

Octobre 2018 – Décembre 2018



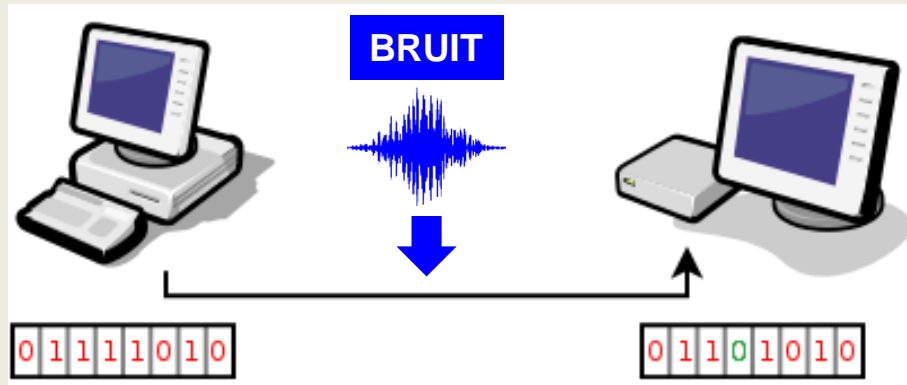
Projet **P**y**T**hon

Simulation de la transmission d'une
image et correction d'erreurs

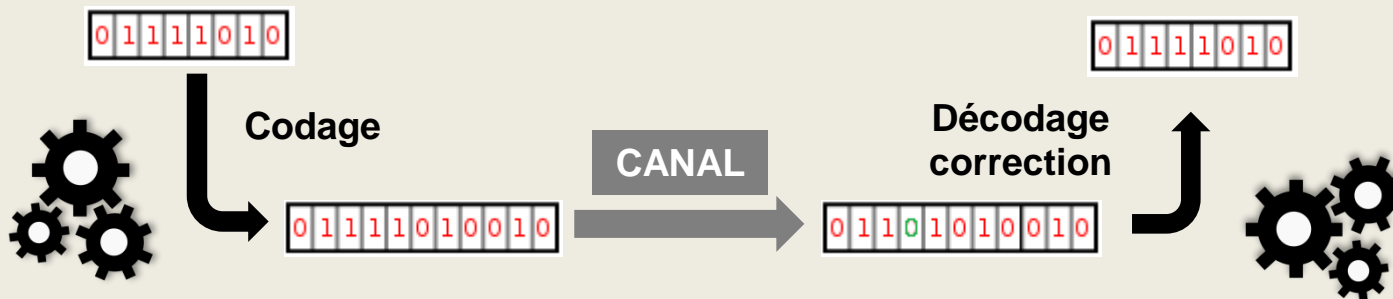
Objectifs



- **A. Simuler un canal de transmission-réception d'un signal binaire avec bruit aléatoire**



- **B. Utiliser un code correcteur d'erreurs lors de cette transmission afin de corriger le signal reçu**



Organisation du projet



Travail en équipe de 3 élèves

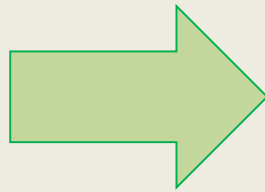


Chef de projet

*Prépare la
synthèse...*



Supervise, aide...



Programmeur 1 :

A. Simulation du canal de transmission



Échanges
de données



Programmeur 2 :

**B. Programmation du code correcteur :
encodage de l'image et décodage**

✓ *Durée du projet : (3 + 1) séances de 2h*

✓ *Dernière séance : présentation et démonstration aux enseignants
des programmes (8-10 min)*

✓ *Envoi des programmes avec instructions aux professeurs pour tests
(1 semaine avant)*

A. Simulation du canal de transmission



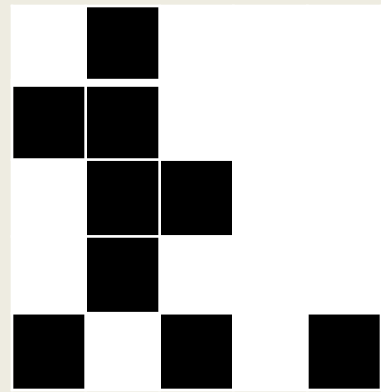
- Le but est de simuler la « couche physique » d'une transmission de données (électrique filaire, sans fil, optique...)
- La « couche physique » comprend :
 - la transformation des bits d'information à transmettre en signaux physique pouvant transporter cette information.
 - Le passage de ce signal dans le canal de transmission entre l'émetteur du signal et le récepteur.
 - La récupération de bits d'information à partir du signal physique reçu.
- Le canal physique de transmission peut être modéliser par :
 - un coefficient multiplicatif constant (si le canal ne change pas lors de la transmission). Il représente l'atténuation de l'amplitude du signal lors de sa propagation.
 - Un terme de bruit dû à une pollution extérieure sur le canal. Ce bruit s'ajoute au signal utile reçu. Il est aléatoire et est différent pour chaque échantillon du signal reçu. Il est généralement modélisé par un « bruit additif blanc gaussien ».
 - Lors de la simulation le coefficient de canal et le bruit simulés ne peuvent pas être connus de l'émetteur ou du recepneur.

A. Simulation du canal de transmission

Emission du signal



Image Noir & Blanc
à transmettre

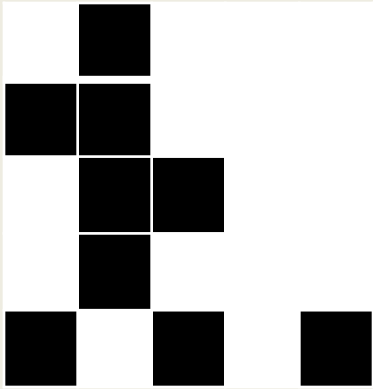


Information
numérique
binaire

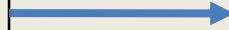
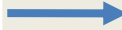


Emission du signal

Image Noir & Blanc
à transmettre



Information
numérique
binaire



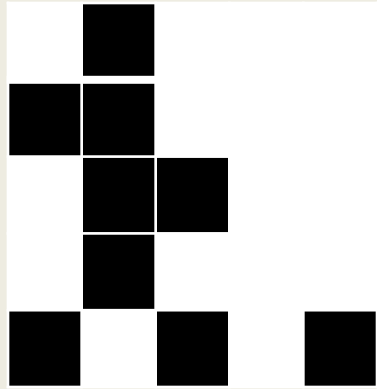
Signal
physique
émis

Information
analogique
binaire

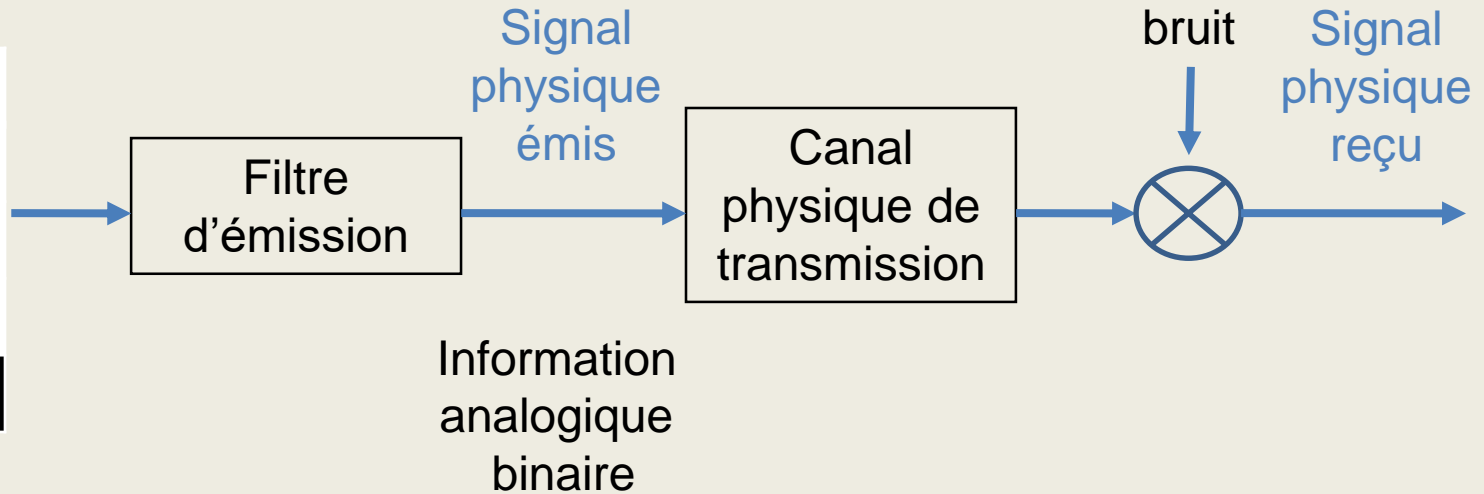


Emission du signal

Image Noir & Blanc
à transmettre



Information
numérique
binaire





Réception du signal

Signal
physique
reçu

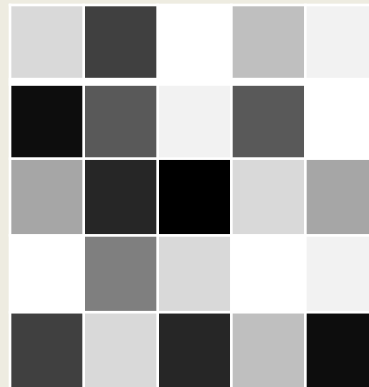
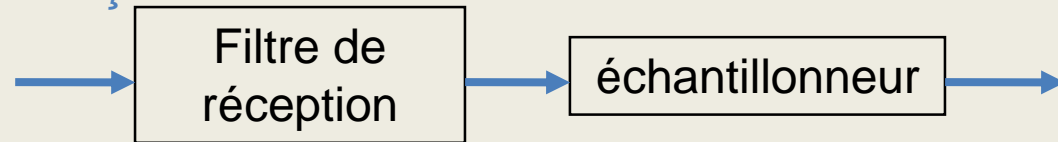
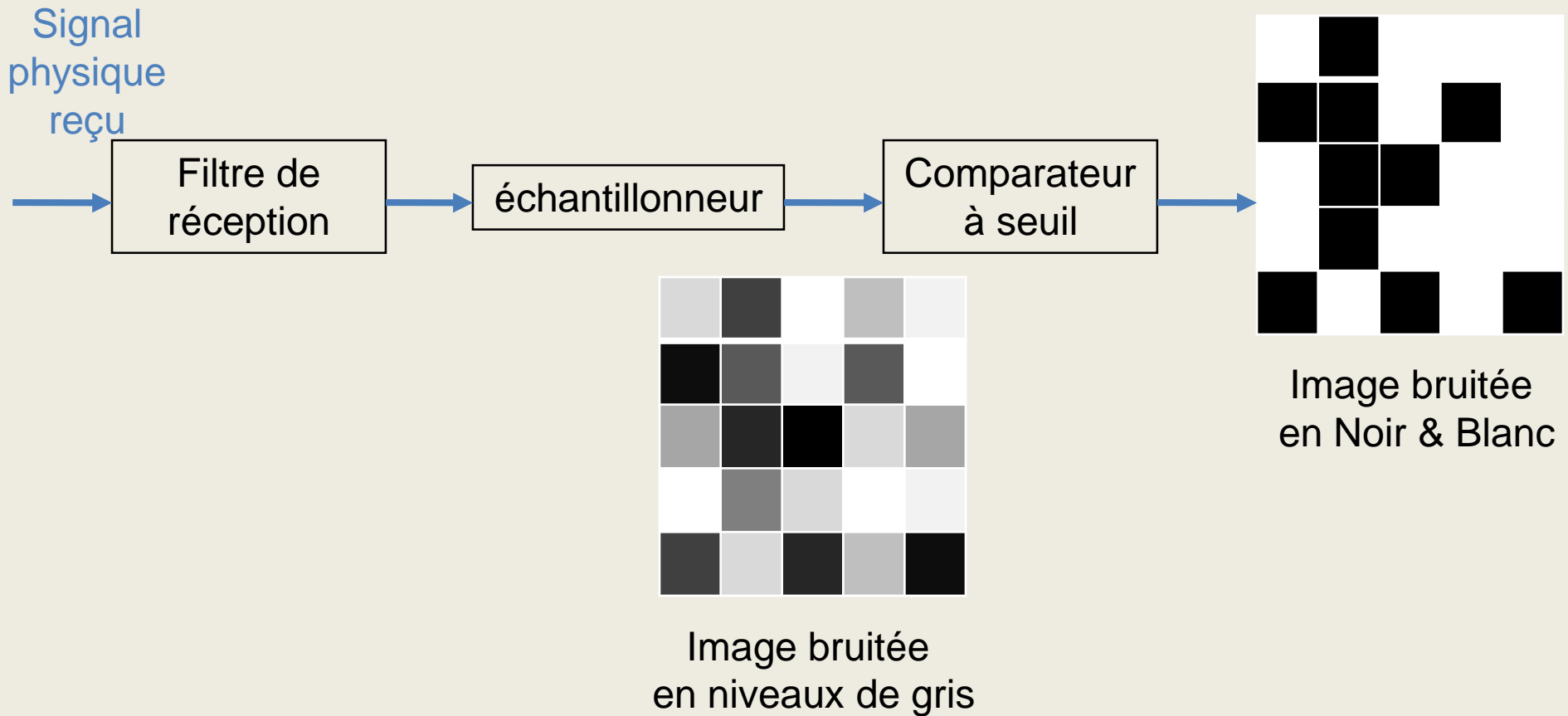


Image bruitée
en niveaux de gris



Réception du signal





Simulation du canal

Image Noir & Blanc
à transmettre

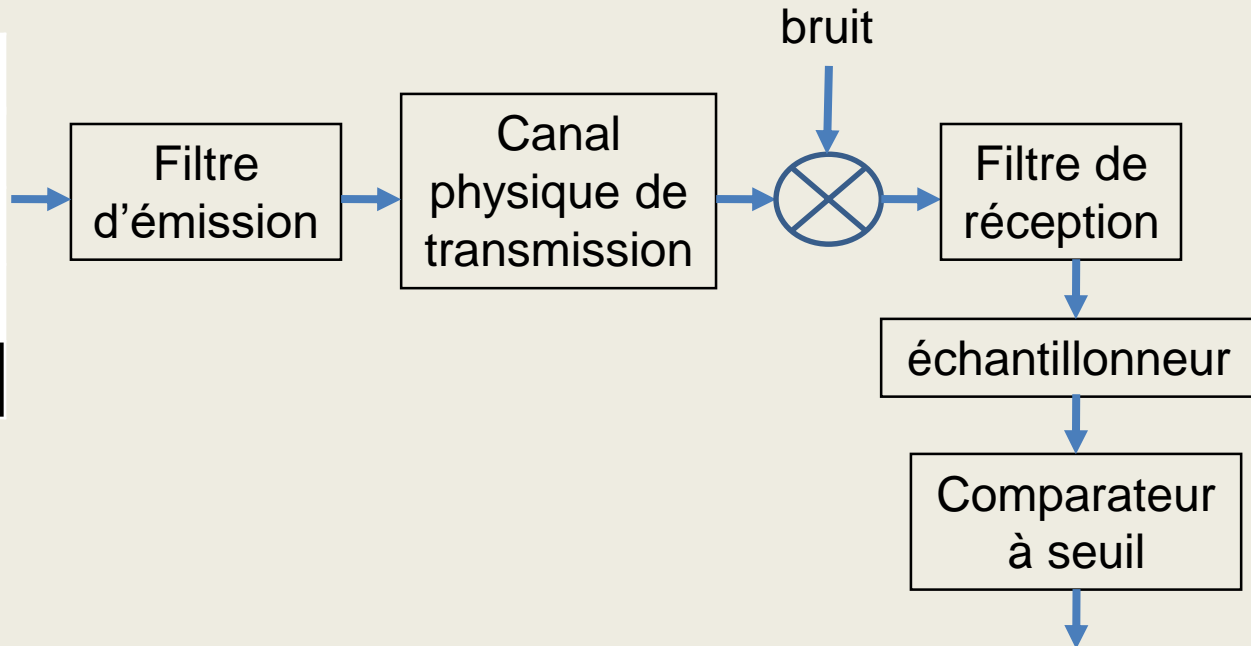
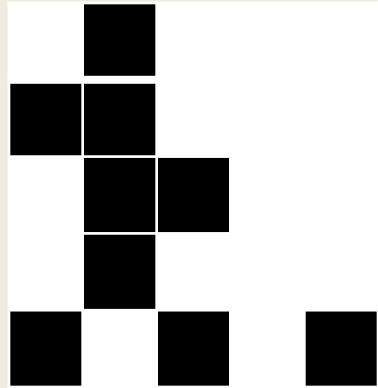
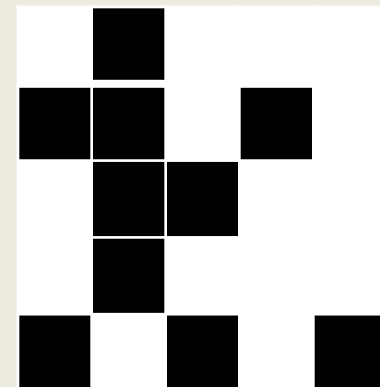


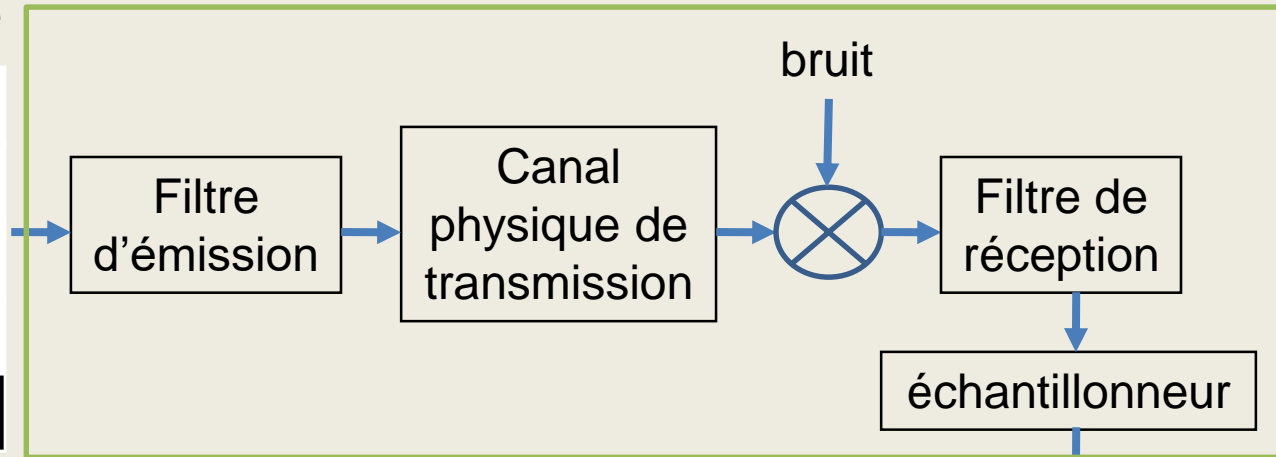
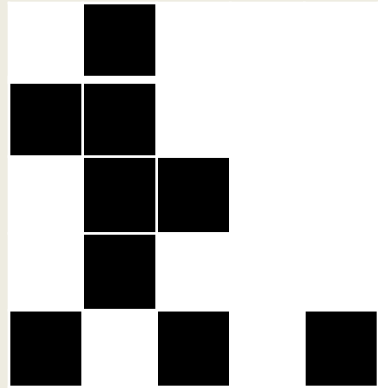
Image Noir & Blanc
reçue





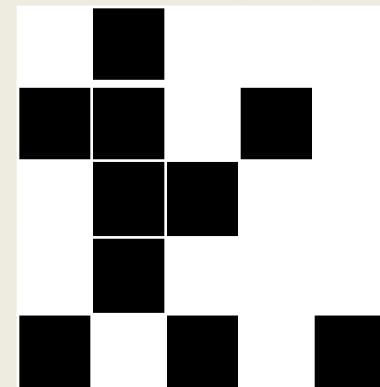
Simulation du canal

Image Noir & Blanc
à transmettre



Simulation de la transmission des
Signaux physiques

Image Noir & Blanc
reçue





B. Codage de l'information

1. Principe général d'un code correcteur :

- ❑ Chaque suite de bits (appelée *trame*) à transmettre **est augmentée** par une autre suite de bits dite de **redondance** ou de **contrôle**.
- ❑ Pour chaque suite transmise de **k** bits, on ajoute **r** bits. On dit alors que l'on utilise un code **C(n,k)** avec **n = k + r**.
- ❑ À la réception, on effectue l'**opération inverse** et les bits ajoutés permettent d'effectuer des **contrôles** à l'arrivée.



Code proposé : Code de Hamming (7,4)



2. Code de Hamming (7,4) (d'après wikipedia) :

- ❑ pour 4 bits de données d_i on utilise 3 bits de contrôle p_i
- ❑ les bits de contrôle p_i sont en position 2^{i-1} avec $i=1,2,\dots$
- ❑ le mot binaire envoyé a ainsi la structure suivante :

	p_1	p_2	d_1	p_3	d_2	d_3	d_4
Position :	1	2	3	4	5	6	7

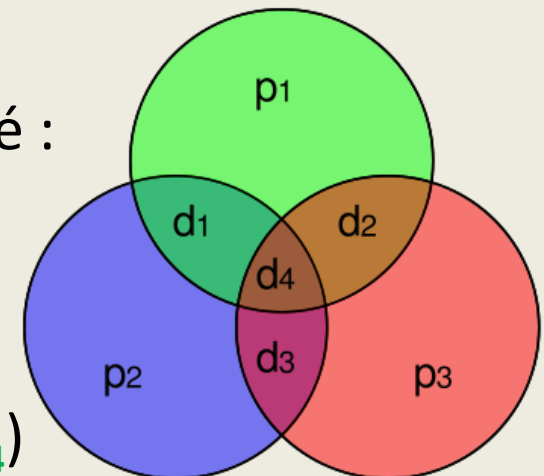
- ❑ les bits p_i sont des bits de contrôle de parité :

- p_1 = parité du cercle vert,
si $d_1 + d_2 + d_4$ est pair $p_1=0$, sinon $p_1=1$

de même :

- p_2 = parité du cercle mauve sur (d_1, d_3, d_4)
- p_3 = parité du cercle rouge sur (d_2, d_3, d_4)

Illustration



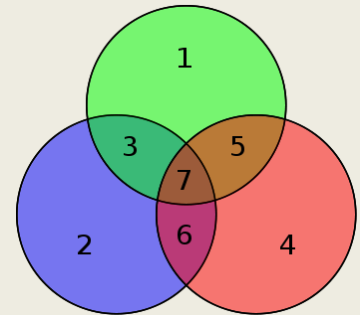


3. Exemple de construction du code de Hamming (7,4) :

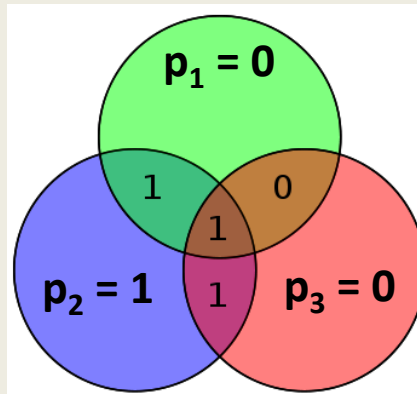
❑ soit 4 bits à transmettre (**1011**)

❑ le mot binaire envoyé a la structure suivante :

p_1 p_2 1 p_3 0 1 1
Position : **1 2 3 4 5 6 7**



❑ plaçons ces bits dans les cercles précédents à leurs positions respectives :

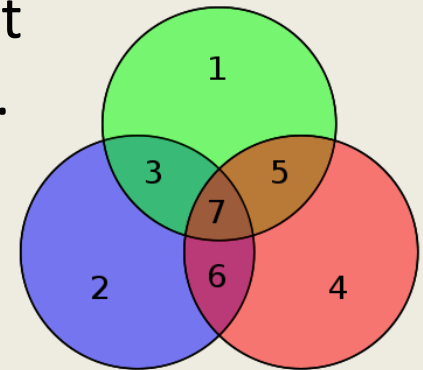
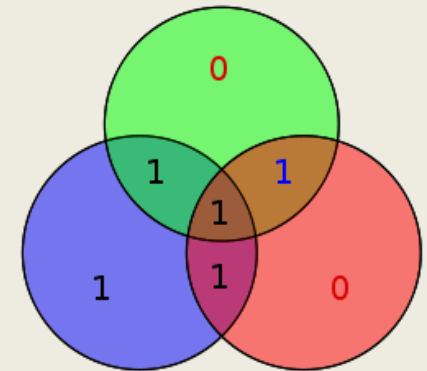


❑ mot transmettre : **0 1 1 0 0 1 1**



4. Détection & correction d'une erreur :

- ❑ Ce code permet la **détection** et la **correction** d'**UNE** erreur.
- ❑ Supposons le bit d_2 altéré lors du transfert : le message reçu n'est plus **0110011** mais **0110111**.
- ❑ 2 conditions, correspondant aux cercles rouge et vert, ne sont plus remplies.
- ❑ L'analyse des positions des bits dans les cercles implique que seul le bit en position 5 est altéré car la parité du cercle mauve est vérifiée.
- ❑ **Conclusion** : l'erreur est détectée et peut ainsi être corrigée.



***Remarques** : si l'erreur se produit sur le bit 7 toutes les parités sont modifiées
si l'erreur se produit sur un bit de parité, seul un cercle est modifié...*

5. Mise en œuvre du code de Hamming (7,4) (d'après wikipedia) :



•Étape 1 : Construction d'une **matrice génératrice notée G** obtenue à partir d'une base canonique de signaux de 4 bits à transmettre :

d1 = 1000, **d2 = 0100**, **d3 = 0010** et **d4 = 0001**

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

•Étape 2 : Pour tout **message D** de 4 bits on applique **G** pour obtenir le **message C** de 7 bits à transmettre

$$D = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad G \cdot D = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad \text{et} \quad C = G \cdot D = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

•**Attention** : la somme utilisée dans ces produits de matrice est un « ou exclusif » c'est-à-dire :

1 + 0 = 1 **1 + 1 = 0** et donc **1 + 1 + 1 = 1**

5. Mise en œuvre du code de Hamming (7,4) (d'après wikipedia) :



•**Étape 3** : Pour le décodage du message reçu on construit une **matrice de contrôle notée H** : elle correspond en fait au code binaire de 1 à 7 où le bit de poids fort est en haut

$$\mathbf{H} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{matrix}$$

Ainsi pour tout **message reçu C** de 7 bits on applique **H** pour obtenir un vecteur de 3 bits nuls si le message est sans erreur...

$$\mathbf{H} \cdot \mathbf{C} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

5. Mise en œuvre du code de Hamming (7,4) (d'après wikipedia) :



•**Étape 4** : Pour tout message reçu C de 7 bits on applique H pour obtenir un vecteur de 3 bits

Supposons que le message reçu n'est plus $C = (0110011)$ mais $X = (0110\mathbf{1}11)$

$$H \cdot X = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

L'erreur est détectée, $H \cdot X$ est un vecteur non nul, correspondant à la valeur $\mathbf{101}_2$ en binaire, soit **5**. La valeur de $H \cdot X$ est appelée **syndrome**.

•**Étape 5** : Un mot correcteur E_5 composé de 6 bits à 0 et du **5^{ème} bit** à 1 permet par addition au vecteur X (« ou exclusif ») de corriger l'erreur

$$C = X + E_5 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{1} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

•**Étape 6** : Après correction il ne reste plus qu'à récupérer le message D initial

6. Amélioration du rendement du codage (partie facultative)



- ❑ Le **rendement** d'un code correspond au rapport :
nombre de bits de données sur nombre de bits envoyés
- ❑ Ainsi le code de **Hamming (7,4)** a un rendement estimé à $4/7=57\%$
- ❑ Le code de **Hamming (15,11)** présente un rendement de $11/15=73\%$
- ❑ Il est construit de façon analogue au code (7,4) :
 - Il utilise 11 bits de données et 4 bits de contrôle
 - ses 4 bits de parité sont en position : 1, 2, 4, 8
 - Le bit 1 contrôle la parité des positions 3, 5, 7, 9, 11, 13 et 15.
 - Le bit 2 contrôle la parité des positions 3, 6, 7, 10, 11, 14 et 15.
 - Le bit 4 contrôle la parité des positions 5, 6, 7, 12, 13, 14 et 15.
 - Le bit 8 contrôle la parité des positions 9, 10, 11, 12, 13, 14 et 15.
- ❑ Programmer ce code & le confronter au précédent

Evaluation



Programme		Barème	Remarques
	tourne sur une matrice créée	0,5	
	tourne sur une image (fournie)	1,5	
	tourne sur d'autres images fournies	1	<i>taille de l'image dans le préambule</i>
	compréhensible avec commentaires détaillés	1	
	facilité d'utilisation, ergonomie	1	<i>prg tourne "tout seul"</i>
Respect du cahier des charges			
Canal physique	bits transmis 1 à 1 (liaison série)	1	
	coefficient de canal et seuillage	1	
	bruit additif blanc gaussien	1	
	apprentissage et détermination du seuil	1	
Code de Hamming	fonction codage hamming	1	
	codage de la trame	1	
	fonction décodage Hamming + correction erreur	1	
	decodage trame avec correction erreur	1	
Organisation du travail d'équipe		1	
BONUS : projet dépassant le cahier des charges		2	<i>Hamming 15/11 ou tout type d'images, interface graphique...</i>
Présentation			
Oral 10min/groupe	pertinence, intérêt...	1	
	clarté, gestion du temps	1	
	visualisations de l'erreur due au canal	1	
	visualisation du canal, de la correction d'erreurs	1	
Respect des délais		2	
date d'envoi des fichiers			