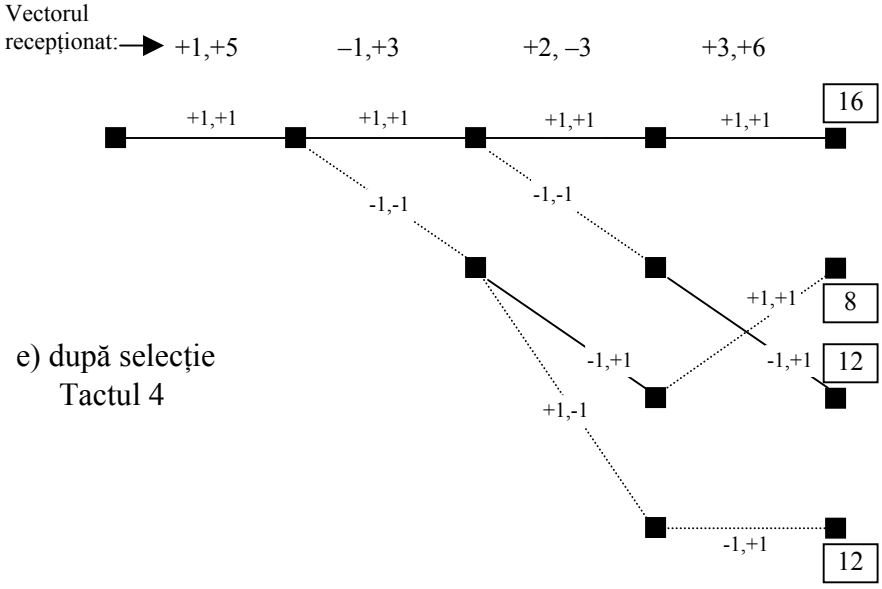
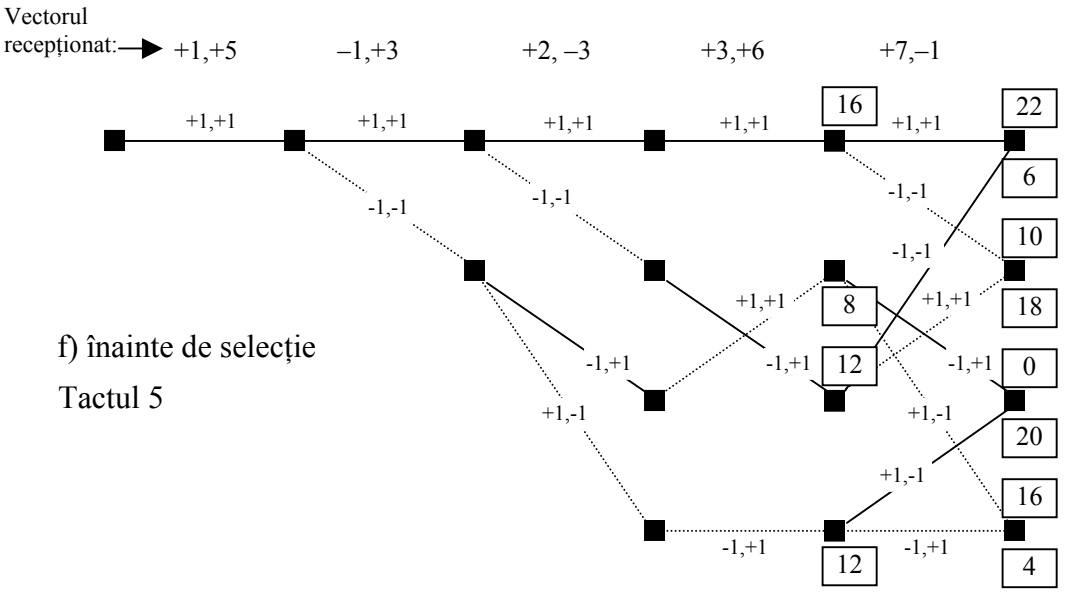


Figura G.6 Diagrama trellis în cazul canalului AWGN



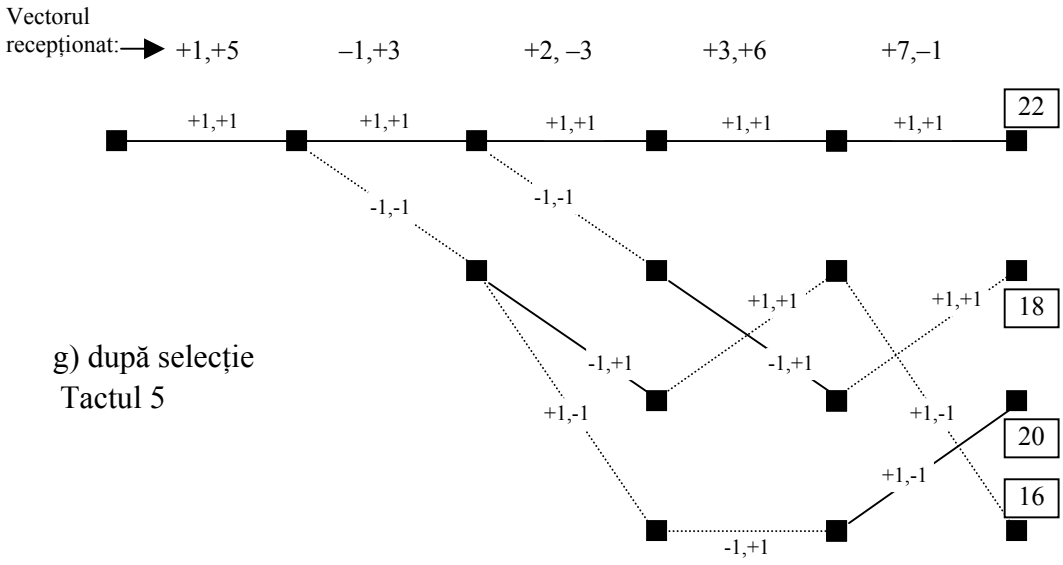


Figura G.6 Diagrama trellis în cazul canalului AWGN