



(12) CERERE DE BREVET DE INVENȚIE

(21) Nr. cerere: a 2017 00118

(22) Data de depozit: 28/02/2017

(41) Data publicării cererii:
29/09/2017 BOPI nr. 9/2017

(71) Solicitant:
• NEGHINĂ MIHAI, BD. VICTORIEI, NR.31A,
AP.21, SIBIU, SB, RO;
• STOIAN RODICA, CAL. PLEVNEI,
NR.145 B, BL.1, ET.6, AP.1061, SECTOR 6,
BUCUREȘTI, B, RO

(72) Inventatori:
• STOIAN RODICA, CALEA PLEVNEI
NR.145B, BL.1, ET.6, AP.1061, SECTOR 1,
BUCUREȘTI, B, RO;
• NEGHINĂ MIHAI, BD.VICTORIEI NR.31A,
AP.21, SIBIU, SB, RO

(54) METODĂ DE CODARE/DECODARE A DATELOR BINARE
ÎN STRUCTURI MULTIDIMENSIONALE PESTE CÂMPURI
GALOIS, IMPLEMENTATĂ CU AJUTORUL
CALCULATORULUI PENTRU STOCAREA DATELOR
ÎN MODULE DASD

(57) Rezumat:

Invenția se referă la o metodă de codare/ decodare a codurilor tridimensionale peste câmpuri Galois $GF(2)$, implementată cu ajutorul calculatorului, pentru stocarea datelor în module DASD (Direct Access Storage Device). Metoda conform invenției constă în: alegerea codului (bloc) $B(n_1, k, t)$ corector de t erori și detector de b erori, capabil să codeze k biți de informație în n_1 biți, cod care să fie folosit de-a lungul dimensiunii x_1 ; alegerea lungimilor structurii, n_1, n_2, n_3 , de-a lungul dimensiunilor x_1, x_2 și, respectiv, x_3 , conform următoarelor constrângeri: n_2 și n_3 să fie prime între ele și $n_1 > n_2 \cdot n_3$; alegerea regulii de generare a diagonalelor generalizate cu ajutorul pasului s , conform constrângerilor: $s = (\alpha_1, \alpha_2, \alpha_3)$ și $\text{cmmdc}(\alpha_i, n_i) = 1, \forall, i$; codarea a $k \cdot (n_2 - 1) \cdot (n_3 - 1)$ biți de informație într-o structură tridimensională $n_1 \cdot n_2 \cdot n_3$; stocarea datelor în module DASD, undefiecare modul DASD conține câte o diagonală generalizată a structurii tridimensionale, iar defecțiunea sau ștergerea unui modul DASD corespunde cu introducerea unui pachet de erori de lungime $n_2 \cdot n_3$; decodarea, cu corecția a până la $t+3$ pachete de erori de lungime $n_2 \cdot n_3$, fiecare.

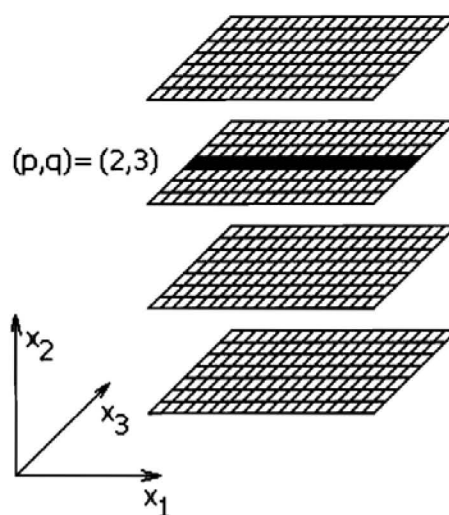
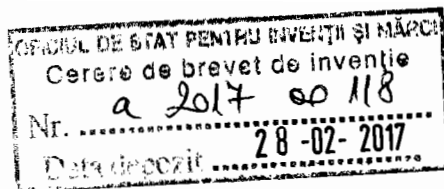


Fig. 3

Revendicări: 2
Figuri: 8

Cu începere de la data publicării cererii de brevet, cererea asigură, în mod provizoriu, solicitantului, protecția conferită potrivit dispozițiilor art.32 din Legea nr.64/1991, cu excepția cazurilor în care cererea de brevet de invenție a fost respinsă, retrasă sau considerată ca fiind retrasă. Întinderea protecției conferite de cererea de brevet de invenție este determinată de revendicările conținute în cererea publicată în conformitate cu art.23 alin.(1) - (3).





Metodă de codare decodare a datelor binare în structuri multidimensionale peste câmpuri Galois, implementată cu ajutorul calculatorului pentru stocarea datelor în module DASD

Prezenta invenție se referă la o metodă de stocare a datelor în module DASD implementată cu ajutorul calculatorului prin codarea și decodarea datelor binare în structuri multidimensionale peste câmpuri Galois $GF(2)$ cu ajutorul calculatorului, capabila sa corecteze pachete de erori.

Cauzele acestor pachete de erori sunt datorate mediilor de stocare care cedează sau sunt distruse, sau când mediul de stocare este propice deteriorării secvențelor de date, cum ar fi de exemplu distrugerea informațiilor optice datorată unei expuneri la lumină/întuneric timp îndelungat. O altă cauză o reprezintă stocarea informațiilor utilizând suporturi sensibile la câmpurile magnetice.

În prezent sunt cunoscute codurile bidimensionale și aplicarea acestora în domeniul stocării de date (US 5271012) document care descrie o metodă de configurare a mediilor de stocare, codarea și recuperarea datelor în cazul în care maxim două module cedează, brevetul prezentând o aplicație a acestora în stocarea de date (DASD- Direct Access Storage Device).

Mai sunt cunoscute din documentul „*A family of efficient burst-correcting array codes*” codurile bidimensionale care menționează unii parametri ai codurilor, dimensiuni, constrângeri, etc, modalitățile de codare, respectiv formarea matricelor de date și decodare, respectiv identificarea și corectarea pachetelor de eroare, sau performanțele comparative față de alte tipuri de coduri bloc.

Documentul „*The burst error correcting capabilities of a simple array code*” prezintă un nou algoritm de decodare pentru coduri bidimensionale corectoare de pachete de erori.

Problema tehnică pe care o rezolvă prezenta invenție constă în creșterea performanțelor de detecție și corecție a erorilor.

1. Metoda de codare decodare a codurilor tridimensionale peste câmpuri Galois $GF(2)$, implementată cu ajutorul calculatorului, pentru stocarea datelor în module DASD conform prezentei invenții revendicate constă în următorii pași:
- Alegerea codului (bloc) $B(n_1, k, t)$ corector de t erori și detector de b erori, capabil să codeze k biți de informație în n_1 biți, cod care să fie folosit de-a lungul dimensiunii (x_1)
 - Alegerea lungimilor structurii n_1, n_2 și n_3 de-a lungul dimensiunilor $(x_1), (x_2)$ și respectiv (x_3) conform constrângerilor:
 - n_2 și n_3 să fie prime între ele: $\text{cmmdc}(n_2, n_3) = 1$
 - $n_1 \gg n_2 \cdot n_3$
 - Alegerea regulii de generare a diagonalelor generalizate cu ajutorul pasului s conform constrângerilor
 - $s = (\alpha_1, \alpha_2, \alpha_3)$
 - $\text{cmmdc}(\alpha_i, n_i) = 1, \forall i$
 - Codarea a $k \cdot (n_2 - 1) \cdot (n_3 - 1)$ biți de informație într-o structură tridimensională $n_1 \times n_2 \times n_3$
 - Stocarea datelor în module DASD, unde fiecare modul DASD conține câte o diagonală generalizată a structurii tridimensionale, iar defecțiunea sau ștergerea unui DASD corespunde cu introducerea unui pachet de erori de dimensiune $n_2 \cdot n_3$.
 - Decodarea cu corecția a până la $t+3$ pachete de erori de lungime $n_2 \cdot n_3$ fiecare
2. Metoda de codare decodare a codurilor D -dimensionale, $D > 3$, peste câmpuri Galois $GF(2)$, implementată cu ajutorul calculatorului, pentru stocarea datelor în module DASD conform prezentei invenții revendicate constă în următorii pași:
- Alegerea codului (bloc) $B(n_1, k, t)$ corector de t erori și detector de b erori, capabil să codeze k biți de informație în n_1 biți, cod care să fie folosit de-a lungul dimensiunii (x_1)
 - Alegerea lungimilor structurii n_i de-a lungul dimensiunilor (x_i) , conform constrângerilor:
 - n_i și n_j să fie prime între ele: $\text{cmmdc}(n_i, n_j) = 1, \forall i \neq j, i > 1, j > 1$
 - $n_1 \gg \prod_{i=2}^D n_i$
 - Alegerea regulii de generare a diagonalelor generalizate cu ajutorul pasului s și a ordinii de parcurgere a acestora cu ajutorul vectorului v (vectorul v dictează ordinea de parcurgere a diagonalelor prin indicarea ordinii primelor elemente ale fiecărei diagonale) în vederea vectorizării (transformării matricii tridimensionale într-un vector de biți), conform constrângerilor
 - $s = (\alpha_1, \alpha_2, \dots, \alpha_D)$
 - $\text{cmmdc}(\alpha_i, n_i) = 1, \forall i$
 - Codarea a $k \cdot \prod_{i=2}^D (n_i - 1)$ biți de informație într-o structură D -dimensională $n_1 \times n_2 \times \dots \times n_D$
 - Stocarea datelor în module DASD, unde fiecare modul DASD conține câte o diagonală generalizată a structurii D -dimensionale, iar defecțiunea sau ștergerea unui DASD corespunde cu introducerea unui pachet de erori de dimensiune $\prod_{i=2}^D n_i$.
 - Decodarea cu corecția a până la $t + 2^{D-1} - 1$ pachete de erori de lungime $\prod_{i=2}^D n_i$ fiecare

Avantajele pe care le aduce prezenta invenție sunt:

- posibilitatea de decodare corectă sau recuperare a datelor deteriorate.
- creșterea capacității de corecție a pachetelor de erori față de codurile bidimensionale;

Pentru o prezentare a metodei de implementare a codurilor tridimensionale peste câmpuri Galois cu ajutorul calculatorului, se va descrie o formă de realizare a acestuia, cu referire și la figurile anexate care reprezintă :

Figura 1 – exemplu de vizualizare pentru un cod tridimensional;

Figura 2 – exemplu grafic a ordinii de calcul a biților în algoritmul de codare;

Figura 3 – exemplu de vizualizare pentru un cod tridimensional cu identificarea explicită a liniei (p,q) ;

Figura 4 – reprezentare 2D a codului tridimensional folosit pentru exemplificare;

Figura 5 – reprezentare 2D a codului tridimensional folosit pentru exemplificare;

Figura 6 – exemple de intervale cu erori (mov) și fără erori (verde) pentru care toate sindromurile verticale nu detectează erori, în reprezentarea 2D a unui cod multidimensional;

Figura 7 – reprezentări 2D succesive ale stării de eronare a biților din codul tridimensional pe parcursul rulării algoritmului de decodare a $b = t+2$ pachete de erori.

Figura 8 – reprezentări 2D succesive ale stării de eronare a biților din codul tridimensional pe parcursul rulării algoritmului de decodare a $b = t+3$ pachete de erori.

Codurile multidimensionale corectoare de pachete de erori rezolvă aceleași tipuri de probleme ca și codurile bidimensionale corectoare de pachete de erori, însă cu performanțe mai bune.

Dându-se un cod D -dimensional $C(n_1 \times n_2 \times \dots \times n_D, s, v, t)$, rata de transmisie/stocare R și rata de corecție R_C sunt:

Numărul biților de informație împărțit la numărul total de biți:	Numărul maxim de biți corecți împărțit la numărul total de biți:
$R = \frac{k \cdot \prod_{i=2}^D (n_i - 1)}{\prod_{i=1}^D (n_i - 1)}$	$R_C = \frac{b \cdot \prod_{i=2}^D (n_i - 1)}{\prod_{i=1}^D (n_i - 1)} \rightarrow \frac{b}{n_1}$

Pentru valori mari ale $n_3 \dots n_D$, rata de transmisie a codurilor multidimensionale se apropie de rata de transmisie/stocare a codurilor bidimensionale. Numărul de pachete de erori corectabile este $b = t + 2^{D-1} - 1$. Particularizând pentru $D=2$ se obține $b=t+1$, formula pentru codurile bidimensionale. Capacitatea de corecție crește exponențial cu dimensionalitatea.

P

- Pentru aceeași rată de transmisie/stocare R , rata de corecție R_C este mai bună
- Pentru aceeași rată de corecție R_C , rata de transmisie/stocare R este mai bună

Domeniile de aplicabilitate ale acestor coduri includ:

- Transmisii de date pe canale care produc pachete de erori (mai degrabă decât erori izolate)
 - Transmisii de pachete de date care pot fi deteriorate/pierdute
 - Transmisii de informații greu accesibile (de exemplu de la sateliți și roboți de explorare) a căror importanță justifică precauțiunile
 - Transmisii pe canale cu bruij sporadic (care deteriorează secvențe de date)
- Medii de stocare, pentru a permite reconstituirea informației pierdute atunci când:
 - mai multe module de stocare cedează/sunt distruse
 - mediul de stocare este propice deteriorării secvențelor de date (de exemplu informații optice distruse prin expunerea timp îndelungat la lumină/întuneric, sau informații stocate pe suporturi sensibile la câmpuri magnetice externe)

Exemplu de codare, stocare și corecție

Introducere

Fie codul multidimensional $C(n_1 \times n_2 \times \dots \times n_D, s, v, t)$, unde:

- D reprezintă numărul de dimensiuni ($D = 3$ în exemplu)
- n_1, \dots, n_D reprezintă numărul de biți de-a lungul dimensiunilor $1, \dots, D$
- s reprezintă pasul (step) de parcurgere a biților, important pentru vectorizare $s = (1,1,1)$
- v este un vector de inițializare, important pentru vectorizare $v = [a_{00..0}, a_{n_1 0..0}, \dots, a_{10..0}]$
- t este capacitatea de corecție a codului $B(n_1, k, t)$ folosit de-a lungul primei dimensiuni
- k este numărul de biți de informație folosiți pentru construcția unui cuvânt din $B(n_1, k, t)$
- $a_{i1,i2,i3}$ este un bit în codul multidimensional (tridimensional)

Pentru vizualizare, în *figura 1* este reprezentat un cod tridimensional $C(5 \times 2 \times 3, s, v, t)$. Codul folosit în exemplu va fi $C(n_1 \times 2 \times 3, s, v, t)$, unde $n_1 \gg n_2 \cdot n_3$. Biții sunt așezați într-o structură $5 \times 2 \times 3$ pentru a putea fi desenați și vizualizați.

Observatie: codurile sunt multidimensionale doar din punctul de vedere al relațiilor matematice între biți. Pentru transmisie sau stocare, blocurile multidimensionale sunt vectorizate, iar prelucrarea vectorului de date (între codare/vectorizare și reasamblare/decodare) nu ține cont de multidimensionalitatea relațiilor între biți.

56

Exemplu de codare

În urma codării a $k \cdot (n_2 - 1) \cdot (n_3 - 1)$ biți de informație într-o structură tridimensională $n_1 \times n_2 \times n_3$, se vor obtine următoarele relații între biții a_{i_1, i_2, i_3} ai structurii tridimensionale:

{ Orice rând de-a lungul primei dimensiuni este un cuvânt de cod din $B(n_1, k, t)$
 { Orice rând de-a lungul celorlalte dimensiuni are suma egală cu 0 modulo 2

- pentru i_2 de la 1 la $(n_2 - 1)$
 pentru i_3 de la 1 la $(n_3 - 1)$
 - Se selectează următorii k biți de informație
 - Se codează cei k biți de informație selectați cu ajutorul codului $B(n_1, k, t)$
 - Se poziționează cuvântul de cod rezultat (n_1 biți) pe rândul i_2 al stratului i_3
- Se calculează valorile biților a_{i_1, i_2, i_3} pentru care $i_2 = 0$ sau $i_3 = 0$

$$a_{i_1, 0, i_3} = \left(\sum_{i_2=1}^{n_2-1} a_{i_1, i_2, i_3} \right) \text{ modulo } 2 \quad a_{i_1, i_2, 0} = \left(\sum_{i_3=1}^{n_3-1} a_{i_1, i_2, i_3} \right) \text{ modulo } 2$$

- Se calculează valorile $a_{i_1, 0, 0} = \left(\sum_{i_2=1}^{n_2-1} a_{i_1, i_2, 0} \right) \text{ modulo } 2 = \left(\sum_{i_3=1}^{n_3-1} a_{i_1, 0, i_3} \right) \text{ modulo } 2$

Conform *figurii 1*, ordinea de completare a biților în tabel ar fi:

- Roșu: biți de informație originali (k pe fiecare linie)
- Portocaliu: biți de control conform $B(n_1, k, t)$
- Galben: biți de paritate de-a lungul celorlalte dimensiuni
- Albastru: biți de paritate ai biților de paritate

Exemplu de stocare pe module DASD

În exemplul aplicației de stocare a datelor vor fi folosite n_1 module DASD. Numărul modulelor DASD poate fi ajustat prin alegerea corespunzătoare a codului $B(n_1, k, t)$, care codează k biți de informație în n_1 biți de informație și control de-a lungul unei linii din codul multidimensional. Dimensiunea modulelor de stocare poate fi controlată prin alegerea corespunzătoare a valorilor n_2, \dots, n_D reprezentând numărul de biți de-a lungul dimensiunilor 2, 3, ..., D . În cazul codului tridimensional $C(n_1 \times 2 \times 3, s, v, t)$, fiecare DASD va stoca $2 \times 3 = 6$ biți.

Fiecare DASD va stoca câte o diagonală generalizată a codului multidimensional. Pentru generarea unei diagonale generalizate se folosește pasul $s = (\alpha_1, \alpha_2, \alpha_3)$ de parcurgere a biților, pentru care $\text{cmmdc}(\alpha_i, n_i) = 1, \forall i$.

$$a_{i,j,k} \cdot a_{(i+\alpha_1) \bmod n_1, (j+\alpha_2) \bmod n_2, (k+\alpha_3) \bmod n_3}$$

În acest exemplu $C(n_1 \times 2 \times 3, s=(1, 1, 1), v, t)$, așadar ordinea biților va fi:

$$a_{i,j,k} \cdot a_{(i+1) \bmod n_1, (j+1) \bmod 2, (k+1) \bmod 3}$$

Fiecare DASD va stoca câte o diagonală generalizată a codului multidimensional, anume biții diagonalei generalizate care pornește din poziția $a_{1,0,0}$ și continuă conform regulii de parcurgere cu pas $s = (1,1,1)$ pentru încă $n_2 \cdot n_3 - 1$ biți.

- DASD_0 va conține biții $a_{0,0,0} \dots a_{1,1,1} \dots a_{2,0,2} \dots a_{3,1,0} \dots a_{4,0,1} \dots a_{5,1,2}$
- DASD_1 va conține biții $a_{1,0,0} \dots a_{2,1,1} \dots a_{3,0,2} \dots a_{4,1,0} \dots a_{5,0,1} \dots a_{6,1,2}$
- ...
- DASD_{(n_1-1)} va conține biții $a_{(n_1-1),0,0} \dots a_{0,1,1} \dots a_{1,0,2} \dots a_{2,1,0} \dots a_{3,0,1} \dots a_{4,1,2}$

Ordinea diagonalelor generalizate, dată de vectorul v este importantă doar pentru vectorizarea completă a datelor în vederea transmisiei lor. Vectorizarea datelor este obținută prin concatenarea diagonalelor generalizate în ordinea indicată de vectorul v .

Noțiuni necesare pentru corecție și decodare

Fie (p,q) cea de-a $(q+1)$ -a linie din al $(p+1)$ -lea strat al codul tridimensional. Cu alte cuvinte, linia (p,q) conține biții $[a_{0,p,q}, a_{1,p,q}, \dots, a_{(n_1-1),p,q}]$.

Reprezentarea 2D a codurilor multidimensionale este ordonarea liniilor din codul multidimensional conform regulii de parcurgere data de pasul $s = (\alpha_1, \alpha_2, \alpha_3)$.

- Linia $((0 \cdot \alpha_2) \bmod n_2, (0 \cdot \alpha_3) \bmod n_3)$
- Linia $((1 \cdot \alpha_2) \bmod n_2, (1 \cdot \alpha_3) \bmod n_3)$
- ...
- Linia $(((n_2 \cdot n_3 - 1) \cdot \alpha_2) \bmod n_2, ((n_2 \cdot n_3 - 1) \cdot \alpha_3) \bmod n_3)$

Figura 3 este similară cu *figura 1*, cu diferențele că este o structură $20 \times 4 \times 7$, forma abstractă a biților $(a_{i,j,k})$ nu mai este scrisă în mod explicit în fiecare dreptunghi și în plus este marcată linia $(p,q) = (2,3)$. Numărătoarea straturilor p și a rândurilor q începe de la 0.

O reprezentare similară se poate obține și pentru codurile multidimensionale cu mai mult de 3 dimensiuni. În cazul exemplului, $C(n_1 \times 2 \times 3, s=(1,1,1), v, t)$, reprezentarea 2D echivalentă va fi conform *figurii 4*, unde culorile identifică biții de informație, de control și de paritate cu același cod de culori ca în *figura 2*.

Diagonalele generalizate, conform *figurii 5*, vor fi acum diagonale adevărate ale reprezentării 2D, după cum reiese din figură. Culorile identifică diagonale generalizate ale codului. Toți biții unei diagonale generalizate vor fi salvați pe aceeași unitate DASD.

Sindromurile orizontale h_{pq} sunt sindromuri de-a lungul fiecărei linii, conform codului bloc $B(n_1, k, t)$ folosit în codare. Fiecare linie (p,q) are un sindrom h_{pq} asociat, din care se poate determina dacă pe linia (p,q) sunt maxim t erori (și unde se găsesc acestea) sau dacă sunt mai mult de t erori (caz în care nu se mai poate determina direct locația acestora).

Sindromurile verticale v_p și w_q sunt $(n_2 + n_3)$ vectori de lungime n_1 fiecare, a căror definiție este:

$v_p[\xi] = \left(\sum_{i_3=0}^{n_3-1} a_{\xi, p, i_3} \right) \text{ modulo } 2$	$w_q[\xi] = \left(\sum_{i_2=0}^{n_2-1} a_{\xi, i_2, q} \right) \text{ modulo } 2$
--	--

Intervalele sunt seturi de coloane adiacente în reprezentarea 2D, pentru care toate sindromurile verticale v_p și w_q nu detectează erori. După cum reiese din *figura 6*, intervalele pot fi fără erori (intervalul verde) sau cu erori (intervalul mov). Deși la decodare nu avem informații despre care intervale sunt fără erori și care sunt cu erori, se poate demonstra că pentru o valoare suficient de mare a n_1 , cel mai mare interval este cu siguranță fără erori. Datorită modului în care este definită trecerea de la un element la altul în diagonalele generalizate, prima și ultima coloană a reprezentării 2D sunt considerate adiacente, astfel încât un interval se poate întinde pe ultimele câteva coloane apoi primele câteva coloane ale reprezentării.

În *figura 6*, pătratele roșii reprezintă biți eronați. Figura a fost introdusă pentru a înlesni înțelegerea noțiunii de „interval fără erori”, care va fi folosită în algoritmul de decodare.

Exemplu de corecție în cazul defecțiunii unui număr de b DASD-uri

Defectarea unui DASD înseamnă pierderea informațiilor dintr-o diagonală generalizată. Codurile tridimensionale sunt capabile să corecteze/refacă informația din până la $b = t + 2$ pachete de erori de lungime $n_2 \cdot n_3$ biți (formula generală = $t + D - 1$) printr-un algoritm mai simplu sau până la $b = t + 3$ pachete de erori de lungime $n_2 \cdot n_3$ biți (formula generală $b = t + 2^{D-1} - 1$) printr-un algoritm mai complicat, în condițiile unei dimensionari corecte:

- Alegerea codului (bloc) $B(n_1, k, t)$ corector de t erori și detector de b erori, capabil să codeze k biți de informație în n_1 biți, cod care să fie folosit de-a lungul dimensiunii (x_1)
- Alegerea lungimilor structurii n_1, n_2 și n_3 de-a lungul dimensiunilor (x_1), (x_2) și respectiv (x_3) conform constrângerilor:
 - n_2 și n_3 să fie prime între ele: $\text{cmmdc}(n_2, n_3) = 1$
 - $n_1 \gg n_2 \cdot n_3$
- Alegerea regulii de parcurgere a biților în vederea generării diagonalelor generalizate cu ajutorul pasului s , conform constrângerilor
 - $s = (\alpha_1, \alpha_2, \alpha_3)$
 - $\text{cmmdc}(\alpha_i, n_i) = 1, \forall i$

Algoritmul de decodare pentru $b = t + 2$

- Se calculează sindromurile orizontale h_{pq} și se corectează toate liniile cu maxim t erori. Se determină linia (p, q) care este cea mai de sus linie care încă mai are erori conform h_{pq}
- Se calculează sindromurile verticale v_p și w_q
- Se găsește cel mai mare interval pentru care sindromurile verticale nu detectează erori. Acest interval se termină la coloana ξ . Se incrementează coordonata ξ .

- Cât timp mai sunt erori (cel puțin un sindrom orizontal sau vertical indică erori)
 - Dacă sindromurile verticale nu detectează erori la coordonata ξ
 - Se incrementează coordonata ξ
 - Altfel, dacă sindromurile verticale detectează un număr impar de erori la coordonata ξ , există o singură eroare care poate fi identificată prin sindromurile verticale v_p și w_q :
 - Se corectează eroarea $a_{\xi,m,n}$
 - Se calculează sindromul orizontal $h_{m,n}$ corespunzător liniei (m,n) ;
 - Dacă se poate, se corectează celelalte erori de pe linia (m,n)
 - Se incrementează coordonata ξ ;
 - Altfel, dacă sindromurile verticale detectează 2 erori
 - Se modifică bitul $a_{\xi,p,q}$
 - Se calculează sindromul orizontal h_{pq}
 - Se corectează celelalte erori de pe linia (p,q)
 - Se corectează cea de-a doua eroare $a_{\xi,m,n}$
 - Se calculează sindromul orizontal $h_{m,n}$ corespunzător liniei (m,n) ;
 - Dacă se poate, se corectează celelalte erori de pe linia (m,n)
 - Se incrementează coordonata ξ ;

Explicații suplimentare despre algoritm:

Sindromul vertical pentru 0 erori este:
$$\begin{cases} v_p[\xi] = 0, \forall p < n_2 \\ w_q[\xi] = 0, \forall q < n_3 \end{cases}$$

Sindromul vertical pentru 1 eroare este:
$$\begin{cases} \sum_{p=0}^{n_2} v_p[\xi] = 1 \\ \sum_{q=0}^{n_3} w_q[\xi] = 1 \end{cases}$$

Sindromul vertical pentru 2 erori este:
$$\begin{cases} \sum_{p=0}^{n_2} v_p[\xi] \in \{0,2\} \\ \sum_{q=0}^{n_3} w_q[\xi] \in \{0,2\} \\ \sum_{p=0}^{n_2} v_p[\xi] + \sum_{q=0}^{n_3} w_q[\xi] > 0 \end{cases}$$

Exemplu grafic – Figura 7

Fie $C(n_1 \times 2 \times 3, s=(1,1,1), v, t=3)$ un cod tridimensional capabil să corecteze $b = t + 2$ pachete de erori. Presupunem ca DASD-urile cu numerele 3, 6, 8, 10 și 13 s-au defectat. După introducerea unor DASD-urilor noi (blank, cu valori aleatoare), o parte din biții codului vor fi eronați. În *figura 7a* este reprezentarea 2D a codului, cu biții eronați colorați în funcție de DASD-ul de care aparțin.

12

Etapele de corectare a biților eronați vor fi:

Se calculează sindromurile horizontale h_{pq}

Se corectează liniile $(1,1)$ și $(0,1)$ care au maxim $t=3$ erori, rezultând starea din *figura 7b*.

Se găsește cel mai mare interval pentru care sindromurile verticale nu detectează erori. Acest interval se termină la coloana $\xi=2$. Se incrementează coordonata ξ .

Pentru $\xi=3$, sindromurile verticale indică o singură eroare, $a_{3,0,0}$, care poate fi corectată. Se calculează sindromul orizontal $h_{0,0}$ și se corectează celelalte 3 erori din linia $(0,0)$, rezultând starea din *figura 7c*.

Se incrementează coordonata până la $\xi=8$. Sindromurile verticale indică 2 erori. Cea mai de sus linie cu erori este acum linia $(0,2)$. Se corectează $a_{8,0,2}$, se calculează sindromul orizontal $h_{0,2}$ și se corectează celelalte 3 erori din linia $(0,2)$. Se recalculează sindromurile verticale pentru coloana 8 și se corectează $a_{8,1,2}$. Se calculează sindromul orizontal $h_{1,2}$ și se corectează celelalte 3 erori din linia $(1,2)$, rezultând starea din *figura 7d*.

Se incrementează coordonata până la $\xi=9$. Se corectează $a_{9,1,0}$ indicată de sindromurile verticale, se calculează apoi sindromul orizontal $h_{1,0}$ și se corectează celelalte 3 erori din linia $(1,0)$.

Algoritmul de decodare pentru $b = t + 3$

- Se calculează sindromurile horizontale h_{pq} și se corectează toate liniile cu maxim t erori. Se determină linia (p,q) care este cea mai de sus linie care încă mai are erori conform h_{pq}
- Se calculează sindromurile verticale v_p și w_q
- Se găsește cel mai mare interval pentru care sindromurile verticale nu detectează erori. Acest interval se termină la coloana ξ . Se incrementează coordonata ξ și $\psi = \xi$.
- Cât timp mai sunt erori (cel puțin un sindrom orizontal sau vertical indică erori) { }
 - Dacă sindromurile verticale nu detectează erori la coordonata ξ
 - Se incrementează coordonatele ξ și $\psi = \xi$
 - Altfel, dacă sindromurile verticale detectează un număr impar de erori la coordonata ξ
 - Se testează sindromul h_{pq} după modificarea bitului $a_{\xi,p,q}$

Dacă linia (p,q) conține mai puțin de t erori, se corectează erorile de pe linia (p,q) și se revine la bucla { }

Altfel, la coordonata ξ trebuie să fie o singură eroare, care poate fi identificată din sindromurile verticale. Se corectează $a_{\xi,\sim,\sim}$ și se revine la bucla { }

o Altfel, dacă sindromurile verticale detectează 2 erori

- Se incrementează coordonata $\psi = \psi + 1$ [**]

- Dacă sindromurile verticale detectează un număr impar de erori la coordonata ψ

- Se testează sindromul h_{pq} după modificarea bitului $a_{\xi,p,q}$
- Se testează sindromul h_{pq} după modificarea bitului $a_{\psi,p,q}$
- Se testează sindromul h_{pq} după modificarea bitilor $a_{\xi,p,q}$ și $a_{\psi,p,q}$

În caz că nici unul dintre teste nu reduce numărul de erori din rândul (p,q) la mai puțin de $t+1$ erori (să poată fi corectate cu sindromul orizontal h_{pq}), atunci la coordonata ψ trebuie să fie o singură eroare, care poate fi identificată din sindromurile verticale. Se corectează $a_{\psi,\sim,\sim}$ și se revine la bucla [**]

Dacă însă rândul (p,q) poate fi corectat din sindromul orizontal h_{pq} , se face corecția, se recalculază sindromurile verticale și se revine la bucla [**]

- Dacă sindromurile verticale detectează 0 erori la coordonata ψ

- Se testează sindromul h_{pq} după modificarea bitului $a_{\psi,p,q}$

În caz că testul nu reduce numărul de erori din rândul (p,q) la mai puțin de $t+1$ erori (să poată fi corectate cu sindromul orizontal h_{pq}), atunci la coordonata ψ erau într-adevăr 0 erori (și se revine la bucla [**]).

Dacă însă rândul (p,q) poate fi corectat din sindromul orizontal h_{pq} , se face corecția, se recalculază sindromurile verticale și se revine la bucla [**]

- Dacă sindromurile verticale detectează 4 erori la coordonata ψ

- Se modifică bitul $a_{\xi,p,q}$
- Se recalculază h_{pq} și se corectează linia (p,q)
- Se corectează celelalte 3 erori de la coordonata ψ
- Se revine la bucla [**]

- Dacă sindromurile verticale detectează 2 erori la coordonata ψ

- Se testează sindromul h_{pq} după modificarea bitului $a_{\xi,p,q}$
- Se testează sindromul h_{pq} după modificarea bitului $a_{\psi,p,q}$
- Se testează sindromul h_{pq} după modificarea bitilor $a_{\xi,p,q}$ și $a_{\psi,p,q}$

În caz că nici unul dintre teste nu reduce numărul de erori din rândul (p,q) la mai puțin de $t+1$ erori (să poată fi corectate cu sindromul orizontal h_{pq}), atunci se revine la bucla :

Dacă însă rândul (p,q) poate fi corectat din sindromul orizontal h_{pq} , se face corecția, se recalculeaza sindromurile verticale și se revine la bucla :

Explicații suplimentare despre algoritm:

Sindromul vertical pentru 0 erori este:
$$\begin{cases} v_p[\xi] = 0, \forall p < n_2 \\ w_q[\xi] = 0, \forall q < n_3 \end{cases}$$

Sindromul vertical pentru un număr impar de erori este:
$$\begin{cases} \sum_{p=0}^{n_2} v_p[\xi] \in \{1,3\} \\ \sum_{q=0}^{n_3} w_q[\xi] \in \{1,3\} \end{cases}$$

Sindromul vertical pentru 2 erori este:
$$\begin{cases} \sum_{p=0}^{n_2} v_p[\xi] \in \{0,2\} \\ \sum_{q=0}^{n_3} w_q[\xi] \in \{0,2\} \\ \sum_{p=0}^{n_2} v_p[\xi] + \sum_{q=0}^{n_3} w_q[\xi] > 0 \end{cases}$$

Sindromul vertical pentru 4 erori este:
$$\begin{cases} \sum_{p=0}^{n_2} v_p[\xi] = 4 \text{ sau} \\ \sum_{q=0}^{n_3} w_q[\xi] = 4 \end{cases}$$

Motivația pentru modificarea anumitor biți și testarea sindromurilor orizontale este dată de faptul că 4 erori bine plasate generează sindromuri verticale identice cu 0 erori, respectiv 3 erori bine plasate generează sindromuri verticale identice cu o eroare.

Exemplu grafic – Figura 8

Fie $C(n_1 \times 2 \times 3, s=(1,1,1), v, t=3)$ un cod tridimensional capabil să corecteze $b = t + 2$ pachete de erori. Presupunem că DASD-urile cu numerele 3, 6, 8, 10, 13 și 16 s-au defectat. După introducerea unor DASD-urilor noi (blank, cu valori aleatoare), o parte din biții codului vor fi eronați. În *figura 8a* este reprezentarea 2D a codului, cu biții eronați colorați în funcție de DASD-ul de care aparțin.

Etapele de corectare a biților eronați vor fi:

Se calculează sindromurile orizontale h_{pq}

Se corectează linia $(0,1)$ care are maxim $t=3$ erori, rezultând starea din *figura 8b*.

Se găsește cel mai mare interval pentru care sindromurile verticale nu detectează erori. Acest interval se termină la coloana $\xi=2$. Se incrementează coordonata ξ .

Pentru $\xi=3$, sindromurile verticale indică un număr impar de erori. Se testează modificarea bitului $a_{3,0,0}$, dar sindromul orizontal $h_{0,0}$ indică în continuare mai mult de 3 erori pe linia $(0,0)$. Așadar, pentru $\xi=3$ există o singură eroare, indicată de sindromurile verticale la $a_{3,0,0}$, rezultând starea din *figura 8c*.

Se incrementează coordonata până la $\xi=8$. Sindromurile verticale indică un număr impar de erori. Se testează modificarea bitului $a_{8,0,0}$, sindromul orizontal $h_{0,0}$ indică 3 erori care pot și vor fi corectate. Cea mai de sus linie cu erori este acum linia $(1,1)$, rezultând starea din *figura 8d*.

În continuare pentru coordonata $\xi=8$, sindromurile verticale indică 2 erori. Se incrementează coordonata ψ . Pentru $\psi=9$, sindromurile verticale indică 2 erori.

- Se testează sindromul $h_{1,1}$ după modificarea bitului $a_{\xi,p,q} = a_{8,1,1} \Rightarrow$ linia $(1,1)$ nu poate fi corectată
- Se testează sindromul $h_{1,1}$ după modificarea bitului $a_{\psi,p,q} = a_{9,1,1} \Rightarrow$ linia $(1,1)$ poate și va fi corectată

Cea mai de sus linie cu erori este acum linia $(0,2)$, ca în *figura 8e*.

În continuare pentru coordonata $\xi=8$, sindromurile verticale indică 2 erori. Se incrementează coordonata ψ . Pentru $\psi=9$, sindromurile verticale indică un număr impar de erori.

- Se testează sindromul $h_{0,2}$ după modificarea bitului $a_{\xi,p,q} = a_{8,0,2} \Rightarrow$ linia $(0,2)$ nu poate fi corectată cu ajutorul sindromului orizontal
- Se testează sindromul $h_{0,2}$ după modificarea bitului $a_{\psi,p,q} = a_{9,0,2} \Rightarrow$ linia $(0,2)$ nu poate fi corectată cu ajutorul sindromului orizontal
- Se testează sindromul $h_{0,2}$ după modificarea bitilor $a_{8,0,2}$ și $a_{9,0,2} \Rightarrow$ linia $(0,2)$ nu poate fi corectată cu ajutorul sindromului orizontal

Așadar la coordonata $\psi=9$ trebuie să fie o singură eroare, indicată de sindromurile verticale ca $a_{9,1,0}$, rezultând starea din *figura 8f*.

Se verifică apoi sindromul orizontal $h_{1,0}$, dar linia $(1,0)$ nu poate fi corectată.

În continuare la coordonata $\xi=8$ sindromurile verticale indică 2 erori. Se incrementează coordonata ψ . Pentru $\psi=10$, sindromurile verticale indică un număr impar de erori.

- Se testează sindromul $h_{0,2}$ după modificarea bitului $a_{\xi,p,q} = a_{8,0,2} \Rightarrow$ linia $(0,2)$ nu poate fi corectată cu ajutorul sindromului orizontal
- Se testează sindromul $h_{0,2}$ după modificarea bitului $a_{\psi,p,q} = a_{10,0,2} \Rightarrow$ linia $(0,2)$ nu poate fi corectată cu ajutorul sindromului orizontal
- Se testează sindromul $h_{0,2}$ după modificarea bitilor $a_{8,0,2}$ și $a_{10,0,2} \Rightarrow$ linia $(0,2)$ poate și va fi corectată cu ajutorul sindromului orizontal

Cea mai de sus linie cu erori este acum linia $(1,0)$, ca în *figura 8g*.

În continuare pentru coordonata $\xi=8$, sindromurile verticale indică un număr impar de erori. Se testează modificarea bitului $a_{8,1,0}$, dar sindromul orizontal $h_{1,0}$ indică în continuare mai mult de 3 erori pe linia $(1,0)$. Așadar, pentru $\xi=8$ există o singură eroare, indicată de sindromurile verticale la $a_{8,1,2}$, ca în *figura 8h*.

Se verifică apoi sindromul orizontal $h_{1,2}$, dar linia $(1,2)$ nu poate fi corectată.

Se incrementează coordonata până la $\xi=11$. Sindromurile verticale indică 2 erori. Se incrementează coordonata ψ . Pentru $\psi=12$, sindromurile verticale indică 0 erori.

- Se testează sindromul $h_{1,0}$ după modificarea bitului $a_{\psi,p,q} = a_{12,1,0} \Rightarrow$ linia $(1,0)$ nu poate fi corectată cu ajutorul sindromului orizontal

Așadar, coordonata $\psi=12$ conține într-adevăr 0 erori. Se incrementează ψ .

Pentru $\psi=13$, sindromurile verticale indică 2 erori.

- Se testează sindromul $h_{1,0}$ după modificarea bitului $a_{\xi,p,q} = a_{11,1,0} \Rightarrow$ linia $(1,0)$ nu poate fi corectată cu ajutorul sindromului orizontal
- Se testează sindromul $h_{1,0}$ după modificarea bitului $a_{\psi,p,q} = a_{13,1,0} \Rightarrow$ linia $(1,0)$ nu poate fi corectată cu ajutorul sindromului orizontal
- Se testează sindromul $h_{1,0}$ după modificarea bitilor $a_{11,1,0}$ și $a_{13,1,0} \Rightarrow$ linia $(1,0)$ poate și va fi corectată cu ajutorul sindromului orizontal

Cea mai de sus linie cu erori este acum linia $(1,2)$, ca în *figura 8i*.

În continuare pentru coordonata $\xi=11$, sindromurile verticale indică un număr impar de erori. Se testează modificarea bitului $a_{11,1,2}$ și se corectează restul erorilor conform sindromului $h_{1,2}$.

Revendicari

1. Metoda de codare decodare a codurilor tridimensionale peste câmpuri Galois $GF(2)$, implementată cu ajutorul calculatorului, pentru stocarea datelor în module DASD conform prezentei invenții revendicate constă în următorii pași:
 - Alegerea codului (bloc) $B(n_1, k, t)$ corector de t erori și detector de b erori, capabil să codeze k biți de informație în n_1 biți, cod care să fie folosit de-a lungul dimensiunii (x_1)
 - Alegerea lungimilor structurii n_1, n_2 și n_3 de-a lungul dimensiunilor $(x_1), (x_2)$ și respectiv (x_3) conform constrângerilor:
 - n_2 și n_3 să fie prime între ele: $\text{cmmdc}(n_2, n_3) = 1$
 - $n_1 \gg n_2 \cdot n_3$
 - Alegerea regulii de generare a diagonalelor generalizate cu ajutorul pasului s conform constrângerilor
 - $s = (\alpha_1, \alpha_2, \alpha_3)$
 - $\text{cmmdc}(\alpha_i, n_i) = 1, \forall i$
 - Codarea a $k \cdot (n_2 - 1) \cdot (n_3 - 1)$ biți de informație într-o structură tridimensională $n_1 \times n_2 \times n_3$
 - Stocarea datelor în module DASD, unde fiecare modul DASD conține câte o diagonală generalizată a structurii tridimensionale, iar defecțiunea sau ștergerea unui DASD corespunde cu introducerea unui pachet de erori de lungime $n_2 \cdot n_3$.
 - Decodarea cu corectia a până la $t+3$ pachete de erori de lungime $n_2 \cdot n_3$ fiecare

2. Metoda de codare decodare a codurilor D-dimensionale ($D > 3$) peste câmpuri Galois $GF(2)$, implementată cu ajutorul calculatorului, pentru stocarea datelor în module DASD conform prezentei invenții revendicate constă în următorii pași:
 - Alegerea codului (bloc) $B(n_1, k, t)$ corector de t erori și detector de b erori, capabil să codeze k biți de informație în n_1 biți, cod care să fie folosit de-a lungul dimensiunii (x_1)
 - Alegerea lungimilor structurii n_i de-a lungul dimensiunilor (x_i) , conform constrângerilor:
 - n_i și n_j să fie prime între ele: $\text{cmmdc}(n_i, n_j) = 1, \forall i \neq j, i > 1, j > 1$
 - $n_1 \gg \prod_{i=2}^D n_i$
 - Alegerea regulii de generare a diagonalelor generalizate cu ajutorul pasului s conform constrângerilor
 - $s = (\alpha_1, \alpha_2, \dots, \alpha_D)$
 - $\text{cmmdc}(\alpha_i, n_i) = 1, \forall i$
 - Codarea a $k \cdot \prod_{i=2}^D (n_i - 1)$ biți de informație într-o structură D -dimensională $n_1 \times n_2 \times \dots \times n_D$

46

- Stocarea datelor în module DASD, unde fiecare modul DASD conține câte o diagonală generalizată a structurii D -dimensionale, iar defecțiunea sau ștergerea unui DASD corespunde cu introducerea unui pachet de erori de lungime $\prod_{i=2}^D n_i$.
- Decodarea cu corecția a până la $t + 2^{D-1} - 1$ pachete de erori de lungime $\prod_{i=2}^D n_i$ fiecare

Lista de figuri

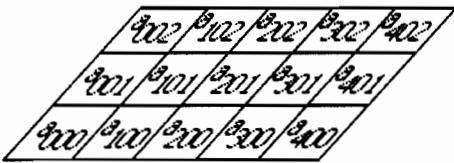
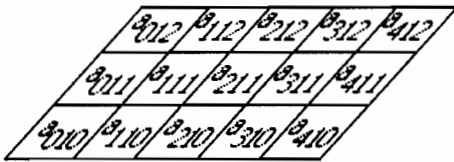


Figura 1.

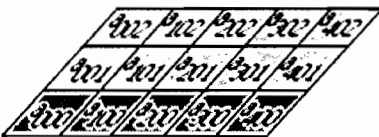


Figura 2.

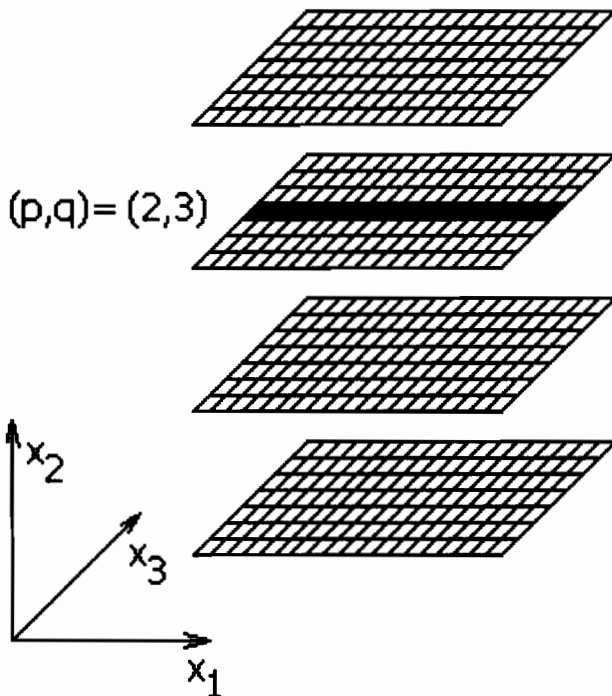


Figura 3.

44

<i>Linia (0,0)</i>						
<i>Linia (1,1)</i>						
<i>Linia (0,2)</i>	$a_{0,0,2}$	$a_{1,0,2}$	$a_{2,0,2}$	$a_{3,0,2}$	$a_{4,0,2}$...
<i>Linia (1,0)</i>	$a_{0,1,0}$	$a_{1,1,0}$	$a_{2,1,0}$	$a_{3,1,0}$	$a_{4,1,0}$...
<i>Linia (0,1)</i>	$a_{0,0,1}$	$a_{1,0,1}$	$a_{2,0,1}$	$a_{3,0,1}$	$a_{4,0,1}$...
<i>Linia (1,2)</i>						

Figura 4.

<i>Linia (0,0)</i>					$a_{3,0,0}$	$a_{4,0,0}$	$a_{5,0,0}$	$a_{6,0,0}$	
<i>Linia (1,1)</i>					$a_{4,1,1}$	$a_{5,1,1}$	$a_{6,1,1}$...	
<i>Linia (0,2)</i>	$a_{0,0,2}$						$a_{5,0,2}$	$a_{6,0,2}$...
<i>Linia (1,0)</i>	$a_{0,1,0}$	$a_{1,1,0}$						$a_{6,1,0}$...
<i>Linia (0,1)</i>	$a_{0,0,1}$	$a_{1,0,1}$	$a_{2,0,1}$...
<i>Linia (1,2)</i>	$a_{0,1,2}$	$a_{1,1,2}$	$a_{2,1,2}$	$a_{3,1,2}$					

Figura 5.

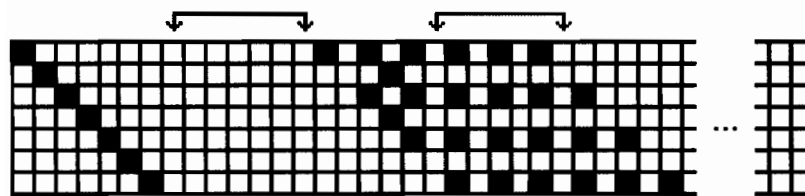


Figura 6.

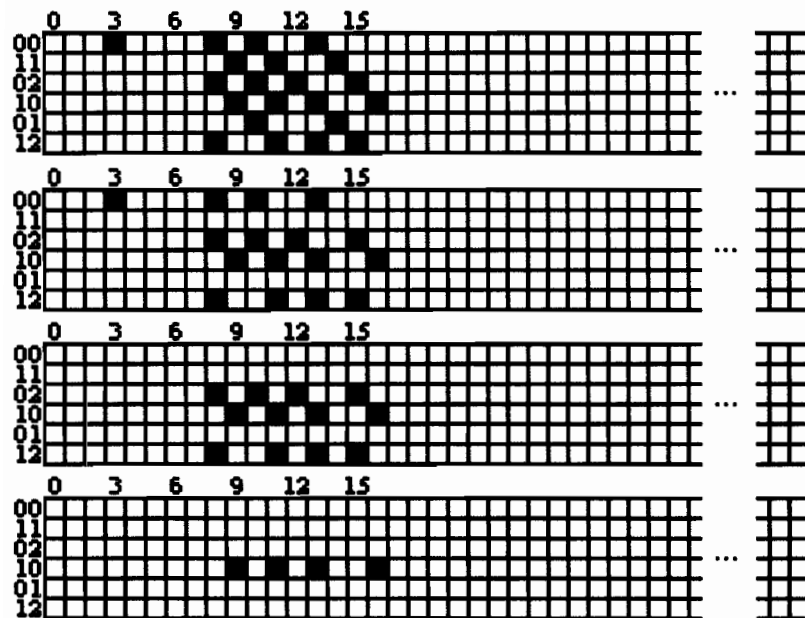


Figura 7.

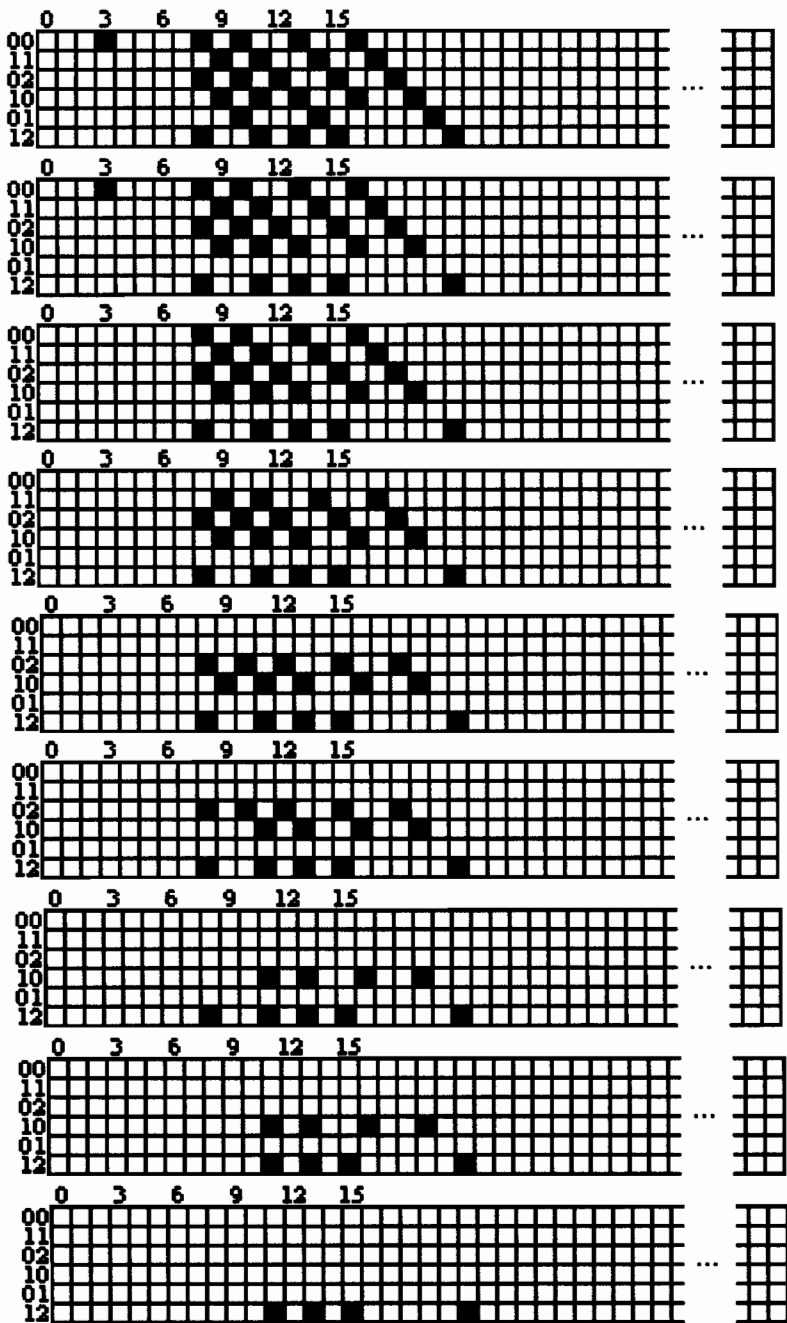


Figura 8.