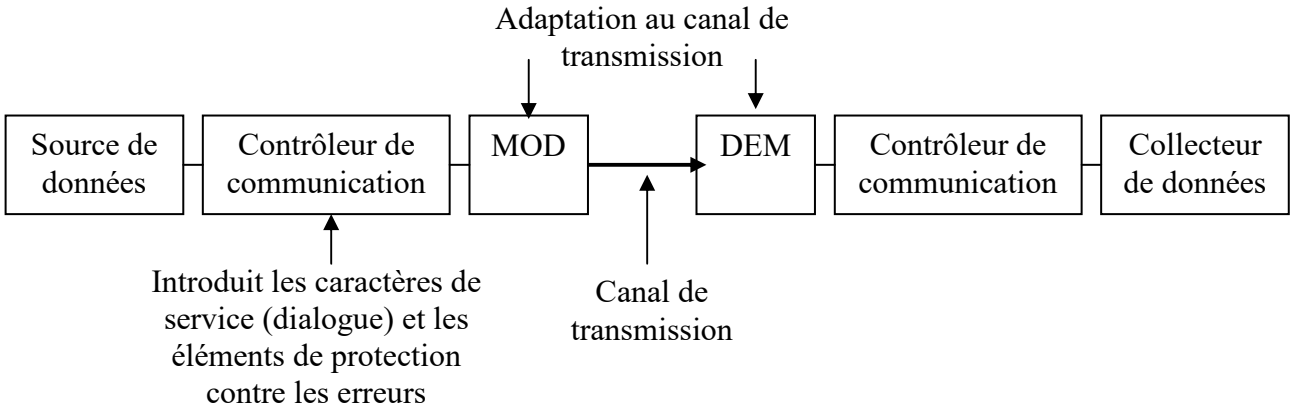
I) Généralités

I.1) Schéma fonctionnel

Pour être acheminée, toute information doit être mise sous la forme d'une suite structurée de symboles. La signification de ces symboles est une affaire de pure convention et est définie par une loi de correspondance.



La protection contre les erreurs de transmission est réalisée grâce à un "surcodage" de chaque symbole de l'information à transmettre. A la réception, le contrôleur de communication peut, suivant les performances de ce "surcodage", détecter et corriger la majorité des erreurs dues aux imperfections du canal de transmission.

L'objet de ce document est l'étude de ce "surcodage" qu'on appellera simplement **codage** par la suite.

Les techniques de contrôle des erreurs sont :

- ARQ : Automatic Repeat Request. Nécessite un canal de retour.
  - Stop & Wait ARQ : le transmetteur envoie un symbole et attend un acquittement positif (ACK) ou négatif (NAK) du récepteur. Dans le premier cas, le transmetteur poursuit la transmission, et dans le second cas, il réémet le même symbole jusqu'à recevoir un ACK.
  - Continous ARQ : l'algorithme est identique mais le transmetteur n'attend pas ce qui permet de gagner du temps. Les contrôleurs de communication sont, par contre, plus complexes.
- FEC : Forward Error Control. La détection et la correction des erreurs sont réalisées à la réception en utilisant les éléments de protection introduits à l'émission. Cette technique de protection est utilisée s'il n'y a pas de canal de retour.

I.2) Les erreurs de transmission

Les imperfections du canal de transmission entraînent des perturbations aléatoires :

- disparition de bit
- adjonction de bit
- inversion de bit

Les deux premières sont facilement détectables mais non corrigibles.
La troisième n'est détectable et corrigible que grâce aux codes de protection, objets de ce document.

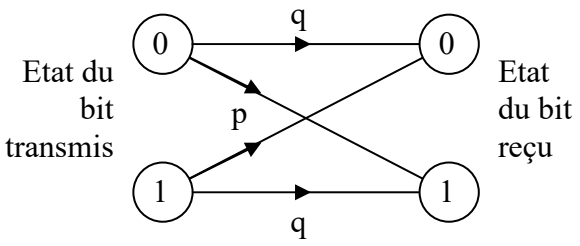
I.2.1) Distribution des erreurs

Les altérations peuvent être classées en 2 types :

- indépendance totale des erreurs : le canal de transmission est qualifié de "**Voie Binaire Symétrique**" ou Binary Symetric Channel (BSC)
- erreurs groupées ou "**paquets d'erreurs**" : une suite de bits plus ou moins longue est altérée.

Les quelques rares calculs statistiques proposés dans ce document ne traiteront que de la voie binaire symétrique.

I.2.1) Voie binaire symétrique



q : probabilité de recevoir le même état que celui transmis.
p : probabilité de recevoir un état opposé à celui transmis.

$p = 1 - q$
$q = 1 - p$

La voie binaire est dite "symétrique" car les probabilités de recevoir le même état ou un état opposé ne dépendent pas de l'état des bits.

La probabilité p caractérise le bruit sur la voie de transmission.

Si on transmet un bloc de n bits, la probabilité de le recevoir avec "d" erreurs est :

$$P_d = p^d \bullet q^{n-d}$$

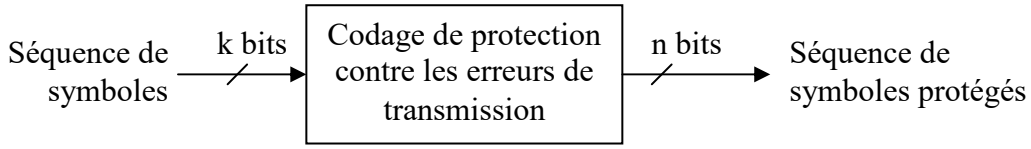
Sans erreur :

$$P_C = q^n = (1 - p)^n$$

Au moins une erreur :

$$P_E = 1 - P_C = 1- (1 - p)^n \cong 1 - (1-n \bullet p) = n \bullet p$$

I.3) Notions de codes redondants



Chaque symbole de l'information à transmettre est codé sur k bits. Le codage de protection ajoute des éléments à chaque symbole pour permettre la détection et/ou la correction, à la réception, des erreurs de transmission.

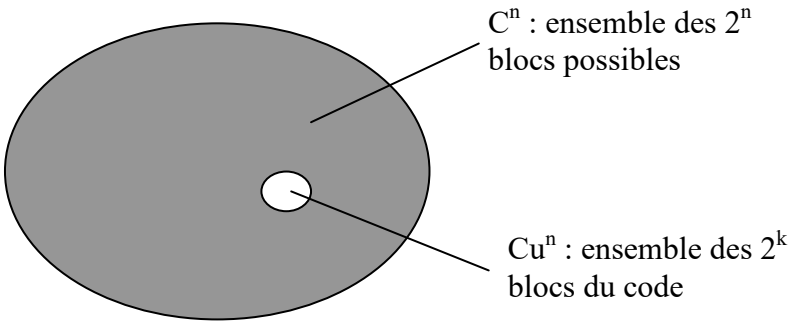
Les symboles protégés sont nommés **blocs** et ont une longueur de n bits.

Deux types de codes sont utilisés :

- codes en bloc : la loi de codage est combinatoire
- codes convolutionnels : la loi de codage est séquentielle. Ce type de code ne sera pas traité dans ce document.

Pour un code en bloc :

- nombre de symboles maximums possible : 2^k
- nombre de symboles protégés (blocs) en sortie du codeur : 2^k (loi combinatoire)
- nombre de blocs possibles : 2^n

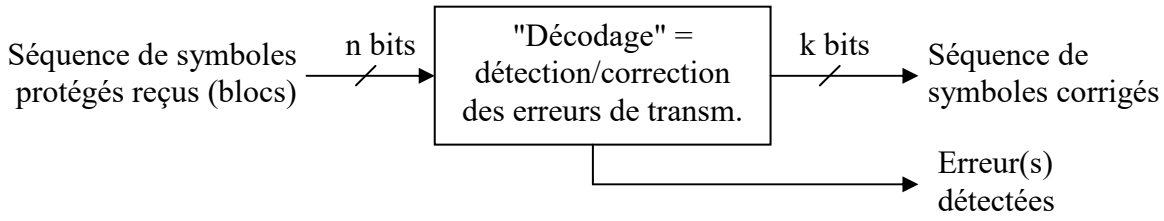


Exemple : code de protection RDS :

- k = 16 ⇒ l'ensemble des blocs du code comporte 65536 éléments
- n = 26 ⇒ l'ensemble C^n comporte 67108864 éléments

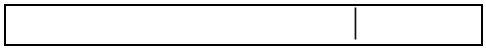
On constate la grande **redondance** du code : l'ensemble Cu^n est 1024 fois plus petit que C^n

Cette redondance permet au récepteur de détecter, voire corriger, les erreurs de transmission.



I.4) Formats des blocs des codes de protection

Pour faciliter, à la réception, l'extraction des symboles dans les blocs reçus, la plupart des codes de protection placent dans chaque bloc le symbole associé non modifié. Deux formats sont utilisés :

- format systématique : bloc  sur n bits

  Info utile (symbole) sur k bits — Info de contrôle sur (n - k) bits
- format non systématique : les bits d'info utile et de contrôle ne sont pas groupés.

Le format systématique est généralement préféré car plus simple à traiter.

Si on veut protéger l'information transmise contre le piratage, les codes correcteurs d'erreur peuvent être conçus de façon à cripter l'information utile. Mais ceci entraîne généralement une dégradation des performances en détection et correction d'erreur. On préfère ajouter la fonction de criptage en amont du codeur de protection contre les erreurs de transmission.

I.5 Exemples de codes protecteurs

I.5.1) Contrôle 2 sur 3

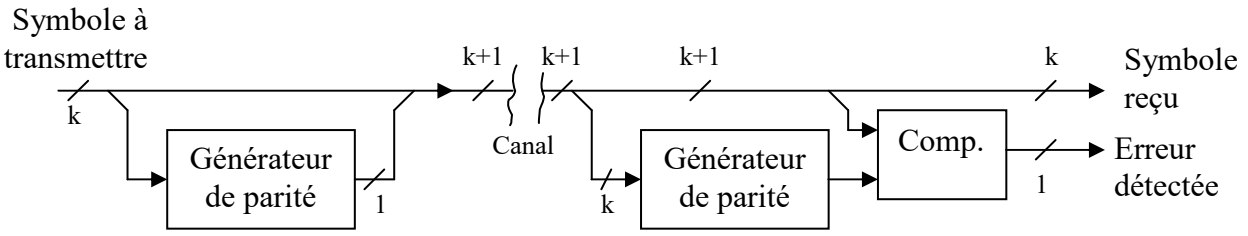
Le même symbole est transmis trois fois de suite. Un bloc du code est donc constitué de trois groupes de k bits et a une longueur de 3•k bits. A la réception, les cas suivants peuvent de présenter :

- les 3 groupes du bloc reçu sont identiques ⇒ il est fort probable que le symbole reçu est celui qui a été transmis.
- 2 groupes sont identiques, le 3° est différent ⇒ il est probable que le symbole transmis soit celui associé aux 2 groupes identiques.
- les 3 groupes sont différents : les erreurs sont détectées mais la correction est impossible.

Ainsi, suivant le nombre d'erreurs, le code est autocorrecteur (une seule erreur) ou simplement autovérificateur (plus d'une erreur).

I.5.2) Contrôle de parité

Schéma

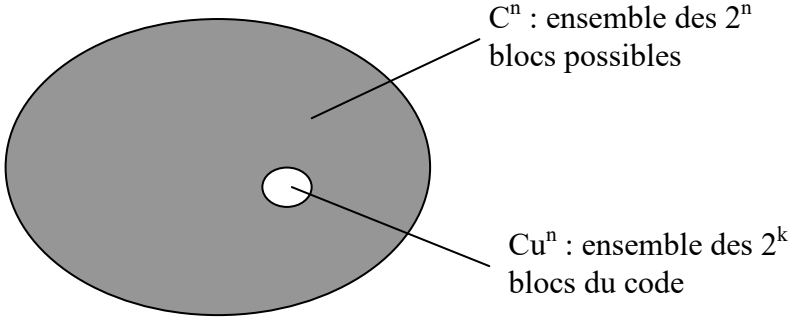


Les générateurs de parité de l'émetteur et du récepteur sont identiques. Le bit de parité calculé à l'émission est ajouté au symbole pour constituer le bloc du code transmis. Le bit de parité calculé à la réception est comparé au bit de parité inclus dans le bloc reçu. Si les 2 états sont identiques, il est probable, mais pas sûr, qu'il n'y a pas d'erreur de transmission. Dans le cas contraire, il y a au moins une erreur.

I.6) Codes détecteurs d'erreurs

Par convention, l'émetteur n'envoie que des blocs appartenant au code. Les éventuelles erreurs de transmission font que le récepteur peut recevoir un bloc différent de celui émis. Dans ce cas :

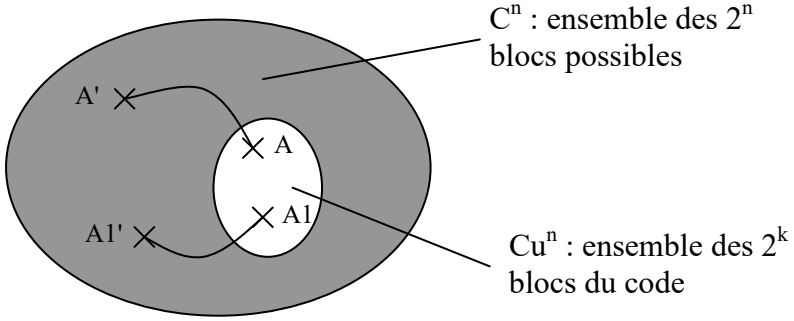
- si le bloc n'appartient pas à l'ensemble Cu^n du code ⇒ détection d'au moins 1 erreur
- si le bloc appartient au code, il est fort probable qu'il n'y a pas d'erreur de transmission. Cette probabilité est d'autant plus forte que le sous-ensemble Cu^n est petit devant C^n.
  En effet, si on transmet le bloc A (donc appartenant à Cu^n), qu'il y a des erreurs de transmission et que l'on reçoit le bloc R, il est fort probable que R soit extérieur au sous-ensemble Cu^n à cause de sa taille par rapport à C^u.



I.7) Codes détecteurs et correcteurs d'erreurs

L'émetteur transmet le bloc A, mais le récepteur reçoit le bloc A' à cause d'erreurs de transmission. A un autre moment, l'émetteur envoie A1 et le récepteur reçoit A1'.

- la correction des erreurs est possible si A' ne peut être obtenu qu'à partir du bloc du code A. L'implication est alors bidirectionnelle et le récepteur peut décider que le bloc transmis est A, alors qu'il a reçu A'.
  Cette implication bidirectionnelle n'est possible que si le nombre d'erreurs est limité. En effet, avec beaucoup d'erreurs, le bloc peut se transformer en bloc A1' à la réception, et le récepteur décidera que c'est A1 qui a été transmis, ce qui est faux. La correction est mise en défaut.
- la détection des erreurs reste possible tant que le nombre d'erreurs ne transforme pas le bloc transmis en un autre bloc du code (on transmet A et on reçoit A1, le récepteur ne peut pas détecter l'erreur).



I.8) Notion de voisinage - Poids et distance de Hamming

I.8.1) Vecteur d'erreur

Si **A** est le bloc émis et **R** le bloc reçu, le vecteur d'erreur **E** est :

$$\mathbf{E = A \oplus R}$$

L'opérateur $\oplus$ signifie "Ou exclusif bit à bit"

Le vecteur d'erreur E comporte autant de bits que les blocs A et R. Les positions des bits de E à "1" indiquent les positions des bits en erreur dans R.

La fonction du décodeur à la réception est de calculer la valeur la plus probable de E en fonction de R pour en déduire le bloc corrigé : $R_{corrigé} = R \oplus E$.
Cette opération inverse l'état des bits en erreur et réalise ainsi la correction du bloc reçu.

Exemple :	l'émetteur transmet	A = (0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0)	
	le récepteur reçoit	R = (0, 1, 0, 1, **0**, 1, 0, 0, 1, **1**, 0)	
	il calcule alors	E = (0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0)	s'il est capable de corriger 2 erreurs
	et en déduit	$R_{corrigé}$ = (0, 1, 0, 1, **1**, 1, 0, 0, 1, **0**, 0)	par inversion des bits identifiés par E

I.8.2) Poids de Hamming

Le poids de Hamming d'un bloc du code est **le nombre d'états "1"** dans ce bloc.

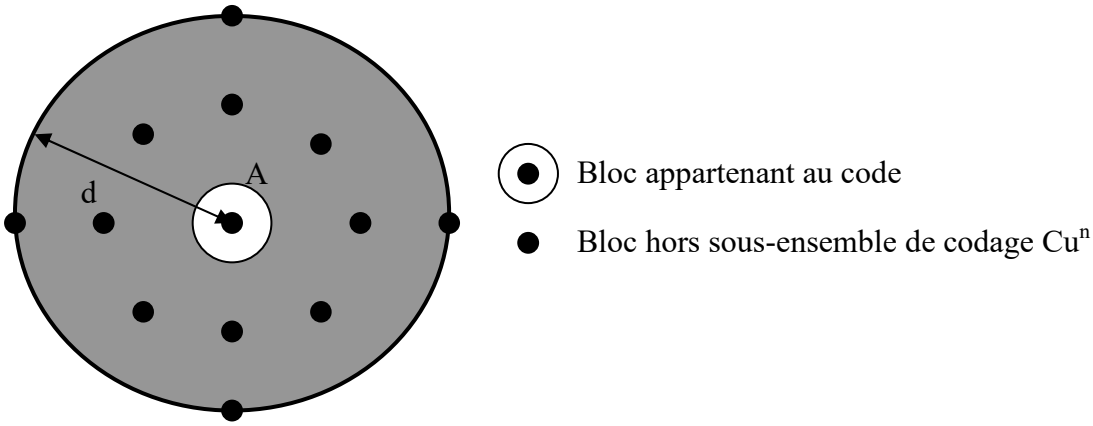
I.8.3) Distance de Hamming

La distance de Hamming entre 2 blocs du code est **le nombre de bits différents** entre ces 2 blocs.

La **distance de Hamming minimum** (notée dmin) est la plus petite distance de Hamming de toutes les paires de blocs du code possibles.

I.8.4) Notion de voisinage

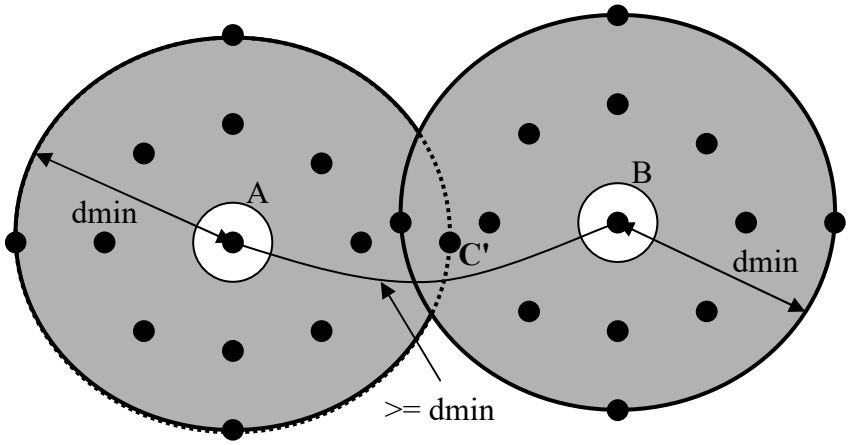
La notion de voisinage définit les règles élémentaires de détection et de correction des erreurs de transmission.



Le bloc A appartient au sous-ensemble de codage Cu^n. Le voisinage de ce bloc est constitué des blocs de l'ensemble C^n ayant une distance de Hamming par rapport à A inférieure ou égale à d.

I.8.5) Principe de la correction

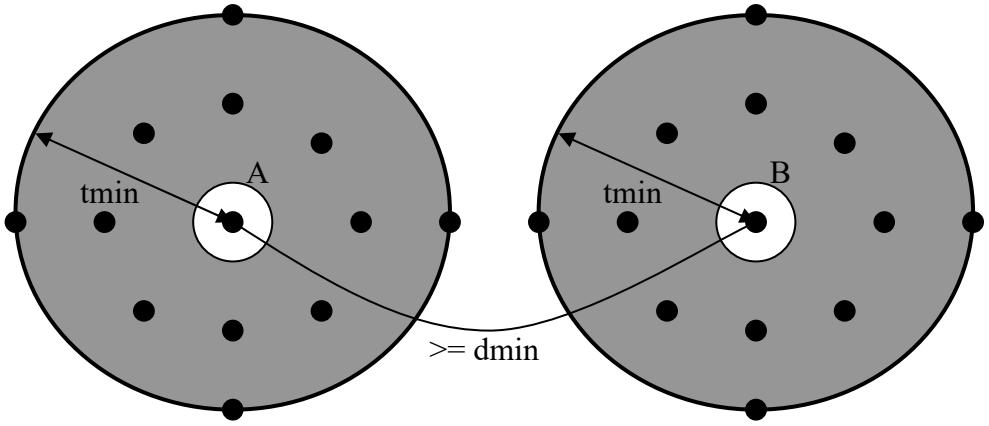
Soient A et B deux blocs appartenant au sous-ensemble de codage Cu^n. On peut définir un voisinage pour chacun de ces blocs :



Voisinage du bloc A Voisinage du bloc B

Les voisinages représentés ci-dessus sont définis à la distance de Hamming minimum dmin. Lorsqu'ils se croisent, il existe des blocs communs aux 2 voisinages. La correction est impossible. En effet, si le récepteur reçoit le bloc C', il est incapable de décider quel est le bloc transmis, A ou B, car C' appartient aux 2 voisinages.

Par contre, si les voisinages des blocs A et B sont distincts, comme représentés ci-dessous, la correction devient possible.



Voisinage du bloc A Voisinage du bloc B

Les voisinages représentés ci-dessus sont définis par la distance de Hamming tmin. Pour que ces voisinages soient distincts et donc permettre la correction des erreurs, la distance tmin doit être telle que :

$$\mathbf{2 \bullet tmin < dmin}$$

Si cette inégalité est respectée, le récepteur sera capable de :

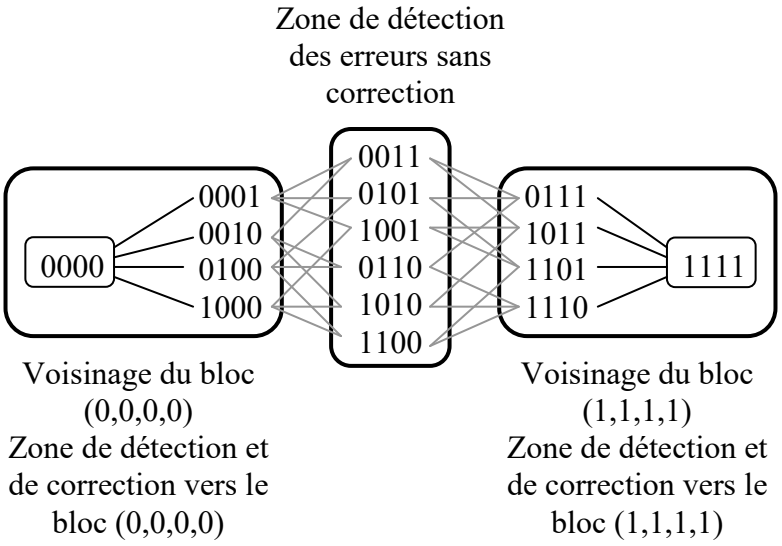
- corriger tmin erreurs simultanées,
- détecter (dmin - 1) erreurs simultanées.

I.9 Exemple

Les symboles à transmettre ont une longueur k = 1 bit. Les blocs du code de protection associés ont une longueur n de 4 bits et sont (0,0,0,0) pour le symbole "0" et (1,1,1,1) pour le symbole "1".

La distance de Hamming minimum vaut donc dmin = 4. On en déduit tmin = 1. Ce codage est donc capable de :

- corriger 1 erreur
- détecter 3 erreurs



I.10) Capacités de la détection et de la correction des erreurs

Notation : le code de protection **(n, k)** produit des blocs de n bits à partir de symboles de k bits en ajoutant (n-k) bits de contrôle.

I.10.1) Capacité de détection

La fonction de détection est mise en défaut si, au cours de la transmission, un bloc du code se transforme en un autre bloc du code.

La probabilité que cela se produit est : $\mathbf{P_D = p^{dmin} \bullet q^{n-dmin} \cong p^{dmin}}$

En effet, dmin représente le nombre minimum de bits différents entre 2 blocs du code. Donc si dmin bits sont altérés au cours de la transmission, le bloc reçu peut faire partie du code.

Exemple : code de Hamming (7,4). Il est caractérisé par dmin = 3 et tmin = 1.
Si le canal de transmission est caractérisé par $p = 10^{-2}$ (1 bit sur 100 inversé) :

$$\Rightarrow P_D = 10^{-6}$$

La probabilité de mettre la détection d'erreur en défaut est de 1 bit sur 1 million.

I.10.1) Capacité de correction

La fonction de correction est mise en défaut si le nombre d'erreurs au cours de la transmission est supérieur à tmin.

La probabilité que cela se produit est : $\mathbf{P_C = p^{tmin+1} \bullet q^{n-(tmin+1)} \cong p^{tmin+1}}$

Exemple : code de Hamming (7,4). Il est caractérisé par dmin = 3 et tmin = 1.
Si le canal de transmission est caractérisé par $p = 10^{-2}$ (1 bit sur 100 inversé) :

$$\Rightarrow Pc = 10^{-4}$$

La probabilité de mettre la correction d'erreur en défaut est de 1 bit sur 10000.

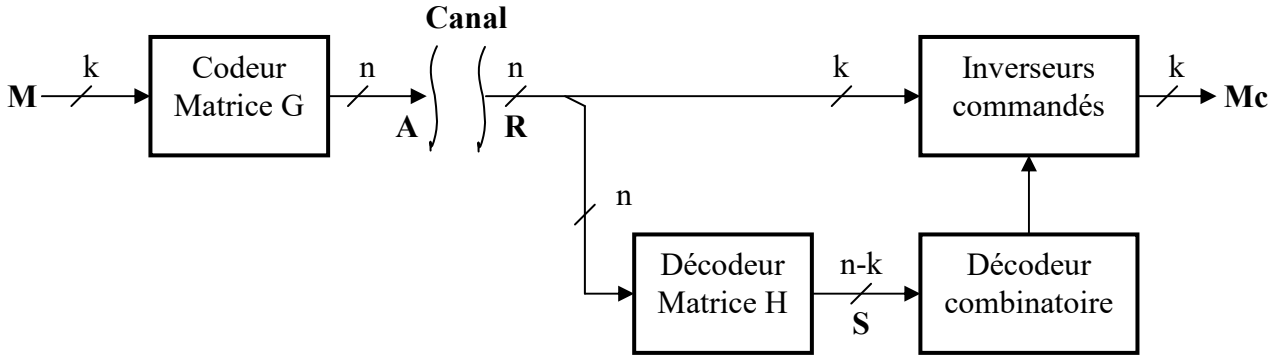
II) Codes linéaires

II.1) Généralités

Notations :

- Code linéaire (n, k)
- k = nombre de bits de chaque symbole
- n = longueur des blocs du code
- le codeur ajoute donc (n-k) bits de contrôle (ou redondants)
- V = k/n : vitesse du code

Schéma :



II.2) Définitions

Le code est dit "linéaire (n, k)" si la somme modulo 2 bit à bit (opérateur ⊕) de deux blocs du code donne un autre bloc du code.

Note : le bloc (0,0,0,0, … ,0) fait partie de tous les codes linéaires car il est le résultat de la somme d'un bloc du code avec lui-même.

Les codes linéaires utilisent le format systématique.

II.3) Codeur

La **matrice de codage G** défini entièrement le codeur linéaire (n, k) largeur = n, hauteur = k :

$$G = \left|\begin{array}{ccccc|ccccc} \multicolumn{5}{c|}{\text{Sous-matrice diagonale (k)}} & \multicolumn{5}{c}{\text{Sous-matrice de parité P (n-k, k)}} \\ 1 & 0 & \dots & 0 & 0 & p_{k-1,n-k-1} & p_{k-1,n-k-2} & \dots & p_{k-1,1} & p_{k-1,0} \\ 0 & 1 & \dots & 0 & 0 & p_{k-2,n-k-1} & p_{k-2,n-k-2} & \dots & p_{k-2,1} & p_{k-2,0} \\ 0 & 0 & \dots & 0 & 0 & p_{k-3,n-k-1} & p_{k-3,n-k-2} & \dots & p_{k-3,1} & p_{k-3,0} \\ \dots & \dots & \dots & 0 & 0 & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & p_{2,n-k-1} & p_{2,n-k-2} & \dots & p_{2,1} & p_{2,0} \\ 0 & 0 & \dots & 1 & 0 & p_{1,n-k-1} & p_{1,n-k-2} & \dots & p_{1,1} & p_{1,0} \\ 0 & 0 & \dots & 0 & 1 & p_{0,n-k-1} & p_{0,n-k-2} & \dots & p_{0,1} & p_{0,0} \end{array}\right|$$

On y reconnaît :

- une sous-matrice diagonale de dimension k
- une sous-matrice de parité (coef. $p_{i,j}$) de hauteur k et de largeur (n-k)

Le bloc A est obtenu en calculant le produit scalaire (ET logique et ⊕) :

$$\mathbf{A = M \blacklozenge G}$$

En notant :

- $M = (m_{k-1}, m_{k-2}, \dots, m_1, m_0)$,
- $A = (a_{n-1}, a_{n-2}, \dots, a_1, a_0)$.

On en déduit :

$a_{n-1} = m_{k-1}$
$a_{n-2} = m_{k-2}$
…
$a_{n-k+1} = m_1$
$a_{n-k} = m_0$
$a_{n-k-1} = m_{k-1}\bullet p_{k-1,n-k-1} \oplus m_{k-2}\bullet p_{k-2,n-k-1} \oplus \dots \oplus m_1\bullet p_{1,n-k-1} \oplus m_0\bullet p_{0,n-k-1}$
$a_{n-k-2} = m_{k-1}\bullet p_{k-1,n-k-2} \oplus m_{k-2}\bullet p_{k-2,n-k-2} \oplus \dots \oplus m_1\bullet p_{1,n-k-2} \oplus m_0\bullet p_{0,n-k-2}$
…
$a_1 = m_{k-1}\bullet p_{k-1,1} \oplus m_{k-2}\bullet p_{k-2,1} \oplus \dots \oplus m_1\bullet p_{1,1} \oplus m_0\bullet p_{0,1}$
$a_0 = m_{k-1}\bullet p_{k-1,0} \oplus m_{k-2}\bullet p_{k-2,0} \oplus \dots \oplus m_1\bullet p_{1,0} \oplus m_0\bullet p_{0,0}$

On constate donc que :

- les k premiers bits du bloc A représentent la copie du symbole M (a_{n-1} à a_{n-k})
- les (n-k) derniers bits du bloc A sont les bits de contrôle (a_{n-k-1} à a_0)

C'est bien un format systématique.

Exemple : matrice génératrice du code linéaire de Hamming (7,4) :

$$G_{(7,4)} = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}$$

Soit $M = (m_3, m_2, m_1, m_0)$ le symbole à encoder
On en déduit :

$a_6 = m_3$
$a_5 = m_2$
$a_4 = m_1$
$a_3 = m_0$
$a_2 = m_3 \oplus m_2 \oplus m_0$
$a_1 = m_3 \oplus m_1 \oplus m_0$
$a_0 = m_2 \oplus m_1 \oplus m_0$

Si $M = (1, 0, 1, 1) \Rightarrow A = (1, 0 , 1, 1, 0, 1, 0)$

II.4) Décodeur

Il n'y a pas de décodage du symbole à proprement parlé puisqu'il se trouve non modifié dans le bloc reçu. La fonction du décodeur est de **détecter et éventuellement identifier les bits altérés** au cours de la transmission.

Pour ce faire, on calcule le **syndrome S**, résultat du produit scalaire entre le bloc reçu (R) et la transformée de la matrice de vérification H (hauteur = n - k, largeur = n) :

$$\mathbf{S = R \blacklozenge H^T}$$

$$H = \left|\begin{array}{ccccc|ccccc} \multicolumn{5}{c|}{\text{Sous-matrice de parité } P^T\ (k,\ n\text{-}k)} & \multicolumn{5}{c}{\text{Sous-matrice diagonale } (n\text{-}k)} \\ p_{k-1,n-k-1} & p_{k-2,n-k-1} & \cdots & p_{1,n-k-1} & p_{0,n-k-1} & 1 & 0 & \ldots & 0 & 0 \\ p_{k-1,n-k-2} & p_{k-2,n-k-2} & \cdots & p_{1,n-k-2} & p_{0,n-k-2} & 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ p_{k-1,1} & p_{k-2,1} & \cdots & p_{1,1} & p_{0,1} & 0 & 0 & \ldots & 1 & 0 \\ p_{k-1,0} & p_{k-2,0} & \cdots & p_{1,0} & p_{0,0} & 0 & 0 & \ldots & 0 & 1 \end{array}\right|$$

La sous-matrice de parité P^T de la matrice H est la **transposée** de la sous-matrice P de la matrice G, de telle façon que : $\mathbf{G \blacklozenge H^T = [0]_{k,n-k}}$

En notant :

- $R = (r_{n-1}, \ldots, r_1, r_0)$, les n bits du bloc reçu et
- $S = (s_{n-k-1}, \ldots, s_1, s_0)$, les (n-k) bits du syndrome,

On en déduit :

$$\begin{array}{lll} s_{n-k-1} & = (r_{n-1} \bullet p_{k-1,n-k-1} \oplus \ \ldots \oplus r_{n-k+1} \bullet p_{1,n-k-1} \ \oplus r_{n-k} \bullet p_{0,n-k-1}) \oplus r_{n-k-1} & \\ \ldots & & \ldots \\ s_1 & = (r_{n-1} \bullet p_{k-1,1} \ \oplus \ \ldots \oplus r_{n-k+1} \bullet p_{1,1} \ \oplus r_{n-k} \bullet p_{0,1}) & \oplus r_1 \\ s_0 & = (r_{n-1} \bullet p_{k-1,0} \ \oplus \ \ldots \oplus r_{n-k+1} \bullet p_{1,0} \ \oplus r_{n-k} \bullet p_{0,0}) & \oplus r_0 \end{array}$$

S'il n'y a **aucune erreur** au cours de la transmission, les blocs émis (A) et reçu (R) sont identiques, soit $r_{n-1} = a_{n-1} = m_{k-1}$, … , $a_{n-k+1} = r_{n-k+1} = m_1$, $r_{n-k} = a_{n-k} = m_0$, $r_{n-k-1} = a_{n-k-1}$, … , $r_1 = a_1$, $r_0 = a_0$ et on en déduit :

$$(r_{n-1} \bullet p_{k-1,n-k-1} \oplus \ldots \oplus r_{n-k+1} \bullet p_{1,n-k-1} \oplus r_{n-k} \bullet p_{0,n-k-1}) = m_{k-1} \bullet p_{k-1,n-k-1} \oplus \ldots \oplus m_1 \bullet p_{1,n-k-1} \oplus m_0 \bullet p_{0,n-k-1} = a_{n-k-1} = r_{n-k-1}$$

…

$$(r_{n-1} \bullet p_{k-1,1} \oplus \ldots \oplus r_1 \bullet p_{1,1} \oplus r_0 \bullet p_{0,1}) = (m_{k-1} \bullet p_{k-1,1} \oplus \ldots \oplus m_1 \bullet p_{1,1} \oplus m_0 \bullet p_{0,1}) = a_1 = r_1$$

$$(r_{n-1} \bullet p_{k-1,0} \oplus \ldots \oplus r_1 \bullet p_{1,0} \oplus r_0 \bullet p_{0,0}) = (m_{k-1} \bullet p_{k-1,0} \oplus \ldots \oplus m_1 \bullet p_{1,0} \oplus m_0 \bullet p_{0,0}) = a_0 = r_0$$

Ce qui donne le syndrome suivant : $S = (s_{n-k-1}, \ldots, s_1, s_0) = (0, 0, 0, 0, \ldots, 0, 0)$

Conclusion : *si tous les bits du syndrome S sont à zéro, on peut en déduire qu'il est très fortement probable qu'il n'y a **pas d'erreur de transmission**.*

En fait, chaque bit s_i du syndrome S est le résultat de la comparaison entre :

- le bit de parité a_i calculé à l'émission sur certains bits de l'information utile, et reçu sous l'indentificateur $\mathbf{r_i}$. L'indice i correspond à une des colonnes de la sous-matrice diagonale de H, soit i compris entre (n-k-1) et 0.
- le bit de parité calculé à la réception sur les mêmes bits d'information utile reçus.

S'il n'y a aucune erreur, les 2 bits de parité ont le même état et s_i = "0". Dans le cas contraire, s_i = "1", ce qui signifie qu'il y a une altération sur l'un des bits contrôlés.

Exemple : La matrice de décodage (ou de vérification) H du code de Hamming (7,4) est

$$H_{(7,4)} = \begin{vmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$$

L'émetteur transmet le bloc A = (1, 0 , 1, 1, 0, 1, 0)
Le récepteur reçoit le bloc R = (1, 0 , 1, 1, 0, 1, 0) , donc sans erreur.

Le produit scalaire $S = R \blacklozenge H^T$ donne :

$s_2 = \boxed{(1\bullet1 \oplus 1\bullet0 \oplus 0\bullet1 \oplus 1\bullet1)} \oplus 1\bullet\boxed{0} \oplus 0\bullet1 \oplus 0\bullet0 = 0$

Calcul de parité sur m_3, m_2, m_1 et m_0 reçus = 0 ; Bit de parité entre m_3, m_2, m_1 et m_0 calculé à l'émission et reçu sans erreur

$s_1 = (1\bullet1 \oplus 0\bullet0 \oplus 1\bullet1 \oplus 1\bullet1) \oplus 0\bullet0 \oplus 1\bullet1 \oplus 0\bullet0 = 0$

$s_0 = (0\bullet1 \oplus 1\bullet0 \oplus 1\bullet1 \oplus 1\bullet1) \oplus 0\bullet0 \oplus 0\bullet1 \oplus 1\bullet0 = 0$

Si maintenant le récepteur reçoit R = (1, 0 , **0**, 1, 0, 1, 0) le bit r_4 étant inversé, le produit scalaire $S = R \blacklozenge H$ donne :

$s_2 = (1\bullet1 \oplus 1\bullet0 \oplus 0\bullet\mathbf{0} \oplus 1\bullet1) \oplus 1\bullet0 \oplus 0\bullet1 \oplus 0\bullet0 = 0$

$s_1 = (1\bullet1 \oplus 0\bullet0 \oplus 1\bullet\mathbf{0} \oplus 1\bullet1) \oplus 0\bullet0 \oplus 1\bullet1 \oplus 0\bullet0 = 1$

$s_0 = (0\bullet1 \oplus 1\bullet0 \oplus 1\bullet\mathbf{0} \oplus 1\bullet1) \oplus 0\bullet0 \oplus 0\bullet1 \oplus 1\bullet0 = 1$

Soit S_{10} (équivalent décimal de S) = 3. Cette valeur permettra au décodeur d'identifier la position 4 pour corriger r_4 (c'est à dire inverser son état).
On peut remarquer que les états de s_2, s_1 et s_0 correspondent à la colonne d'indice 4 de la matrice H. Ce n'est pas un hasard !

II.5) <u>Décodage des codes linéaires</u>

Il s'agit d'étudier la manière d'utiliser le syndrome S pour évaluer la valeur la plus probable du vecteur d'erreur E.

Rappels : $A = (a_{n-1}, a_{n-2}, \dots, a_1, a_0)$ est le bloc émis,
$R = (r_{n-1}, r_{n-2}, \dots, r_1, r_0)$ est le bloc reçu,
$E = (e_{n-1}, e_{n-2}, \dots, e_1, e_0)$ est le vecteur d'erreur.

Par définition : $E = A \oplus R$
On en déduit : $R = A \oplus E$
Le syndrome est : $S = R \blacklozenge H^T = (A \oplus E) \blacklozenge H^T = (A \blacklozenge H^T) \oplus (E \blacklozenge H^T)$
Or, par définition : $A \blacklozenge H^T = 0$ car A est un bloc du code et le produit scalaire avec la matrice de vérification est toujours nul.
On en déduit : $\mathbf{S = E \blacklozenge H^T}$

Le syndrome est le résultat du produit scalaire du bloc reçu R avec H^T, mais c'est aussi, comme le montre la relation ci-dessus, **le résultat du produit scalaire du vecteur E avec la transposée de la matrice de vérification.**

En utilisant le contenu de la matrice H, on obtient :

$s_{n-k-1} = e_{n-1} \bullet p_{k-1,n-k-1} \oplus \ldots \oplus e_{n-k+1} \bullet p_{1,n-k-1} \oplus e_{n-k} \bullet p_{0,n-k-1} \oplus e_{n-k-1}$

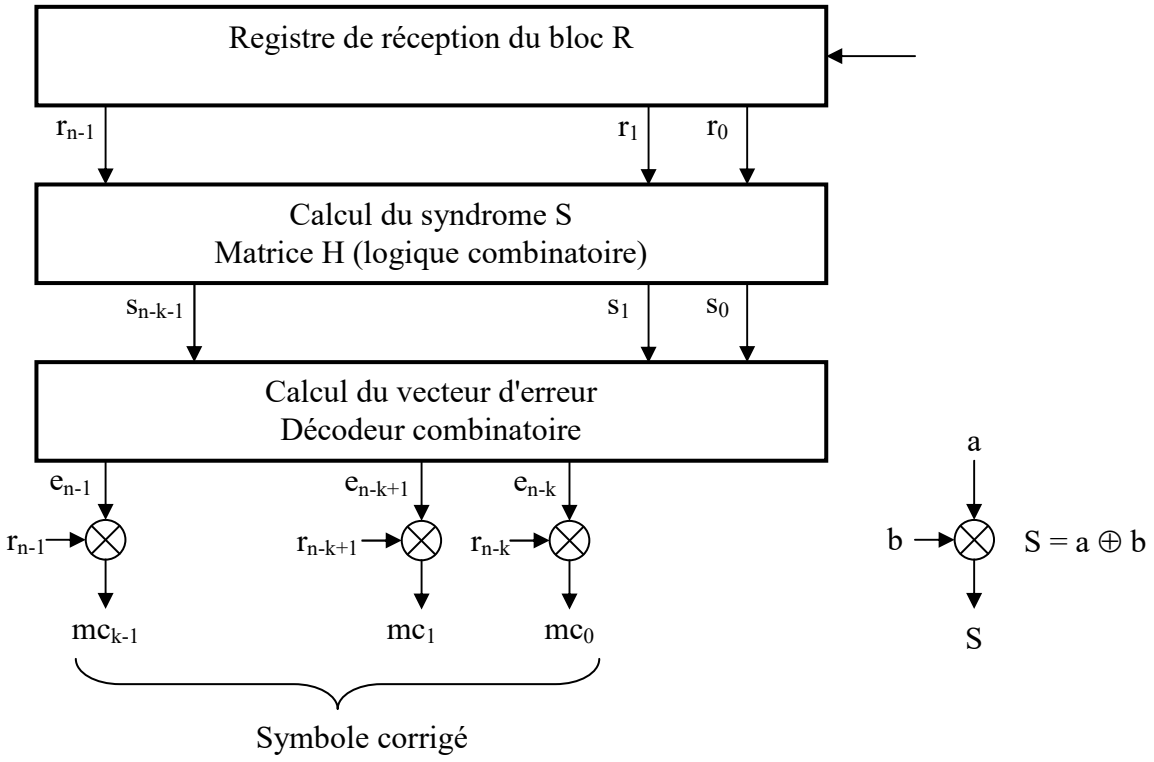
...

$s_1 = e_{n-1} \bullet p_{k-1,1} \oplus \ldots \oplus e_{n-k+1} \bullet p_{1,1} \oplus e_{n-k} \bullet p_{0,1} \ldots \oplus e_1$

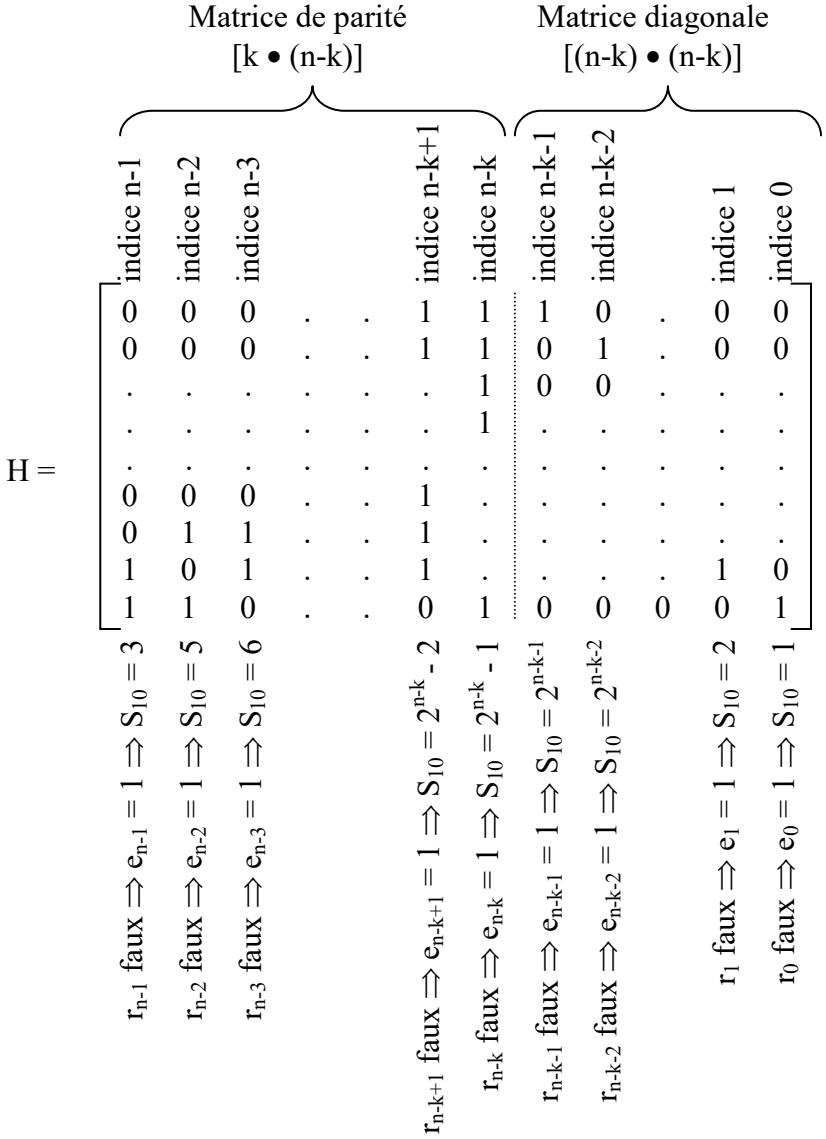
$s_0 = e_{n-1} \bullet p_{k-1,0} \oplus \ldots \oplus e_{n-k+1} \bullet p_{1,0} \oplus e_{n-k} \bullet p_{0,0} \oplus e_0$

On constate que les états des bits du syndrome sont chacun une combinaison linéaire des bits d'erreur.
En conséquence, **le syndrome contient une information sur les positions (les indices) des bits altérés**.
Si on se contente d'une capacité de correction d'une seule erreur (un seul des bits de E est à "1"), un simple décodeur combinatoire permet d'identifier la position en erreur, et donc de déterminer la valeur du vecteur d'erreur E.

Schéma du détecteur/correcteur d'erreur



Le contenu de la matrice H peut maintenant être complètement défini :



Si on admet qu'il y a qu'une seule erreur dans le bloc reçu, le syndrome obtenu correspond à la colonne d'indice du bit altéré de la matrice H. Pour permettre l'identification de cet indice à partir de S, il suffit de choisir les états de la matrice de parité de manière à ce que les valeurs de S_{10} soient uniques. Les valeurs en puissance entières de 2 sont réservées par la matrice diagonale, les autres valeurs peuvent être choisies de manière quelconque.

II.6) <u>Codes de Hamming corrigeant une erreur</u>

Dans ce cas, le vecteur d'erreur obtenu après décodage aura **au maximum** une composante non nulle.

Puisque le syndrome S de (n-k) positions (nombre de bits redondants) doit permettre d'indiquer, soit l'absence d'erreur, soit la position altérée dans le bloc reçu, la valeur (n-k) de bits de contrôle doit respecter l'inégalité :

$$\mathbf{2^{n-k} - 1 \geq n}$$

On en déduit le tableau suivant :

Nombres de positions de contrôle	n-k	2	3	4	5	6	7	8	…
Nombre maximum de positions dans le bloc	$n = 2^{n-k} - 1$	3	7	15	31	63	127	255	…
Nombre maximum de bits d'information utile	k	1	4	11	26	57	120		

II.7) Exercice

Etude du code linéaire de Hamming (15,11)

1. Le code est capable de corriger une erreur. Quelle est sa capacité de détection d'erreur ? En déduire sa distance de Hamming minimum.

2. Proposer le contenu de la matrice de vérification H. En déduire le contenu de la matrice génératrice G.

3. Proposer le schéma logique du codeur (entrée M sur 11 bits, sortie A sur 15 bits).

4. Proposer le schéma logique du décodeur (entrée R sur 15 bits, sortie corrigée sur 11 bits)

III) Codes cycliques

III.1) Définitions

Un code linéaire est dit cyclique si une transposition circulaire à droite ou à gauche d'un bloc du code donne un autre bloc du code.

Ainsi si $A^{(0)} = (a_0, a_1, \dots, a_{n-1})$ est un bloc du code
alors $A^{(1)} = (a_{n-1}, a_0, a_1, \dots, a_{n-2})$ est aussi un bloc du code.

Les codes cycliques sont intéressants sur plusieurs points :

- l'encodage à l'émission et le calcul du vecteur d'erreur à la réception sont facilement réalisables sous les formes logicielle ou matérielle,
- ils sont capables de corriger les paquets d'erreurs. Un paquet d'erreurs de longueur L est une succession de L bits, corrects ou faux, le premier et le dernier étant toujours erronés.

III.2) Représentations

Le symbole M à protéger est représenté sous une forme polynomiale :

$$\mathbf{M(X) = m_{k-1} \bullet X^{k-1} + \dots + m_1 \bullet X^1 + m_0 \bullet X^0}$$

Le polynôme associé au bloc du code $A = (a_0, a_1, \dots, a_{n-1})$ s'écrit :

$$\mathbf{A(X) = a_{n-1} \bullet X^{n-1} + \dots + a_1 \bullet X^1 + a_0 \bullet X^0}$$

A(X) est un **bloc polynomial**.

Notes : l'opérateur + est en fait l'opérateur $\oplus$. Il n'a pas été utilisé pour améliorer la lisibilité.
l'opérateur $\bullet$ est le ET logique.

III.3) Polynôme générateur

Les codes cycliques (n,k) sont entièrement définis par un polynôme générateur G(X) ayant la forme générale suivante :

$$\mathbf{G(X) = X^{n-k} + g_{n-k-1} \bullet X^{n-k-1} + \dots + g_1 \bullet X^1 + 1}$$

G(X) doit posséder les propriétés suivantes :

- non nul ce qui justifie le dernier terme ($g_0 = 1$),
- de degré (n-k), ce qui justifie le premier terme ($g_{n-k} = 1$),
- facteur de $(X^n + 1)$,
- premier.

Note : Les méthodes d'élaboration des polynômes générateurs sortent du domaine d'étude de ce document.

III.4) Définition de l'encodage

Chaque bloc du code est un multiple polynomial de G(X) : $\mathbf{A(X) = f(X) \bullet G(X)}$

III.4.1) Encodage simple

Dans ce cas : **f(X) = M(X)**

Le bloc polynomial A(X) associé au symbole M est le résultat du produit polynomial de M(X) par G(X) :

$$\mathbf{A(X) = M(X) \bullet G(X)}$$

Exemple : si on respecte cette définition mathématique, on peut calculer tous les blocs du code cyclique (7,4) défini par le polynôme générateur $G(X) = X^3 + X^1 + 1$:

Symboles				Blocs du code						
m_3	m_2	m_1	m_0	a_6	a_5	a_4	a_3	a_2	a_1	a_0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	1	1
0	0	1	0	0	0	1	0	1	1	0
0	0	1	1	0	0	1	1	1	0	1
0	1	0	0	0	1	0	1	1	0	0
0	1	0	1	0	1	0	0	1	1	1
0	1	1	0	0	1	1	1	0	1	0
0	1	1	1	0	1	1	0	0	0	1
1	0	0	0	1	0	1	1	0	0	0
1	0	0	1							
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0							
1	1	1	1							

Exemples de calculs :

$M(X) = 0 \quad \Rightarrow A(X) = M(X) \bullet G(X) = 0$

$M(X) = X^2 + X \quad \Rightarrow A(X) = M(X) \bullet G(X) = X^5 + X^2 + X^4 + X^2 + X^3 + X^5$

$A(X) = X^5 + X^4 + X^3 + X^1$ soit A = (0, 1, 1, 1, 0, 1, 0)

De ce tableau, on déduit dmin = 3 car, hormis le premier bloc, il y au moins 3 positions à 1 dans les autres blocs. Le code est donc capable de corriger 1 erreur et d'en détecter 2.

Cette méthode de codage **n'est pas utilisée** car le format obtenu n'est pas systématique (symboles non apparents dans les blocs du code).

III.4.2) Encodage des codes cycliques

Le polynôme f(X) est élaboré pour obtenir des blocs au **format systématique**.

Le symbole à encoder est $M = (m_{k-1}, \ldots, m_1, m_0)$.

Le symbole polynomial correspondant est : $M(X) = m_{k-1} \bullet X^{k-1} + \ldots + m_1 \bullet X^1 + m_0 \bullet X^0$

On multiple M(X) par X^{n-k} : $X^{n-k} \bullet M(X) = m_{k-1} \bullet X^{n-1} + \ldots + m_1 \bullet X^{n-k-1} + m_0 \bullet X^{n-k}$

C'est un polynôme de degré inférieur ou égal à (n-1).

Si maintenant on divise $X^{n-k} \bullet M(X)$ par G(X), on obtient :

(1) $X^{n-k} \bullet M(X) = Q(X) \bullet G(X) + \rho(X)$

avec : Q(X) : quotient de la division. Degré ≤ n-1-(n-k) = k-1

ρ(X) : reste de la division. Degré ≤ (n-k-1)

Comme le degré de G(X) est (n-k), le degré de ρ(X) est nécessairement inférieur ou égal à (n-k-1). En effet, le reste d'une division est toujours de degré inférieur au diviseur.

Soit : $\rho(X) = \rho_{n-k-1} \bullet X^{n-k-1} + \ldots + \rho_1 \bullet X^1 + \rho_0$

Le + de la relation (1) ci-dessus signifie en fait ⊕ (ou exclusif sur les bits de même poids). Cette égalité peut alors s'écrire :

(2) $X^{n-k} \bullet M(X) + \rho(X) = Q(X) \bullet G(X)$

Cette relation indique que " $X^{n-k} \bullet M(X) + \rho(X)$ " est un multiple de G(X) et est de degré inférieur ou égal à (n-1).

Ainsi " $X^{n-k} \bullet M(X) + \rho(X)$ " est le bloc polynomial A(X) associé au symbole M(X).

En développant " $X^{n-k} \bullet M(X) + \rho(X)$ " , on obtient :

$$X^{n-k} \bullet M(X) + \rho(X) = m_{k-1} \bullet X^{n-1} + \ldots + m_1 \bullet X^{n-k+1} + m_0 \bullet X^{n-k} + \rho_{n-k-1} \bullet X^{n-k-1} + \ldots + \rho_1 \bullet X^1 + \rho_0$$

qui correspond au bloc du code : A = ($\boxed{m_{k-1}, \ldots, m_1, m_0}$, $\boxed{\rho_{n-k-1}, \ldots, \rho_1, \rho_0}$)

k bits du symbole (n-k) bits de contrôle

On reconnaît un **format systématique**. Les (n-k) bits de contrôle sont les coefficients de ρ(X)

En résumé, le codage consiste en 3 étapes :

- Multiplier le symbole M(X) par X^{n-k}. On obtient ainsi les k premiers bits du bloc du code A,
- Calculer le reste ρ(X) de la division de $X^{n-k} \bullet M(X)$ par G(X). On obtient ainsi les (n - k) bits de contrôle,
- Concaténer $X^{n-k} \bullet M(X)$ et $\rho(X)$ pour obtenir les n bits du bloc du code A associé au symbole.

Exemple

Soit un code cyclique (7,4) défini par $G(X) = X^3 + X^1 + 1$

Le symbole à encoder s'écrit : $M(X) = m_3 \bullet X^3 + m_2 \bullet X^2 + m_1 \bullet X^1 + m_0 \bullet X^0$

Les blocs du code sont obtenus en calculant le reste ρ(X) de la division de $X^3 \bullet M(X)$ par G(X) :

$$A(X) = X^3 \bullet M(X) + \rho(X)$$

Le tableau ci-dessous donne tous les blocs du code :

Symboles				$X^{n-k} \bullet M(X)$	Reste	Blocs du code						
m_3	m_2	m_1	m_0	$X^3 \bullet M(X)$	$\rho(X)$	a_6	a_5	a_4	a_3	a_2	a_1	a_0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	X^3	$X^1 + X^0$	0	0	0	1	0	1	1
0	0	1	0	X^4	$X^2 + X^1$	0	0	1	0	1	1	0
0	0	1	1	$X^4 + X^3$	$X^2 + X^0$	0	0	1	1	1	0	1
0	1	0	0	X^5	$X^2 + X^1 + X^0$	0	1	0	0	1	1	1
0	1	0	1	$X^5 + X^3$	X^2	0	1	0	1	1	0	0
0	1	1	0	$X^5 + X^4$	X^0	0	1	1	0	0	0	1
0	1	1	1	$X^5 + X^4 + X^3$	X^1	0	1	1	1	0	1	0
1	0	0	0	X^6	$X^2 + 1$	1	0	0	0	1	0	1
1	0	0	1	$X^6 + X^3$		1	0	0	1			
1	0	1	0	$X^6 + X^4$		1	0	1	0			
1	0	1	1	$X^6 + X^4 + X^3$		1	0	1	1			
1	1	0	0	$X^6 + X^5$		1	1	0	0			
1	1	0	1	$X^6 + X^5 + X^3$		1	1	0	1			
1	1	1	0	$X^6 + X^5 + X^4$		1	1	1	0			
1	1	1	1	$X^6 + X^5 + X^4 + X^3$		1	1	1	1			

III.5) <u>Décodage des codes cycliques</u>

Soient $A = (a_0, a_1, \dots, a_{n-1})$ le bloc du code émis
et $R = (r_0, r_1, \dots, r_{n-1})$ le bloc reçu.
Ils sont associés aux polynômes :

$$A(X) = a_{n-1} \bullet X^{n-1} + \dots + a_1 \bullet X^1 + a_0 \bullet X^0$$
$$R(X) = r_{n-1} \bullet X^{n-1} + \dots + r_1 \bullet X^1 + r_0 \bullet X^0$$

Comme dans les codes linéaires, on défini un vecteur d'erreur :

$$E = (e_{n-1}, e_{n-2}, \dots, e_1, e_0)$$
tel que $E = A \oplus R$

qui est associé au polynôme $E(X) = e_{n-1} \bullet X^{n-1} + \dots + e_1 \bullet X^1 + e_0 \bullet X^0$

On en déduit : $E(X) = A(X) + R(X)$

En divisant R(X) par G(X) on obtient :

$$R(X) = Q'(X) \bullet G(X) + S(X)$$

S(X) est le polynôme associé au syndrome S de degré strictement inférieur à celui de G(X).
Q'(X) est le quotient de la division de degré $\leq n-1-(n-k) = k-1$

Soit $E(X) = A(X) + Q'(X) \bullet G(X) + S(X)$

Or A(X) est, par définition, un bloc polynomial généré avec G(X), soit :

$$A(X) = Q(X) \bullet G(X)$$ (voir relation (2) ci-dessus)

Et **$E(X) = [Q(X) + Q'(X)] \bullet G(X) + S(X)$**

On constate, avec cette dernière relation, que le **syndrome S(X) est le reste de la division du polynôme d'erreur E(X) avec G(X),** mais aussi (et c'est comme cela qu'il est calculé), le reste de la division du polynôme associé au bloc reçu, R(X), avec G(X).

Cette propriété permet de définir le type d'erreur pouvant être détectée :

- Pas d'erreur : $R(X) = A(X) \Rightarrow E(X) = 0$
  Le reste de la division de E(X) par G(X) est nul $\Rightarrow S(X) = 0$
  S(X) = 0 est caractéristique d'une transmission sans erreur
- Une seule erreur : $E(X) = X^i$ avec $0 \leq i \leq n-1$.
  S(X) est nécessairement ≠ 0 (sauf si G(X) = 1 ou $G(X) = X^j$, ce qui ne peut être le cas car ces polynômes ne sont par facteurs de $(X^n + 1)$).
- Paquet d'erreurs de longueur L. Un paquet d'erreurs de longueur L est une succession de L bits, corrects ou faux, le premier et le dernier étant toujours erronés.
  Le degré de E(X) est nécessairement supérieur ou égal à (L-1). Ce polynôme peut donc se mettre sous la forme :

  $$E(X) = X^i \bullet B(X)$$

  B(X), de degré (L-1) est appelé **modèle d'erreur** :

  $$B(X) = X^{L-1} + b_{L-2} \bullet X^{L-2} + \ldots + b_1 \bullet X^1 + X^0$$

  Or, tout polynôme générateur de degré supérieur à (L-1) ne peut diviser le modèle d'erreur. On choisira donc un polynôme générateur G(X) de degré (n-k) tel que $(n-k) \geq L$ pour détecter des paquets d'erreurs de longueur inférieure ou égale à L.

Exemple

Soit un code cyclique (7,4) défini par $G(X) = X^3 + X^1 + 1$
La distance minimum de Hamming (dmin) vaut 3. Le code est donc capable de détecter 2 erreurs et de corriger 1 erreur.

S'il y a une seule erreur, le vecteur d'erreur sera de la forme : $E(X) = X^i$, avec $0 \leq i \leq 6$
Avec cette hypothèse, on peut calculer les syndromes obtenus pour chacune des erreurs possibles :

Erreur à l'indice	Polynôme E(X) associé	Reste S(X) de $E(X)/(X^3 + X^1 + 1)$
0	X^0	1
1	X^1	X^1
2	X^2	X^2
3	X^3	$X^1 + 1$
4	X^4	$X^2 + X^1$
5	X^5	$X^2 + X^1 + 1$
6	X^6	$X^2 + 1$

En rappelant que $S(X) = s_2 \bullet X^2 + s_1 \bullet X^1 + s_0$, on peut réaliser un décodeur qui donne la position en erreur, donc qui élabore le vecteur d'erreur E :

s_2	s_1	s_0	S_{10}	Indice de l'erreur	Bit du symbole correspondant
0	0	0	0	Pas d'erreur	-
0	0	1	1	0	Bit de contrôle
0	1	0	2	1	Bit de contrôle
1	0	0	4	2	Bit de contrôle
0	1	1	3	3	m_0
1	1	0	6	4	m_1
1	1	1	7	5	m_2
1	0	1	5	6	m_3

Si la position pointée correspond à un des bits du symbole transmis, il suffira de l'inverser pour corriger l'erreur (voir schéma page 13).

III.6) Analogie avec les codes linéaires

Un code cyclique est un cas particulier de code linéaire. Les codes cycliques peuvent également être définis par leurs matrices de codage G et de vérification H.

En reprenant le code cyclique (7,4) de l'exemple ci-dessus, on déduit la matrice de vérification H en utilisant la méthode décrite pages 13 et 14 :

$$H(7,4) = \begin{vmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{vmatrix}$$

$$S_{10} \quad 5 \quad 7 \quad 6 \quad 3 \quad 4 \quad 2 \quad 1$$

On reconnaît la forme générale (sous-matrices de parité et diagonale). On en déduit :

$$G_{(7,4)} = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{vmatrix}$$

III.7) Corrections d'erreurs multiples

Les méthodes de décodage décrites jusqu'ici ne permettent de corriger qu'une seule erreur dans chaque bloc reçu.

Pour augmenter cette capacité il faut augmenter la proportion de bits de contrôle. Le nombre de combinaisons du syndrome est : 2^{n-k}. Pour que la correction soit possible, chacun de ces syndromes ne doit être obtenu qu'à partir d'un vecteur d'erreur unique. Dans ce cas, l'implication peut être inversée : le récepteur peut évaluer le vecteur d'erreur à partir du syndrome.

- Pour corriger 1 erreur, il faut respecter : $2^{n-k} - 1 \geq n$ (voir codes de Hamming page 14)
- Pour corriger 1 ou 2 erreurs en positions quelconques, il faut respecter :
  $2^{n-k} - 1 \geq n+(n-1)+(n-2)+\ldots+(n-n+1)+(n-n) = n+n \bullet n - (1+2+3+ \ldots + n) = n \bullet (n+1)/2$
- Pour corriger 1 erreur isolée ou 2 erreurs groupées, il faut respecter :
  $2^{n-k} - 1 \geq n + n - 1 = 2n - 1$
- Pour corriger 1 à 3 erreurs en positions quelconques, il faut respecter :
  $2^{n-k} - 1 \geq n \bullet (n+1)/2 + n \bullet n \bullet n + \ldots$
- Pour corriger 1 à 3 erreurs groupées, il faut respecter :
  $2^{n-k} - 1 \geq n + (n - 1) + (n - 2) = 3n - 3$

Conclusion : la correction des erreurs isolées conduit rapidement à un nombre de bits de contrôle excessif.

Exemples :

- pour n = 16 :
  - correction d'1 erreur isolée $\Rightarrow 2^{n-k}-1 \geq 16 \Rightarrow$ **n-k ≥ 5 et k ≤ 11**
  - correction de 2 erreurs isolées $\Rightarrow 2^{n-k}-1 \geq 136 \Rightarrow$ **n-k ≥ 8 et k ≤ 8**

  Il y a déjà plus de bits de contrôle que de bits utiles !
- pour n = 26 (code RDS par exemple)
  - correction d'1 erreur isolée $\Rightarrow 2^{n-k}-1 \geq 26 \Rightarrow$ **n-k ≥ 5 et k ≤ 21**
  - correction de 2 erreurs isolées $\Rightarrow 2^{n-k}-1 \geq 351 \Rightarrow$ **n-k ≥ 9 et k ≤ 17**
  - paquet d'erreurs de longueur ≤ 5 $\Rightarrow 2^{n-k}-1 \geq 26+25+24+23+22 = 120$ soit **n-k ≥ 7**

Attention, les conditions établies ci-dessus sont nécessaires mais pas suffisantes.

III.8) Longueur du code

La quantité $2^{n-k}-1$ est la **longueur du code**.

La plupart des codes cycliques n'exploitent pas toute la longueur du code pour corriger les erreurs car cela conduirait à des circuits très complexes. On les qualifie de **codes cycliques abrégés**.

III.9) Réalisation des codeurs et décodeurs

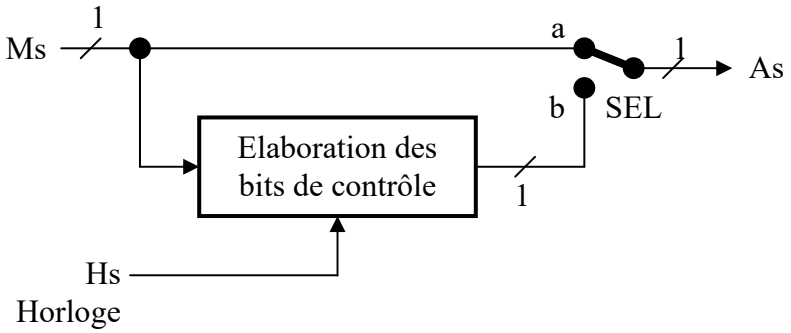
Les codeurs et les décodeurs sont généralement réalisés par des structures matérielles pour soulager le processeur de communication. Mais dans certains cas, on pourra préférer une structure logicielle.

Les structures matérielles sont très fréquemment basées sur des circuits séquentiels car :

- les blocs sont toujours transmis au format série,
- la réalisation est simple et facile à implanter dans des PLD.

III.9.1) Codage séquentiel

Principe général :



Les k bits du symbole M sont transmis en série sur Ms en synchronisme avec l'horloge. Le premier bit est toujours le poids le plus fort du symbole, à savoir m_{k-1}.

- Première étape :
  L'inverseur SEL est en position "a" pour transmettre les bits d'information utile sur le canal.
  Pendant ce temps, le diviseur élabore les (n-k) bits de contrôle à partir des k bits du symbole injectés au format série synchrone.
- Deuxième étape :
  Après k impulsions d'horloge, le commutateur est placé en position "b".
  Le codeur fourni alors les bits de contrôle sur les (n-k) impulsions d'horloge suivantes.

Le chronogramme du signal As est donc le suivant :

m_{k-1}	…	m_0	ρ_{n-k-1}	…	ρ_0

$t \rightarrow$

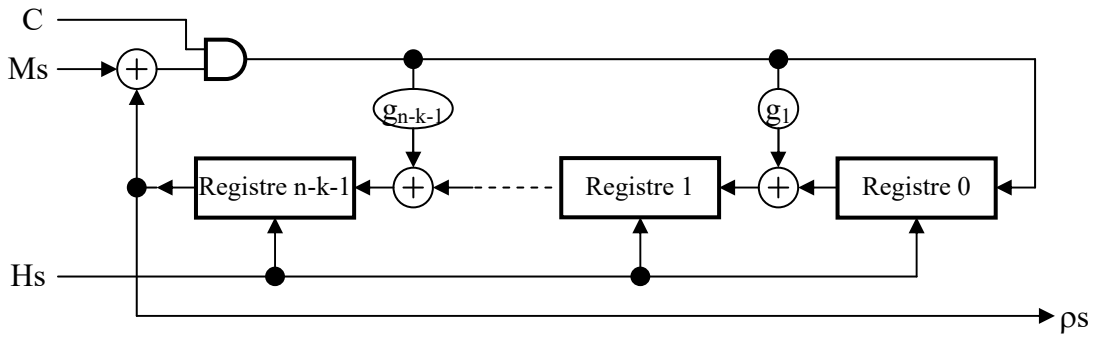
Elaboration des bits de contrôle :

La structure proposée réalise la division de polynôme définie au chapitre III.4.2 :

$$\frac{X^{n-k} \bullet M(X)}{G(X)}$$

Le reste ρ(X) de cette division donne les états des (n-k) bits de contrôle.

Schéma du codeur



⊕ = ou exclusif (g_i) = multiplication binaire par g_i

Principe :

Rappels : Polynôme associé au reste : $\rho(X) = \rho_{n-k-1} \bullet X^{n-k-1} + \ldots + \rho_1 \bullet X^1 + \rho_0$

Polynôme générateur : $G(X) = X^{n-k} + g_{n-k-1} \bullet X^{n-k-1} + \ldots + g_1 \bullet X^1 + 1$

En posant : $\rho'(X) = \rho_{n-k-2} \bullet X^{n-k-2} + \ldots + \rho_1 \bullet X^1 + \rho_0$

$G'(X) = g_{n-k-1} \bullet X^{n-k-1} + \ldots + g_1 \bullet X^1 + 1$

on déduit : $G(X) = G'(X) + X^{n-k}$ ou $X^{n-k} = G(X) + G'(X)$

$\rho(X) = \rho'(X) + \rho_{n-k-1} \bullet X^{n-k-1}$

En multipliant par X les 2 expressions de cette dernière égalité, on obtient :

$X \bullet \rho(X) = X \bullet \rho'(X) + \rho_{n-k-1} \bullet X^{n-k}$

$X \bullet \rho(X) = X \bullet \rho'(X) + \rho_{n-k-1} \bullet [G(X) + G'(X)]$

$X \bullet \rho(X) = \rho_{n-k-1} \bullet G(X) + X \bullet \rho'(X) + \rho_{n-k-1} \bullet G'(X)$ (1)

On constate que le degré de $[X \bullet \rho'(X) + \rho_{n-k-1} \bullet G'(X)]$ est toujours ≤ n-k-1

⇒ **Le reste de la division de $X \bullet \rho(X)$ par G(X) est**

$X \bullet \rho'(X) + \rho_{n-k-1} \bullet G'(X)$ **(2)**

Soient P(X) un polynôme de degré ≤ (n-1) et ρ(X) le reste de la division de P(X) par G(X) :

$P(X) = Q(X) \bullet G(X) + \rho(X)$ (3)

Supposons qu'à l'instant considéré, les registres du diviseur contiennent le reste $\rho(X)$. A l'impulsion d'horloge suivante, on introduit un nouveau bit m_i du symbole M(X). Le polynôme associé au nouveau dividende est :

$$P_1(X) = P(X)\bullet X + m_i \bullet X^{n-k} \qquad (4)$$

Note : la pondération (n-k) de m_i est due au niveau d'injection de Ms.

On cherche maintenant à exprimer le nouveau reste $\rho_1(X)$ de la division de $P_1(X)$ par G(X) en fonction du précédent $\rho(X)$ pour justifier la structure utilisée.

D'après la relation (3) : $X\bullet P(X) = X\bullet Q(X) \bullet G(X) + X\bullet\rho(X)$ (5)

et, d'après (1) : $X\bullet P(X) = X\bullet Q(X) \bullet G(X) + \rho_{n-k-1}\bullet G(X) + X\bullet\rho'(X) + \rho_{n-k-1}\bullet G'(X)$

soit : $X\bullet P(X)+ m_i \bullet X^{n-k} = G(X)\bullet[X\bullet Q(X)+\rho_{n-k-1}]+X\bullet\rho'(X)+\rho_{n-k-1}\bullet G'(X)+ m_i \bullet X^{n-k}$

Or $X^{n-k} = G(X) + G'(X)$ (voir ci-dessus)

On en déduit : $X\bullet P(X)+ m_i \bullet X^{n-k} = G(X)\bullet[X\bullet Q(X)+\rho_{n-k-1}+ m_i]+X\bullet\rho'(X)+G'(X)\bullet(\rho_{n-k-1}+ m_i)$

Le degré de $[X\bullet\rho'(X) + G'(X)\bullet(\rho_{n-k-1} + m_i)]$ est toujours inférieur à (n-k-1), c'est donc bien le reste de $X\bullet P(X)+ m_i \bullet X^{n-k}$

Conclusion :

Si $\rho(X)$ est le reste de la division de P(X) par G(X), alors le reste $\rho_1(X)$ de la division de $P_1(X) = X\bullet P(X) + m_i \bullet X^{n-k}$ par G(X) est :

$$\rho_1(X) = X\bullet\rho'(X) + G'(X)\bullet(\rho_{n-k-1} + m_i)$$

Ce traitement est réalisé par le circuit proposé. En effet, supposons que les registres contiennent les états de $\rho(X)$ à un instant donné. Les entrées "datas" des registres valent alors :

$$D_i(X) = \boxed{X\bullet\rho'_i(X)} + \boxed{(Ms + \rho_{i/n-k-1})}\bullet\boxed{G'(X)}$$

Chaînage des registres	Porte ⊕	G(X) avec bit MSB à "0"

A l'impulsion d'horloge suivante : $\rho_{i+1}(X) = D_i(X)$

On constate que $\rho_{i+1}(X) = \rho_1(X) \Rightarrow$ *le circuit proposé réalise bien une division séquentielle polynomiale par G(X).*

Fonctionnement :

- les registres sont initialement à "0"
- le signal C est à "1" pendant les k premières impulsions d'horloge pour rendre le diviseur opérationnel.
  - Avant la 1° impulsion d'horloge :

    $P_0(X) = 0$

    $\rho_0(X)$ = reste de $P_0(X)$ /G(X) = 0

    $As = M_{k-1}$

- Après la 1° impulsion d'horloge et avant la 2° :
  $P_1(X) = X \bullet P_0(X) + M_{k-1} \bullet X^{n-k} = M_{k-1} \bullet X^{n-k}$
  $\rho_1(X) = X \bullet \rho'_0(X) + G'(X) \bullet (0 + M_{k-1}) = G'(X) \bullet M_{k-1}$ = reste de $P_1(X)/G(X)$
  $As = M_{k-2}$
- Après la 2° impulsion d'horloge et avant la 3° :
  $P_2(X) = X \bullet P_1(X) + M_{k-2} \bullet X^{n-k} = M_{k-1} \bullet X^{n-k+1} + M_{k-2} \bullet X^{n-k}$
  $\rho_2(X) = X \bullet \rho'_1(X) + G'(X) \bullet (\rho_{1,n-k-1} + M_{k-2})$ = reste de $P_2(X)/G(X)$
  $As = M_{k-3}$
- …
- Après la (k-1)° impulsion d'horloge et avant la dernière :
  $P_{k-1}(X) = M_{k-1} \bullet X^{n-k+k-2} + M_{k-2} \bullet X^{n-k+k-3} + \ldots + M_1 \bullet X^{n-k}$
  $\rho_{k-1}(X)$ = reste de $P_{k-1}(X)$
  $As = M_0$
- Après la k° impulsion d'horloge :
  $P_k(X) = M_{k-1} \bullet X^{n-k+k-1} + M_{k-2} \bullet X^{n-k+k-2} + \ldots + M_1 \bullet X^{n-k+1} + M_0 \bullet X^{n-k}$
  $= M_{k-1} \bullet X^{n-1} + M_{k-2} \bullet X^{n-2} + \ldots + M_1 \bullet X^{n-k+1} + M_0 \bullet X^{n-k}$
  $= M(X) \bullet X^{n-k}$
  $\rho_k(X)$ = reste de $P_k(X)/G(X)$ soit reste de $M(X) \bullet X^{n-k}/G(X)$
  $As = 0$

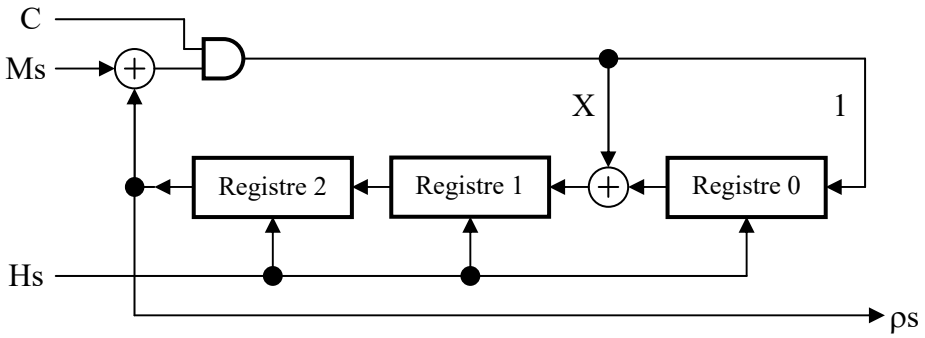
Ainsi, après la $k^{ième}$ impulsion d'horloge, les registres contiennent le reste de :

$$\frac{X^{n-k} \bullet \left(X^{k-1} \bullet m_{k-1} + X^{k-2} \bullet m_{k-2} + \ldots + X^1 \bullet m_1 + m_0\right)}{G(X)}$$

c'est à dire **les (n - k) bits de contrôle : ρ_{n-k-1} à ρ_0.**

- le signal C est à "0" pendant les (n-k) impulsions suivantes. Le diviseur n'est plus opérationnel et les registres constituent un simple registre à décalage.
  Ces (n-k) impulsions d'horloge vident le registre à décalage en commençant par ρ_{n-k-1} et jusqu'à ρ_0 ce qui correspond au fonctionnement désiré.

Exemple : soit le code cyclique (7,4) défini par le polynôme générateur $G(X) = X^3 + X + 1$



Avec le schéma de principe général, on obtient les blocs du code suivants :

Symboles				Blocs du code						
m_3	m_2	m_1	m_0	a_6	a_5	a_4	a_3	$a_2=\rho_2$	$a_1=\rho_1$	$a_0=\rho_0$
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	1	1
0	0	1	0	0	0	1	0	1	1	0
0	0	1	1	0	0	1	1	1	0	1
0	1	0	0	0	1	0	0	1	1	1
0	1	0	1	0	1	0	1	1	0	0
0	1	1	0	0	1	1	0	0	0	1
0	1	1	1	0	1	1	1	0	1	0
1	0	0	0	1	0	0	0	1	0	1
1	0	0	1	1	0	0	1	1	1	0
1	0	1	0	1	0	1	0	0	1	1
1	0	1	1	1	0	1	1	0	0	0
1	1	0	0	1	1	0	0	0	1	0
1	1	0	1	1	1	0	1	0	0	1
1	1	1	0	1	1	1	0	1	0	0
1	1	1	1	1	1	1	1	1	1	1

III.9.2) Décodage séquentiel simple

Comme cela a déjà été vu précédemment et à plusieurs reprises, la première étape du décodage consiste à élaborer le syndrome. Dans les codes cycliques, le syndrome S(X) est le reste de la division polynomiale du bloc reçu R(X) avec G(X). Le chapitre III.5 (page 18) démontre cette affirmation.

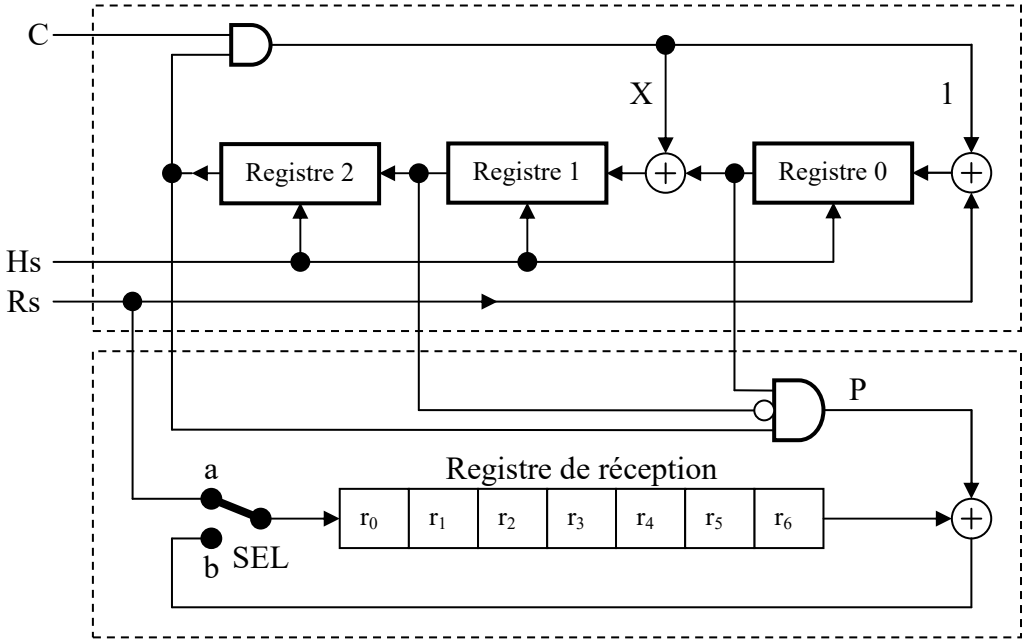
Ce calcul peut être aisément réalisé avec une structure très proche de celle utilisée dans le codeur. En effet, on réalise ici aussi une division par G(X).

Pour compléter le décodage, cette opération de division doit être associée à une fonction d'élaboration du vecteur d'erreur à partir du syndrome pour corriger les erreurs identifiées. Cette fonction peut devenir très complexe si les capacités de correction du code sont élevées.

On s'intéresse, dans un premier temps, aux décodeurs ayant une capacité de correction de **1 erreur seulement**. Le chapitre suivant sera consacré à la correction des paquets d'erreurs.

Pour ne pas alourdir la notation, l'étude est faite sur un exemple, à savoir le décodage du code cyclique (7,4) défini par $G(X) = X^3 + X + 1$.

Schéma complet



Rs : bloc reçu au format série (bit MSB d'abord)
Hs : horloge bits

La partie supérieure du schéma représente le diviseur séquentiel par G(X) et la partie inférieure, le registre de réception et le circuit de correction.

A noter que chaque bit r_i du bloc est ici injecté à l'entrée du registre d'indice "0" du syndrome. Il n'y a donc pas de multiplication par X^{n-k} comme c'était le cas dans le codeur séquentiel.

Fonctionnement :

On suppose qu'une simple erreur apparaît en position "i" dans le bloc reçu R. Le vecteur d'erreur correspondant a donc toutes ses positions à "0", sauf la position "i" qui est à "1". Ce vecteur est associé au polynôme :

$$E(X) = X^i \text{ avec } 0 \leq i \leq n-1$$

- **Première étape** : C=1 et SEL en position "a"
  Les "n" (7 dans l'exemple) bits du bloc reçu sont fournis sur Rs au format série en synchronisme avec l'horloge.
  Le décodeur calcule le syndrome S(X) en divisant le bloc reçu R(X) par G(X). Cette opération est réalisée par le diviseur séquentiel dans lequel on injecte sous forme série les n bits du bloc R au rythme de l'horloge.
  A la fin de cette étape :
  - Le registre de réception contient les 7 bits du bloc reçu. Un de ceux-ci est faux.
  - Les registres du diviseur contiennent les bits du syndrome.

  L'état du syndrome obtenu permet d'identifier la position "i" en erreur. Un banal décodeur combinatoire associé à des inverseurs commandés pourrait être utilisé pour corriger l'état du bit altéré (voir page 13). En fait la structure séquentielle mise en œuvre pour calculer le syndrome permet de réaliser un circuit de correction encore plus simple.

- **Deuxième étape** : C=1 et SEL en position "b"
  Le diviseur séquentiel est toujours opérationnel.
  On applique n impulsions d'horloge supplémentaires au diviseur séquentiel, l'entrée Rs étant maintenue à "0".
  A la première impulsion on peut considérer que le bloc reçu est décalé à gauche et a maintenant (n+1) bits, soit 8 pour l'exemple, et que l'erreur est en position (i +1). Le nouveau syndrome obtenu est donc le reste de la division de $E(X) = X^{i+1}$ par G(X). Il indique alors la position (i +1).

  On continue d'activer le circuit de division jusqu'à ce que le syndrome indique la dernière position du bloc, soit (n-1), c'est à dire 6 dans notre exemple. Le nombre d'impulsions d'horloge supplémentaire est de (n-1-i). Ce nombre peut varier entre 0 (position 6 en erreur) et 6 (position 0 en erreur).
  On utilise un décodeur combinatoire pour détecter le syndrome correspondant à une erreur en dernière position. Pour le code cyclique de notre exemple, le signal P passe à "1" si le syndrome prend la valeur 5, donc quand l'erreur est en position 6.

  Pour chacune de ces impulsions d'horloge supplémentaire, le registre de réception du circuit de correction est bouclé sur lui-même via une porte "ou exclusif" de correction commandée par P. Ainsi, si l'erreur est en position "i" :
  - P est à "0" pendant les (n-1-i-1) premières impulsions d'horloge supplémentaire et passe à "1" à l'impulsion suivante. Le bit r_i est alors dans la dernière position droite du registre de réception.
  - le bit ri est inversé, c'est à dire corrigé, avant d'être réinjecté dans le registre de réception.

  Les (i +1) impulsions d'horloge restantes (sur n) replacent les bits du registre de réception à leurs places, le **bit r_i étant corrigé**.

Remarque : l'économie de circuits autorisé par cette méthode de décodage devient nettement plus évidente quand la longueur des blocs est plus grande que l'exemple utilisé. Par contre, cette économie est faite au détriment de la vitesse, car il faut doubler les impulsions d'horloge pour réaliser la correction. Un circuit combinatoire réaliserait cette fonction beaucoup plus rapidement.

III.9.3) Décodage séquentiel des paquets d'erreurs (application au code RDS)

Certains codes cycliques sont très performants pour corriger les paquets d'erreurs.

Le support de l'étude qui suit est le code cyclique utilisé pour protéger les données numériques transmises dans le système RDS. Il est capable de :

- détecter toutes les erreurs binaires simples et doubles dans un bloc de 26 bits,
- détecter toute salve d'erreurs de longueur inférieure à 10 bits,
- détecter 99,8% des salves d'erreurs de longueur inférieure à 11 bits,
- détecter 99,9% des salves d'erreurs de longueur supérieure,
- corriger des paquets d'erreurs de longueur inférieure ou égale à 5 bits.

Le code de protection RDS est basé sur le code cyclique (341,331) défini par le polynôme générateur suivant :

$$\mathbf{G(X) = X^{10} + X^8 + X^7 + X^5 + X^4 + X^3 + 1}$$

Un bloc du code d'origine est donc constitué de 331 bits d'information utile et de 10 bits de contrôle.

Le code réellement utilisé est le code cyclique (341,331) abrégé à 26 bits dans le but de réduire la complexité des codeurs et décodeurs.

L'étude qui suit traite les décodages des codes cycliques d'origine et abrégé. Le codage ne pose pas de problème particulier, on peut, par exemple, utiliser la structure du diviseur séquentiel vu au chapitre III.9.1 (page 22).

III.9.3.1) Outils théoriques

Rappels de notation :

R(X) : polynôme associé au bloc reçu R de degré ≤ n-1,
E(X) : polynôme associé au vecteur d'erreur de degré ≤ n-1,
G(X) : polynôme générateur de degré n-k,
S(X) : polynôme associé au syndrome de degré ≤ n-k-1.

La deuxième étape élabore la valeur la plus probable du vecteur d'erreur E(X) pour finalement corriger le bloc reçu en calculant : Rc(X) = R(X) + E(X).
L'étude qui suit montre comment est utilisé le diviseur séquentiel pour réaliser cette fonction.

Principe d'élaboration du vecteur d'erreur

La fonction du décodeur est d'élaborer la valeur la plus probable du vecteur d'erreur E(X).

Première étape :

La **première étape** du décodage consiste toujours à élaborer le syndrome S(X) en calculant le reste de la division polynomiale de R(X) par G(X). On utilisera pour cela la structure classique d'un diviseur séquentiel (le codeur utilise d'ailleurs la même):

$$R(X) = Q(X) \bullet G(X) + S(X)$$

Cette première opération nécessite donc 341 (n) impulsions d'horloge.

Le syndrome est aussi le reste de la division de E(X) par G(X) (voir page 19) :

$$E(X) = Q_E(X) \bullet G(X) + S(X)$$

Imaginons un vecteur d'erreur $E^*(X)$ tel que :

$$E^*(X) = S(X)$$

Ce vecteur d'erreur a les particularités suivantes :

- le degré est ≤ n-k-1 = 9 (le nombre d'erreurs est ≤ 10)
- les k = 331 bits de poids forts sont à "0"

On en déduit que :

- **le syndrome de $E^*(X)$ est S(X)** , à cause du degré de $E^*(X)$,
- et que **E(X) et $E^*(X)$ ont le même syndrome**.

Si le poids de Hamming de $E^*(X)$ est inférieur ou égal à la capacité de correction "t" du code, alors :

- *le poids de Hamming de son syndrome S(X) est aussi ≤ t (car $S(X) = E^*(X)$)*
- *E(X) = $E^*(X)$ car il ne peut exister deux vecteurs d'erreur différents ayant le même syndrome (voisinages distincts).*

On en déduit une méthode de correction désignée sous le nom "error trapping" décrite ci-après.

Deuxième étape :

Le diviseur séquentiel par G(X) est toujours opérationnel dans cette 2° étape. L'entrée "datas" est simplement maintenue à "0".

Un nouveau syndrome est donc calculé à chaque impulsion d'horloge. A la $i^{ème}$ impulsion, $S_i(X)$ est le reste de la division de $X^i \bullet R(X)$ par G(X) :

$$X^i \bullet R(X) = Q_i \bullet G(X) + S_i(X)$$

Une fonction évalue le poids de Hamming de chaque syndrome $S_i(X)$. Si ce poids est inférieur ou égal à la capacité de correction t du code, la correction est possible et est réalisée de la manière suivante :

- A la $i^{ème}$ impulsion : $S_i(X)$: syndrome de $R_i(X) = X^i \bullet R(X)$
  : syndrome de $E_i(X) = X^i \bullet E(X)$
  $R_i(X)$: R(X) avec i transpositions circulaires à gauche
  $E_i(X)$: E(X) avec i transpositions circulaires à gauche
- Si le poids de Hamming de $S_i(X)$ est ≤ t, alors, d'après la règle ci-dessus (en italiques) :
  $E_i(X) = E^*_i(X) = S_i(X) \Rightarrow X^i \bullet E(X) = S_i(X)$
  $\rightarrow X^{n-i} \bullet X^i \bullet E(X) = X^{n-i} \bullet S_i(X)$
  Or $X^n \bullet E(X) = E(X)$ car cette opération consiste en une rotation à gauche de n bits sur un mot de n bits.
  Soit finalement : **$E(X) = X^{n-i} \bullet S_i(X)$, le vecteur d'erreur est élaboré.**

  Pour corriger le bloc reçu il suffit de réaliser : Rc(X) = R(X) + E(X)

<u>Cas particulier des paquets d'erreurs</u>

Dans le cas de la correction de paquets d'erreurs de longueur inférieure ou égale à L, E(X) peut être mise sous la forme (on suppose que L ne dépasse pas la capacité "t" de correction du code) :

$$E(X) = X^j \bullet B(X) \text{ avec } B(X) = b_{L-1} \bullet X^{L-1} + \ldots + b_1 \bullet X^1 + b_0 \bullet X^0 \text{ le modèle d'erreur}$$

Supposons qu'à la $i^{ème}$ impulsion d'horloge, le bit altéré b_{L-1} se trouve à la position (n-k-1) de $E_i(X)$, soit :

$$E^*_i(X) = X^{n-k-L} \bullet B(X) \text{ de degré} \leq n-k-1 \quad \text{(degré d'un syndrome)}$$

Dans ce cas, le syndrome $S_i(X)$ de $E^*_i(X)$ est égal à $X^{n-k-L} \bullet B(X)$ et a nécessairement un poids de Hamming inférieur ou égal à L (donc à "t"). On peut alors appliquer la méthode de correction décrite ci-dessus :

$$S_i(X) = E^*_i(X) = X^{n-k-L} \bullet B(X) = E_i(X)$$

- Le syndrome est donc égal au modèle d'erreur décalé de (n-k-L) positions à gauche,
- Il en résulte que les (n-k-L) premières positions (de poids faible) du syndrome sont toutes à "0".

Cette dernière caractéristique permet d'élaborer un circuit de correction relativement simple :

- à chaque impulsion d'horloge dans la 2° étape, on détecte si les (n-k-L) bits de poids faible du syndrome $S_i(X)$ sont à "0". Cette fonction est facilement réalisée par une porte OU de (n-k-L) entrées.
- si c'est le cas, le modèle d'erreur B(X) est contenu dans les L bits de poids fort du syndrome $S_i(X)$.
- on calcule alors $E(X) = X^{n-i} \bullet S_i(X)$.
- on corrige le bloc reçu en réalisant $Rc(X) = R(X) + E(X)$

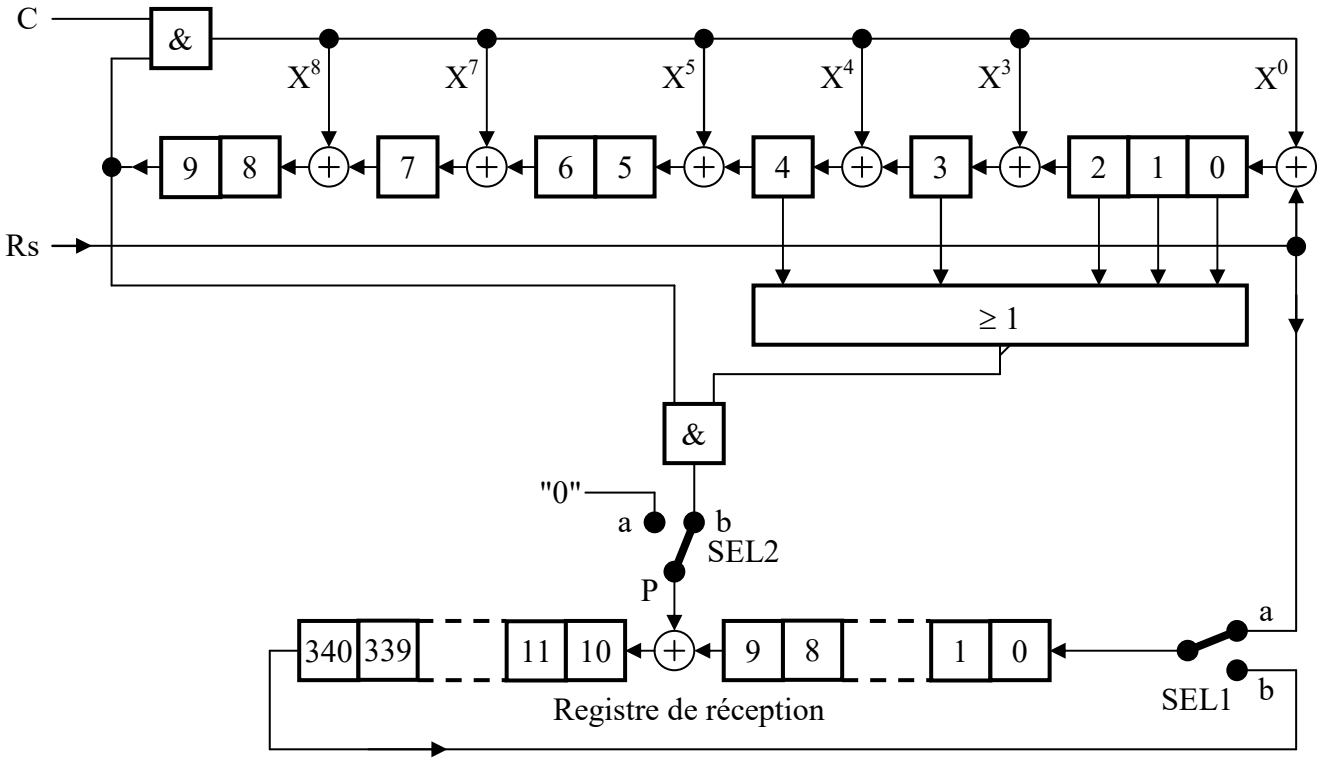
III.9.3.2) Décodage du code cyclique (341,331) du système RDS

Le circuit séquentiel proposé utilise les résultats de l'étude théorique ci-dessus.

Le code cyclique (341,331) a les caractéristiques suivantes :

- longueur des blocs : n = 341
- nombre de bits utiles : k = 331
- nombre de bits de contrôle : n-k = 10
- capacité de correction : un paquet d'erreurs de longueur ≤ 5 (soit t = 5)
- polynôme générateur : $G(X) = X^{10} + X^8 + X^7 + X^5 + X^4 + X^3 + 1$

Schéma du décodeur



Note : l'horloge Hs n'a pas été représentée pour la clarté du schéma.

On reconnaît la structure d'un diviseur séquentiel déjà vue au chapitre III.9.1

Fonctionnement :

- Première étape : C="1" et SEL1 et SEL2 en position "a".

  Les 341 bits du bloc R(X) sont fournis sur Rs au format série synchrone, en commençant par les poids forts.

  A la fin de cette étape :
  - les 10 registres du diviseur contiennent le syndrome S(X) de R(X),
  - le registre de réception contient les 341 bits de R(X).

  Imaginons que le bloc reçu soit entaché d'un paquet d'erreurs de longueur inférieure ou égale à 5 en position "j". Le vecteur d'erreur associé est alors :

  $$E(X) = X^j \bullet B(X) \text{ avec } B(X) = b_{L-1} \bullet X^{L-1} + \ldots + b_1 \bullet X^1 + b_0 \bullet X^0 \quad \text{le modèle d'erreur}$$
  $$B(X) = b_4 \bullet X^4 + b_3 \bullet X^3 + b_2 \bullet X^2 + b_1 \bullet X^1 + b_0 \bullet X^0$$

  Le syndrome S(X) associé est tel que :

  $$X^j \bullet B(X) = Q_E(X) \bullet G(X) + S(X)$$

- Deuxième étape : SEL1 en position "b". L'entrée Rs est maintenue à "0".

  L'inverseur SEL2 passe en position "b" et le signal C reste à l'état "1" jusqu'à la détection de la salve d'erreurs. Le diviseur est donc toujours opérationnel.
  Le registre de réception est bouclé sur lui-même avec un circuit de correction entre les positions 9 et 10.

  On continue d'appliquer des impulsions d'horloge jusqu'à ce que les 5 bits de poids faible du syndrome soient à "0". Soit "i", le nombre d'impulsions supplémentaires pour obtenir ce résultat.

  Les registres du diviseur contiennent alors le syndrome $S_i(X)$ de $E_i(X) = X^i \bullet E(X)$,
  et le registre de réception : $R_i(X) = X^i \bullet R(X)$ par transpositions circulaires.

  D'après l'étude théorique, les 5 registres de poids fort du syndrome contiennent le modèle d'erreur B(X), soit :

  $$S_i(X) = X^{n-k-L} \bullet B(X) = X^5 \bullet (b_4 \bullet X^4 + b_3 \bullet X^3 + b_2 \bullet X^2 + b_1 \bullet X^1 + b_0 \bullet X^0)$$
  $$S_i(X) = b_4 \bullet X^9 + b_3 \bullet X^8 + b_2 \bullet X^7 + b_1 \bullet X^6 + b_0 \bullet X^5$$

  On constate bien que les 5 bits de poids faible de $S_i(X)$ sont à "0" (indices 4 à 0).

  Au même moment : $E_i(X) = S_i(X)$, ce qui signifie que la salve d'erreur se trouve entre les indices 9 et 5 de $R_i(X)$.

  A partir de ce moment, le signal logique C passe à "0" pour transformer le diviseur séquentiel en un registre à décalage à gauche non bouclé.

  Durant les 5 impulsions d'horloge suivantes, le signal P prendra les états successifs des bits b_4 à b_0 du modèle d'erreur pour corriger les bits d'indice 9 à 5 de $R_i(X) = X^i \bullet R(X)$.

On termine le traitement en appliquant le reliquat d'impulsions d'horloge (sur n) avec SEL2 en position "a", pour que le bloc reçu et corrigé se retrouve en ordre dans le registre de réception.

III.9.3.3) Décodage du code abrégé (26,16) du système RDS

Le code de protection utilisé par le système RDS est issu du code cyclique (341,331) décrit ci-dessus. Il utilise le même polynôme générateur :

- longueur des symboles : k = 16 bits,
- longueur des blocs du code : n = 26 bits,
- nombre des bits de contrôle : n - k = 10,
- polynôme générateur : $G(X) = X^{10} + X^8 + X^7 + X^5 + X^4 + X^3 + 1$

Notes sur le codeur :

La structure du codeur reste identique. En effet, on peut imaginer que l'on utilise le code cyclique (341,331) et que les 16 bits du symbole sont placés dans les 16 positions de poids faible des 331 bits d'information utile du code d'origine; les 315 bits de poids fort étant tous à zéro.

Ainsi, le diviseur séquentiel déduit du schéma général de la page 33 donnera les mêmes 10 bits de contrôle :

- si on injecte les 331 bits du symbole du code cyclique (341,331) avec 331 impulsions d'horloge. Les 315 premières impulsions ne changent pas le contenu initial du registre car elles correspondent à des bits de données à "0".
- si on injecte les 16 bits du symbole du code abrégé avec 16 impulsions d'horloge.

On préférera évidemment la deuxième méthode !

Décodeur :

On pourrait utiliser la structure classique du diviseur séquentiel pour calculer le syndrome du bloc R(X) de 26 bits reçu.

Par contre, le circuit de correction proposé au chapitre III.9.3.2 ne fonctionne plus car il exploite les *propriétés cycliques* du code d'origine. Et le code abrégé *n'est plus cyclique*.

Le problème peut être contourné en remarquant que seule la correction des 16 bits d'information utile est intéressante.

On peut imaginer de placer les 16 bits du bloc reçu R(X) dans un bloc virtuel de 341 bits comme s'il faisait partie du code d'origine :

	⌊ 0	0	…	0	r_{25}	r_{24}	…	r_{10}	r_9	r_8	…	r_1	r_0 ⌋
Indice	340	339		26	25	24		10	9	8		1	0
					← 16 bits d'info utile →			← 10 bits de contrôle →					

En utilisant le circuit de correction du chapitre III.9.3.2, les dix premières impulsions d'horloge de la phase de correction servent à identifier et corriger une salve d'erreurs dans les 10 bits de contrôle, ce qui est inutile.

A cause de la permutation circulaire sur 341 bits, il faudrait appliquer 315 impulsions d'horloge pour que le bit r_{25} se trouve en position 0. Du point de vue de la détection et de la correction des erreurs, ces impulsions sont inutiles car les positions traitées sont toutes à zéro.

La détection et la correction d'une salve d'erreurs sur les 16 bits d'information utile s'effectuera sur les 16 dernières impulsions d'horloge. Mais auparavant, il a fallut appliquer 325 impulsions *inefficaces* mais *nécessaires* !

Un pré-traitement adéquat sur le bloc reçu permet de détecter et corriger la salve d'erreur sur l'information utile sans appliquer les 325 impulsions d'horloge préliminaires : il *suffit* de réaliser une permutation circulaire de 325 positions sur le bloc de 341 bits, soit l'opération polynomiale :

$$\mathbf{R(X)\bullet X^{N-k} = R(X)\bullet X^{325}} \quad \text{avec } N = 341 \text{ et } k = 16$$

Malheureusement ce traitement nécessite une structure matérielle assez complexe, surtout si on choisit un traitement combinatoire pour diminuer le temps de calcul.

Nous allons montrer qu'une astuce permet de contourner le problème avec peu de moyens matériels :

En divisant X^{N-k} par G(X), on obtient :

$X^{N-k} = Q_N(X)\bullet G(X) + P(X)$

Avec $Q_N(X)$: quotient de degré ≤ N-k-n+k = N-n

$P(X)$: reste de degré ≤ n-k-1

Soit $P(X) = X^{N-k} + Q_N(X)\bullet G(X)$

En divisant le produit $R(X)\bullet P(X)$ par G(X) on obtient :

$R(X)\bullet P(X) = Q_{RP}(X)\bullet G(X) + S_{RP}(X)$

Avec $R(X)\bullet P(X)$: dividende de degré ≤ n-1+n-k-1 = 2n-k-2

$Q_{RP}(X)$: quotient de degré ≤ 2n-k-2-n+k = n-2

$S_{RP}(X)$: reste de degré ≤ n-k-1

En remplaçant P(X) par son expression :

$X^{N-k}\bullet R(X) + R(X)\bullet Q_N(X)\bullet G(X) = Q_{RP}(X)\bullet G(X) + S_{RP}(X)$

soit $X^{N-k}\bullet R(X) = G(X)\bullet [\, R(X)\bullet Q_N(X) + Q_{RP}(X)] + S_{RP}(X)$

Avec $X^{N-k}\bullet R(X)$ de degré ≤ N+n-k-1

$R(X)\bullet Q_N(X) + Q_{RP}(X)$: quotient de degré ≤ n-1+N-n = N-1

$S_{RP}(X)$: reste de degré ≤ n-k-1

Conclusions :

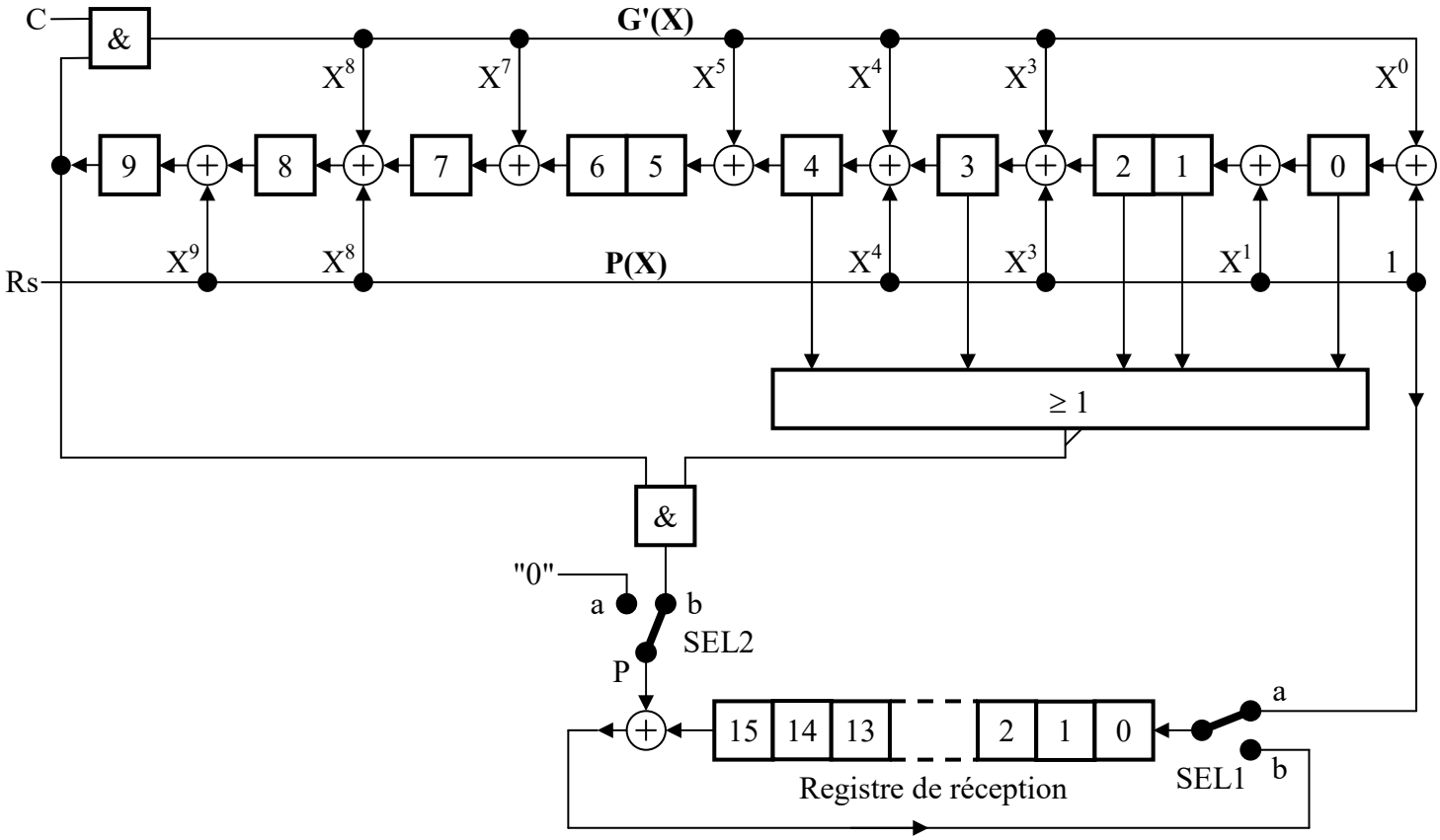
- **le reste $S_{RP}(X)$ de la division de $R(X)\bullet P(X)$ par G(X) est égal au reste de la division de $X^{N-k}\bullet R(X)$ par G(X)**
- **$S_{RP}(X)$ est le syndrome de $X^{N-k}\bullet R(X)$**

On en déduit la méthode de calcul du syndrome dans le code abrégé :

- On élabore le produit polynomial **$R(X)\bullet P(X)$**
  P(X) est le reste de la division polynomiale de X^{N-k} (= X^{325}) par G(X), soit :
  $$\mathbf{P(X) = X^9 + X^8 + X^4 + X^3 + X^1 + 1}$$
- On calcule ensuite le reste de la division de ce produit par G(X) :
  c'est le syndrome de $X^{N-k}\bullet R(X)$.

En fait, les opérations de multiplication et de division polynomiales sont réalisées simultanément comme le montre le schéma ci-dessous :

Schéma du décodeur du code cyclique abrégé RDS



Fonctionnement :

- Première étape : chargement du registre de réception et calcul du syndrome pendant les 26 premières impulsions d'horloge.

  Les registres du diviseurs sont initialement à "0".
  Les inverseurs SEL1 et SEL2 sont en position "a" et le signal C est à "1".
  Les 26 bits du bloc reçu (r_{25} à r_0) sont injectés sur Rs au rythme de l'horloge.

  Les 16 bits d'information utile se trouvent dans le registre de réception après la $16^{ième}$ impulsion d'horloge. Le registre de réception est alors bloqué pour les 10 impulsions restantes de cette première étape (cette fonction n'est pas représentée sur le schéma).

  Rappels :
  On note S(X) le polynôme associé au contenu des 10 registres du diviseur (partie supérieure du schéma).
  Soit N(X) le dividende de la division tel que : $N(X) = Q(X) \bullet G(X) + S(X)$
  - $G'(X) = G(X) + X^{n-k} = G(X) + X^{10} = X^8 + X^7 + X^5 + X^4 + X^3 + X^0$
  - Le reste de la division de $X \bullet S(X)$ par G(X) est : $X \bullet S'(X) + s_9 \bullet G'(X)$
    Avec $S'(X) = s_8 \bullet X^8 + s_7 \bullet X^7 + s_6 \bullet X^6 + s_5 \bullet X^5 + s_4 \bullet X^4 + s_3 \bullet X^3 + s_2 \bullet X^2 + s_1 \bullet X^1 + s_0 \bullet X^0$

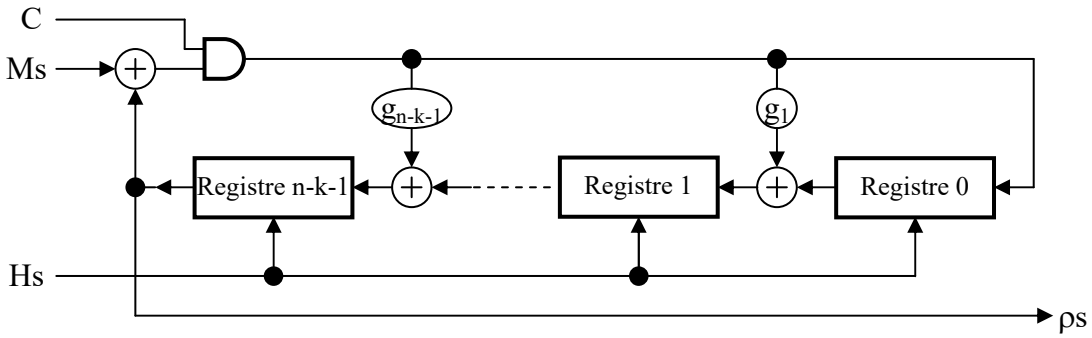
  Le tableau ci-dessous donne les contenus successifs de S(X) à chaque impulsion d'horloge. On en déduit, à chaque étape, le dividende N(X) correspondant.

Impuls. d'horloge	**Syndrome $S_i(X)$**	**Dividende $N_i(X)$**
Etat initial	$S_0(X) = 0$	$N_0(X) = 0$
1	$S_1(X) = X \bullet S'_0(X) + s_9^0 \bullet G'(X) + r_{25} \bullet P(X)$ $= r_{25} \bullet P(X)$	$N_1(X) = r_{25} \bullet P(X)$
2	$S_2(X) = X \bullet S'_1(X) + s_9^1 \bullet G'(X) + r_{24} \bullet P(X)$	$N_2(X) = X \bullet N_1(X) + r_{24} \bullet P(X)$ $= r_{25} \bullet X \bullet P(X) + r_{24} \bullet P(X)$
3	$S_3(X) = X \bullet S'_2(X) + s_9^2 \bullet G'(X) + r_{23} \bullet P(X)$	$N_3(X) = X \bullet N_2(X) + r_{23} \bullet P(X)$ $= r_{25} \bullet X^2 \bullet P(X) + r_{24} \bullet X^1 \bullet P(X) + r_{23} \bullet P(X)$
…	…	…
25	$S_{25}(X) = X \bullet S'_{24}(X) + s_9^{24} \bullet G'(X) + r_1 \bullet P(X)$	$N_{25}(X) = X \bullet N_{24}(X) + r_1 \bullet P(X)$ $= r_{25} \bullet X^{24} \bullet P(X) + \ldots + r_2 \bullet X^1 \bullet P(X) + r_1 \bullet P(X)$
26	$S_{26}(X) = X \bullet S'_{25}(X) + s_9^{25} \bullet G'(X) + r_0 \bullet P(X)$	$N_{26}(X) = X \bullet N_{25}(X) + r_0 \bullet P(X)$ $= r_{25} \bullet X^{25} \bullet P(X) + \ldots + r_1 \bullet X^1 \bullet P(X) + r_0 \bullet P(X)$ $= P(X) \bullet (r_{25} \bullet X^{25} + \ldots + r_1 \bullet X^1 + r_0)$ $= P(X) \bullet R(X)$

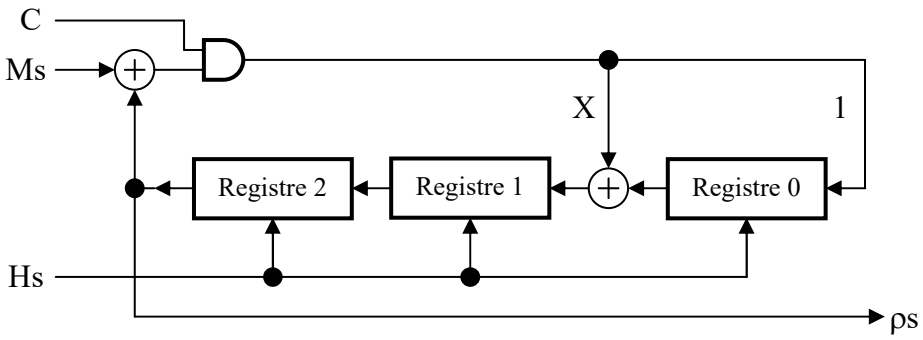
Conclusion :

A la 26^{ième} impulsion d'horloge, le syndrome de R(X)•P(X), donc de X^{N-k} •R(X) se trouve dans les 10 registres du diviseur. *CQFD.*

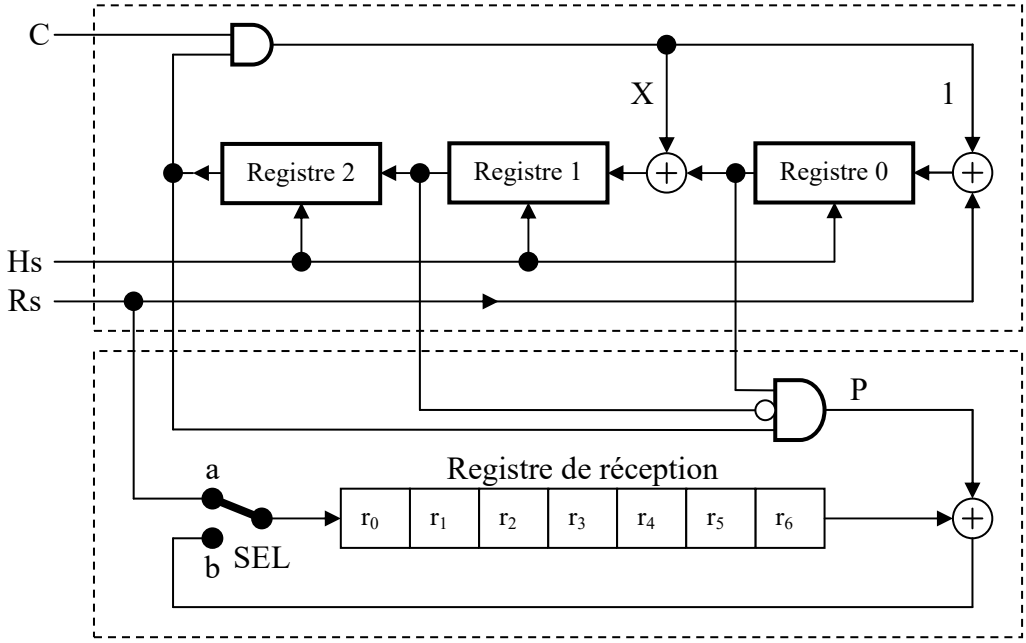
- Deuxième étape : correction du bloc reçu.
  Le principe de fonctionnement est identique à celui déjà décrit pour le code cyclique non abrégé (page 32). Les différences sont :
  - la taille du registre de réception (16 bits à la place de 341),
  - quand une salve d'erreurs est détectée, celle-ci se trouve dans les 5 bits de poids fort du registre de réception. En effet, grâce à la transposition circulaire X^{N-k} , le bit r_{25} se trouve en position 9 dans le bloc virtuel de 341 bits (25+325-341=9) au début de la phase de correction. Or, d'après l'étude théorique, quand les 5 bits de poids faible du syndrome sont à "0", la salve d'erreurs de longueur inférieure ou égale à 5 se trouve entre les indices 9 et 5 du bloc virtuel de 341 bits (voir page 32). La correction agit donc sur les bits d'information utile *dès le début* de cette 2° étape.
  - le nombre d'impulsions d'horloge nécessaire (16 maximum à la place de 341).



⊕ = ou exclusif (g_i) = multiplication binaire par g_i

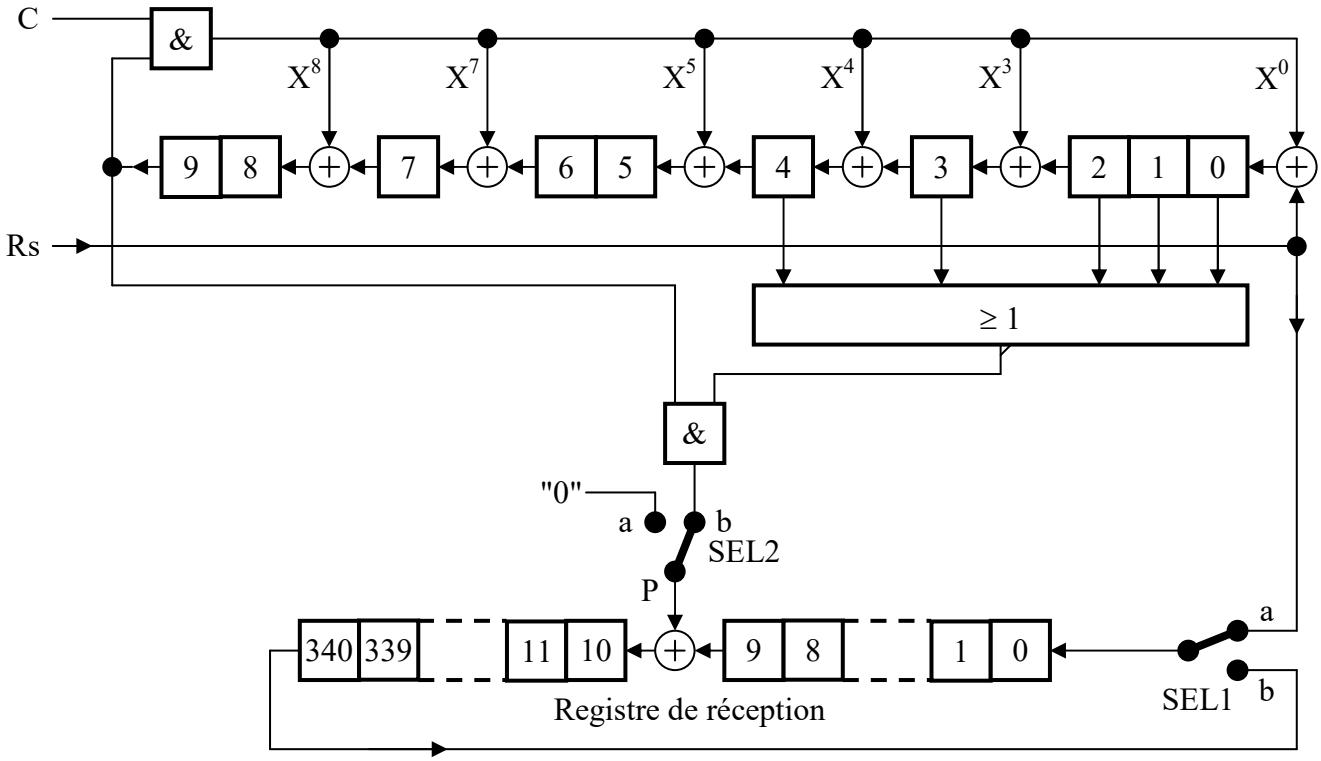


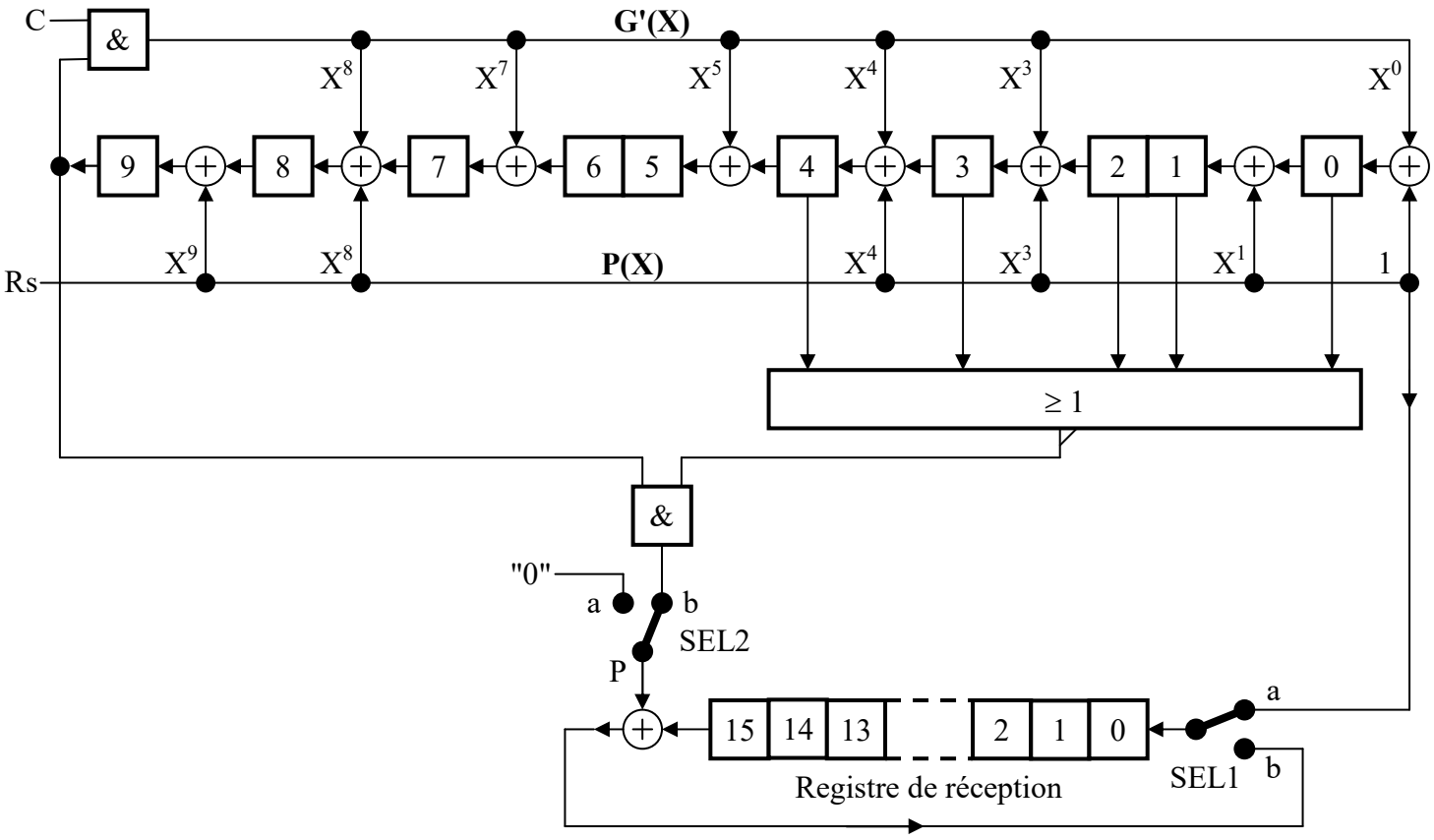
Symboles				Blocs du code						
m_3	m_2	m_1	m_0	a_6	a_5	a_4	a_3	$a_2=\rho_2$	$a_1=\rho_1$	$a_0=\rho_0$
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	1	1
0	0	1	0	0	0	1	0	1	1	0
0	0	1	1	0	0	1	1	1	0	1
0	1	0	0	0	1	0	0	1	1	1
0	1	0	1	0	1	0	1	1	0	0
0	1	1	0	0	1	1	0	0	0	1
0	1	1	1	0	1	1	1	0	1	0
1	0	0	0	1	0	0	0	1	0	1
1	0	0	1	1	0	0	1	1	1	0
1	0	1	0	1	0	1	0	0	1	1
1	0	1	1	1	0	1	1	0	0	0
1	1	0	0	1	1	0	0	0	1	0
1	1	0	1	1	1	0	1	0	0	1
1	1	1	0	1	1	1	0	1	0	0
1	1	1	1	1	1	1	1	1	1	1



Si le poids de Hamming de $E^*(X)$ est inférieur ou égal à la capacité de correction t du code, alors :

- *le poids de Hamming de son syndrome S(X) est aussi $\leq t$ (car $S(X) = E^*(X)$)*
- *$E(X) = E^*(X)$ car il ne peut exister deux vecteurs d'erreur différents ayant le même syndrome (voisinages distincts).*


C
&
X8
X7
X5
X4
X3
X0
9
8
7
6
5
4
3
2
1
0
Rs
≥ 1
&
"0"
a
b
SEL2
P
340
339
11
10
9
8
1
0
a
b
SEL1
Registre de réception


C
&
G'(X)
X8
X7
X5
X4
X3
X0
9
8
7
6
5
4
3
2
1
0
Rs
X9
X8
P(X)
X4
X3
X1
1
≥ 1
&
"0"
a
b
SEL2
P
15
14
13
2
1
0
a
b
SEL1
Registre de réception