**Logique Combinatoire**

**TD 2 (27/10/2010)**

**Exercice 4. Système de transmission numérique avec correction d’erreur.**

***Rappels des points importants de l’énoncé :***

***Note*** : *je ne reprends pas ici l’énoncé intégral du polycopié, merci de le lire attentivement !*

* *Code de Hamming*
* *4 éléments binaires correspondant à un chiffre du système décimal* ***m1, m2, m3, m4****.*
* *3 éléments binaires de contrôle de parité* ***k1, k2, k3****.*
* *La structure du mot :*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Position du bit | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Nom du bit | k1 | k2 | k3 | m1 | m2 | m3 | m4 |
| Exemple | 0 | 1 | 1 | 0 | 0 | 1 | 0 |

*L’exemple numérique donné dans le tableau ci-dessus correspond à N=2 dans le code de Hamming !* ***Vérifiez****!*

* *3 tests de parité* 
  + *T1(test sur k1) se fait sur les bits : 1, 4, 5, 7 (c-à-d : k1, m1, m2, m4)*
  + *T2(test sur k2) se fait sur les bits : 2, 4, 6, 7 (c-à-d : k2, m1, m3, m4)*
  + *T3(test sur k3) se fait sur les bits : 3, 5, 6, 7 (c-à-d : k3, m2, m3, m4)*
* *Test de parité* ***paire*** *(le résultat du test est égal à 0 si le nombre de 1 dans la zone considéré est pair, ce qui revient à dire qu’il n’y a pas d’erreur !)*
* *(T3T2T1)est un mot en code BCD qui donne en décimal la position du bit erroné*
* *1 seul bit au maximum peut-être erroné et l’erreur est cantonné au niveau des bits d’information « mi » (pas d’erreur sur les k)*

**Questions**

4.1 Donner le schéma du dispositif « émetteur » permettant de générer les bits k1, k2 et k3 :

On va utiliser le tableau du code de Hamming pour établir les expressions des bits de contrôle de parité ki en fonction des bits d’information m1, m2, m3 et m4. On trace les tableaux de Karnaugh des bits de contrôle de parité. On remplit les cases avec les données de Hamming.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **k1** |  |  |  | |  | m4  m1m2 | 0 | 1 | |  | 00 | 0 | 1 | |  | 01 | 1 | 0 | |  | 11 | X0 | X1 | |  | 10 | 1 | 0 | | |  |  |  |  | | --- | --- | --- | --- | | **k2** |  |  |  | |  | m4  m1m3 | 0 | 1 | |  | 00 | 0 | 1 | |  | 01 | 1 | 0 | |  | 11 | X | X1 | |  | 10 | 1 | 0 | | |  |  |  |  | | --- | --- | --- | --- | | **k3** |  |  |  | |  | m4  m2m3 | 0 | 1 | |  | 00 | 0 | 1 | |  | 01 | 1 | 0 | |  | 11 | 0 | 1 | |  | 10 | 1 | 0 | |

Explication de remplissage de tableau de Karnaugh de k1 : on sait que k1 est regroupé avec les bits numéros 4, 5 et 7 (le test de parité T1 sur k1 se fait sur les bits 1, 4, 5 et 7), c'est-à-dire m1, m2 et m4. Pour la 1ère case (1ère ligne et 1ère colonne) on repère dans le code de Hamming la ligne où m1m2m4 = 000 et on inscrit dans la case la valeur de k1 correspondante, pour la 2ème case (1ère ligne et 2ème colonne) on repère la ligne où m1m2m4 = 001 et on inscrit dans la case la valeur de k1 correspondante, ainsi de suite… Lorsque vient le tour des 5ème et 6ème cases, on doit rechercher les lignes du code de Hamming où m1m2m4 = 110 et 111, or ces lignes n’existent pas ! Alors dans ce cas on peut se permettre d’y inscrire X et lui attribuer ensuite la valeur (0 ou 1) qui convient la mieux pour une plus grande simplification de l’expression recherchée.

On avait vu que les 1 qui se disposent en diagonal dans un tableau de Karnaugh correspondent à une expression qui relie les variables avec l’opérateur OU Exclusif. Donc ici on a intérêt à remplacer le X de la case 5 par un 0 et le X de la case 6 par un 1.

D’où **k1 = m4 m2 m1** (Je vous conseille de faire la démonstration !)

Il en va de même pour k2 et le résultat est semblable **k2 = m4 m3 m1**

Toutes les combinaisons possibles de m2m3m4 étant disponibles dans le code de Hamming le tableau de Karnaugh de k3 ne contient pas de X (vérifiez !), on a donc **k3 = m4 m3 m2**

Les expressions obtenues nous permettent de proposer le schéma du dispositif émetteur permettant de générer les bits de contrôle de parité.

Emetteur

m1

m2

m3

m4

k1

k2

k3

m1

m2

m3

m4

=1

=1

=1

=1

=1

4.2 Donner le schéma du dispositif « récepteur » permettant de générer les bits T1, T2 et T3 :

On va tracer les tableaux de Karnaugh des bits de détection d’erreur T1, T2 et T3 afin d’obtenir leur expression en fonction des ki et mj. On sait que T1 dépend de *k1, m1, m2, m4*, T2 dépend de *k2, m1, m3, m4* et T3 dépend de *k3, m2, m3, m4*. On va donc remplir les tableaux de Karnaugh pour chaque Ti en fonction des 4 bits auxquels il est lié ; et on mettra 0 pour chaque combinaison de bits contenant un nombre pair de 1, et on mettra 1 pour le cas inverse (ce qui correspond à une erreur car on est dans le contrôle de parité paire)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **T1** |  |  |  |  |  | |  | m’2m’4  k1m’1 | 00 | 01 | 11 | 10 | |  | 00 | 0 | 1 | 0 | 1 | |  | 01 | 1 | 0 | 1 | 0 | |  | 11 | 0 | 1 | 0 | 1 | |  | 10 | 1 | 0 | 1 | 0 | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **T2** |  |  |  |  |  | |  | m’3m’4  k2m’1 | 00 | 01 | 11 | 10 | |  | 00 | 0 | 1 | 0 | 1 | |  | 01 | 1 | 0 | 1 | 0 | |  | 11 | 0 | 1 | 0 | 1 | |  | 10 | 1 | 0 | 1 | 0 | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **T3** |  |  |  |  |  | |  | m’3m’4  k3m’2 | 00 | 01 | 11 | 10 | |  | 00 | 0 | 1 | 0 | 1 | |  | 01 | 1 | 0 | 1 | 0 | |  | 11 | 0 | 1 | 0 | 1 | |  | 10 | 1 | 0 | 1 | 0 | |

On a encore une structure qui donne ***XOR***.

**T1 = k1 m’1 m’2 m’4**

**T2 = k2 m’1 m’3 m’4**

**T3 = k3 m’2 m’3 m’4**

Les mi à la sortie de l’émetteur ayant « voyagé » lors de la transmission ont subi des perturbations et sont devenus m’i.

Récepteur

T1

=1

=1

T2

=1

T3

=1

=1

=1

=1

=1

m’1

m’2

m’3

m’4

m’1

m’2

m’3

m’4

k1

k2

k3

4.3 Dispositif simple réalisant la correction du bit erroné (1 seul bit erroné au maximum parmi uniquement les mi).

Nous savons que (T2T1T0)2 indique le numéro de l’élément binaire erroné, donc celui qu’il faut corriger. Si par exemple (T2T1T0)2 = (101)2 ça veut dire que c’est l’élément numéro 5 qui est erroné (m2). La correction consiste simplement à changer l’élément binaire faux à l’aide d’un circuit complément.

Pour transmettre l’information du numéro de l’élément binaire erroné du code BCD en décimal il faut un décodeur BCD-Décimal. Donc s’il y a erreur, c’est la sortie correspondant au numéro du bit erroné qui sera à l’état haut (1) toutes les autres resteront à l’état bas (0). Dans ce cas il faut rechercher l’opération qui inverse l’entrée lorsque la référence (la deuxième entrée connecté à la sortie du décodeur) est à 1 (donc si erreur) et qui la recopie lorsque la référence est à 0 (pas d’erreur). C’est le « OU exclusif » qui est soit un opérateur transparent soit un opérateur inverseur suivant sa commande (ou référence).

En effet, soit E = entrée et R = Référence, la sortie S = E R obéit au tableau de Karnaugh suivant :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **S** |  |  |  | |  | R  E | 0 | 1 | |  | 0 | 0 | 1 | |  | 1 | 1 | 0 | | On voit bien sur ce tableau que la 1ère colonne (la référence est égale à 0) est identique à E donc c’est une opération transparente et la 2ème colonne est bien l’inverse de E (R = 1). |

D’où le circuit correcteur :

T1

T2

T3

=1

=1

=1

=1

m1

m2

m3

m4

m’2

m’3

m’4

m’1

bits corrigés

0 1 2 3 4 5 6 7