Département d'Informatique
DUT Informatique 2A
Traitement de l'information

TD1 - Correction des exercices

1 Double parité

- On désire envoyer le mot `HELLO`. Pour ce faire le bloc de bits correspondant est obtenu en mettant sur chaque colonne le code ASCII d'un caractère à transmettre, en commençant par le premier caractère sur la première colonne.
- Pour détecter une erreur (voire plusieurs erreurs) sur ce bloc de bits, on utilise une double parité, en effectuant :
  - un contrôle de parité paire sur les lignes (contrôle longitudinal) ;
  - un contrôle de parité paire sur les colonnes (contrôle transversal), ce contrôle s'applique également aux bits de parité du contrôle longitudinal.
- On vous demande de donner la séquence binaire en octal effectivement transmise.

Indication : codes ASCII sur 7 bits.
- `E` $\rightarrow 69 = 1000101_2$;
- `H` $\rightarrow 72 = 1001000_2$;
- `L` $\rightarrow 76 = 1001100_2$;
- `O` $\rightarrow 79 = 1001111_2$.

Correction

Bit	*Caracteres*					*Parite paire*		
	1	*2*	*3*	*4*	*5*			
	H	E	L	L	O	LRC		
0	0	1	0	0	1	0	<=>	***22***
1	0	0	0	0	1	1	<=>	***03***
2	0	1	1	1	1	0	<=>	***36***
3	1	0	1	1	1	0	<=>	***56***
4	0	0	0	0	0	0	<=>	***00***
5	0	0	0	0	0	0	<=>	***00***
6	1	1	1	1	1	1	<=>	***77***
VRC	0	1	1	1	1	0	<=>	***36***
	Parite paire							

On obtient donc la séquence suivante : 22033656000077736_8.

2 Code de Hamming

- On considère un code de Hamming avec parité impaire pour contrôler une transmission.
- On vous demande de trouver le message qui sera envoyé, sachant que les données à transmettre sont $A9C3_{16}$.
- Déterminer le message initial, en effectuant éventuellement la correction d'une erreur, si le message reçu est 6130014_8.

Correction

① Détermination du message à transmettre

— Comme $A9C3_{16} = 1010100111000011_2$ on en déduis que le nombre de bits de données est $n = 16$. Il est alors possible de calculer le nombre k de bits de contrôle qui sont nécessaires. En notant $t = n + k$, on a k qui vérifie :

$$\begin{aligned} 2^k \geq t+1 \quad &\Leftrightarrow \quad 2^k \geq n+k+1 \\ &\Leftrightarrow \quad 2^k - k \geq n+1 \end{aligned}$$

Or la plus petite valeur entière pour k qui satisfait cette relation est $k = 5$.
On conclut donc qu'il faut 5 bits de contrôle et que le nombre total de bits de la séquence binaire à transmettre est $t = 21$ bits (les positions des bits étant numérotées de 1 à 21).

— Détermination des bits de contrôle avec la méthode basique
Le code binaire des positions sur 5 bits permet de trouver quel(s) bit(s) de contrôle, notés k_5, k_4, k_3, k_2, k_1 vérifie(nt) une position.

1	00001	10	01010	19	10011
2	00010	11	01011	20	10100
3	00011	12	01100	21	10101
4	00100	13	01101		
5	00101	14	01110		
6	00110	15	01111		
7	00111	16	10000		
8	01000	17	10001		
9	01001	18	10010		

On en déduis que :
— k_1 contrôle les bits 1,3,5,7,9,11,13,15,17,19,21 ;
— k_2 contrôle les bits 2,3,6,7,10,11,14,15,18,19 ;
— k_3 contrôle les bits 4,5,6,7,12,13,14,15,20,21 ;
— k_4 contrôle les bits 8,9,10,11,12,13,14,15 ;
— k_5 contrôle les bits 16,17,18,19,20,21.
Finalement, il reste à déterminer la valeur de chaque bit de contrôle à partir des positions qu'il contrôle, sachant que la parité voulue est impaire.

1	0	1	0	1	**0**	0	0	1	1	1	0	0	**0**	0	0	1	**1**	1	**0**	**0**
21	20	19	18	17	**16**	15	14	13	12	11	10	9	**8**	7	6	5	**4**	3	**2**	**1**

On obtient donc :

— $k_5 = 0$;
— $k_4 = 0$;
— $k_3 = 1$;
— $k_2 = 0$;
— $k_1 = 0$.

— Détermination des bits de contrôle avec la méthode simplifiée
On utilise les codes binaire naturel des positions, correspondant à des données, qui sont à 1 dans la séquence. On effectue ensuite une addition modulo 2 inversée, puisque la parité est impaire, pour obtenir directement les bits de contrôle.
Addition modulo 2 inversée :
— 1 pour un nombre pair de 1 ;
— 0 pour un nombre impair de 1 ;

$$\begin{array}{rcl} 21 & = & 1\,0\,1\,0\,1 \\ 19 & = & 1\,0\,0\,1\,1 \\ 17 & = & 1\,0\,0\,0\,1 \\ 13 & = & 0\,1\,1\,0\,1 \\ 12 & = & 0\,1\,1\,0\,0 \\ 11 & = & 0\,1\,0\,1\,1 \\ 5 & = & 0\,0\,1\,0\,1 \\ 3 & = & 0\,0\,0\,1\,1 \\ \hline & & 0\,0\,1\,0\,0 \end{array}$$

En définitive, on trouve avec les deux méthodes la séquence suivante :

$$101010001110000011100_2 = 151C1C_{16}$$

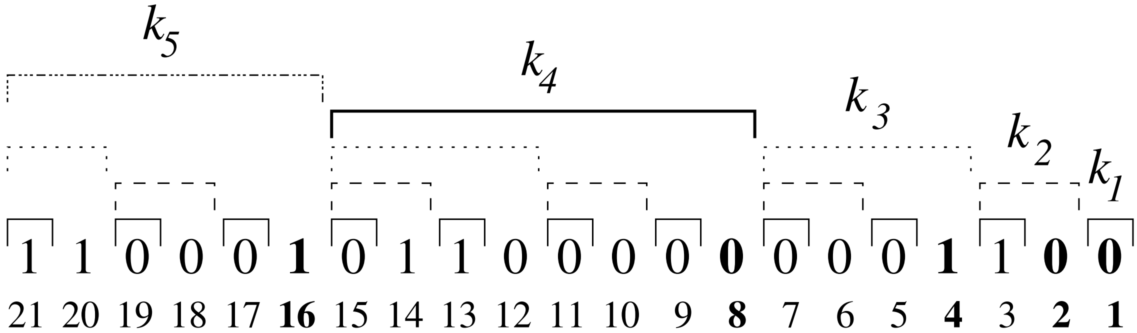
② Vérification du message reçu et extraction des données

— Le message reçu est :

$$6130014_8 = 110001011000000001100_2$$

On a donc $t = 21 \Rightarrow k = 5$ (car $2^k \geq t + 1$) $\Rightarrow n = 16$. La séquence comporte ainsi 5 bits de contrôle en positions 1,2,4,8 et 16. On rappelle que c'est un contrôle de parité impaire qui est utilisé.

— Vérification avec la méthode basique



On recalcule les bits de contrôle et on les compare à leur valeur respective dans la séquence :
- $k_5 = 1$; recalculé $k_5 = 1 \rightarrow$ juste $\rightarrow C_5 = 0$;
- $k_4 = 0$; recalculé $k_4 = 1 \rightarrow$ faux $\rightarrow C_4 = 1$;
- $k_3 = 1$; recalculé $k_3 = 1 \rightarrow$ juste $\rightarrow C_3 = 0$;
- $k_2 = 0$; recalculé $k_2 = 1 \rightarrow$ faux $\rightarrow C_2 = 1$;
- $k_1 = 0$; recalculé $k_1 = 0 \rightarrow$ juste $\rightarrow C_1 = 0$.

Comme $C_5C_4C_3C_2C_1 = 01010_2 = 10$ n'est pas égal à 0, on en déduit qu'il y a une erreur et que c'est le bit en position 10 qui est faux.
Le message corrigé est ainsi :

$$11000101100\mathbf{1}000001100_2$$

et en enlevant les bits de contrôle on obtient les données :

$$1100001100100001_2 = 141441_8$$

— Vérification avec la méthode simplifiée
Le principe est le même que pour le calcul des bits de contrôle de la séquence binaire à transmettre.

$$\begin{array}{rcl}
21 & = & 1\,0\,1\,0\,1 \\
20 & = & 1\,0\,1\,0\,0 \\
16 & = & 1\,0\,0\,0\,0 \\
14 & = & 0\,1\,1\,1\,0 \\
13 & = & 0\,1\,1\,0\,1 \\
4 & = & 0\,0\,1\,0\,0 \\
3 & = & 0\,0\,0\,1\,1 \\
\hline
 & & 0\,1\,0\,1\,0 \\
 & & C_5\,C_4\,C_3\,C_2\,C_1
\end{array}$$

Comme $C_5C_4C_3C_2C_1 \neq 00000$, il y a une erreur dont la position est comme précédemment 10.