Câmpuri Galois

Proiectarea codurilor de dimensiuni finite, deci cu un număr limitat de elemente în alfabet, impune definirea unor câmpuri numerice finite, cu operații interne de adunare și multiplicare, așa-numitele **câmpuri Galois** (*Galois Field* - GF). De exemplu, codurile binare sunt definite în câmpul Galois cu doar două elemente, 0 și 1.

✍ **T1**. Care sunt elementele câmpului GF(8)? Dar cele din GF(16)?

Elementele unui câmp Galois pot fi reprezentate sub trei forme: zecimală, binară și polinomială.

Operația de **adunare** în GF implică sumarea modulo-2 bit cu bit a numerelor exprimate binar.

De exemplu, în GF(8):

$a = 7; b = 5; a = 111_{(2)}; b = 101_{(2)}; a + b = 010_{(2)} = 2.$

✍ **T2.** Scrieți tabela de adunare a elementelor din GF(4). Completați un tabel de forma:

TABEL I

Sum GF(4)	0	1	2	3
0				
1				
2				
3				

❖ În MATLAB este definită funcția *binar(a,m)* (Anexa I) de exprimare pe *m* biți a numărului *a* din câmp. Sumarea elementelor din GF se va face cu funcția *sumgf(a,b,m)* (Anexa I).

```
% Aplicati functia sumgf
% pentru a calcula 143+236 in GF(256)
.
```

✍ **T3.** Completați tabelele de sumare a elementelor din GF(8), respectiv din GF(16). Scrieți în MATLAB algoritmul de calcul a sumei a 2 elemente din fiecare câmp pentru toate valorile posibile.

Produsul în GF(2^m) este definit cu ajutorul unui polinom *p(x)* ireductibil, cu coeficienți binari, de grad *m*, care asigură existența elementului invers pentru orice element nenul din câmp.

În continuare se vor utiliza pentru GF(8):

$$p(x) = x^3 + x + 1 \tag{1}$$

iar pentru GF(16):

$$p(x) = x^4 + x + 1. \tag{2}$$

Produsul a două elemente *a* și *b* din GF se realizează prin multiplicarea polinoamelor asociate acestora și reducere modulo-p(x).

$$c = a \cdot b \rightarrow c(x) = a(x)b(x) modulo [p(x)] \tag{3}$$

✍ **T4.** Calculați $5 \cdot 6$ în GF(8) și în GF(16).

❖ În MATLAB este implementat algoritmul de multiplicare a două elemente din GF(2^m) prin funcția *prodgf(a,b,m)* (Anexa I).

✍ **T5.** Completați tabelele de multiplicare în GF(8), respectiv în GF(16), folosind funcția *prodgf*. Specificați elementele inverse în fiecare caz.

Prin **ordinul** unui element nenul se înțelege puterea minimă la care trebuie ridicat acesta pentru a obține valoarea 1.

✍ **T6.** Deduceți ordinul fiecărui element nenul din GF(8) și GF(16) utilizând funcția *powergf (a,n,m)* (Anexa I). Care sunt elementele de ordin 7 din GF(8)? Dar cele de ordin 3 sau 5 din GF(16)?

Comparativ cu câmpul numerelor complexe, în GF se păstrează:

- metodele de rezolvare a sistemelor liniare de ecuații;
- regulile de calcul matricial;
- transformata Fourier discretă (DFT) și inversă (IDFT) care există în GF(N) numai pentru anumite dimensiuni (*n*) ale vectorilor de semnal și anume numai dacă *n* divide N-1.

Definiție: Transformata Fourier discretă, directă și inversă, a unui vector **c** cu *n* elemente, se calculează în GF(N) cu relațiile :

$$\text{n-DFT}: C_j = \sum_{i=0}^{n-1} c_i w^{ij}\ ,\ j = \overline{0, n-1} \tag{4}$$

$$\text{n-IDFT}: c_i = \sum_{j=0}^{n-1} C_j w^{-ij}\ ,\ i = \overline{0, n-1} \tag{5}$$

unde w este o rădăcină a unității de ordin n în GF(N) dacă n divide N:

$$w^n = 1 \tag{6}$$

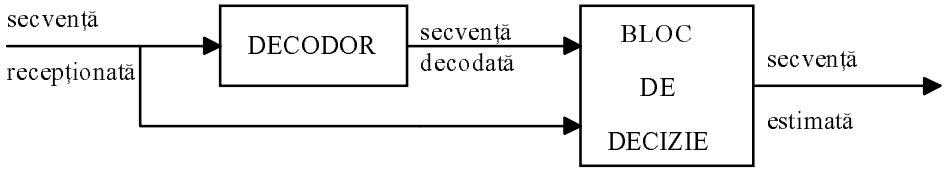

Fig.1 Structura blocului de corecție a erorilor

❖ Definiți în MATLAB funcțiile:

dft7(c) - pentru calculul DFT a vectorului **c** cu 7 elemente din GF(8);

idft7(c) - pentru calculul IDFT a vectorului **c** cu 7 elemente din GF(8).

Similar se pot realiza algoritmi în MATLAB pentru calculul DFT și IDFT în orice alt GF.

Verificați funcțiile de mai sus prin calculul:

DFT(1; 0; 2; 4; 0; 1; 0) și IDFT(6; 2; 1; 4; 1; 2; 5).

✍ **T7.** Să se sintetizeze cu porți logice și bistabile circuitele de sumare și de multiplicare pentru GF(8) (în varianta paralel-paralel).