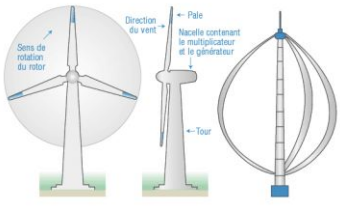
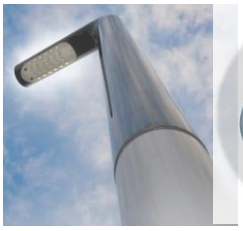
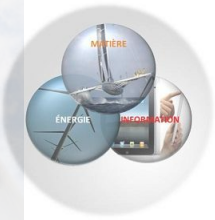






Transmission numérique codes autocorrecteurs

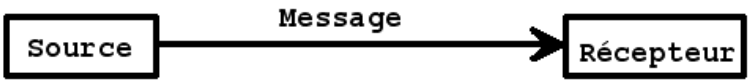
à l'usage des STI2D

1 La transmission de l'information

De nos jours beaucoup d'équipements électroniques échangent des informations.

Ces informations sont transmises d'un dispositif à un autre par différents procédés,

le cas idéal est représenté ci-dessous :



La source émet un message, celui-ci est reçu par le récepteur.

Le cas idéal n'est pas toujours aussi simple, en effet la source et le récepteur peuvent être éloignés l'un de l'autre et la communication doit utiliser des moyens particuliers citons :

- des liaisons filaires sur de longues distances.
- des liaisons radio.

Code de signaux

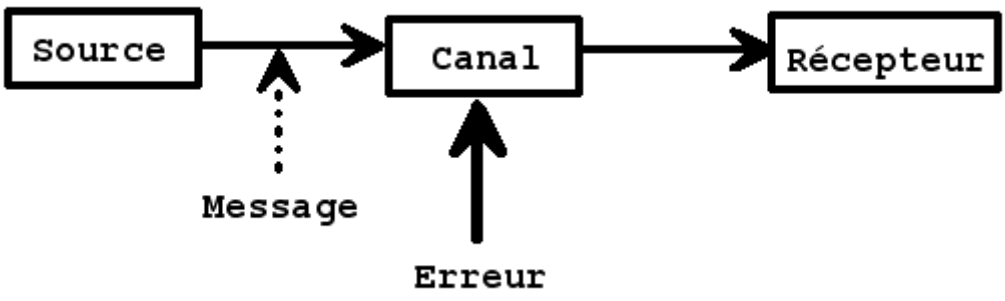
Code barres

Code informatique

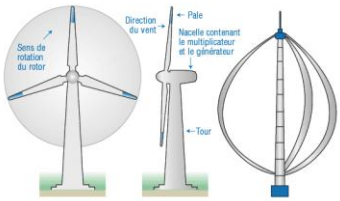
Code postal

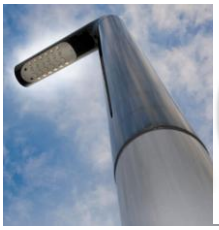
Code génétique

le cheminement du message depuis la source jusqu'au récepteur traverse des milieux physiques particuliers. Ces milieux sont résumés ici par le terme de canal de transmission. Le support du message devra être adapté aux milieux traversés, un courant électrique dans un fil, une onde radio, un son …



Le canal de transmission introduit des perturbations qui peuvent modifier le contenu du message, le message peut alors contenir des erreurs.



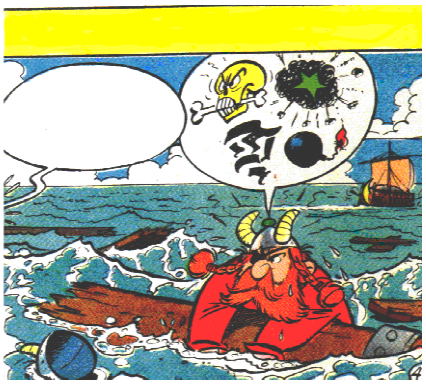


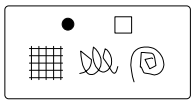
L'objectif du cours est de donner un aperçu des techniques permettant, malgré les erreurs, de retrouver l'intégrité du message envoyé par la source.

2 Coder une information

Pour échanger des informations, nous utilisons des codes. Les codes sont une convention établie entre la source et le destinataire d'un message.

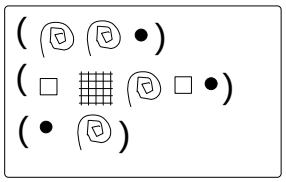
Pour créer un code nous devons définir un alphabet, puis les mots utilisés pour le code :



Alphabet 

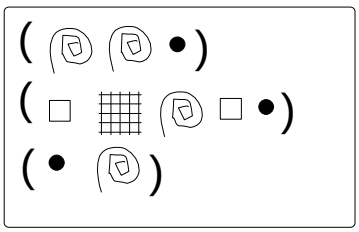
Mot 

Vocabulaire

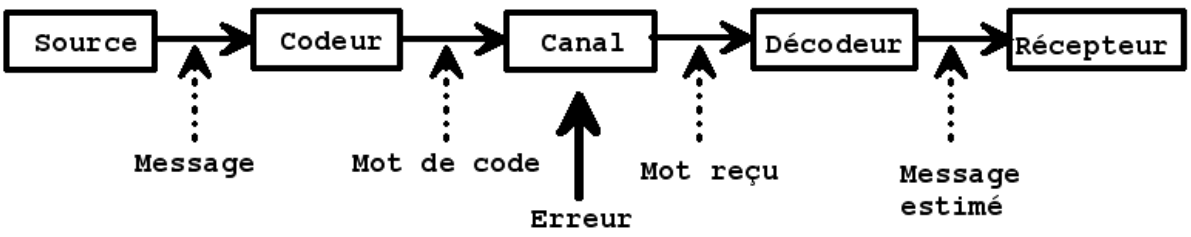


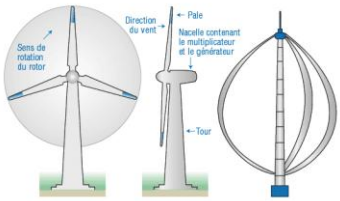
Il suffit ensuite d'établir la correspondance entre les mots du code et les messages à transmettre :

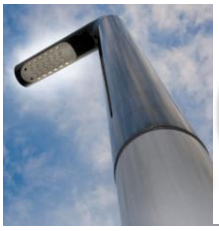
Un code

SOS	
GULPS	
OK	

La transmission du message est alors réalisée selon le schéma ci-dessous :







La source réalise le codage des mots du message, une fois transmis et reçus le récepteur décode ces mots. Le message a pu être perturbé par la traversé du canal de transmission. Il faudra alors être capable de détecter la présence de l'erreur et si possible de la corriger. Nous étudierons cette question avec les codes numériques.

3 Rappel sur les codes numériques

3.1 Introduction.

Les codes peuvent être de nature quelconque, nous étudierons ici les codes numériques binaires. L'information à transmettre est préalablement transformée en une suite de caractère 0 et 1.

Question : quel est l'alphabet d'un tel code ?

3.2 Les codes numériques simples.

Avant d'aborder quelques techniques de codage plus complexes résumons les caractéristiques de quelques codes numériques :

A. Le codage binaire naturel.

cest le codage numérique en base 2.L'expression générale d'un nombre écrit en base B s'écrit :

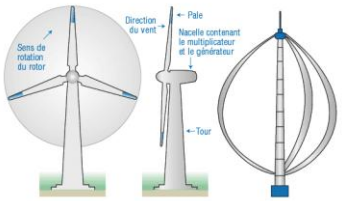
$$(N)_B = a_n.b^n + a_{n-1}.b^{n-1} + \text{------} + a_1.b^1 + a_0.b^0$$

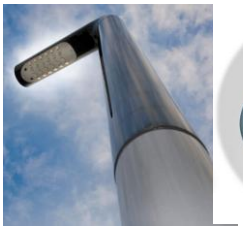
Question : donner les codes binaires naturels des 16 premiers entiers, valeurs comprises entre 0 et 15.

B. Le codage GRAY, exemple d'un code binaire réfléchi.

Codage Gray

0	0 0 0 0	8	1 1 0 0
1	0 0 0 1	9	1 1 0 1
2	0 0 1 1	1 0	1 1 1 1
3	0 0 1 0	1 1	1 1 1 0
4	0 1 1 0	1 2	1 0 1 0
5	0 1 1 1	1 3	1 0 1 1
6	0 1 0 1	1 4	1 0 0 1
7	0 1 0 0	1 5	1 0 0 0





Le code est construit de manière à ce que deux nombres successifs ne diffèrent que par une position, ***code continu au sens large***. Ceci est vrai aussi pour la première et la dernière valeur du code, ***code continu au sens strict***.

C. Le codage des nombres avec le code Décimal Codé Binaire.

Le codage DCB est un code où chaque nombre à coder **est codé chiffre par chiffre**. Ce codage est rapide puisqu'il n'y a aucun calcul à faire simplement un remplacement du chiffre par son codage binaire.

```
Exemple : 102 codé en DCB et code binaire naturel donne :

102 (10) => 0001 0000 0010  (DCB binaire naturel)
```

Le code Aiken permet de connaître le complément à 9 d'un nombre simplement en inversant tous les bits du nombre.

Le code 2 parmi 5 augmente la sécurité du code si par hasard un bit du mot reçu est inversé le mot résultant n'est plus un mot du code ceci permet de détecter la présence de l'erreur.

	Aiken	2 parmi 5
0	0 0 0 0	1 1 0 0 0
1	0 0 0 1	0 0 0 1 1
2	0 0 1 0	0 0 1 0 1
3	0 0 1 1	0 0 1 1 0
4	0 1 0 0	0 1 0 0 1
5	1 0 1 1	0 1 0 1 0
6	1 1 0 0	0 1 1 0 0
7	1 1 0 1	1 0 0 0 1
8	1 1 1 0	1 0 0 1 0
9	1 1 1 1	1 0 1 0 0

3.3 Les codes autocorrecteurs.

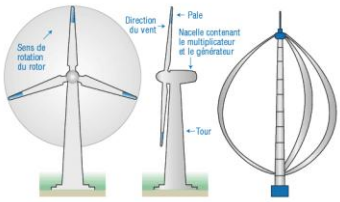
Dans ces codes on ajoute un moyen de détecter si le contenu binaire est corrompu . Deux grandes possibilités technologiques existent :

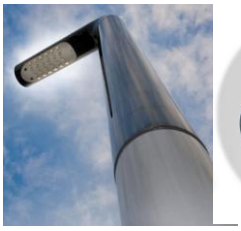
- détection globale de la présence d'une erreur, si l'erreur est détectée alors le message entier est rejeté.
- détection précise de l'erreur et donc dans ce cas il suffit d'inverser le bit faux pour qu'il devienne vrai.

A. Les bits de contrôle (détection globale).

Ajouter des bits de contrôles consiste à ajouter des éléments supplémentaires dans le codage du nombre. Ces éléments ne transportent aucune information, ils permettent simplement de donner au codage des performances de détection et éventuellement de correction des erreurs.

Voici un principe fondamental du codage, plus la sécurité d'utilisation du code devra être grande plus le code sera sophistiqué.





B. Le bit de parité (détection globale).

Le contrôle le plus simple consiste à ajouter un bit de parité au nombre binaire. Ceci de manière à ajuster la parité globale du mot complet à une valeur convenue à l'avance. Nous voulons transmettre le nombre binaire 0011 avec une parité impaire, nous ajoutons le bit de parité 1 pour donner le mot complet 00111.

Question : coder de la même manière les mots 0101, 1011

4 Exercice sur le codage

4.1 Codage DCB

Le code DCB sert à coder des nombres en codant chaque chiffre du nombre séparemment, sans aucun calcul. Ceci est obtenu en remplaçant chaque chiffre par sa représentation binaire. Si la représentation binaire de cahque chiffre correspond au binaire naturel on appelle ce codage Décimal Codé Binaire. D'autres représentation sont possibles selon des besoins particuliers comme les codes Gray, Aiken, Excès3.

Exemple : le nombre 245 est codé 0010 0100 0101 car 0010 est le code du chiffre 2, 0100 est le code du chiffre 4 et 0101 est le code du chiffre 5.

a) Coder en code DCB les nombres suivants : 1023 925 84

b) Coder 896 en DCB avec les codages : 2 parmi 5, gray, aiken.

4.2 Contrôle de parité

Nous étudions içi le prinçipe du contrôle de parité. Ce contrôle permet la détection d'une erreur dans une transmission d'une suite de bits. Pour cela un bit supplémentaire appelé ***bit de parité*** est adjoint au groupe de bits à transmettre. Ce bit est calculé de manière a fixer la parité globale du mot émis.

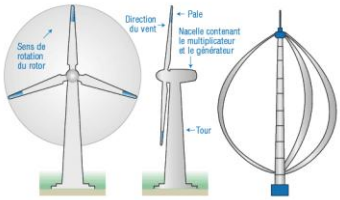
-> On parlera de Parité si le mot émis est pair.

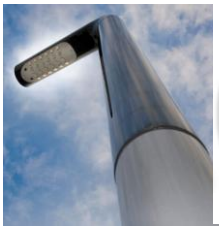
Cela signifie que le nombre total de bits à 1 dans le mot est pair.(0,2,4,6,...), EVEN en anglais.

-> On parlera d'Imparité si le mot émis est impair.

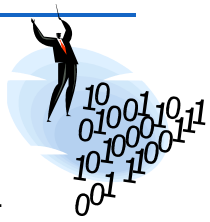
Cela signifie que le nombre total de bits à 1 dans le mot est impair.(1,3,5,7,...), ODD en anglais.

En supposant l'émission de mots de quatre bits calculer le bit de contrôle d'imparité appelé I.





Code auto correcteur de Hamming



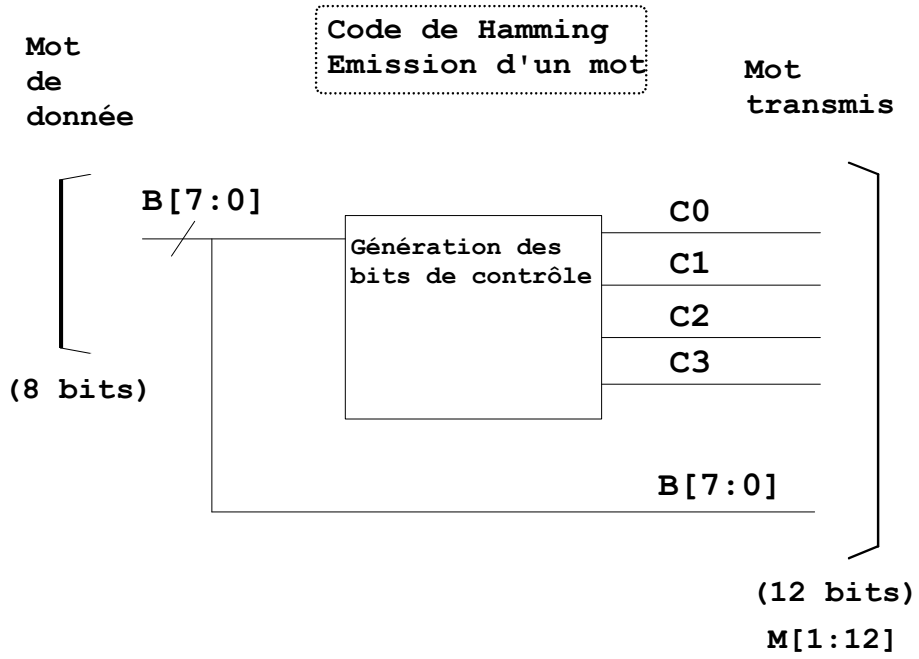
1 Présentation du code autocorrecteur de Hamming.

Le codage de Hamming est un code auto correcteur. Cela signifie qu'il est capable de procéder à la détection puis à la correction d'une erreur dans un mot de plusieurs bits.

Selon le nombre de bits d'informations à transmettre il faut ajouter un certain nombre de bits dits bits de contrôle. Ces bits permettront au décodeur de vérifier si il y a eu une erreur et de corriger celle-ci.

2 Code de Hamming, émission du message.

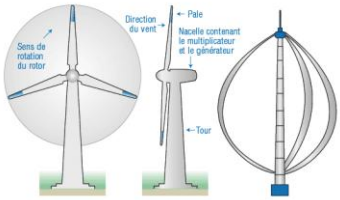
Nous considérons que l'on souhaite détecter et corriger une erreur dans un mot d'information transmis de huit bits de long. Le travail du codeur est résumé dans le schéma ci-dessous :

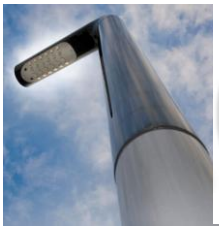


Nous voyons que pour un mot de donnée de huit bits de long nous ajoutons quatre bits de contrôle C0,_,C3. La structure d'un mot du message est donc la suivante:

Organisation d'un mot de message

M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1
B7	B6	B5	B4	C3	B3	B2	B1	C2	B0	C1	C0





Observons que l'ordre dans lequel les bits sont placés n'est pas indifférent. Nous verrons l'utilité de cela lors de l'étude du décodeur.

Calcul des bits de contrôle : le calcul est fait selon les indications du schéma ci-dessous. L'équation de C3 est donnée en exemple.

Calcul des bits de contrôles

	B7	B6	B5	B4	B3	B2	B1	B0	C3	C2	C1	C0
S3	X	X	X	X					X			
S2	X				X	X	X			X		
S1		X	X		X	X		X			X	
S0		X		X	X		X	X				X

$$C3 = B7 \oplus B6 \oplus B5 \oplus B4$$

2.1 Donner les équations des bits de contrôle C0,C1,C2.

La fonction $\oplus$ représente le ou exclusif.

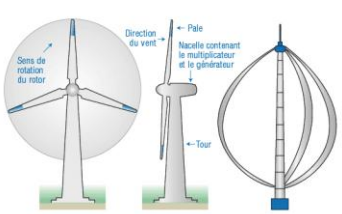
3 Code de Hamming, réception du message.

Le travail du décodeur consiste tout d'abord à détecter si il y a une erreur dans le mot transmis. Pour cela on calcule des bits appelés bits de syndromes. La construction de ces bits de syndromes a comme propriété de donner le numéro du bit du message faux :

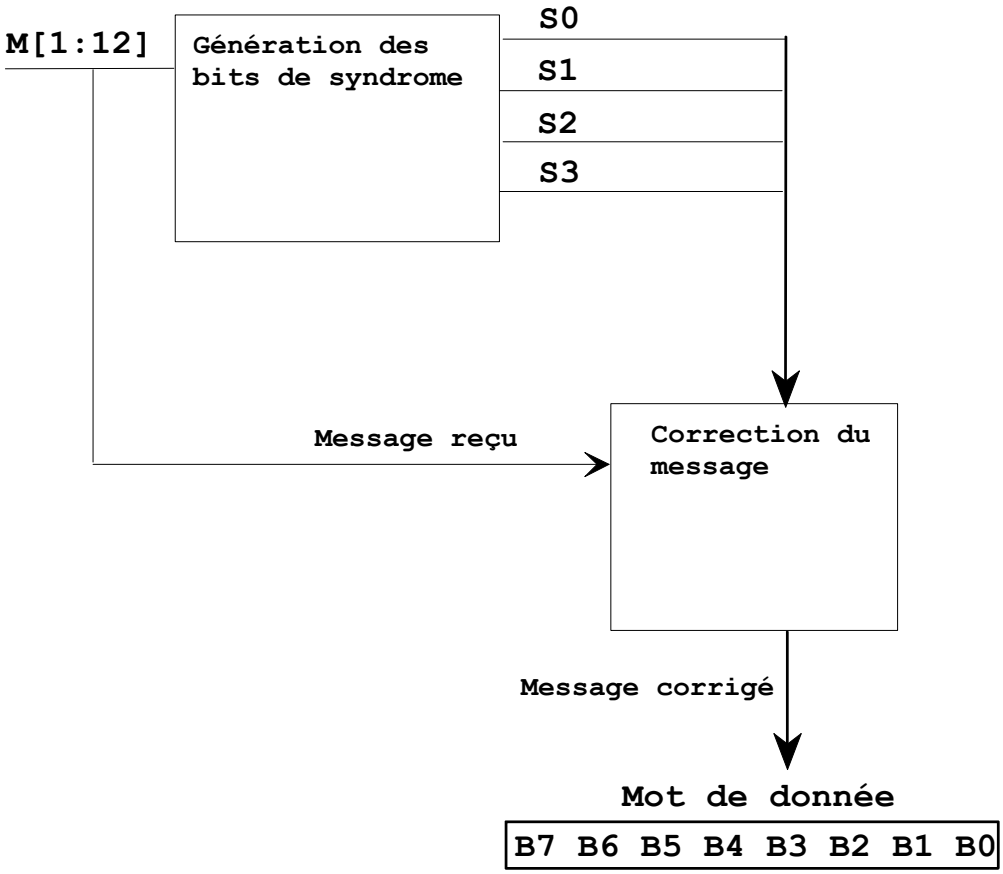
Si on calcule S3 S2 S1 S0 = 1 0 0 1 alors c'est le bit M9 qui est faux car $1001_2=9_{10}$

Si il n'y a pas d'erreur dans le mot reçu alors: S3 S2 S1 S0 = 0 0 0 0.

Les bits de syndrome permettent donc de déterminer si il y a une erreur dans le message reçu ou non. L'erreur si elle a lieu est corrigée dans le bloc correction du message.



Code de Hamming
Réception d'un mot



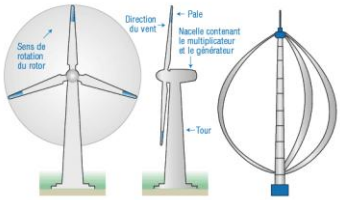
Calcul des bits de syndromes, le calcul des bits de syndromes a lieu selon le tableau ci-dessous :

Calcul des bits de syndrome

	B7	B6	B5	B4	B3	B2	B1	B0	C3	C2	C1	C0
S3	X	X	X	X					X			
S2	X				X	X	X			X		
S1		X	X		X	X		X			X	
S0		X		X	X		X	X				X

$$S3 = B7 \oplus B6 \oplus B5 \oplus B4 \oplus C3$$

3.1 Donner les équations des bits de syndrome S2 S1 S0.



Une fois les bits de syndrome calculés il faut procéder à la correction du message. Pour cela on détermine si le bit étudié est faux. Un exemple de calcul est donné pour le bit de message M3 correspondant à B0.

$$\text{Bit M3 faux} = \overline{S3} * \overline{S2} * S1 * S0$$

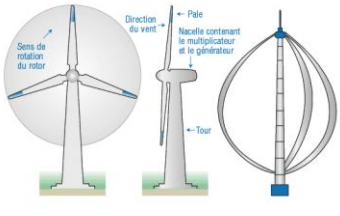
Si l'équation ci-dessus est vraie cela signifie que le bit de message doit être corrigé. On appellera le résultat de ces équations : bit de correction.

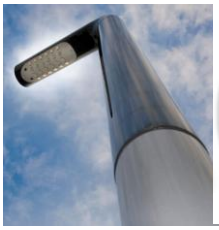
3.2 Donner les équations des bits de correction pour les bits du message correspondant aux bits de données B1 à B7.

Une fois déterminé la valeur du bit de correction on peut procéder à l'équation finale corrigeant le bit du message uniquement si cela s'avère nécéssaire.

La correction d'un bit consiste en son inversion car il n'y a que deux valeurs possibles 0 ou 1.

3.3 Quelle est la fonction logique qui permet d'inverser un bit uniquement si cela doit être réalisé. (bit de correction à 1)





Exemples de codage et décodage avec Hamming

Mise en œuvre du code de Hamming.

Pour faire ce travail il faut utiliser les résultats du paragraphe précédent.

Emission d'un message codage du mot en ajoutant les quatre bits de contrôle

Q1 => Compléter le tableau ci-dessus en calculant les bits de contrôle.

Emission du message : codage du mot.

M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1
B7	B6	B5	B4	C3	B3	B2	B1	C2	B0	C1	C0
0	0	1	0		1	1	0		0		
1	0	1	0		1	0	0		1		
0	0	0	1		0	1	1		1		

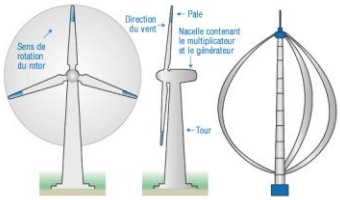
Réception du message, il faut déterminer si le message est juste. Dans le cas contraire le corriger.

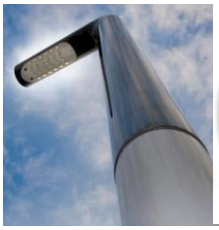
Q2 => Compléter le tableau ci-dessus indiquer la présence d'une erreur, corriger le message si-besoin..

Réception du message : test et correction du mot

M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1				
B7	B6	B5	B4	C3	B3	B2	B1	C2	B0	C1	C0	S3	S2	S1	S0
0	0	1	0	1	1	1	0	0	0	1	1				
0	0	1	0	0	1	0	0	0	1	1	0				
0	0	0	1	0	0	1	1	0	1	0	1				
1	0	1	0	0	1	0	0	0	0	1	0				

Q3 => Cas de deux erreurs, étudier le cas où il y a deux erreurs dans le mot émis. Que concluez vous ?





Fiche résumé codes et codes auto correcteurs

Codes DCB

Codage Gray

0	0 0 0 0	8	1 1 0 0
1	0 0 0 1	9	1 1 0 1
2	0 0 1 1	10	1 1 1 1
3	0 0 1 0	11	1 1 1 0
4	0 1 1 0	12	1 0 1 0
5	0 1 1 1	13	1 0 1 1
6	0 1 0 1	14	1 0 0 1
7	0 1 0 0	15	1 0 0 0

Aiken

0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	1 0 1 1
6	1 1 0 0
7	1 1 0 1
8	1 1 1 0
9	1 1 1 1

2 parmi 5

0	1 1 0 0 0
1	0 0 0 1 1
2	0 0 1 0 1
3	0 0 1 1 0
4	0 1 0 0 1
5	0 1 0 1 0
6	0 1 1 0 0
7	1 0 0 0 1
8	1 0 0 1 0
9	1 0 1 0 0

Codage de Hamming

Organisation d'un mot de message

M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1
B7	B6	B5	B4	C3	B3	B2	B1	C2	B0	C1	C0

Calcul des bits de contrôles

	B7	B6	B5	B4	B3	B2	B1	B0	C3	C2	C1	C0
S3	X	X	X	X					X			
S2	X				X	X	X			X		
S1		X	X		X	X		X			X	
S0		X		X	X		X	X				X

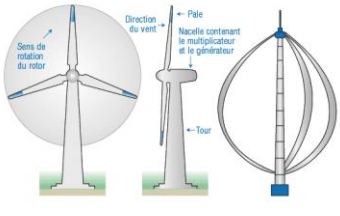
$$C3 = B7 \oplus B6 \oplus B5 \oplus B4$$

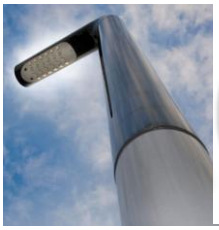
Calcul des bits de syndrome

	B7	B6	B5	B4	B3	B2	B1	B0	C3	C2	C1	C0
S3	X	X	X	X					X			
S2	X				X	X	X			X		
S1		X	X		X	X		X			X	
S0		X		X	X		X	X				X

$$S3 = B7 \oplus B6 \oplus B5 \oplus B4 \oplus C3$$

Les bits S3.S2.S1.S0 donnent la position de l'erreur si ≠ de 0000.





Synthèse d'un codeur-décodeur de Hamming

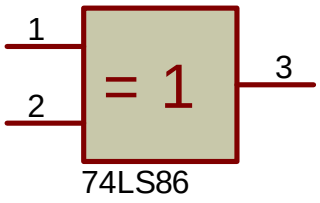
Utilisation d'un logiciel de simulation pour la mise au point d'un codeur décodeur de Hamming.

1 Synthèse du codage

xor = ou exclusif

Les équations du codeur sont rapellées ci-dessous :

```
C0 <= B6 xor B4 xor B3 xor B1 xor B0;
C1 <= B6 xor B5 xor B3 xor B2 xor B0;
C2 <= B7 xor B3 xor B2 xor B1;
C3 <= B7 xor B6 xor B5 xor B4;
```



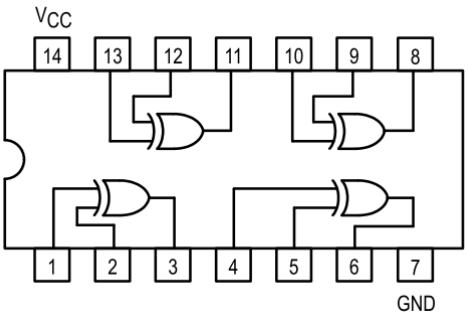
TRUTH TABLE

IN		OUT
A	B	Z
L	L	L
L	H	H
H	L	H
H	H	L

Réaliser la synthèse de ces équations avec le logiciel de simulation proteus Isis.

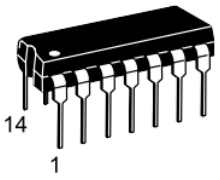
Vérifier le bon fonctionnement sur les exemples ci-dessous :

M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1
B7	B6	B5	B4	C3	B3	B2	B1	C2	B0	C1	C0
1	1	1	1	0	1	1	1	0	1	1	1
0	0	1	1	0	1	0	0	1	1	1	1
0	0	1	0	1	0	0	0	0	1	0	1



2 Synthèse du décodage

```
S0 <= M11 xor M9 xor M7 xor M5 xor M3 xor M1;
S1 <= M11 xor M10 xor M7 xor M6 xor M3 xor M2;
S2 <= M12 xor M7 xor M6 xor M5 xor M4;
S3 <= M12 xor M11 xor M10 xor M9 xor M8;
```



Réaliser la synthèse de ces équations avec le logiciel de simulation proteus Isis.

Vérifier le bon fonctionnement sur les exemples ci-dessous :

M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1				
B7	B6	B5	B4	C3	B3	B2	B1	C2	B0	C1	C0	S3	S2	S1	S0
1	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0
1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1
1	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0