heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

TBA – LABORATOIRE N°2

PROFESSEUR

M. Hervé Dedieu

AUTEURS

Steve LIENHARD
Arnaud BURKALTHER

TEMPS A DISPOSITION : 2 séances
DATE DE DEBUT: 19 Février 2010
DATE DE FIN : 07 Mai 2010

Hes·so
Haute Ecole Spécialisée de Suisse occidentale

Table des matières

Introduction

Ce laboratoire a pour but de mettre en pratique les parties théoriques abordées durant les cours, tout particulièrement le chapitre 4, et donc de nous familiariser avec les codes détecteurs et correcteurs d'erreur. Ce laboratoire est principalement axé sur les capacités de détections et corrections d'erreur.

TP sur génération de codes blocs et calcul de la capacité de détection

Partie pratique sur la capacité de détection

L'ensemble des mots code est le suivant :

000
001
010
011
100
101
110
111

L'ensemble des mots info est le suivant :

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

Partie théorique sur la capacité de détection

Vérification que la somme de deux mots-code est un mot-code

Prenons les mot-codes 0010111 et 0101110.
L'addition nous donne (0010111 + 0101110 = 0111001) se qui est bien un mot-code (celui résultant du mot info 011).

Démonstration que la somme de deux mots-code est un mot-code

Prenons les mots-code A et B, et considérons le résultat S obtenu par l'addition de ces deux.

$S = A + B$

A et B étant des mots-codes, ils sont issu de K-uples et peuvent s'écrire sous la forme suivante $A = A_d * G$ et $B = B_d*G$ où A_d et B_d sont des mots-info possible et G la matrice génératrice.

Nous avons donc :

$S = A_d * G + B_d*G = G (A_d + B_d)$

L'addition étant modulo 2, la somme de mots-code reste toujours un mot-code. Cela résulte du fait que le champ de Gallois est tel qu'on ne sort jamais de celui-ci, lors d'addition ou de multiplication. $A_d + B_d$ peut donc être remplacé par C_d et nous aboutissons donc à $S = C_d*G$ avec S qui est donc bien un mot-code.

Démonstration que la somme de deux mots-code est un mot-code

Lors de calcul en modulo 2, la différence étant identique à l'addition, nous ne présenterons pas cette démonstration du fait qu'elle est faite de manière identique à la démonstration précédente.

Déduire de la question précédente que la capacité de détection est le nombre minimum de 1 dans le mot-code qui en compte le moins si l'on excepte le mot-code nul.

Du fait que toute combinaison linéaire de mot code produit un autre mot code, la différence de mot du code est encore un mot du code. Par conséquent, comme indiqué dans la question précédente, les différences entre mots du code sont des mots-code. Il devient donc logique que la distance minimal entre deux mots du code est le vecteur de moindre poids dans tous les mots-code, à l'exception du vecteur nul.

Réflexion sur le sens de la notion de mot-code

Créer un mot-code c'est créer une loi de dépendance entre des bits d'information et des bits de redondance.

Pour la matrice génératrice précédemment donnée, en désignant sous la forme $[x_1, x_2, x_3]$ les bits d'information, et sous la forme $[y_1, y_2, y_3, y_4, y_5, y_6, y_7]$ les bits de redondance.

Place des bits d'information et bits de redondance dans les mots-code

En observant les mots-informations et les mots-code en résultant :

0 0 0	-->	0 0 0 0 0 0 0
0 0 1	-->	0 0 1 0 1 1 1
0 1 0	-->	0 1 0 1 1 1 0
0 1 1	-->	0 1 1 1 0 0 1
1 0 0	-->	1 0 0 1 0 1 1
1 0 1	-->	1 0 1 1 1 0 0
1 1 0	-->	1 1 0 0 1 0 1
1 1 1	-->	1 1 1 0 0 1 0

Nous pouvons nous apercevoir que la même séquence de bits que le mot info se trouve en en-tête de chaque mot-code. Les bits d'information sont conservés en en-tête de chaque mot code et il s'agit donc d'un code systématique. Nous pouvons donc écrire :

Bits d'information	$Y_1 = x_1$
	$Y_2 = x_2$
	$Y_3 = x_3$
Bits de redondance	Y_4
	Y_5
	Y_6
	Y_7

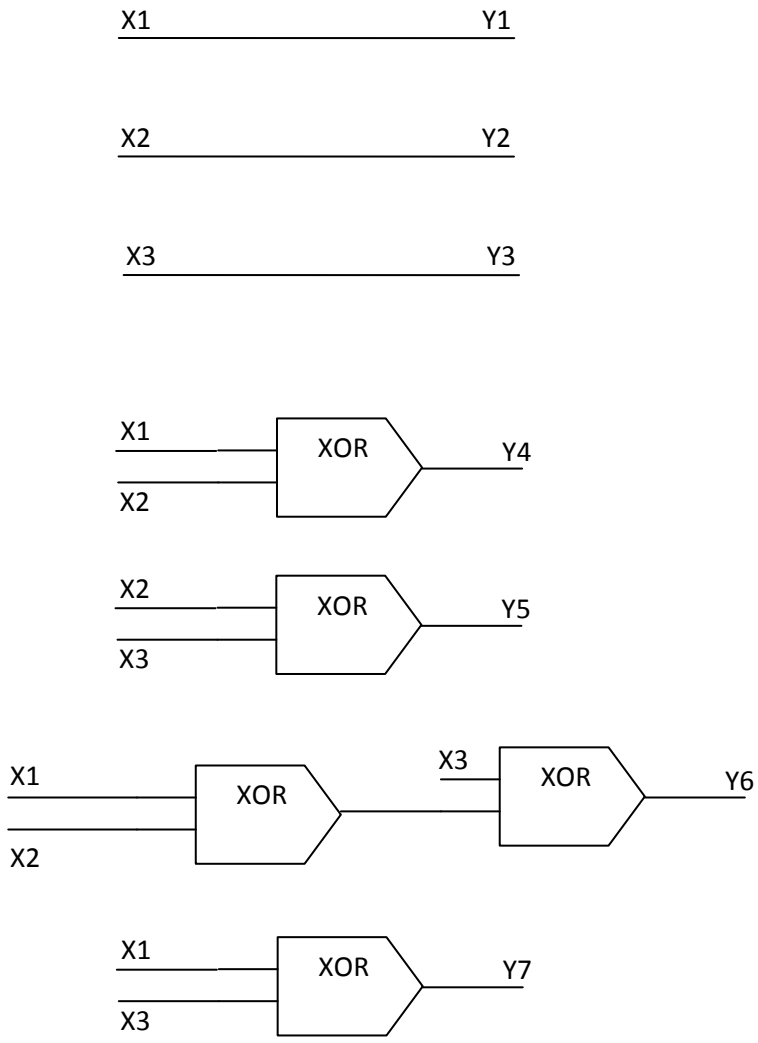
Loi mathématique reliant les bits de redondance aux bits de mots-code

$$[x_1, x_2, x_3] * \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} = [x_1 , x_2 , x_3 , x_1+x_2 , x_2+x_3 , x_1+x_2+x_3 , x_1+x_3]$$

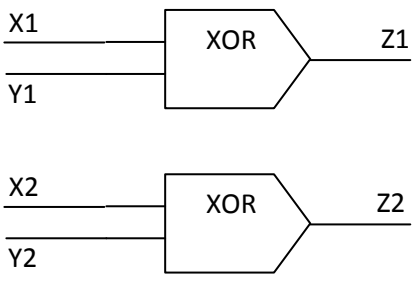
Nous pouvons donc aussi écrire les relations sous la forme suivante :

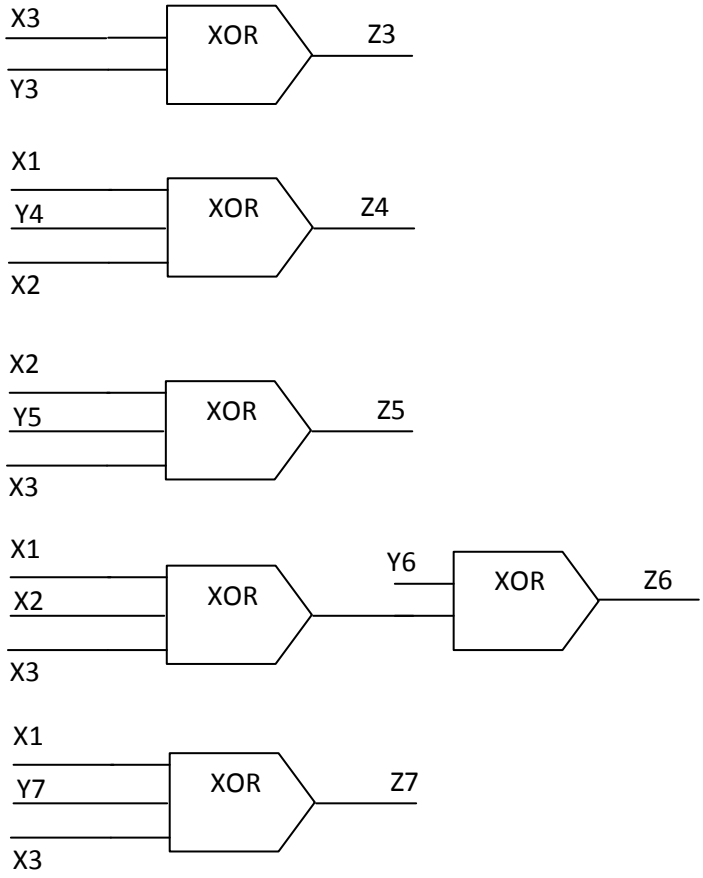
$Y_1 = x_1$
$Y_2 = x_2$
$Y_3 = x_3$
$Y_4 = x_1 + x_2$
$Y_5 = x_2 + x_3$
$Y_6 = x_1 + x_2 + x_3$
$Y_7 = x_1 + x_3$

Schéma des dépendances entre bits d'information et bit de redondance



Compléter le schéma précédant pour faire un détecteur d'erreurs qui signal à la réception la présence d'une erreur





Si une des sortie Z vaut 1, il y a donc une erreur

Détectera-t-on toutes les erreurs ?

Non, toutes les erreurs ne seront pas détectées, uniquement un nombre d'erreur inférieur ou égal à $d_{min} - 1$ pourra être décelé. Un nombre d'erreur supérieur à cette valeur passera inaperçu du fait que le mot-code incorrect ainsi obtenu sera vu comme un autre mot-code valide.

TP sur la notion de la capacité de correction

Définition personnelle de la Capacité de correction :

La capacité de correction d'un code est la capacité qu'a celui-ci à retrouver la valeur correcte d'un code, estimée incorrect grâce à la capacité de détection.

Démonstration de Capa Correction

D_{min} représente la distance minimal existante entre deux mots-code, en d'autre termes, le nombre de bits minimum qu'il faudrait modifier pour obtenir un autre mot-code valide.

La modification de d_{min} bits donnant un autre mot-code, la modification de dmin-1 bits nous donne un mot-code incorrect, se qui est le nombre de bits séparant deux mots-code ou aussi connu comme étant la capacité de détection.

Il parait donc logique qu'une modification d'un nombre de bits égal à une valeur se trouvant dans l'intervalle de zéro à $\frac{dmin-1}{2}$ modifiera le mot-code de telle sorte que celui-ci sera à une distance moindre de son état initial que d'un autre mot-code valide.
Alors qu'une modification d'un nombre de bits se trouvant dans l'intervalle de $\frac{dmin-1}{2}$ à d_{min}-1 donnera un mot-code incorrect avec une distance plus proche d'un autre mot-code que de l'état initial du mot.
Nous pouvons donc en déduire que la capacité de correction est de $\left\lfloor \frac{dmin-1}{2} \right\rfloor$.
Avec la partie inférieure utile dans le cas d'une valeur impaire.

Capacité de correction pour la matrice G définie plus haut

0 0 0	-->	0 0 0 0 0 0 0
0 0 1	-->	0 0 1 0 1 1 1
0 1 0	-->	0 1 0 1 1 1 0
0 1 1	-->	0 1 1 1 0 0 1
1 0 0	-->	1 0 0 1 0 1 1
1 0 1	-->	1 0 1 1 1 0 0
1 1 0	-->	1 1 0 0 1 0 1
1 1 1	-->	1 1 1 0 0 1 0

La distance minimum entre deux mot-code = dmin = 3

$\left\lfloor \frac{dmin-1}{2} \right\rfloor = \left\lfloor \frac{3-1}{2} \right\rfloor = 1$

TP sur la notion de syndrome

Construction d'une matrice H orthogonal à G

G * H = 0
G = [$\prod_k$,P]
H = [$\frac{P}{\prod}$]

G = [3 x 7] = $\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$

La matrice H sera de degré [7 x 4] du fait que le résultat de la multiplication de celle-ci avec un mot-code doit donner un vecteur [1 * 4]. La valeur quatre représentant le nombre de bits de redondance présents dans le mot-code.

$$H = [7 \times 4] = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$[x_1, x_2, x_3, y_4, y_5, y_6, y_7] * \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = [x_1+x_2+y_4, x_2+x_3+y_5, x_1+x_2+x_3+y_6, x_1+x_3+y_7] = [0,0,0,0]$$

$$\begin{array}{lcl} x_1 + x_2 + y_4 = 0 & & y_4 = x_1 + x_2 \\ x_2+x_3 + y_5 = 0 & \Rightarrow & y_5 = x_2+x_3 \\ x_1+x_2+x_3+y_6= 0 & & y_6= x_1+x_2+x_3 \\ x_1+x_3 + y_7 = 0 & & y_7 = x_1+x_3 \end{array}$$

L'information de H est identique à celle de G donc H est correcte.

Déduction que C x H' = 0

$$I \xrightarrow{G} C \xrightarrow{H} S = C * H = 0$$

$[x_1, x_2, x_3]$

C = I *G
S = C * H = I * G * H = 0 (car G * H = 0)

Déduction que C' x H' ≠ 0

De la question précédente, nous pouvons déduire que cette affirmation est correcte, car étant donné que C découle de G et que H est la matrice identité de G (soit G*H = 0), il va de soit que C' n'étant pas produit à l'aide de G ne sera pas orthogonal à H.

Règle de calcul du syndrome

En nous référant aux points ci-dessus, nous pouvons déduire que la règle de calcul du syndrome, correspond à une multiplication du mot-info par H'.

Conclusion

Ce laboratoire nous a permis tout d'abord de nous familiariser avec les notions de détection et de corrections d'erreur dans les séquences de bits transmises par un canal de communication. Nous avons pu mettre en pratique les concepts appris au cours de la théorique, entre autre les matrices génératrices et correctrices d'erreur. Il nous a également permis d'apprendre à utiliser matlab pour effectuer nos algorithmes de calculs. Ceci nous sera particulièrement utile pour la suite de notre cursus.