PROIECTAREA ŞI IMPLEMENTAREA CODECULUI CODULUI MATROID

D. Bodean, G. Bodean
Universitatea Tehnică a Moldovei

INTRODUCERE

În [1] au fost prezentate aspectele teoretice ale codurilor matroide – o clasă nouă de coduri nonbinare corectoare de erori. *Obiectivul* acestei lucrării sunt procedurile de codare şi decodare asincronă (instantanee) a codului matroid. *Scopul lucrării* constă în proiectarea coderului-decoderului (codecului) codului matroid cu parametrii (n,k,t)= (8,4,2) asupra câmpului Galois **GF** de ordinul 2^4, unde n – lungimea cuvintelor de cod, k – numărul simbolurilor în combinaţia de intrare, t – capacitatea corectoare de erori. *Sarcinile lucrării*: 1) de generat matricea matroidului respectiv; 2) de elaborat algoritmul de decodare a codului matroid; 4) de elaborat entitatea de generare a multiplicatorului la constantă asupra câmpului Galois; 5) de elaborat proiectele entităţilor coderului şi decoderului codului matroid; 6) de implementat proiectul codecului elaborat; 7) de estimat câştigul de la codarea matroidă în canalele binare cu modulaţia în fază.

1. CODAREA

Pentru generarea codului matroid (sau M-codul) se foloseşte matricea care urmează structura matroidului uniform [2]. Matroidul uniform $U_{k,n}$ poate fi prezentat printr-o matrice de forma $\mathbf{G}_{k\times n}=[g_{ij}]_{k\times n}$, unde $g_{ij}\in$ **GF**(2^m). În această interpretare vectorii M-codului vor fi generaţi prin aplicarea operaţiei matriceale:

$$\mathbf{v}=\langle v_1,...,v_n\rangle=\mathbf{x}\cdot\mathbf{G}, \qquad (1)$$

unde $\mathbf{x}=\langle x_1,...,x_n\rangle$ este vectorul original, $x\in$ **GF**(2^m).

Fie M-codul cu parametrii (n, k, t)= (8, 4, 2) asupra **GF**(2^4) cu polinomul $p(x)= x^4+x+1$. Asupra **GF**(2^4)={0,1,...,15} cu ajutorul instrumentarului, prezentat în [3], a fost generată matricea matroidului de structura standard:

$$\mathbf{G}=\begin{bmatrix}1&0&0&0&1&2&4&8\\0&1&0&0&1&4&3&12\\0&0&1&0&1&8&12&10\\0&0&0&1&1&3&5&15\end{bmatrix}.$$

Un exemplu de codare: pentru vectorul de intrare $\mathbf{x}$ = <1, 2, 3, 4> se obţine următorul vector de ieşire: $\mathbf{v}=\langle v_1, v_2,...,v_8\rangle=\mathbf{x}\cdot\mathbf{G}=$ <1,2,3,4,4,1,2,7>.

Deci, M-coderul trebuie să implementeze transformarea matriceală (1), care reprezintă un sistem de ecuaţii liniare:

$$\begin{cases} x_1 = v_1,\\ x_2 = v_2,\\ x_3 = v_3,\\ x_4 = v_4,\\ x_1+x_2+x_3+x_4 = v_5,\\ 2x_1+4x_2+8x_3+3x_4 = v_6,\\ 4x_1+3x_2+12x_3+5x_4 = v_6,\\ 8x_1+12x_2+10x_3+15x_4 = v_6, \end{cases} \qquad (2)$$

unde suma şi înmulţirea se execută modulo $p(x)$.

Modul de implementare a RTL-entităţii coderului este determinat de instrumentarul de proiectare asistată de calculator, în special, cel de proiectare pe circuite programabile. Pentru a fi independenţi de platforma utilizată vom codifica structura coderului în limbajul VHDL. Entitatea coderului M-codului (8,4,2) este următoarea:

```
ENTITY Coder IS
  PORT
    (   InWord      : IN   bit_vector( 0 to 15);
        OutWord     : OUT  bit_vector( 0 to 31)
    );
END Coder;

ARCHITECTURE aCoder OF Coder IS
BEGIN
  c0: MultMatrix
        GENERIC MAP(
                    Matr => CoderMatr,
                    Rows => WordSize,
                    Columns => CodeSize,
                    SymSize => Degree
              )
        PORT MAP(   Word => InWord,
                    CodeWord => OutWord);
END aCoder.
```

Pentru generarea coderului se foloseşte o entitate (componentă) universală, *MultMatrix*, care va fi utilizată şi pentru generarea structurii decoderului M-codului.

2. DECODAREA

1.1. Detectarea şi localizarea erorilor

Decoderul M-codului trebuie să efectueze următoarele operaţii: *detectarea* erorii, *localizarea* erorii, *corectarea* erorii şi "*restabilirea*" mesajului (vectorului original) transmis. În lucrarea [1] a fost prezentat un algoritm de decodare a codului matroid bazat pe *principiul logicei majoritare* (de prag). Decodarea se execută asincron, într-un singur tact şi permite detectarea, localizarea şi corectarea integrată a erorilor *t*-uple în combinaţia de cod recepţionată. M-codul analizat are capacitatea (abilitatea) corectoare *t* egală cu 2. Într-un cuvânt de cod de lungimea n = 8 pot fi 8 erori singulare şi $C_8^2 = 28$ erori duble.

Detectarea erorii se efectuează cu ajutorul matricei de control. Conform definiţiei [4] matricea de control **H** trebuie să satisfacă egalitatea:

$$\mathbf{G} \cdot \mathbf{H}_{(\cdot)}^T = 0, \tag{3}$$

iar sindromul **S** este calculat din relaţia:

$$\mathbf{H}_{(\cdot)} \cdot \mathbf{v}^T = \mathbf{S}^{(\cdot)}. \tag{4}$$

Dacă **S**= 0 se presupune că vectorul analizat nu conţine erori admisibile (detectabile sau corectabile).

În relaţiile (3) şi (4) una din dimensiunile matricei **H** nu-i cunoscută. Conform *decodării logicei majoritare* capacitatea detectoare a matricei de control nu poate depăşi redondanţa codului, adică valoarea determinată de diferenţa $n– k$. În cazul analizat avem $n– k$ =8- 4= 4. Deci, dimensiunile matricei de control vor fi 8×4, iar structura ei va fi de tip standard şi anume:

$$\mathbf{H}_{(1)} = \begin{bmatrix} \mathbf{I}_{4\times4} \\ \mathbf{Q}_{4\times4} \end{bmatrix} \tag{5}$$

unde **I** este matricea unitate, iar **Q** se va determina din relaţia (3).

Însă având structura (3) sindromul **S** va conţine numai $n – k$ = 4 elemente, ceea ce nu acoperă toată lungimea n a vectorului **v**. De aceea trebuie de elaborat un algoritm optimal de acoperire a ansamblului de erori duble în cuvântul de cod.

În [1] a fost propusă şi analizată, pentru un caz particular, o metodă de "testare" prin deplasarea secvenţială a matricei-unitate **I** în spaţiul matricei de control (5). Vom extinde metoda pentru M-codul analizat. Conform metodei logicei de prag, aplicată pentru M-decodare şi exprimată de relaţia (4), avem următorul criteriu.

***Regula* 1**: pentru a corecta (până la) *t* erori este necesar ca sindromul **S** să conţină *nu mai mult* de *t* simboluri semnificative (nonzero).

Pe de altă parte sindromul **S** va localiza erorile în poziţiile vectorului recepţionat **v**, indicii componentelor căruia coincid cu indicii liniei matricei-unitate din (5). Astfel, matricea de control cu structura (5) va acoperi numai (şi numai) erorile din prima jumătate a vectorului **v**, adică erorile în componentele v_1,..., v_4 . Numărul erorilor duble detectate este egal cu $C_4^2 = 6$.

Dacă "deplasăm", efectuăm o permutare a matricelor în (5), se obţine matricea de control (a doua) cu structura:

$$\mathbf{H}_{(2)} = \begin{bmatrix} \mathbf{Q}' \\ \mathbf{I} \end{bmatrix}. \tag{6}$$

Matricea (6) va "acoperi" erorile din a două jumătate a vectorului **v**, adică erorile asupra simbolurilor v_5 ,..., v_8 . Numărul erorilor duble va fi egal cu $C_4^2 = 6$.

În total avem: $C_4^2 + C_4^2 = 12$ erori duble detectate. Celelalte erori duble ramase, adică $C_8^4 - 2C_4^2 = 16$ vor fi detectate cu încă 4 matrice de control în care matricea-unitate este *distribuită* (dispersată). Aceste matrice de control au următoarele structuri:

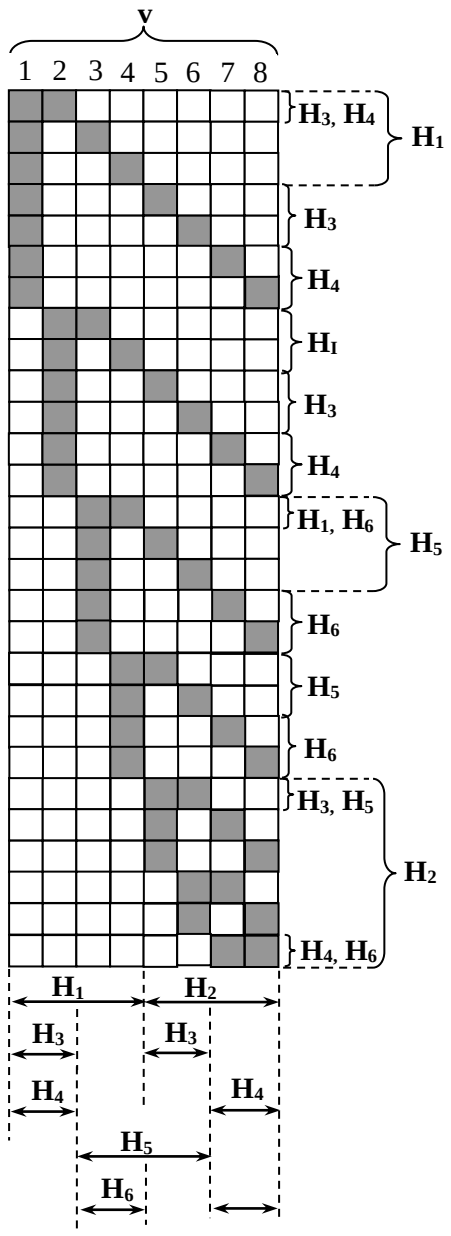
$$\mathbf{H}_{(3)} = \begin{bmatrix} 1 & 0 & q_{11} & q_{12} & 0 & 0 & q_{13} & q_{14} \\ 0 & 1 & q_{21} & q_{22} & 0 & 0 & q_{23} & q_{24} \\ 0 & 0 & q_{31} & q_{32} & 1 & 0 & q_{33} & q_{34} \\ 0 & 0 & q_{41} & q_{42} & 0 & 1 & q_{43} & q_{44} \end{bmatrix},$$

$$\mathbf{H}_{(4)} = \begin{bmatrix} 1 & 0 & q_{11} & q_{12} & q_{13} & q_{14} & 0 & 0 \\ 0 & 1 & q_{21} & q_{22} & q_{23} & q_{24} & 0 & 0 \\ 0 & 0 & q_{31} & q_{32} & q_{33} & q_{34} & 1 & 0 \\ 0 & 0 & q_{41} & q_{42} & q_{43} & q_{44} & 0 & 1 \end{bmatrix},$$

$$\mathbf{H}_{(5)} = \begin{bmatrix} q_{11} & q_{12} & 1 & 0 & 0 & 0 & q_{13} & q_{14} \\ q_{21} & q_{22} & 0 & 1 & 0 & 0 & q_{23} & q_{24} \\ q_{31} & q_{32} & 0 & 0 & 1 & 0 & q_{33} & q_{34} \\ q_{41} & q_{42} & 0 & 0 & 0 & 1 & q_{43} & q_{44} \end{bmatrix},$$

$$\mathbf{H}_{(6)} = \begin{bmatrix} q_{11} & q_{12} & 1 & 0 & q_{13} & q_{14} & 0 & 0 \\ q_{21} & q_{22} & 0 & 1 & q_{23} & q_{24} & 0 & 0 \\ q_{31} & q_{32} & 0 & 0 & q_{33} & q_{34} & 1 & 0 \\ q_{41} & q_{42} & 0 & 0 & q_{43} & q_{44} & 0 & 1 \end{bmatrix}.$$

În figura 1 sunt prezentate convenţional combinaţiile de erori duble într-un cuvânt de cod şi graficul acoperii acestor erori de matricele de control $\mathbf{H}_{(1)}$, ..., $\mathbf{H}_{(6)}$. Precum se observă din grafic unele erori sunt detectate de câteva matrice de control.



□ - simbol corect; ■ - poziţie eronată.

Fig. 1. Diagrama acoperirii erorilor duble de matricele de control $\mathbf{H}_{(\cdot)}$

În acest mod, pentru *detectarea* şi *localizarea* erorii duble, (ca urmare, şi a erori singulare) se vor execută următorii paşi:

- la început, pentru o matrice arbitrară de control se va calcula sindromul **S** conform (4);
- dacă $\mathbf{S} \neq 0$ şi numărul de simboluri nonzero $\leq t$, atunci secvenţial sunt calculate sindroamele $\mathbf{S}^{(i)}$, unde $\mathbf{S}^{(i)} = \mathbf{H}_{(i)} \cdot \mathbf{v}^T$, $i \in \{1, \ldots, 6\}$, până când nu va fi satisfăcută Regula 1 (ori $i \notin \{1,...,6\}$).

După ce eroarea a fost localizata se trece la *corectarea* ei. Corectarea erorii *t*-uple se realizează în mod standard, prin adunarea modulo 2 (XOR), bit-cu-bit a componentelor sindromului-corector, $\mathbf{S}^{(i)}$ cu respectivele componente ale vectorului recepţionat $\mathbf{v}'$. Astfel, va fi obţinut vectorul transmis **v**, componentele căruia v_1 ,..., v_8 sunt coeficienţii liberi ai sistemului de ecuaţii liniare (2).

1.2. Restabilirea mesajului

Prin *restabilirea* mesajului sursei se înţelege rezolvarea unui subsistem de *k* ecuaţii ai sistemului (2) faţă de componentele necunoscute $x_1,...,x_4$. Deoarece este indiferent care din celea $C_8^4 = 70$ subsisteme de ecuaţii liniare va fi ales, pentru simplitatea calculelor selectăm primele k=4 ecuaţii din setul de relaţii (2). Atunci vectorul original se va restabili prin atribuirea $x_1=v_1$, $x_2=v_2$, $x_3= v_3$, şi $x_4= v_4$.

Deci, pentru codului matroid (8,4,2) se va aplica schema de decodare, prezentată în figura 2. Prin μ^{-1} convenţional este marcată operaţia de restabilire a mesajului original.

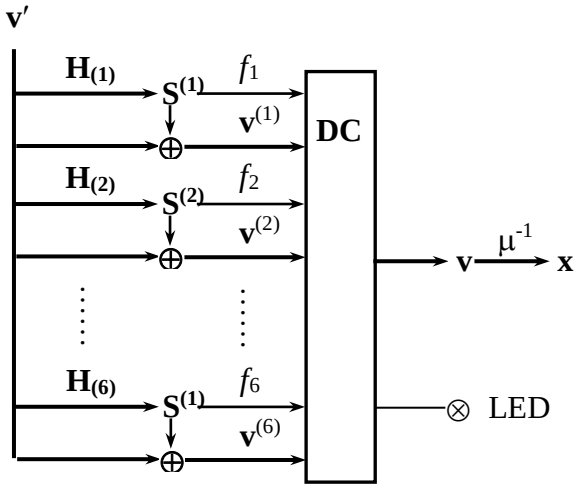


Fig. 2. Bloc-diagrama decodării M-codului (8,4,2)

Funcţia $f_i(\mathbf{S}^{(i)})$ ia valoarea logică *True* (sau *False*) dacă este satisfăcută (sau Nu) *Regula* 1. Valorile $f_i(\mathbf{S}^{(i)})$ sunt intrările de control ale decoderului DC şi comutează una din intrările $\mathbf{v}^{(i)}$ la ieşirea **v**, $i = \overline{1,6}$. Precum se observă din diagrama figurii 2, la ieşirea **v** a decoderului DC pot fi comutate concomitent (până la) 3 intrări. Aici nu apare nici o contradicţie. Valorile (cuvintele de cod) de la intrările comutate sunt identice.

Dacă eroarea este inadmisibilă (necorectabilă), atunci toate funcţiile $f_i(\mathbf{S}^{(i)})$ iau valoarea *False*, ceea ce va rezultă starea "Z", starea flotantă (de înaltă impedanţă) la ieşirea decoderului DC. Totodată, această stare va fi semnalizată de indicatorul LED. Trebuie de evidenţiat că toate operaţiile de decodare, prezentate în figura 2, se execută asincron. Timpii de executare a operaţiilor vor depinde de timpii de propagare a semnalelor în proiectul implementat.

3. IMPLEMENTAREA

3.1. Calculul matricelor de control

Pentru generarea matricelor de control $\mathbf{H}^{(i)}$, necesare pentru implementarea M-decodării, vom aplica relaţia (3). Pentru codul matroid (8, 4, 2) matricea **G** este de dimensiunea $k \times n = 4 \times 8$, iar matricea de control **H** este de dimensiunea $(n - k) \times n = 4 \times 8$. Vom prezenta prin exemplu modul de calcul al unei matrice de control.

Fie matricea de control $\mathbf{H}_{(1)}$ cu structura (5); avem: $\mathbf{G} \cdot \mathbf{H}_{(1)}^{T} = 0$, de unde rezultă 4 sisteme de ecuaţii liniare:

$$\begin{cases} q_{11} + 2q_{21} + 4q_{31} + 8q_{41} = 1, \\ q_{11} + 4q_{21} + 3q_{31} + 12q_{41} = 0, \\ q_{11} + 8q_{21} + 12q_{31} + 10q_{41} = 0, \\ q_{11} + 3q_{21} + 5q_{31} + 15q_{41} = 0. \end{cases}$$

$$\begin{cases} q_{12} + 2q_{22} + 4q_{32} + 8q_{42} = 0, \\ q_{12} + 4q_{22} + 3q_{32} + 12q_{42} = 1, \\ q_{12} + 8q_{22} + 12q_{32} + 10q_{42} = 0, \\ q_{12} + 3q_{22} + 5q_{32} + 15q_{42} = 0. \end{cases}$$

$$\begin{cases} q_{13} + 2q_{23} + 4q_{33} + 8q_{43} = 0, \\ q_{13} + 4q_{23} + 3q_{33} + 12q_{43} = 0, \\ q_{13} + 8q_{23} + 12q_{33} + 10q_{43} = 1, \\ q_{13} + 3q_{23} + 5q_{33} + 15q_{43} = 0. \end{cases}$$

$$\begin{cases} q_{14} + 2q_{24} + 4q_{34} + 8q_{44} = 0, \\ q_{14} + 4q_{24} + 3q_{34} + 12q_{44} = 0, \\ q_{14} + 8q_{24} + 12q_{34} + 10q_{44} = 0, \\ q_{14} + 3q_{24} + 5q_{34} + 15q_{44} = 1. \end{cases}$$

care se rezolvă, de exemplu, prin metoda Gauss. Luând în consideraţie că operaţiile aditive şi multiplicative se efectuează **modulo** $x^4 + x + 1$, în rezultat se obţine următoarea matrice de control:

$$\mathbf{H}_{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 7 & 2 & 13 & 2 \\ 0 & 1 & 0 & 0 & 4 & 6 & 5 & 10 \\ 0 & 0 & 1 & 0 & 1 & 10 & 2 & 5 \\ 0 & 0 & 0 & 1 & 3 & 14 & 10 & 13 \end{bmatrix}$$

În acelaşi mod vor fi calculate şi matricele de control $\mathbf{H}_{(2)},\ldots,\mathbf{H}_{(6)}$.

3.2. Multiplicatorul vector la matrice

Multiplicatorul vector la matrice (sau VM-multiplicatorul) este un dispozitiv de înmulţire modulară a unui vector la matrice. Coderul codului matroid este în esenţă un VM- multiplicator, iar M-decoderul conţine un set prestabilit de VM-multiplicatoare.

Proiectul VM-multiplicatorului la nivel comportamental (*behavioral*) este definit de relaţia (1). Conform (1) alcătuim modelul de structură a VM-multiplicatorului. Modelul de structură va fi descris în limbajul VHDL.

La nivelul cel mai înalt al ierarhiei se va situa modulul, în care vor fi declaraţi parametrii (*genericul*) şi porturile de intrare-ieşire ale entităţii. În genericul entităţii, evident, va fi specificată matricea generatoare (sau de control) **Matr** cu parametrii specifici: numărul de linii **Rows** şi de coloane **Columns**, binaritatea simbolurilor **SymSize**, iar intrarea şi ieşirea vor fi respectiv vectorul, rezultat de la înmulţirea vectorului de intrare la matricea **Matr**. Pentru VM-multiplicatorul coderului vectorul de intrare este secvenţa de simboluri originale (informaţionale) numită **Word**, iar vectorul de ieşire este cuvântul de cod **CodeWord**. Parametrii **Rows** şi **Columns** definesc lungimile acestor vectori.

Deci, VHDL-interfaţa modulului principal se declară în modul următor:

```
ENTITY MultMatrix IS
  GENERIC
    ( Matr: TMatrix:= (others=> 0);
      Rows: natural:= 2;
      Columns: natural:= 4;
      SymSize: natural:= 2
    );
  PORT
    (Word: IN bit_vector(0 to Rows*SymSize-1);
     CodeWord: OUT bit_vector( 0 to Columns*
                                   SymSize-1)
    );
END MultMatrix;
```

Parametrul **Matr** din generic este un tablou (*array*) unidimensional de tipul:

```
type Tmatrix is array (0 to MatrixSize –1) of States,
```

unde **MatrixSize** este dimensiunea în bituri a matricei, iar tipul de date **States** este tipizat în modul următor:

```
type States is range 0 to 2**Degree –1,
```

unde **Degree**= **deg** $p(x)$. Tipul **States** defineşte mulţimea elementelor în forma zecimală ale câmpului Galois.

În entitatea secundară (*architecture*) a modulului principal va fi descris algoritmul de generare a structurii multiplicatorului vector la matrice. Arhitectura entităţii **MultMatrix**, interfaţa căreia a fost declarată anterior, este următoarea:

```
ARCHITECTURE aMultMatrix OF MultMatrix IS
SIGNAL T: bit_vector( 0 TO
            Rows* SymSize *Columns *SymSize- 1);
SIGNAL V: bit_vector( 0 TO
             Rows*SymSize*Columns*SymSize- 1);
BEGIN
  Column: for j in 0 to Columns-1 generate
    Row: for i in 0 to Rows-1 generate
      s1: MultByConst
      generic map( Matr( Rows*j+ i))
      port map(
        Word(SymSize* i to SymSize*( i+1)- 1),
        T( Rows* SymSize*j+SymSize* i to
            Rows* SymSize* j+(SymSize*(i+1)-1)));
      first: if i= 0 generate
        V( Rows* SymSize*j+SymSize*i  to
        Rows* SymSize*j+(SymSize*(i+1)-1))<=
          T( Rows* SymSize*j+SymSize*i
          to Rows*SymSize*j+(SymSize*(i+1)-1));
      end generate;    -- if first
      rest: if i>0 generate
      s2: Xoring
      port map(
       dataA=> V(Rows*SymSize*j+SymSize*( i-1)
                 to Rows* SymSize*j+ SymSize*i-1),
       dataB=> T( Rows* SymSize*j+ SymSize* i
              to Rows* SymSize*j+ SymSize*(i+1)-1),
       result=> V( Rows* SymSize*j+ SymSize* i
             to Rows* SymSize*j+ SymSize*(i+1)-1));
      end generate;    -- if rest
    end generate;      -- for i
    s3:
    CodeWord( SymSize*j to SymSize*(j+1)-1)<=
      V(Rows*SymSize*(j+1)-SymSize to
                          Rows*SymSize*(j+1)-1);
  end generate;        -- for j
END aMultMatrix;
```

În arhitectura **aMultMatrix** a entităţii **MultMatrix** recurent sunt generate componentele VM-multiplicatorului M-codecului analizat. Pentru aceasta suplimentar în antetul declaraţiei tipurilor de date sunt introduse două variabile de tip **SIGNAL**, care specifică conexiunile dintre componentele VM-multiplicatorului.

Generarea structurii se face cu ajutorul instrucţiunilor generatoare **`for generate`** şi **`if generate`**.

Circuitul generat de modulul **MultMatr** are o structură omogenă regulară (fig. 3), de tip "systolic array" [5]. Nodurile diagramei din figura 3 reprezintă multiplicatoarele la constantă, unde constantele sunt coeficienţii matricei **Matr**. Multiplicatoarele la constantă asupra **GF**(2^m) sunt generate de componenta **MultByConst** conform metodei descrisă în [6].

Arhitectura VM- multiplicatorului elaborată este compactă şi universală.

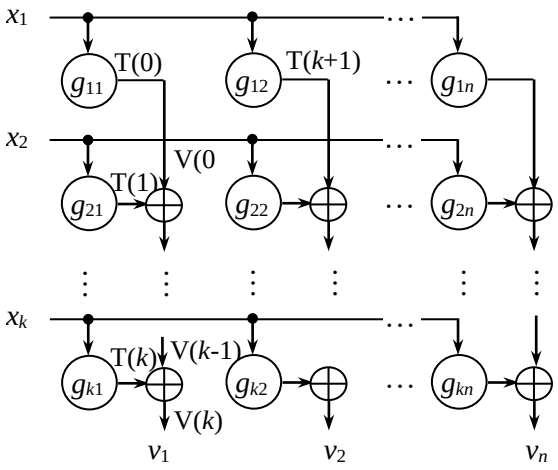


Fig. 3. Bloc-diagrama multiplicatorului vector la matricea de dimensiunea 4×8

3.3. Codecul

În majoritatea sistemelor de comunicaţie se utilizează regimul duplex, când într-un canal datele se transmit în ambele direcţii. De aceea, pentru implementarea proiectului codecului codului matroid (în continuare M-codec) se va utiliza schema prezentată în figura 4.

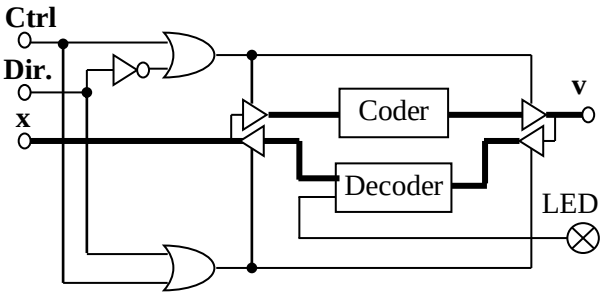


Fig. 4. Bloc-diagrama M-codecului cu magistrală multiplexată

CAD-sistemul de lucru este Quartus 6.0 (firma Altera). Cu ajutorul **Graphic Design Editor** culegem schema prezentată în figura 4. Intrarea **Ctrl** activează dispozitivul, iar intrarea **Dir** specifică direcţia de transmitere a datelor.

Proiectul coderului codului matroid este implementat conform bloc-diagramei, prezentată în figura 3. Proiectul schemotehnic al decoderului este mai complex şi trebuie să realizeze bloc-diagrama prezentată în figura 2. Însă, datorită faptului că implementarea a fost realizată în limbajul VHDL, procesul de proiectare a durat timp de câteva zile. În rezultat descrierea integrală a decoderului conţine 5 entităţi cu volumul total de instrucţiuni aproximativ 160.

Pentru implementarea M-codecului (8,4,2) a fost selectat circuitul programabil EPM570T100C3 din familia MAX II. Parametrii implementării sunt:

Total logic elements:	452/ 570 (79%)
Total pins:	51/ 76 (67%)
Longest tpd :	16,918 ns.

Reieşind din timpul maximal de reţinere (parametrul Longest tpd), rezultă că frecvenţa de funcţionarea a codecului proiectat se preconizează a fi de ≈ 59 MHz.

Proiectul codecului codului matroid (8,4,2) asupra **GF**(2^4) a fost verificat cu ajutorul **WaveformEditor**. În cadrul verificării au fost simulate cât erori admisibile atât şi celea necorectabile, adică cu t>2.

4. CÂŞTIGUL DE LA M-CODARE

Codecul codului matroid proiectat este destinat să funcţioneze într-un sistem de telecomunicaţii. Conform structurii clasice a sistemului de comunicaţie [4], secvenţele generate de codec sunt transmise via modulator în canalul de comunicaţie. Cât modulatorul (demodulatorul), atât şi canalul de comunicaţie pot avea diferite caracteristici. De regulă, când tipul canalului nu-i cunoscut apriori, se alege modelul AWGN (*Additive White Gaussian Noise*). Tipul modulării trebuie, cel puţin, să coreleze cu modul de codare a datelor.

Câştigul energetic de la codare G, conform definiţiei, determină câştigul în stabilitatea la perturbaţii când se aplică codurile corectoare de erori într-un sistem de telecomunicaţie. Câştigul se estimează pentru o valoare fixată a probabilităţii de decodare. Pentru a obţine nivelul indicat al probabilităţii de recepţionare eronată a unui simbol P_b în secvenţa transmisă, la ieşirea demodulatorului receptorului trebuie de asigurat un nivel minimal acceptabil al ratei semnal/ zgomot E_b/ N_0 (*ratio of bit energy to noise power spectral density*). Diferenţa dintre ratele E_b/ N_0 **cu** şi **fără** utilizarea codului corector determină valoarea câştigului energetic G exprimat în decibeli (dB). Estimarea câştigului G, de regulă, se face în raport cu rata biturilor (simbolurilor) eronate.

Pentru determinarea dependenţei dintre caracteristica canalului: rata eronării per bit (**BER**) şi parametrul principal al tractului de transmisie, şi anume, E_b/ N_0, vom utiliza instrumentarul **Bit Error Rate Analysis Tool** (*MatLAB, Release* 14). Graficele dependenţei P_b de E_b/ N_0, generate de aplicaţia **BERTool**, sunt prezentate în figura 5.

Pentru un canal cu rata eronării **BER**= 10^{-6}, caracteristică pentru canalele de comunicaţie

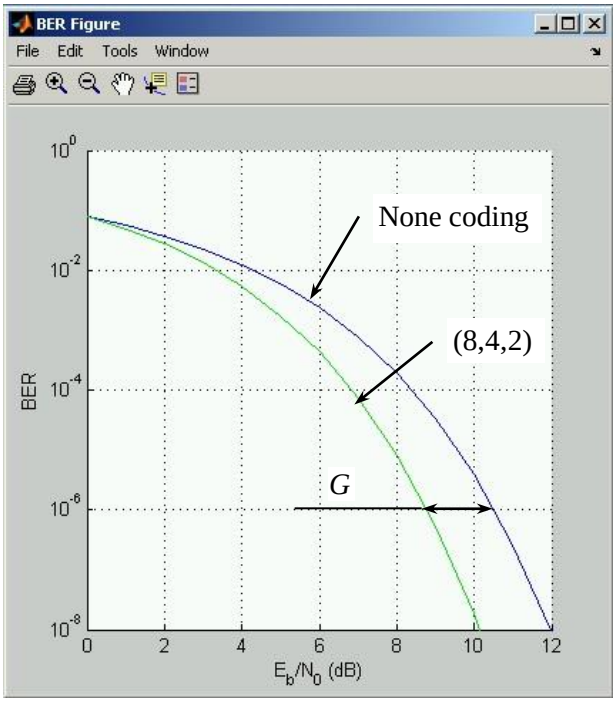


Fig. 5. Graficele dependenţei probabilităţii eronării **BER** versus E_b/ N_0 cu modularea 2PSK asupra canalului AWGN fără codificare (None coding) şi cu M-codarea (8,4,2)

contemporane, câştigul energetic G de la M-codare constituie G≈ 2 dB. Câştigul de la codare duce la micşorarea puterii emiţătorului, dimensiunilor antenei sau la mărirea vitezei de transmisiune şi constituie circa 1..10 mln dolari per 1 dB [7].

Bibliografie

1. Bodean G. *Aspecte teoretice ale codurilor matroide. Meridian Ingineresc Nr.2, Chişinau, pag.35..42, 2005.*

2. Welsh D. *Matroid Theory. London: Academic Press, 1976.*

3. Bodean G., Bodean D., Sorokin G. *Instrumentarul de cercetare a codurilor matroide // Conferinta internationala BIT+ "Tehnologii Informationale", Chişinau, pag. 100, 2006.*

4. Spataru A. *Teoria transmisiunii informaţiei. Coduri şi decizii statistice. Buc.: Ed. Tehnica, 1971.*

5. Ullman J. *Computational aspects of VLSI. - New York, NY: Computer Science Press, 1984.*

6. Bodean G. *Generarea multiplicatoarelor asupra câmpurilor Galois. – Meridian Ingineresc, Nr. 3, pp.14 ..21, 2006.*

7. Zolotarev V., Ovechkin G. *Effektivnye algoritmy pomexoustojchivogo kodirovaniya dlya cifrovyx system svyazi // Elektrosvyaz`, Nr. 3, 2003.*

Aprobat spre publicare: 09.01.2007