Exemple :

Le symbole ⊕ désigne l'addition modulo 2. On code le mot binaire **b1b2b3** au moyen du mot de code **b1b2b3p1p2p3** dans lequel :

p1 = b1 ⊕ b2 , p2 = b2 ⊕ b3 , p3 = b1 ⊕ b2 ⊕ b3.

1- Le codage est il systématique ? Déterminer sa dimension **k** , le nombre de bits de contrôle **r,** la longueur **n** et le rendement **ρ.**
2- Démonter que le codage est linéaire, écrire sa matrice génératrice.
3- Etablir la liste des mots de codes et trouver la distance minimale du code.
4- Combien d'erreurs sont détectées de façon certaine ? Combien d'erreurs sont corrigées de façon certaine ?
5- Déterminer la matrice de parité et la transposée de la matrice de contrôle du code.
6- Calculer le syndrome du message R = 011101. Est-ce que R est un mot de code ?
7- Construire une liste de syndromes et indiquer comment on corrige R avec cette liste.

Solution :

1- concernant les propriétés du codage, oui le codage est systématique car se sont des blocs constitués de 3 bits qu'on code donc la dimension k = 3 , on ajoute 3 bits de contrôle r = 3 $\underbrace{\underbrace{\text{b1 b2 b3}}_{\mathbf{k}}\ \underbrace{\text{p1 p2 p3}}_{\mathbf{r}}}_{\mathbf{n}}$

La longueur du code n = k+r = 3+3= 6 et le rendement **ρ =** k/n = 3/6 = 0,5.

2- le codage est-il linéaire ?

Si on a deux bloc b1 b2 b3 et d1 d2 d3 on peut les coder et on obtient deux mots de code b1 b2 b3 p1 p2 p3 et d1 d2 d3 q1 q2 q3

Si on fait la somme des deux blocs

$$\begin{array}{rl} & \text{b1 b2 b3} \\ \oplus & \text{d1 d2 d3} \\ \hline & \text{f1 f2 f3} \end{array}$$

puis on code le résultat qui nous donne f1 f2 f3 R1 R2 R3

Puis on fait la somme des mots de code générés :

$$\begin{array}{rl} & \text{b1 b2 b3 p1 p2 p3} \\ \oplus & \text{d1 d2 d3 q1 q2 q3} \\ \hline & \text{f1 f2 f3 B1 B2 B3} \end{array}$$

Si on trouve le même résultat c'est-à-dire f1 f2 f3 R1 R2 R3 = f1 f2 f3 B1 B2 B3

On dit que le codage est linéaire c'est-à-dire le codage de la somme est égale à la somme des codages.
On peut vérifier ça :
Dans les deux cas b1 b2 b3 $\oplus$ d1 d2 d3 = f1 f2 f3 il nous reste les bits de contrôle.
B1 = (b1 $\oplus$ b2) $\oplus$ (d1 $\oplus$ d2) = (b1 $\oplus$ d1) $\oplus$ (b2 $\oplus$ d2)
R1 = (f1 $\oplus$ f2) et f1 = (b1 $\oplus$ d1) et f2 = (b2 $\oplus$ d2) donc R1 = B1 et ainsi de suite.

Pour écrire la matrice génératrice, on écrit la base canonique
1 0 0 $\longrightarrow$ 1 0 0 1 0 1
0 1 0 $\longrightarrow$ 0 1 0 1 1 1
0 0 1 $\longrightarrow$ 0 0 1 0 1 1

La matrice génératrice G = $\begin{pmatrix} 1&0&0&1&0&1 \\ 0&1&0&1&1&1 \\ 0&0&1&0&1&1 \end{pmatrix}$

3- Pour avoir la liste des mots de code il faut écrire tous les blocs puis on les code :

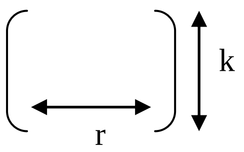
0 0 0	0 0 0 0 0 0
0 0 1	0 0 1 0 1 1
0 1 0	0 1 0 1 1 1
0 1 1	0 1 1 1 0 0
1 0 0	1 0 0 1 0 1
1 0 1	1 0 1 1 1 0
1 1 0	1 1 0 0 1 0
1 1 1	1 1 1 0 0 1

Pour avoir la distance minimale du code , on calcule le poids de chaque mot de code :

0 0 0 0 0 0	W=0
0 0 1 0 1 1	W=3
0 1 0 1 1 1	W=4
0 1 1 1 0 0	W=3
1 0 0 1 0 1	W=3
1 0 1 1 1 0	W=4
1 1 0 0 1 0	W=3
1 1 1 0 0 1	W=4

4- La distance minimale du code est le plus petit poids non nul, donc d = 3.
Si on connaît que d = 3 alors on connaît aussi que si un message qui a une ou deux erreurs est bien détecté comme faux. Car d-1 = 2.
Et un message qui a une erreur (donc détecté comme faux) est bien corrigée car : (d-1)/2 = 1.

5- La matrice de parité est une matrice qui a k ligne et r colonnes

P = 

Pour l’obtenir on écrit la matrice

$$G = \begin{array}{cc} \begin{pmatrix} 1&0&0&1&0&1\\ 0&1&0&1&1&1\\ 0&0&1&0&1&1 \end{pmatrix} & \updownarrow k \\ \overleftrightarrow{\;k\;}\;\overleftrightarrow{\;r\;} & \end{array}$$

Donc la matrice p = $\begin{pmatrix} 1&0&1\\ 1&1&1\\ 0&1&1 \end{pmatrix}$

Pour obtenir la matrice transposée de H (la matrice du contrôle du code) appelée H’ en haut on met la matrice de parité et en bas la matrice identité.

$$H' = \begin{pmatrix} 1&0&1\\ 1&1&1\\ 0&1&1\\ 1&0&0\\ 0&1&0\\ 0&0&1 \end{pmatrix}$$

6- Le syndrome du message R = 011101, pour cela on écrit la matrice H’ et verticalement on écrit les bits de R, on prend les bits de R qui sont égaux à 0 et on efface les lignes de H’ correspondantes.

$$H' = \begin{array}{cc} & R \\ \begin{pmatrix} \cancel{1\ 0\ 1}\\ 1\ 1\ 1\\ 0\ 1\ 1\\ 1\ 0\ 0\\ \cancel{0\ 1\ 0}\\ 0\ 0\ 1 \end{pmatrix} & \begin{array}{c} 0\\1\\1\\1\\0\\1 \end{array} \end{array}$$

Les lignes restantes sont ajoutées et le résultat de cette addition est le syndrome de R

$$\begin{array}{cc} & 1\ 1\ 1\\ & 0\ 1\ 1\\ \oplus & 1\ 0\ 0\\ & 0\ 0\ 1\\ \hline & 0\ 0\ 1 \end{array}$$

Donc le syndrome de R est σ (R) = 0 0 1.

Comme σ (R) n’est pas nul donc R n’est pas un mot de code.

7- Pour obtenir la liste des syndromes, on range par ordre tous les mots binaires de longueur r .

0 0 0	0 0 0 0 0 0
0 0 1	0 0 0 0 0 1 v
0 1 0	0 0 0 0 1 0
0 1 1	0 0 1 0 0 0
1 0 0	0 0 0 1 0 0
1 0 1	1 0 0 0 0 0
1 1 0	0 0 0 1 1 0
1 1 1	0 1 0 0 0 0

$\xleftrightarrow{r}$

Chacun de ses mots est un syndrome de plusieurs messages de longueurs n. On commence par 0 0 0 0 0 0 (le mot avec le poids le plus petit) son syndrome est nul donc on le place en haut de la table, en suite on prend les mots binaires de poids W= 1 qui est le 0 0 0 0 0 1 son syndrome est 0 0 1. On fait ça avec tous les autres.

A la fin, on remarque q'on vient de ranger tous les mots sauf le mot qui a comme syndrome 1 1 0. Donc il faut chercher ce mot. Comment faire ? On remarque que 110 est la somme de 010 et 100, alors on additionne les mots binaires associés à ces deux syndromes c'est-à-dire :
000010 et 000100 on obtient 000110

Pour corriger R = 0 1 1 1 0 1 on calcule son syndrome σ (R) = 0 0 1 on va vers le syndrome de la table 0 0 1 on aura un vecteur erreur v = 000001. Maintenant on additionne v et R.
0 1 1 1 0 1 $\oplus$ 0 0 0 0 0 1 = 0 1 1 1 0 0 on obtient le mot de code qui va servir à corriger R.