Communications Numériques TP 2

PolyTech INFO3 - Basset - Tourancheau

1 Code (détecteur et) correcteur de Hamming

1. Quelle est la capacité de correction d'erreur d'un code de Hamming ?
2. Est-ce qu'un code de Hamming est linéaire ?
3. Quel est le surcoût r en bits, introduit par un code de Hamming, sur un mot de source de k bits ?
4. Donner en fonction de k l'efficacité maximum des codes de Hamming $[n, k]$ avec $n = k + r$ pour $r = 3, 4, 5$ et $n = 2^k - 1$.
5. Donner la limite de cette efficacité maximum lorsque $k \to \infty$
6. Dans un code de Hamming $[7, 4]$, soit le mot de source $m = m_1m_2m_3m_4$, on numérote en binaire de $(001)_2 = 1$ à $(111)_2 = 7$ les indices du message codé x : $x = x_{001}x_{010}x_{011}x_{100}x_{101}x_{110}x_{111}$. Les bits de contrôle $x_{001}, x_{010}, x_{100}$ se situe aux positions puissance de 2, ceux du message sont gardés dans le même ordre: $x = x_{001}x_{010}m_1x_{100}m_2m_3m_4$ Exprimer la valeurs des bits de parité en fonction de $m_1m_2m_3m_4$.
7. Comment détecte-t-on et corrige-t-on une erreur à la réception ? Illustrer par un exemple.
8. Donner la matrice généatrice pour ce code $[7, 4]$.
9. Donner la matrice de contrôle pour ce code $[7, 4]$.
10. Programmez un codeur $[n, k]$ pour des mots de source tels que $n + 1 = 2^{n-k}$.
11. Testez votre codeur sur les exemples de mots suivants:
    0000; 1111; 1011; 10101010101; 11110000101.
12. Programmez un décodeur-correcteur $[n, k]$ avec $n + 1 = 2^{n-k}$.
13. Testez votre décodeur sur les exemples de mots suivants:
    0000001; 0000010;0100000;1111111; 1011010; 101010101010101; 111100001010101.

2 Détection d'erreur CRC

1. Le code obtenu avec l'ajout d'un CRC est-il un code systématique ?
2. Qu'est ce qu'un code cyclique ?
3. Soit le message 10011101, Quelle est sa représentation polynomiale $D(x)$?
4. On veut "protéger" $D(x)$ avec le polynôme générateur $G(x) = x^4 + x^2 + x + 1$. Combien de bits va-t-on ajouter ?
5. Quels sont les bits à ajouter pour former le mot codé $C(x)$ à transmettre (donner votre calcul manuel) ?
6. Vérifiez que $C(x)$ est un multiple de $G(x)$ (donner votre calcul manuel).
7. G(x) permet de détecter toutes les erreurs de 1 bit, pourquoi ?
8. G(x) permet de détecter toutes les erreurs de 2 bits, pourquoi ?
9. G(x) permet de détecter toutes les erreurs d'un nombre impair de bits, pourquoi ?
10. G(x) permet de détecter toutes les erreurs en rafales d'un nombre inférieur à 4 bits, pourquoi ?
11. Ecrire un programme de calcul du code CRC pour un mot binaire D(x) et un polynome générateur G(x) donnés, il fournira C(x) en résultat.
12. Validez votre programme avec $G(x)$ et les mots de source : 11001101, 10101011, et 00110101.

13. Ecrire un programmes de vérification du CRC prenant en entrée un mot de code C(x) et un polynôme générateur G(x) et retournant la détection ou non d'erreur(s).

14. Validez votre programme avec $G(x)$ et les mots de code : 10001101000, 11101011101, et enfin les 3 mots de code générés dans la validation question 12.