Tarik YILDIZ
Ouassim MESSAGIER

Compte-Rendu TPs CODAGE

TP 1 : Code d'étalement

Objectif :

L'objectif de ce TP est de mettre en œuvre un codeur de Hadamard en langage C.

Définition :

Le codeur de Hadamard (ou matrice de Hadamard) est une matrice carrée (puissances de 2) contenant seulement les coefficients 1 ou -1. Pour créer cette matrice, il faut que la première ligne soit « normale » (des 1 partout), que la première colonne soit aussi normale, mais que l'élément restant (c'est-à-dire le seconde colonne de la seconde ligne) doit être l'inverse des autres éléments. Pour chaque puissance de 2 (taille du tableau) nous avons juste à le diviser en 4 parties égales et appliquer l'algorithme décrit juste avant.

$$\begin{pmatrix} H & H \\ H & -H \end{pmatrix}$$

Fig. 1 : Matrice de Hadamard

Exemple avec une matrice de 4 (2^2) :

$$H_4 = \left(\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right)$$

Fig. 2 : Matrice H4 de Hadamard

Nous pouvons donc voir à quoi cela ressemble d'après l'explication donnée.

Création de la matrice de Hadamard en fonction du nombre d'utilisateurs :

La taille du tableau correspond au nombre d'utilisateurs voulus. Un utilisateur est associé à une ligne.

Pour ce faire, la matrice devra être de taille paire et qui est une puissance de deux, donc par exemple, si nous voulons trois utilisateurs, nous mettrons un tableau de taille 4 (2^2) comme présenté ci-dessus (cf. voir figure 2).

Pour créer celle-ci en langage C, on choisi le nombre d'utilisateurs voulus, par rapport à cela on crée une matrice de la taille du nombre d'utilisateurs choisis (en largeur et hauteur) et on initialise toutes les cases à une valeur nulle (0).

Ensuite, pour générer la matrice de Hadamard par rapport au nombre d'utilisateurs voulu, nous initialisons le premier élément de la première ligne de la matrice à 1, puis nous avons juste à faire deux boucles (une pour les lignes, l'autre pour les colonnes), et enfin nous créons la matrice à partir de la première puissance de deux possible (c'est-à-dire 1 donc $2^1 = 2$) et on ajoute celle-ci à elle même à la fin des deux itérations pour passer à la puissance suivante jusqu'à qu'elle soit égale au nombre d'utilisateurs voulus.

Sur la première itération, nous créerons la matrice H2 comme montré ci dessous :

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Fig. 3 Matrice H2 de Hadamard.

Cette matrice est créée par le code montré ci-dessous :

```
void hadamard(int nbUtilisateurs, int M[nbUtilisateurs][nbUtilisateurs]){
  // création de la matrice de hadamard
  M[0][0] = 1;
  int i,j;
  int facteur=1;
  while(facteur< nbUtilisateurs){
    for(i=0; i<facteur ; i++){
      for(j=0;j<facteur; j++){
        M[i+facteur][j] = M[i][j];
        M[i][j+facteur] = M[i][j];
        M[i+facteur][j+facteur] = - M[i][j];
      }
    }
    facteur+= facteur;
  }
}
```

Fig. 4 : Fonction de création de la matrice de Hadamard

Pour plus de compréhension, nous préférons mettre 1 à la place de 0 au niveau des itérations pour les explications.

Nous voyons donc que :
- que pour l'élément [1,1] (première ligne et première colonne) est égal à 1
- que l'élément [1+facteur,1] est égal à 1
- que l'élément [1,1+facteur] est égal à 1
- que l'élément [1+facteur,1+facteur] est égal à -1

Ce qui génère la matrice Fig. 3, puis si le nombre d'utilisateurs est supérieur à 2, la fonction créera la matrice 4 montrée en Fig. 2, etc.

Pour une matrice de 4 par exemple (2^2), la matrice générée par le programme sera :

```
********** Matrice de Hadamard **********

    1    1    1    1
    1   -1    1   -1
    1    1   -1   -1
    1   -1   -1    1

********** Fin Matrice de Hadamard **********
```

Fig. 5: Matrice de Hadamard générée par le programme

Nous pouvons remarquer sur la Figure 5 que la matrice est correctement générée.

Séquences d'étalement de Hadamard :

Définition :

Une séquence d'étalement est le fait d'émettre la trame d'un utilisateur mais codée par un/des bit(s) préalablement choisi au lancement du programme.

Ces bits représentent la modification ou non de la ligne à laquelle ce bit est donné, exemple :
Nous avons une matrice Hadamard de taille 4, la première ligne sera :

|1|1|1|1|

Et nous voulons faire une séquence d'étalement de celle ci en 2 bits qui sont 1 puis 0, ce qui donnera une séquence d'étalement égale à :

|1|1|1|1|-1|-1|-1|-1|

Si nous analysons cette dernière trame nous pouvons voir qu'elle à une taille qui est double à celle de départ car on la duplique suivant le nombre de bits choisis. Enfin nous voyons aussi qu'elle est divisé en deux parties, une seulement avec des 1 et une autre avec des -1, cela se produit comme cela car le premier bit choisi est 1 ce qui donne la matrice de départ et le second étant 0 donne l'inverse de chaque élément
(1 → -1 et -1→ 1).

Résultat des séquences d'étalement :

Le résultat de toutes les séquences d'étalement est égal à la somme de chaque colonnes comme montré ci-dessous pour la matrice de Hadamard de taille 4 (Cf. Fig. 5) :

```
********** Messages Codés **********

M1 : |    1|    1|    1|    1|   -1|   -1|   -1|   -1|
M2 : |   -1|    1|   -1|    1|   -1|    1|   -1|    1|
M3 : |   -1|   -1|    1|    1|   -1|   -1|    1|    1|
M4 : |   -1|    1|    1|   -1|   -1|    1|    1|   -1|

**********  Fin des Messages Codés  **********


**********   Résultat   **********

   -2    2    2    2   -4    0    0    0
```

Fig. 6 Séquences d'étalement et résultat de celles-ci

Désétallement :

Définition :

Le désétallement consiste à trouver les bits d'étalement grâce au résultat de la séquence d'étalement finale (Cf. Figure 6), ce procédé renvoie les bits d'étalement à la fin de son fonctionnement.

Le procédé consiste donc à mettre dans une matrice temporaire les valeurs du résultat de la séquence d'étalement, elles sont dans chaque ligne et nous multiplions les éléments de chaque ligne par les éléments à la même position mais dans la matrice de départ (Hadamard), l'élément sera donc du signe opposé ou intact.

L'addition des éléments de chaque ligne entre eux donnera donc un nombre positif ou négatif, traduit donc par 1 ou 0.

Voici l'exemple en images pour la matrice H4 :

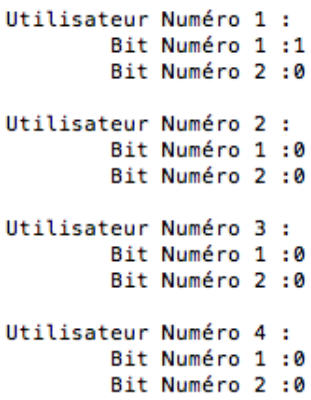

Fig. 7 : Séquence d'étallement

Nous pouvons constater sur la figure ci-dessus que la séquence d'étalement choisie est la même (nombres négatifs sont vu comme un 0).

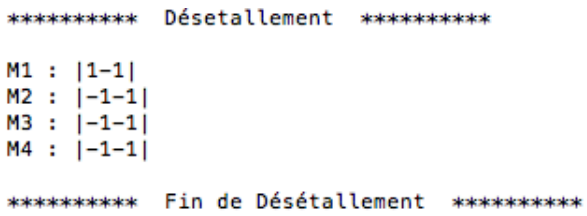

Fig. 8 : Résultat de Désétallement

TP 2 : Gold & JPL

Fonctionnement du code de Gold

Nous avons plusieurs messages de même taille mais codés de différentes manières, c'est à dire que nous avons plusieurs code LM dont le polynôme est différent. Par exemple le premier message aura pour polynôme [5,2] et le deuxième [5,4,2,1]. Nous attribuons leurs dernières colonnes dans un tableau et effectuer un OU exclusif (XOR) entre m1 et m2 qui sont respectivement les messages 1 et messages 2.

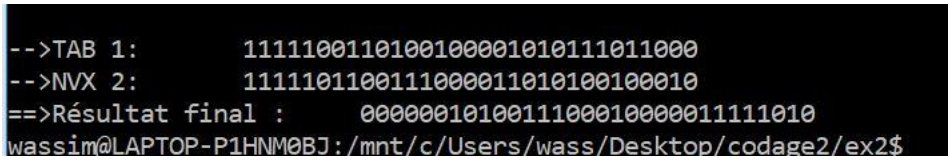

Fig. 9 : Résultat de l'exemple précédent (Gold)

Fonctionnement du code JPL

Le code JPL est un peu plus complexe est permet d'avoir un résultat différent du code de Gold. Le code JPL effectue un traitement sur des messages pouvant avoir différentes tailles ce qui permet d'être plus utile comparer au code Gold qui lui doit avoir des messages de mêmes longueurs. La contrainte de celui-ci est que le nombre d'étage doit être un nombre premier par rapport aux étages des autres messages. Le traitement effectuer est assez simple. Comme pour le code de Gold la dernière colonne des matrices est copiée dans différents tableaux. Les tableaux n'ont donc pas forcément la même taille est donc nous effectuons un Ou-exclusif(XOR) entre chaque case des tableaux.

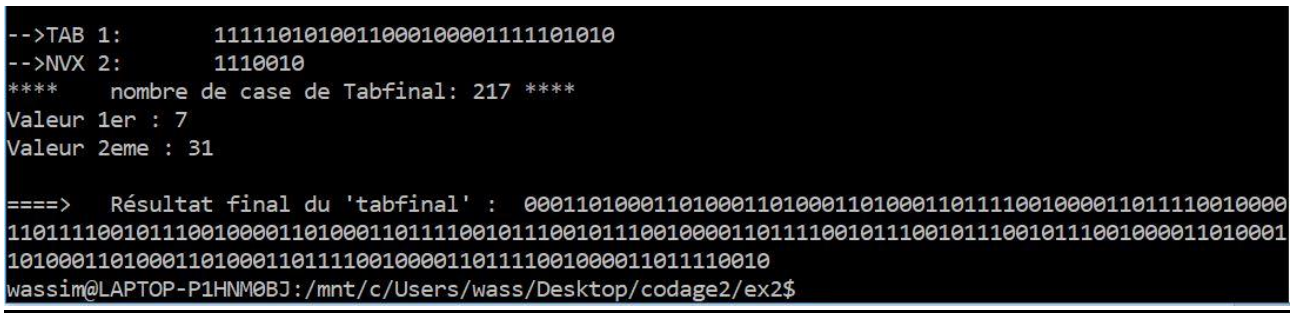

Fig.10 : Résultat pour un exemple du code JPL avec [5,1] [3,2] comme valeurs.