CNRS, UMR 6285

Étude des décodeurs LDPC non-binaires

Oussama ABASSI

Directeur de thèse : Emmanuel BOUTILLON
Co-directeur de thèse : Laura CONDE-CANENCIA

Thèse soutenue le 27 juin 2014 devant le jury composé de :

Président : Jean-François HELARD
Rapporteur : Charly POULLIAT
Rapporteur : Christophe JEGO
Examinateur : Pierre PENARD

Intérêt du codage canal

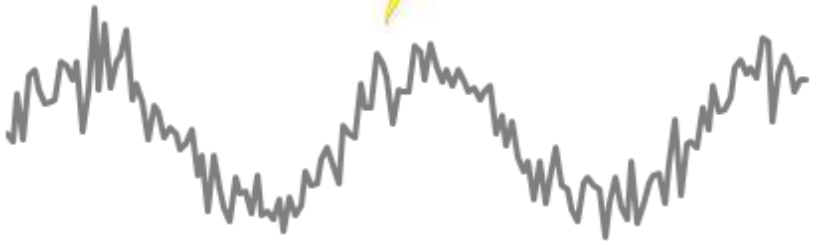
1111

1111

Intérêt du codage canal

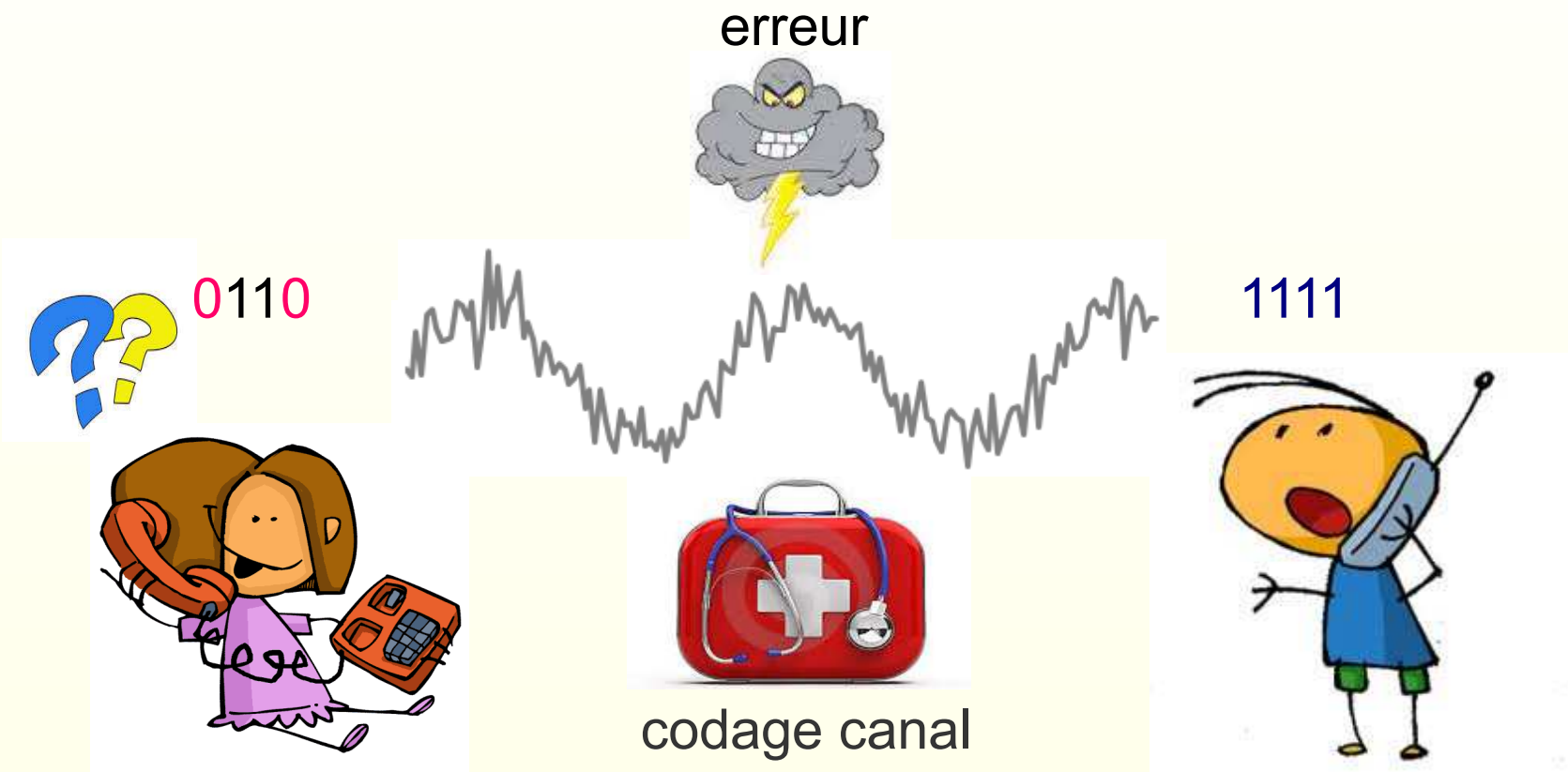
erreur

0110



1111

Intérêt du codage canal



Évolution du codage canal

Historique

1946 - Code de Hamming

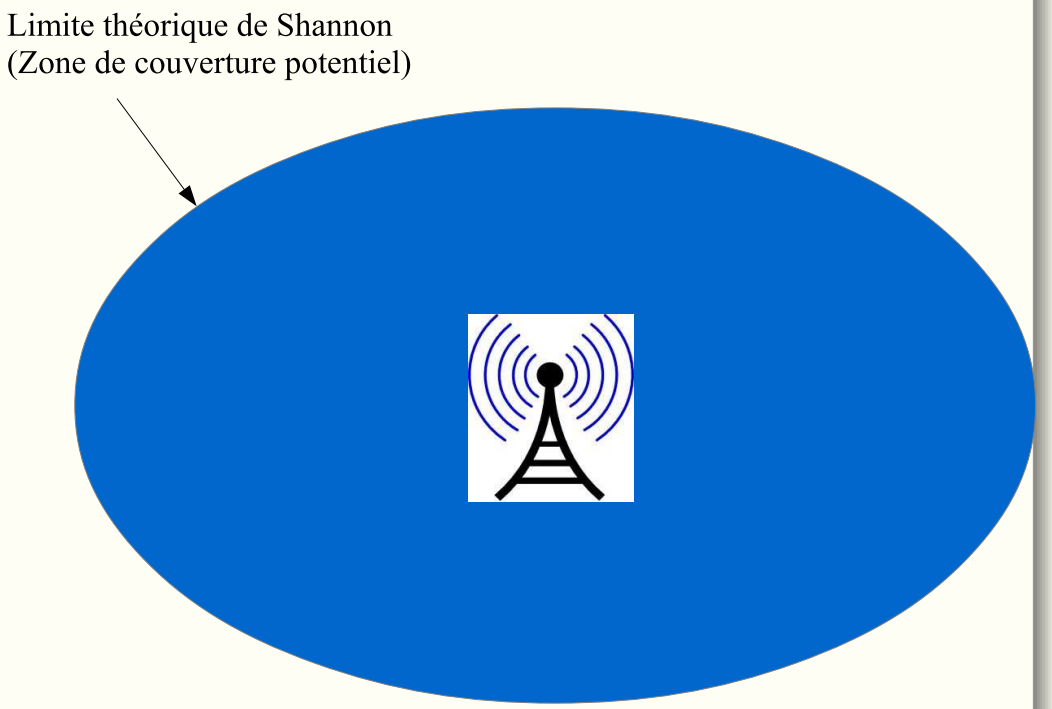
Performances

Évolution du codage canal

Historique

1946 - Code de Hamming
1948 - Naissance de la théorie de l'information

Performances

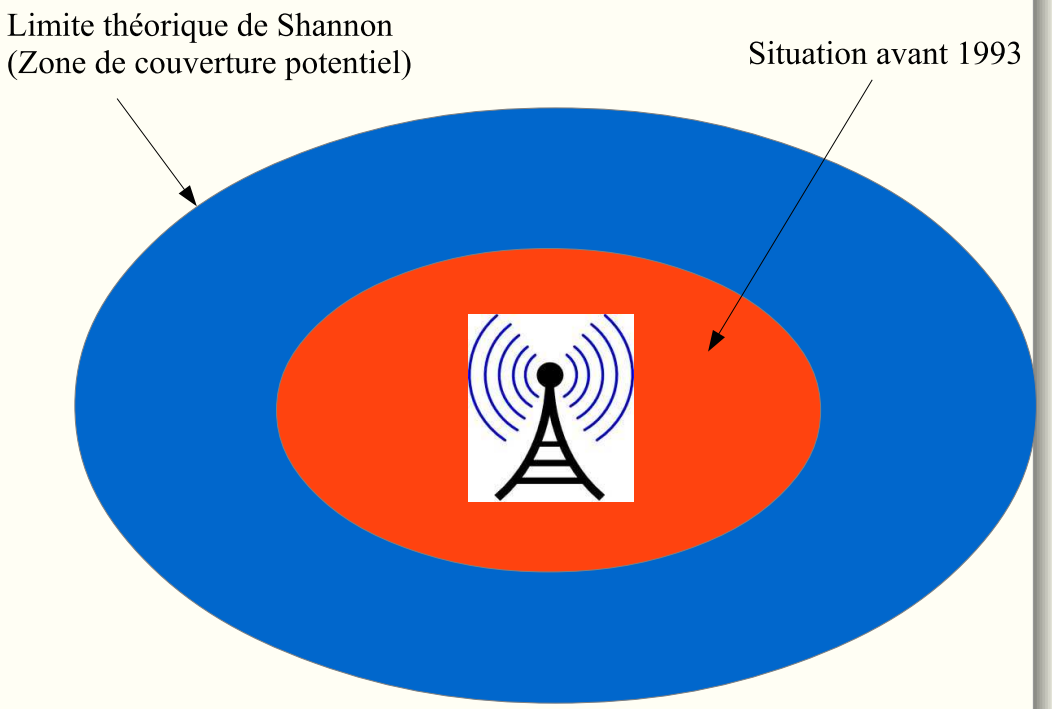


Évolution du codage canal

Historique

1946 - Code de Hamming
1948 - Naissance de la théorie de l'information
1949 - Code de Golay
1954 - Codes de Reed-Muller
1955 - Codes convolutifs
1957 - Codes cycliques (ou codes CRC)
1959 - Codes BCH
1960 - Codes de Reed-Solomon (Extension des codes BCH au cas non-binaire)
1967 - Algorithme de Viterbi

Performances

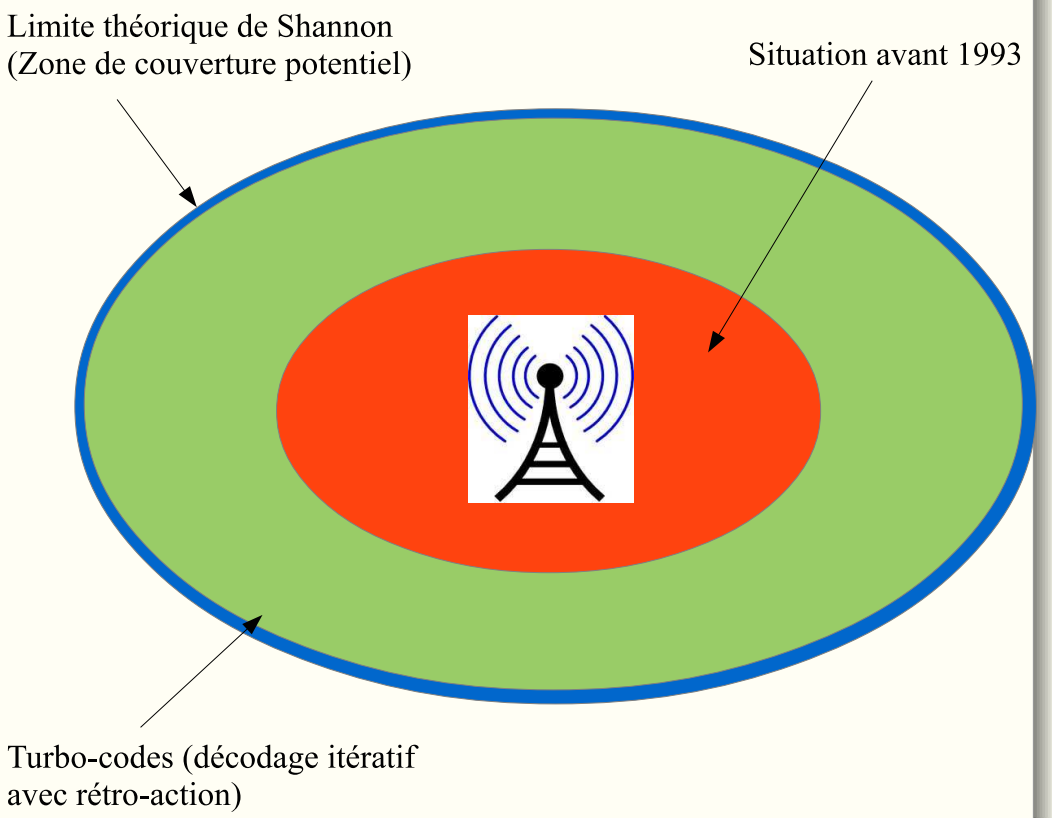


Évolution du codage canal

Historique

1946 - Code de Hamming
1948 - Naissance de la théorie de l'information
1949 - Code de Golay
1954 - Codes de Reed-Muller
1955 - Codes convolutifs
1957 - Codes cycliques (ou codes CRC)
1959 - Codes BCH
1960 - Codes de Reed-Solomon (Extension des codes BCH au cas non-binaire)
1967 - Algorithme de Viterbi
1993 - Principe de décodage Turbo

Performances

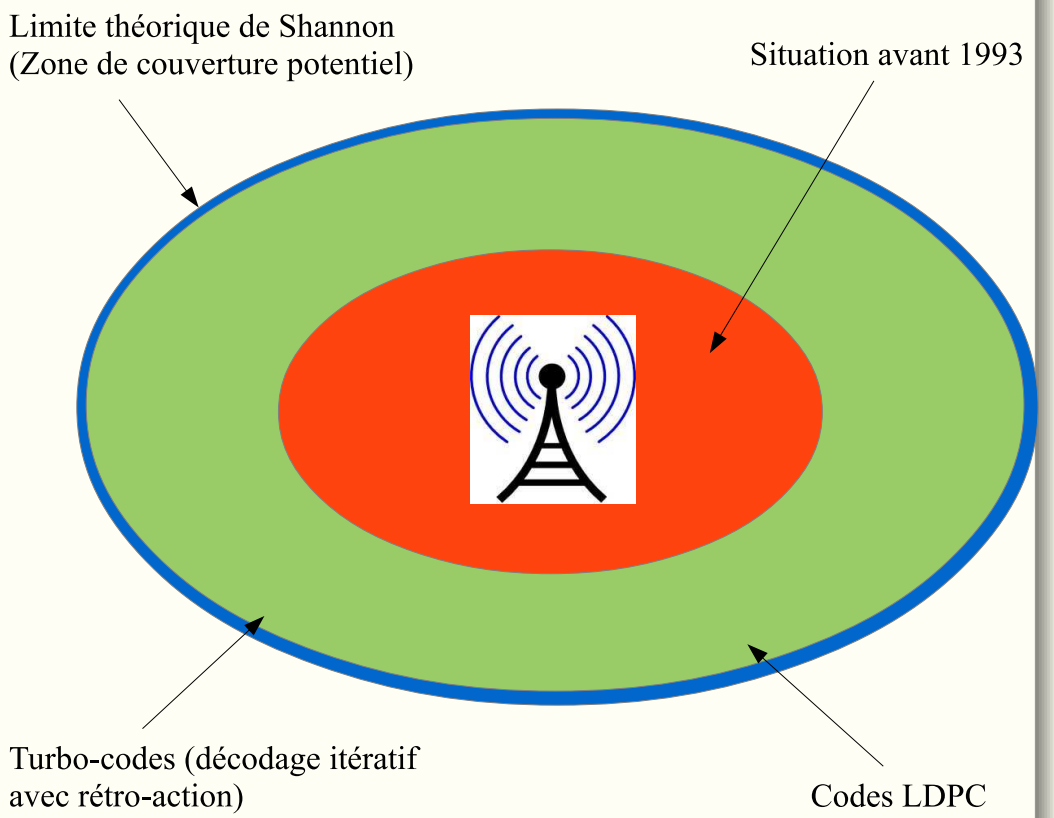


Évolution du codage canal

Historique

1946 - Code de Hamming
1948 - Naissance de la théorie de l'information
1949 - Code de Golay
1954 - Codes de Reed-Muller
1955 - Codes convolutifs
1957 - Codes cycliques (ou codes CRC)
1959 - Codes BCH
1960 - Codes de Reed-Solomon (Extension des codes BCH au cas non-binaire)
1967 - Algorithme de Viterbi
1993 - Principe de décodage Turbo
1995 - Les codes LDPC

Performances



Compétition entre les Turbo-codes et les codes LDPC binaires (UMTS, ADSL, LTE, CMMB, DVB, WiMAX...)

Définition d'un code LDPC

Représentation algébrique

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

$M =$ nombre de PC

$N =$ nombre de bits

- $X = [x_0, x_1, x_2, x_3, x_4, x_5, x_6]^T$ est mot de code $\Leftrightarrow H \cdot X = 0$
- nombre de 1 sur une ligne $= d_c =$ nombre de bits associés à une PC
- nombre de 1 sur une colonne $= d_v =$ nombre de PC associées à un bit
- Rendement du code : $R = \frac{N-M}{N}$

Code LDPC $=$ moins de 1% des bits de la matrice de parité sont égaux à 1

Définition d'un code LDPC

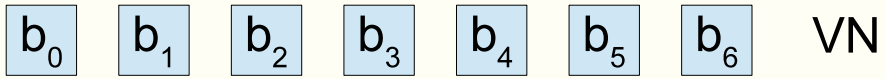
Représentation graphique

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

Définition d'un code LDPC

Représentation graphique

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

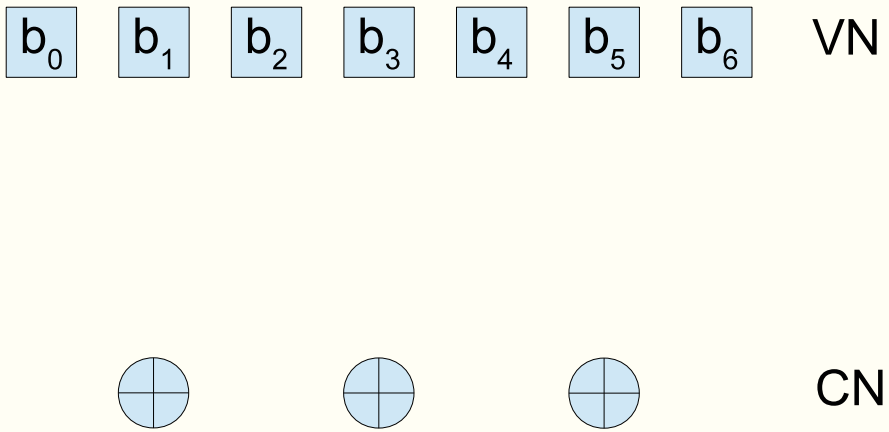


Graphe de Tanner

Définition d'un code LDPC

Représentation graphique

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

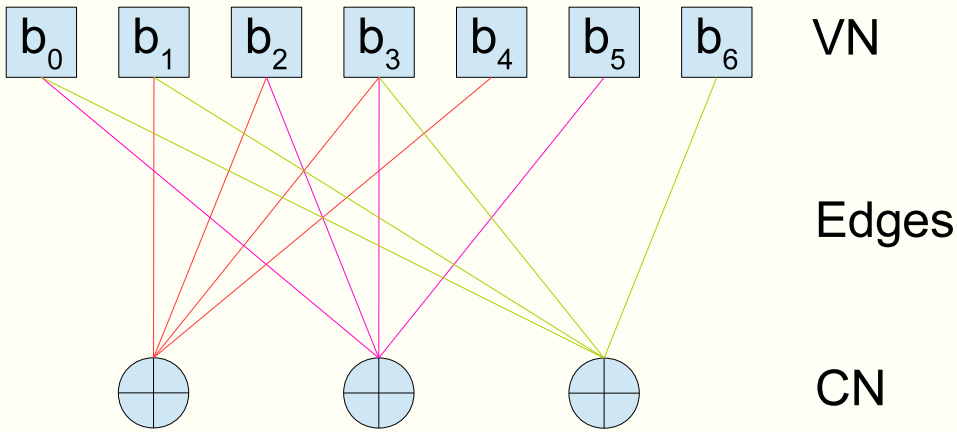


Graphe de Tanner

Définition d'un code LDPC

Représentation graphique

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



Graphe de Tanner

Inconvénient des codes LDPC binaires

Pour s'approcher de la limite de Shannon, il faut considérer des codes de taille infinie (supérieure à 100000 bits).

$\Rightarrow$ Décodeur parallèle

- Routage très complexe.
- Surface d'implantation importante.

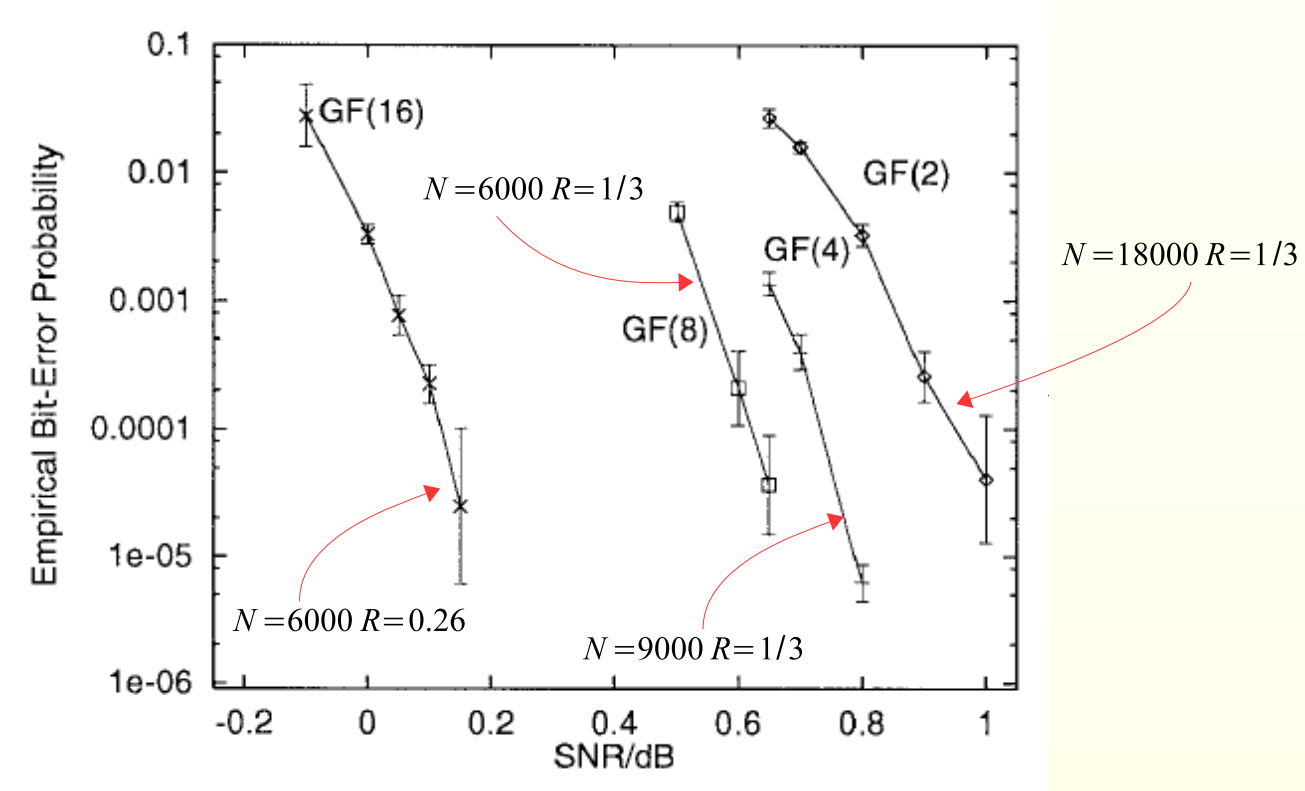
$\Rightarrow$ Décodeur série ou mixte

- Mémoire de taille infinie.

Réduire la taille des mots de code s'accompagne d'une dégradation significative des performances.

Les travaux de Mackay'98

Les codes LDPC non-binaires sont de bon candidats pour des mots de codes de petites et moyennes tailles ($500 \leq N \leq 3000$ bits)



Performances des codes LDPC dans un canal Gaussien [Mackay'98]

Codes LDPC non-binaires : un remède contre les erreurs par paquets

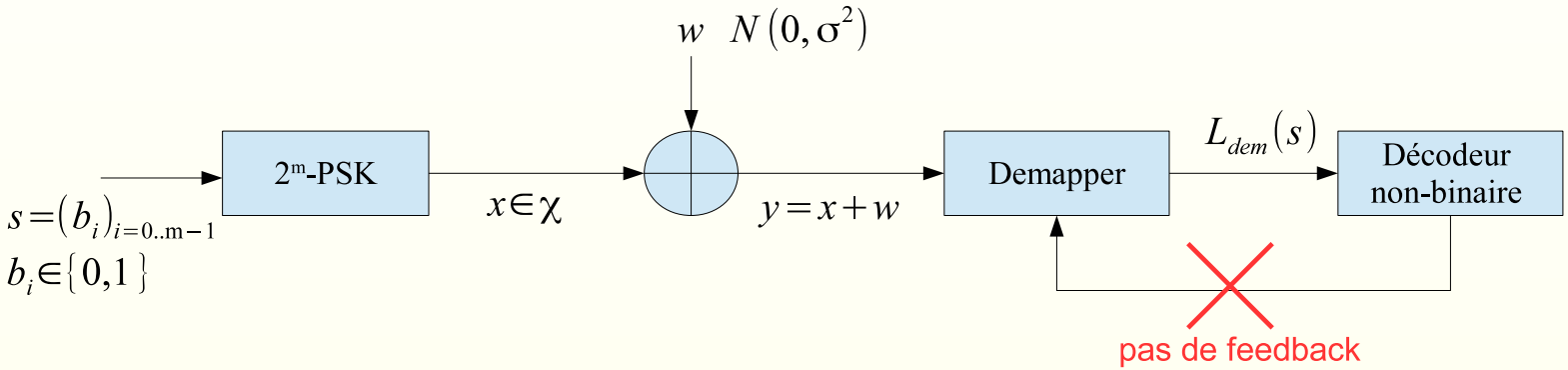
Les codes LDPC non-binaires sont de bon candidats pour remplacer les codes de Reed-Solomon dans les applications de stockage (CD, DVD . . .).

Bo Zhou ; Li Zhang ; Jingyu Kang ; Qin Huang ; Tai, Y.Y. ; Shu Lin ; Meina Xu, "Non-binary LDPC codes vs. Reed-Solomon codes," Information Theory and Applications Workshop, 2008.

Hongzin Song ; Cruz, J.R., "Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording," , IEEE Transactions on Magnetics, 2003.

autre avantage des codes LDPC non-binaires

Une disposition naturelle à la concaténation avec des modulations d'ordre élevé



- Chaque point de la constellation est associé à un symbole du corps de Galois.
- Le LLR d'un symbole correspond la distance entre l'observation du canal et un unique point de constellation.
- Le décodage se fait directement par les LLRs des symboles.

⇒ Demapping simple.
⇒ Pas de perte d'information entre le demapper et le décodeur.

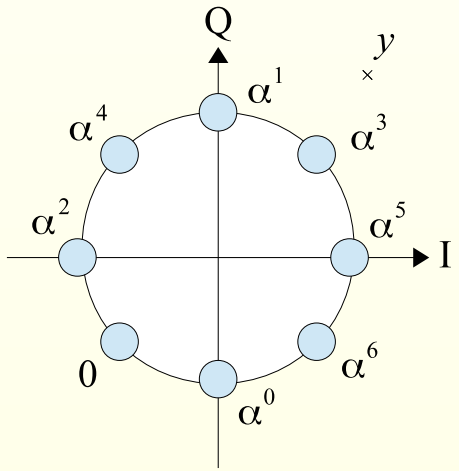


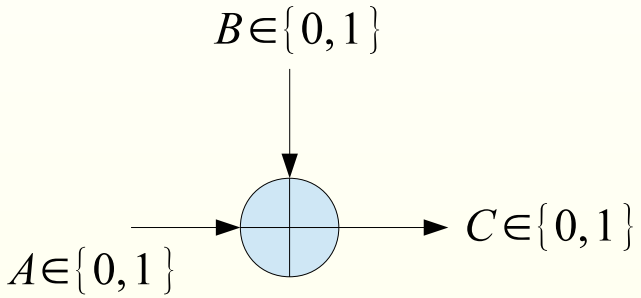
Diagramme de constellation 8-PSK

$GF(8)$	
0	000
α^0	000
α^1	100
α^2	001
α^3	110
α^4	011
α^5	111
α^6	101

Performant mais . . .

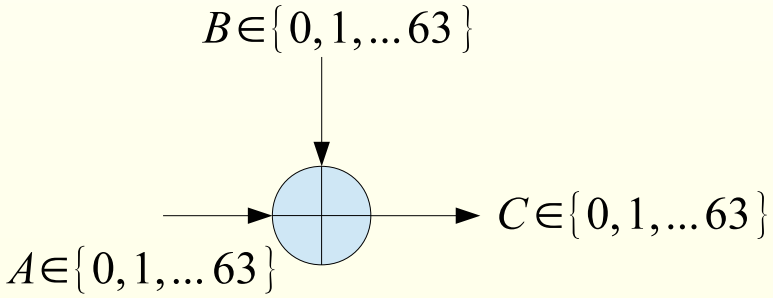
Les bonnes performances des codes LDPC non-binaires sont au prix d'un accroissement significatif de la complexité de décodage en particulier pour des ordres de corps élevés

Cas binaire :



2 valeurs de sortie.
2 combinaisons entrée par valeur de sortie.
$\rightarrow$ 2x2 = 4 combinaisons à évaluer.

Cas GF(64) :

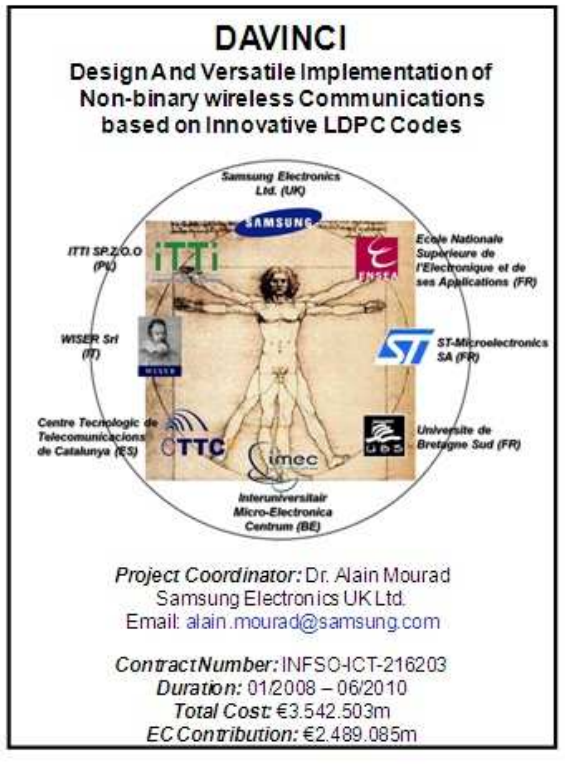


64 valeurs de sortie.
64 combinaisons entrée par valeur de sortie.
$\rightarrow$ 64x64 = 4096 combinaisons à évaluer.

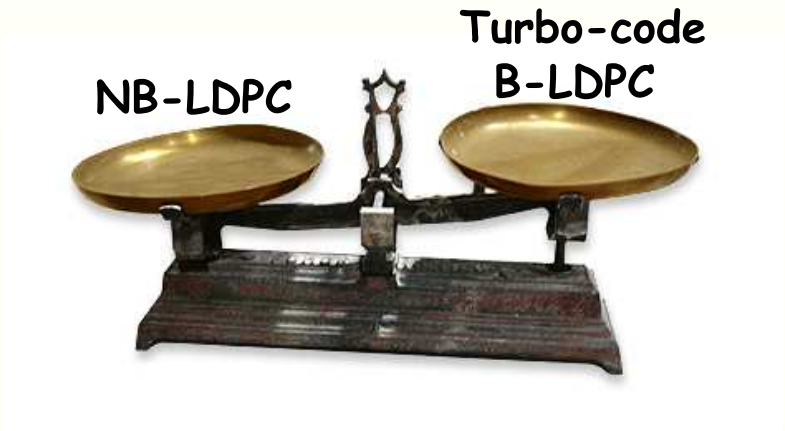
c'est faisable, nous avons la preuve !

- Le premier prototype FPGA d'un décodeur LDPC défini dans un corps de Galois $\mathbb{GF}(64)$ a été conçu en 2009 par notre laboratoire de recherche dans le cadre du projet européen DAVINCI.

- Un brevet déposé en 2009 dont la licence a été achetée par FRANCE BREVET en 2013.



Objectifs de la thèse



Nos contributions

- Proposer une architecture de décodeur LDPC non-binaire à faible complexité.
- Étudier les avantages d'associer la modulation CCSK avec un code LDPC non-binaire.

Sommaire

Sommaire

Sommaire

Sommaire

Sommaire

Corps de Galois $\mathbb{GF}(q = 2^m)$

- Un code LDPC peut être défini dans un corps de Galois $\mathbb{GF}(q = 2^m)$.
- Rappelons ce qu'est un corps de Galois :

Définition

Un corps de Galois $\mathbb{GF}(q = 2^m)$ est un ensemble fini de q éléments qui possède la structure algébrique d'un corps, c'est-à-dire :

une loi d'addition $(\mathbb{GF}(q = 2^m), +)$
une loi de multiplication $(\mathbb{GF}(q = 2^m), \times)$
. . . et d'autres élégantes propriétés associées.

- Par convention, un corps de Galois $\mathbb{GF}(q = 2^m)$ se présente sous la forme $\{0, \alpha^0, \alpha^1, \cdots \alpha^{q-2}\}$

Corps de Galois $\mathbb{GF}(q = 2^m)$

- Un corps de Galois $\mathbb{GF}(q = 2^m)$ possède aussi une représentation binaire.

Exemple du corps de Galois $\mathbb{GF}(8)$

0	000
α^0	100
α^1	010
α^2	001
α^3	110
α^4	011
α^5	111
α^6	101

Addition :

$x = x_1x_2x_3$ et $y = y_1y_2y_3 \in \mathbb{GF}(8)$

$x + y = (x_1 \text{ XOR } y_1)(x_2 \text{ XOR } y_2)(x_3 \text{ XOR } y_3)$

Exemple :

$$\begin{aligned}\alpha^2 + \alpha^5 &= (0 \text{ XOR } 1)(1 \text{ XOR } 1)(0 \text{ XOR } 1)\\ &= 101\\ &= \alpha^6\end{aligned}$$

Multiplication :

$0 \times \alpha^i = 0$

$\alpha^i \times \alpha^j = \alpha^{(i+j) \bmod (q-1)}$

Exemple :

$$\begin{aligned}\alpha^3 \times \alpha^5 &= \alpha^{(3+5) \bmod 7}\\ &= \alpha^1\end{aligned}$$

Représentation algébrique d'un code LDPC non-binaire

LDPC binaire

mot de code : $c = [c_i \in \mathbb{GF}(2), 0 \leq i < 6]$

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

Contraintes de parité :

$$\begin{cases} c_0 \oplus c_1 \oplus c_3 & = & 0 \\ c_0 \oplus c_1 \oplus c_2 \oplus c_5 & = & 0 \\ c_1 \oplus c_2 \oplus c_4 & = & 0 \end{cases}$$

LDPC non-binaire

mot de code : $c = [c_i \in \mathbb{GF}(q = 2^m), 0 \leq i < 6]$

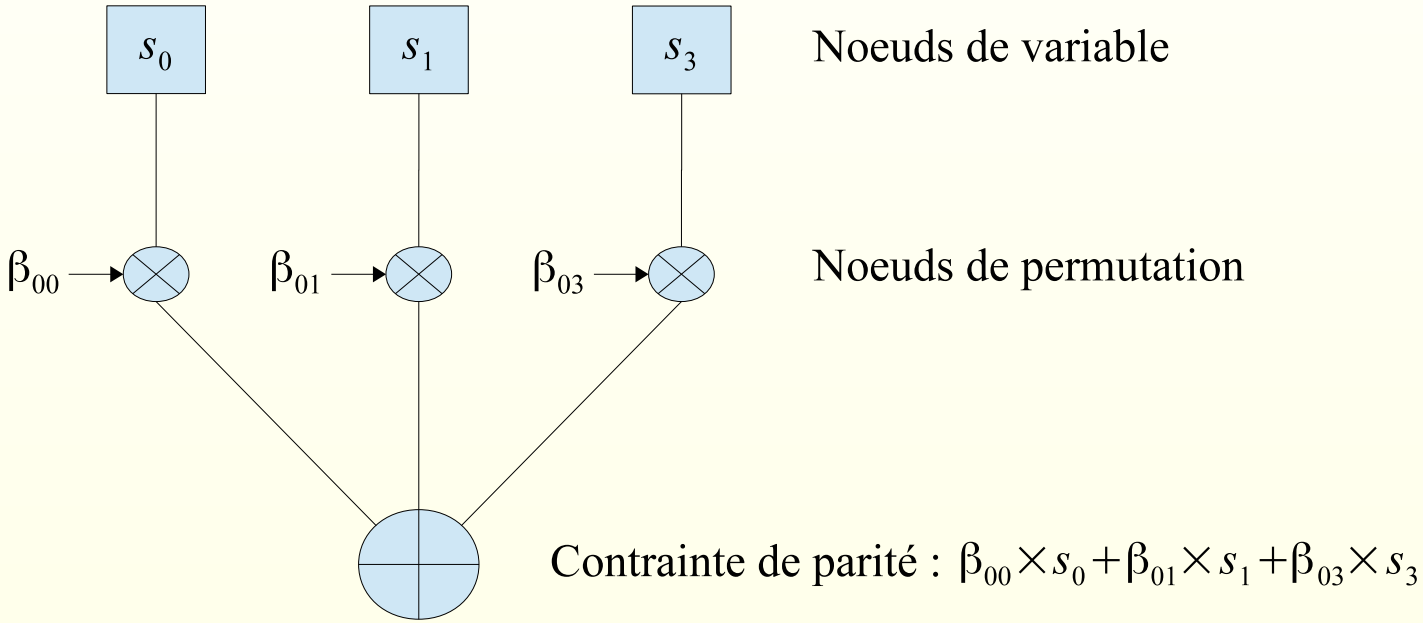
$$H = \begin{pmatrix} \beta_{00} & \beta_{01} & 0 & \beta_{03} & 0 & 0 \\ \beta_{10} & \beta_{11} & \beta_{12} & 0 & 0 & \beta_{15} \\ 0 & \beta_{21} & \beta_{22} & 0 & \beta_{24} & 0 \end{pmatrix}_{\beta_{ij} \in \mathbb{GF}(q) - 0}$$

Contraintes de parité :

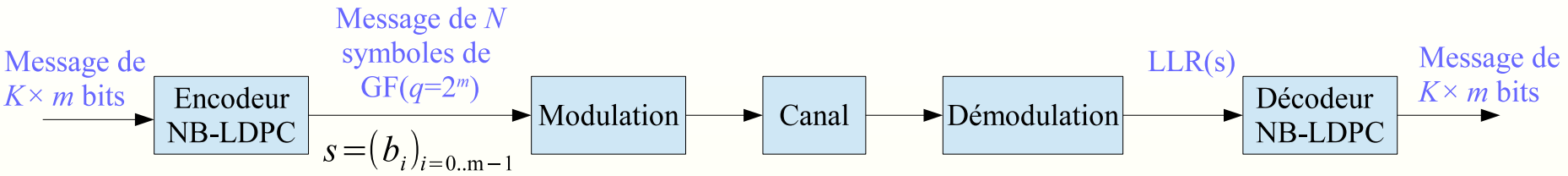
$$\begin{cases} \beta_{00} \times c_0 + \beta_{01} \times c_1 + \beta_{03} \times c_3 & = & 0 \\ \beta_{10} \times c_0 + \beta_{11} \times c_1 + \beta_{12} \times c_2 + \beta_{15} \times c_5 & = & 0 \\ \beta_{21} \times c_1 + \beta_{22} \times c_2 + \beta_{24} \times c_4 & = & 0 \end{cases}$$

Représentation graphique d'un code LDPC non-binaire

Nous devons ajouter des nœuds de permutation au graphe de Tanner pour modéliser la multiplication des symboles du mot de code par les éléments non nuls de la matrice de parité.



Représentation de l'information intrinsèque



LDPC binaire

$$(P(b=0),P(b=1)) \Rightarrow LLR_b = \ln \frac{P(b=1)}{P(b=0)}$$

LDPC dans $\mathbb{GF}(q=2^m)$

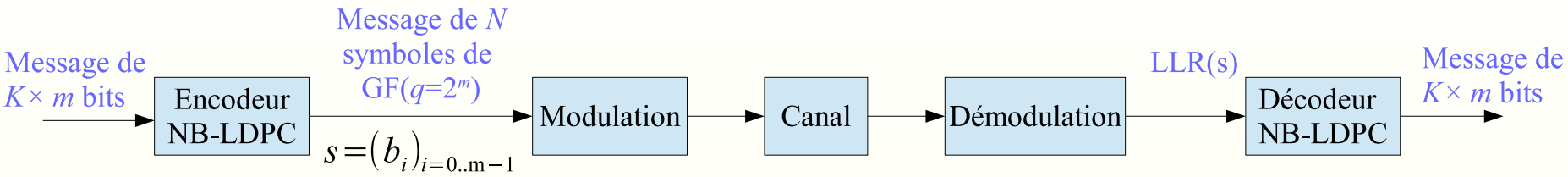
Domaine des probabilités :

$$P_s = \left[P(s=0),P(s=\alpha^0),P(s=\alpha^1),\cdots P(s=\alpha^{q-2})\right]$$

Domaine des logarithmes :

$$LLR_s = -ln(P_s) + \ln\left(\max_{\beta\in\mathbb{GF}(q=2^m)} P_s(\beta)\right)$$

Représentation de l'information intrinsèque



LDPC binaire

$$(P(b=0),P(b=1)) \Rightarrow LLR_b = \ln \frac{P(b=1)}{P(b=0)}$$

LDPC dans $\mathbb{GF}(q=2^m)$

Domaine des probabilités :

$$P_s = \left[P(s=0),P(s=\alpha^0),P(s=\alpha^1),\cdots P(s=\alpha^{q-2})\right]$$

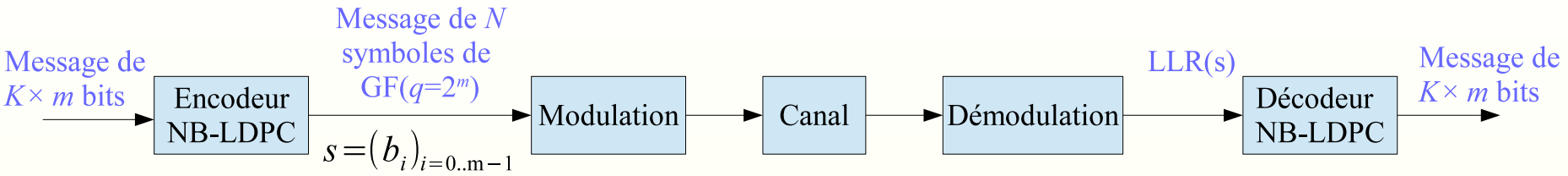
Domaine des logarithmes :

$$LLR_s = -ln(P_s) + \ln\left(\max_{\beta\in\mathbb{GF}(q=2^m)} P_s(\beta)\right)$$

Exemple dans $\mathbb{GF}(8)$

	0	α^0	α^1	α^2	α^3	α^4	α^5	α^6
P_s	0.1	0.85	10^{-3}	10^{-7}	10^{-10}	0.05	10^{-10}	10^{-10}
$-\ln P_s$								
LLR								

Représentation de l'information intrinsèque



LDPC binaire

$$(P(b=0),P(b=1)) \Rightarrow LLR_b = \ln \frac{P(b=1)}{P(b=0)}$$

LDPC dans $\mathbb{GF}(q=2^m)$

Domaine des probabilités :

$$P_s = \left[P(s=0),P(s=\alpha^0),P(s=\alpha^1),\cdots P(s=\alpha^{q-2})\right]$$

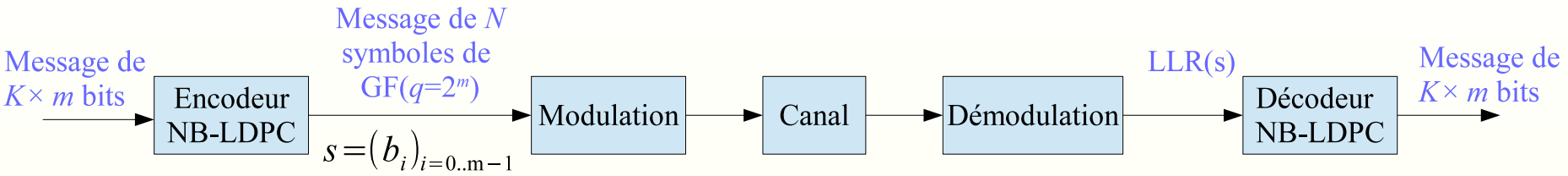
Domaine des logarithmes :

$$LLR_s = -ln(P_s) + \ln\left(\max_{\beta\in\mathbb{GF}(q=2^m)} P_s(\beta)\right)$$

Exemple dans $\mathbb{GF}(8)$

	0	α^0	α^1	α^2	α^3	α^4	α^5	α^6
P_s	0.1	0.85	10^{-3}	10^{-7}	10^{-10}	0.05	10^{-10}	10^{-10}
$-\ln P_s$	2.3	0.2	6.9	16.1	23.0	3.0	23.0	23.0
LLR								

Représentation de l'information intrinsèque



LDPC binaire

$$(P(b=0),P(b=1)) \Rightarrow LLR_b = \ln \frac{P(b=1)}{P(b=0)}$$

LDPC dans $\mathbb{GF}(q=2^m)$

Domaine des probabilités :

$$P_s = \left[P(s=0),P(s=\alpha^0),P(s=\alpha^1),\cdots P(s=\alpha^{q-2})\right]$$

Domaine des logarithmes :

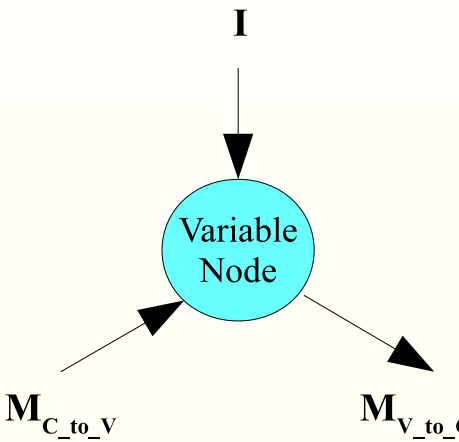
$$LLR_s = -ln(P_s) + \ln\left(\max_{\beta\in\mathbb{GF}(q=2^m)} P_s(\beta)\right)$$

Exemple dans $\mathbb{GF}(8)$

	0	α^0	α^1	α^2	α^3	α^4	α^5	α^6
P_s	0.1	0.85	10^{-3}	10^{-7}	10^{-10}	0.05	10^{-10}	10^{-10}
$-\ln P_s$	2.3	0.2	6.9	16.1	23.0	3.0	23.0	23.0
LLR_s	2.1	0	6.7	15.9	22.9	2.8	22.8	22.8

Principe de décodage LDPC non-binaire

Traitement de Nœud de Variable



$M_{V\_to\_C}$ =Addition terme à terme de I et $M_{C\_to\_V}$

Exemple dans $\mathbb{GF}(4)$

I		+	$M_{C\_to\_V}$		=	$M_{V\_to\_C}$	
0	3		0	1		0	4
α^0	8		α^0	0		α^0	8
α^1	0		α^1	11		α^1	11
α^2	5		α^2	9		α^2	14

Principe de décodage LDPC non-binaire

Traitement d'une contrainte de parité élémentaire (ECN)

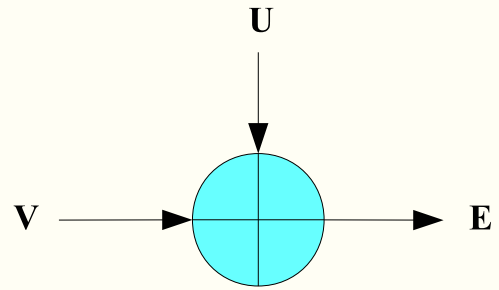
Exemple dans $\mathbb{GF}(4)$

$\mathbf{GF_V}$

$\mathbf{GF_U}$	+	0	α^0	α^1	α^2
0	0	α^0	α^1	α^2
α^0	α^0	0	α^2	α^1
α^1	α^1	α^2	0	α^0
α^2	α^2	α^1	α^0	0

$\mathbf{LLR_V}$

$\mathbf{LLR_U}$	+	18	7	9	0
3	21	10	12	3
0	18	7	9	0
12	30	19	21	12
6	24	13	15	6



$\mathbf{U+V+E=0 \longrightarrow U+V=E}$

Principe de décodage LDPC non-binaire

Traitement d'une contrainte de parité élémentaire (ECN)

Exemple dans $\mathbb{GF}(4)$

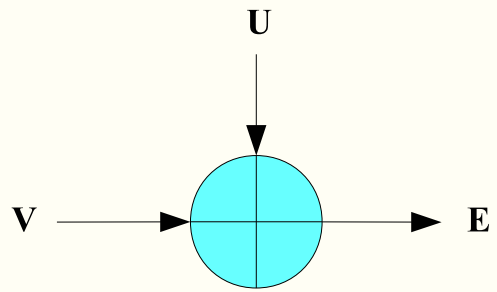
$\mathbf{GF_V}$

$\mathbf{GF_U}$	+	0	α^0	α^1	α^2
0	0	0	α^0	α^1	α^2
α^0	α^0	α^0	0	α^2	α^1
α^1	α^1	α^1	α^2	0	α^0
α^2	α^2	α^2	α^1	α^0	0

$\mathbf{LLR_V}$

$\mathbf{LLR_U}$	+	18	7	9	0
3	3	21	10	12	3
0	0	18	7	9	0
12	12	30	19	21	12
6	6	24	13	15	6

$$\mathbf{LLR_E(0) = 21 + 7 + 21 + 6 = 54}$$



$$\mathbf{U+V+E=0 \longrightarrow U+V=E}$$

Principe de décodage LDPC non-binaire

Traitement d'une contrainte de parité élémentaire (ECN)

Exemple dans $\mathbb{GF}(4)$

GF_V

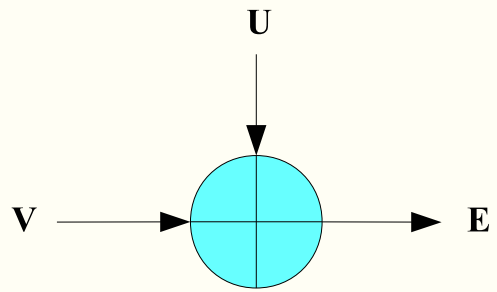
GF_U	+	0	α^0	α^1	α^2
0	0	0	α^0	α^1	α^2
α^0	α^0	α^0	0	α^2	α^1
α^1	α^1	α^1	α^2	0	α^0
α^2	α^2	α^2	α^1	α^0	0

LLR_V

LLR_U	+	18	7	9	0
3	21	10	12	3	
0	18	7	9	0	
12	30	19	21	12	
6	24	13	15	6	

$$LLR_E(0) = 21 + 7 + 21 + 6 = 54$$

$$LLR_E(\alpha^0) = 10 + 18 + 12 + 15 = 55$$



$$U+V+E=0 \longrightarrow U+V=E$$

Principe de décodage LDPC non-binaire

Traitement d'une contrainte de parité élémentaire (ECN)

Exemple dans $\mathbb{GF}(4)$

GF_V

GF_U	+	0	α^0	α^1	α^2
0	0	0	α^0	α^1	α^2
α^0	α^0	α^0	0	α^2	α^1
α^1	α^1	α^1	α^2	0	α^0
α^2	α^2	α^2	α^1	α^0	0

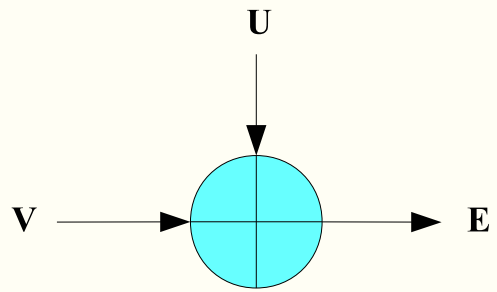
LLR_V

LLR_U	+	18	7	9	0
3	3	21	10	12	3
0	0	18	7	9	0
12	12	30	19	21	12
6	6	24	13	15	6

$$LLR_E(0) = 21 + 7 + 21 + 6 = 54$$

$$LLR_E(\alpha^0) = 10 + 18 + 12 + 15 = 55$$

$$LLR_E(\alpha^1) = 12 + 0 + 30 + 13 = 55$$



$$U+V+E=0 \longrightarrow U+V=E$$

Principe de décodage LDPC non-binaire

Traitement d'une contrainte de parité élémentaire (ECN)

Exemple dans $\mathbb{GF}(4)$

$\mathbf{GF_V}$

$\mathbf{GF_U}$	+	0	α^0	α^1	α^2
0	0	0	α^0	α^1	α^2
α^0	α^0	α^0	0	α^2	α^1
α^1	α^1	α^1	α^2	0	α^0
α^2	α^2	α^2	α^1	α^0	0

$\mathbf{LLR_V}$

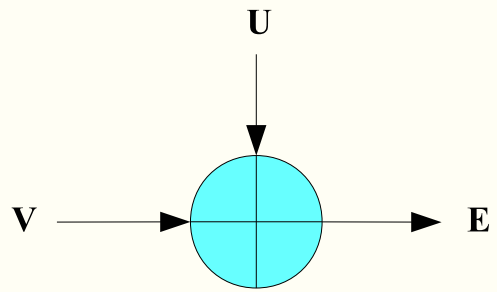
$\mathbf{LLR_U}$	+	18	7	9	0
3	3	21	10	12	3
0	0	18	7	9	0
12	12	30	19	21	12
6	6	24	13	15	6

$$\mathbf{LLR_E(0) = 21 + 7 + 21 + 6 = 54}$$

$$\mathbf{LLR_E(\alpha^0) = 10 + 18 + 12 + 15 = 55}$$

$$\mathbf{LLR_E(\alpha^1) = 12 + 0 + 30 + 13 = 55}$$

$$\mathbf{LLR_E(\alpha^2) = 3 + 9 + 19 + 24 = 55}$$



$$\mathbf{U+V+E=0 \longrightarrow U+V=E}$$

Principe de décodage LDPC non-binaire

Traitement d'une contrainte de parité élémentaire (ECN)

Exemple dans $\mathbb{GF}(4)$

GF_V

GF_U	+	0	α^0	α^1	α^2
0	0	0	α^0	α^1	α^2
α^0	α^0	0	α^2	α^1	
α^1	α^1	α^2	0	α^0	
α^2	α^2	α^1	α^0	0	

LLR_V

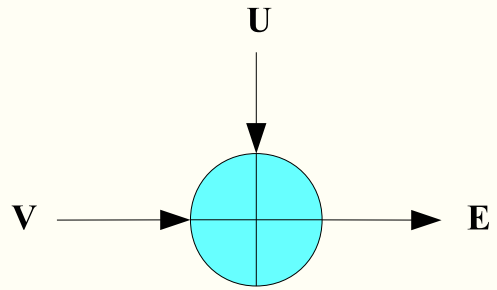
LLR_U	+	18	7	9	0
3	21	10	12	3	
0	18	7	9	0	
12	30	19	21	12	
6	24	13	15	6	

$$LLR_E(0) = \min(21, 7, 21, 6) = 6$$

$$LLR_E(\alpha^0) = \min(10, 18, 12, 15) = 10$$

$$LLR_E(\alpha^1) = \min(12, 0, 30, 13) = 0$$

$$LLR_E(\alpha^2) = \min(3, 9, 19, 24) = 3$$



$$U+V+E=0 \longrightarrow U+V=E$$

Complexité :

$2q^2$ additions + q opérations *min* sur q valeurs

Algorithme Extended Min-Sum (EMS)

Traitement d'un ECN : $$LLR_E(\alpha^k) \approx \min_{\alpha^i,\alpha^j \in \mathbb{GF}(q)^2 / \alpha^i + \alpha^j = \alpha^k} LLR_U(\alpha^i) + LLR_V(\alpha^j)$$

Observation : les valeurs LLR plus élevés de U et V sont rarement, voire jamais, utilisées dans la sortie.

Idée : ne garder que les $n_m \ll q$ valeurs LLR les plus petits triées dans l'ordre croissant pour simplifier le calcul de l'ECN.

Exemple

$\mathbf{GF_V}$

$\mathbf{GF_U}$ \ +	0	α^0	α^1	α^2
0	0	α^0	α^1	α^2
α^0	α^0	0	α^2	α^1
α^1	α^1	α^2	0	α^0
α^2	α^2	α^1	α^0	0

$\mathbf{LLR_V}$

$\mathbf{LLR_U}$ \ +	18	7	9	0
3	21	10	12	3
0	18	7	9	0
12	30	19	21	12
6	24	13	15	6

Algorithme Extended Min-Sum (EMS)

Traitement d'un ECN : $$LLR_E(\alpha^k) \approx \min_{\alpha^i,\alpha^j \in \mathbb{GF}(q)^2 / \alpha^i + \alpha^j = \alpha^k} LLR_U(\alpha^i) + LLR_V(\alpha^j)$$

Observation : les valeurs LLR plus élevés de U et V sont rarement, voire jamais, utilisées dans la sortie.

Idée : ne garder que les $n_m \ll q$ valeurs LLR les plus petits triées dans l'ordre croissant pour simplifier le calcul de l'ECN.

Exemple

$\mathbf{GF_V}$

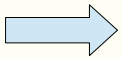
+	0	α^0	α^1	α^2
0	0	α^0	α^1	α^2
α^0	α^0	0	α^2	α^1
α^1	α^1	α^2	0	α^0
α^2	α^2	α^1	α^0	0

$\mathbf{GF_U}$

$\mathbf{LLR_V}$

+	18	7	9	0
3	21	10	12	3
0	18	7	9	0
12	30	19	21	12
6	24	13	15	6

$\mathbf{LLR_U}$



+	0, α^2	7, α^0
0, α^0	0, α^1	7, 0
3, 0	3, α^2	10, α^0

Extraire les n_m valeurs plus petites parmi les $n_m{}^2$ valeurs du tableau

Algorithme Extended Min-Sum (EMS)

Traitement d'un ECN : $$LLR_E(\alpha^k) \approx \min_{\alpha^i,\alpha^j \in \mathbb{GF}(q)^2 / \alpha^i + \alpha^j = \alpha^k} LLR_U(\alpha^i) + LLR_V(\alpha^j)$$

Observation : les valeurs LLR plus élevés de U et V sont rarement, voire jamais, utilisées dans la sortie.

Idée : ne garder que les $n_m \ll q$ valeurs LLR les plus petits triées dans l'ordre croissant pour simplifier le calcul de l'ECN.

Exemple

$\mathbf{GF_V}$

+	0	α^0	α^1	α^2
0	0	α^0	α^1	α^2
α^0	α^0	0	α^2	α^1
α^1	α^1	α^2	0	α^0
α^2	α^2	α^1	α^0	0

$\mathbf{GF_U}$

$\mathbf{LLR_V}$

+	18	7	9	0
3	21	10	12	3
0	18	7	9	0
12	30	19	21	12
6	24	13	15	6

$\mathbf{LLR_U}$

+	$0, \alpha^2$	$7, \alpha^0$
$0, \alpha^0$	$0, \alpha^1$	7, 0
3, 0	$3, \alpha^2$	$10, \alpha^0$

E=[(0, α^1) ; (3, α^2)]

Complexité : $2q^2 \Rightarrow 4n_m$ additions (algorithme L-Bubble)

Sommaire

Traitement de nœud de variable

I

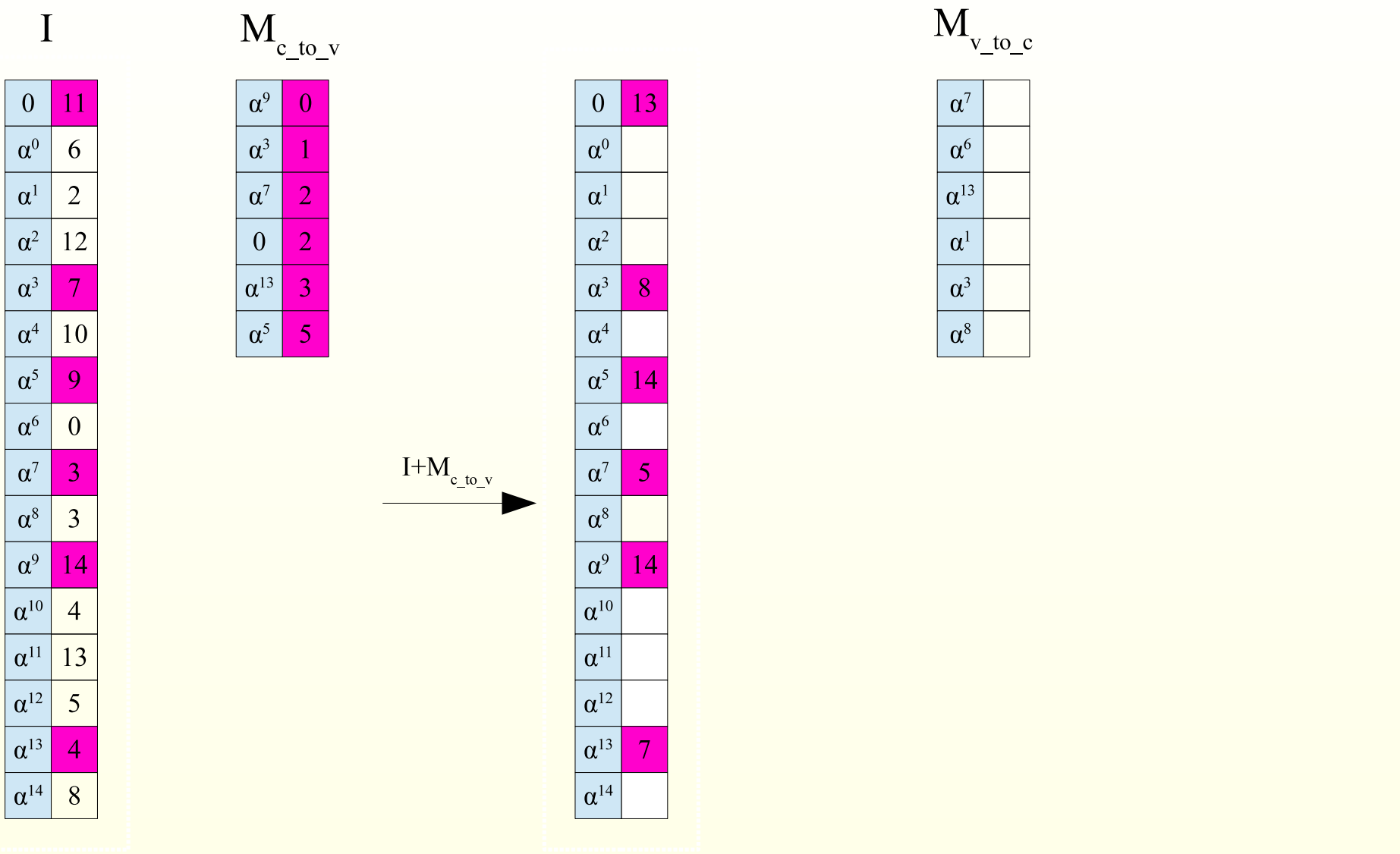
0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

$M_{c\_to\_v}$

α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

$M_{v\_to\_c}$

Traitement de nœud de variable



I	
0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

$M_{c\_to\_v}$	
α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

$I+M_{c\_to\_v}$	
0	13
α^0	
α^1	
α^2	
α^3	8
α^4	
α^5	14
α^6	
α^7	5
α^8	
α^9	14
α^{10}	
α^{11}	
α^{12}	
α^{13}	7
α^{14}	

$M_{v\_to\_c}$	
α^7	
α^6	
α^{13}	
α^1	
α^3	
α^8	

Traitement de nœud de variable

I

0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

$M_{c\_to\_v}$

α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

I+offset

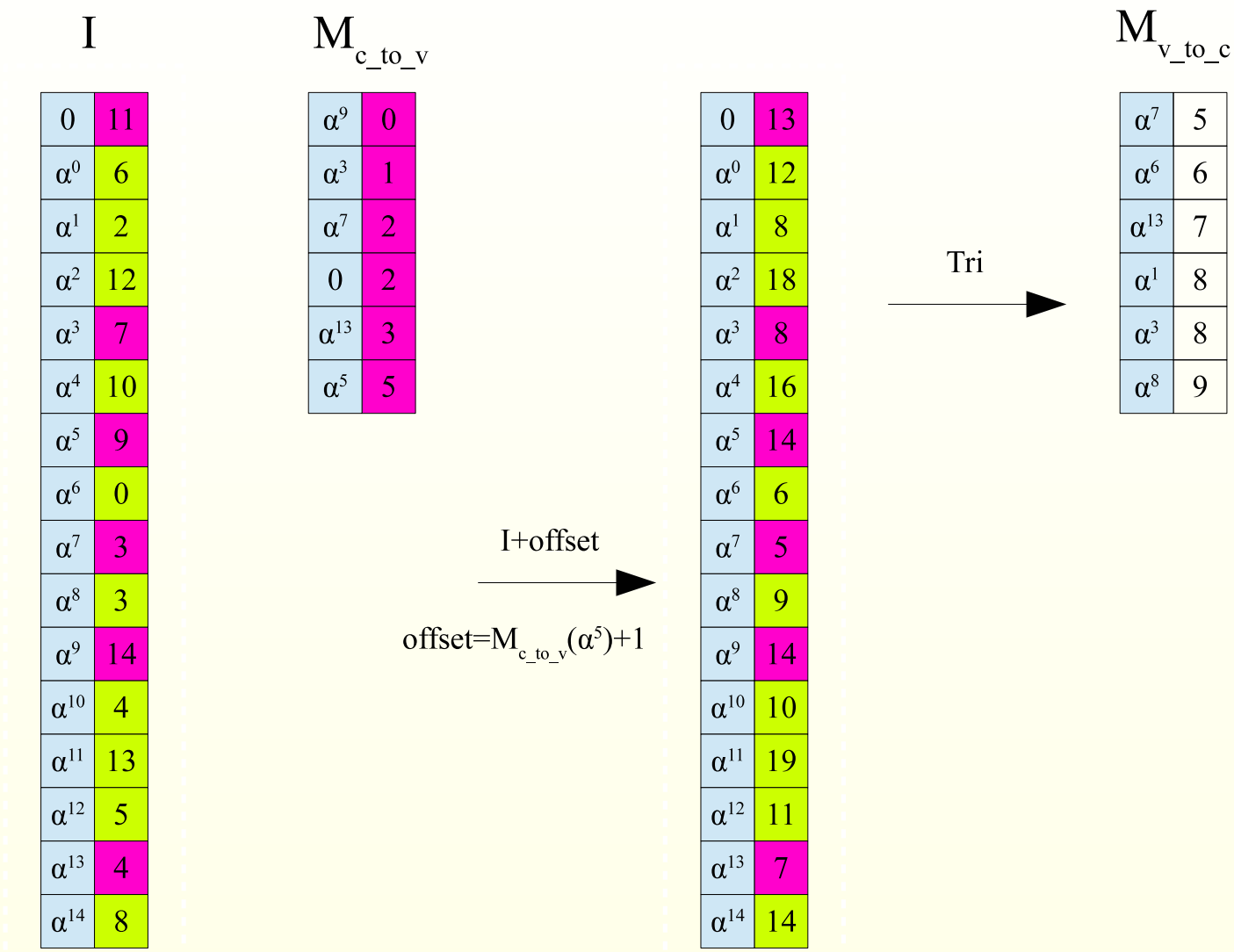
$\text{offset}=M_{c\_to\_v}(\alpha^5)+1$

0	13
α^0	12
α^1	8
α^2	18
α^3	8
α^4	16
α^5	14
α^6	6
α^7	5
α^8	9
α^9	14
α^{10}	10
α^{11}	19
α^{12}	11
α^{13}	7
α^{14}	14

$M_{v\_to\_c}$

α^7	
α^6	
α^{13}	
α^1	
α^3	
α^8	

Traitement de nœud de variable



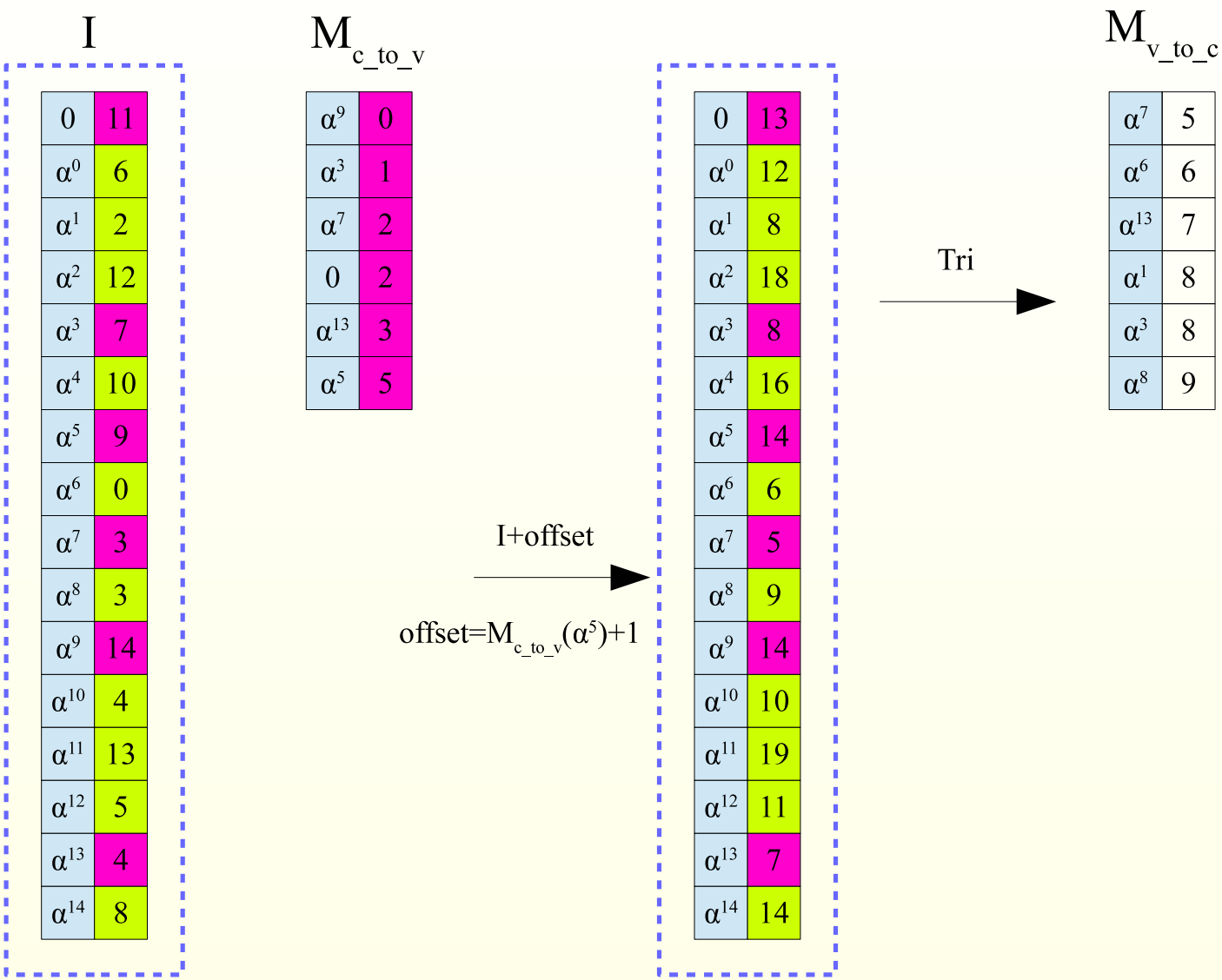
I	
0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

$M_{c\_to\_v}$	
α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

0	13
α^0	12
α^1	8
α^2	18
α^3	8
α^4	16
α^5	14
α^6	6
α^7	5
α^8	9
α^9	14
α^{10}	10
α^{11}	19
α^{12}	11
α^{13}	7
α^{14}	14

$M_{v\_to\_c}$	
α^7	5
α^6	6
α^{13}	7
α^1	8
α^3	8
α^8	9

Traitement de nœud de variable



I	
0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

$M_{c\_to\_v}$	
α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

0	13
α^0	12
α^1	8
α^2	18
α^3	8
α^4	16
α^5	14
α^6	6
α^7	5
α^8	9
α^9	14
α^{10}	10
α^{11}	19
α^{12}	11
α^{13}	7
α^{14}	14

$M_{v\_to\_c}$	
α^7	5
α^6	6
α^{13}	7
α^1	8
α^3	8
α^8	9

Traitement sous-optimal du nœud de variable

I

0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

$M_{c\_to\_v}$

α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

$M_{v\_to\_c}$

Traitement sous-optimal du nœud de variable

I

α^0	6
α^1	2
α^2	12
α^4	10
α^6	0
α^8	3
α^{10}	4
α^{11}	13
α^{12}	5
α^{14}	8

$M_{c\_to\_v}$

α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

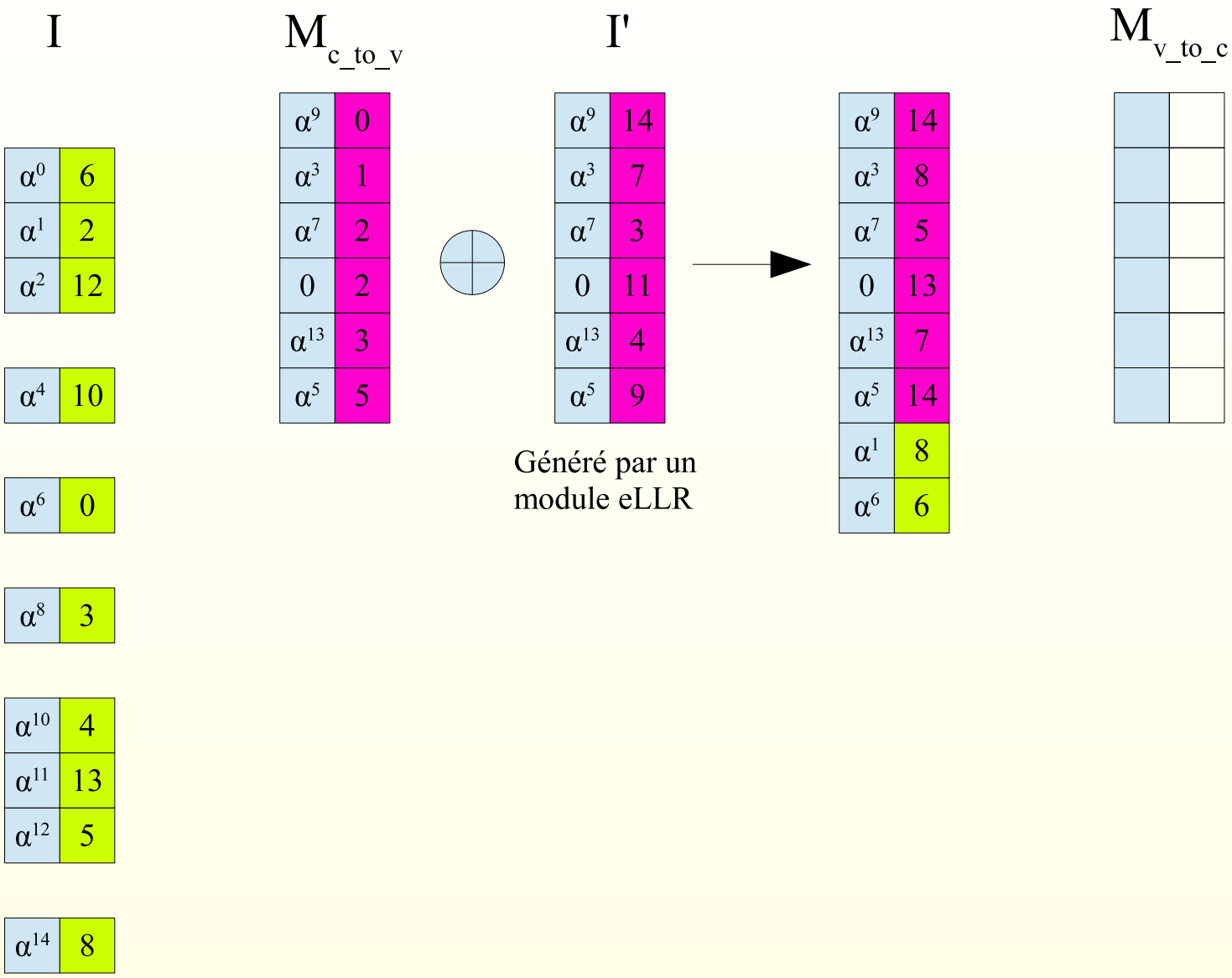
I'

α^9	14
α^3	7
α^7	3
0	11
α^{13}	4
α^5	9

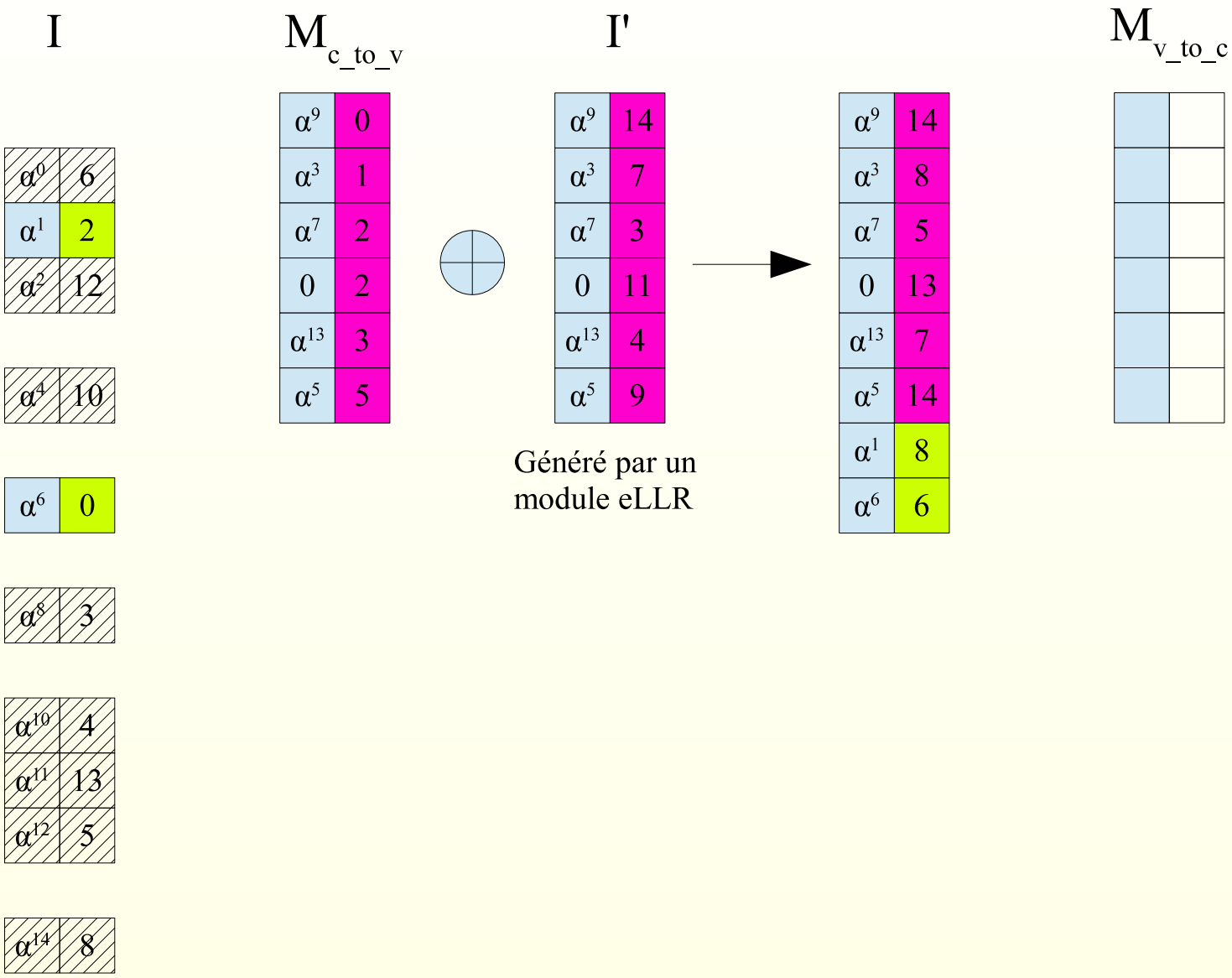
Généré par un module eLLR

$M_{v\_to\_c}$

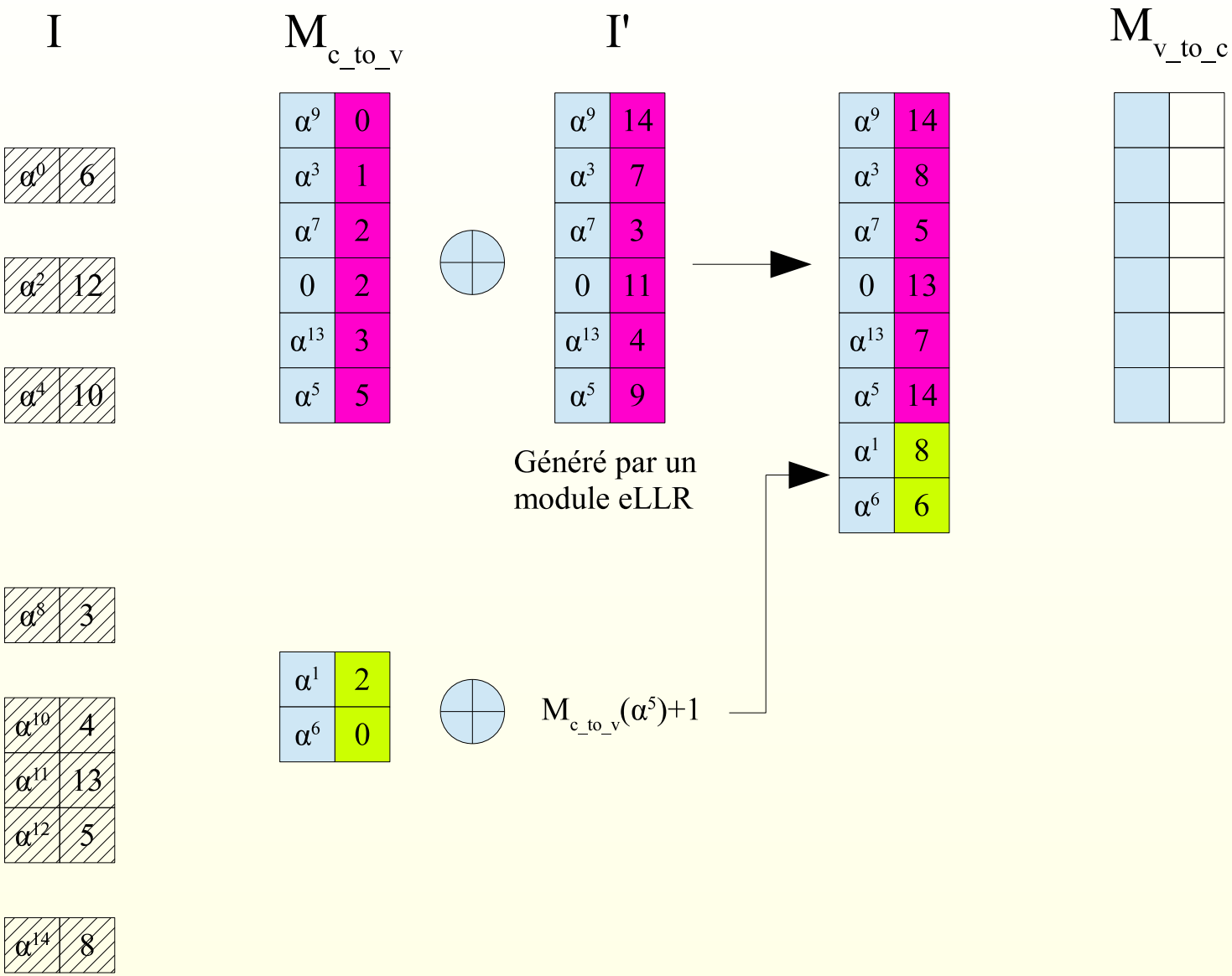
Traitement sous-optimal du nœud de variable



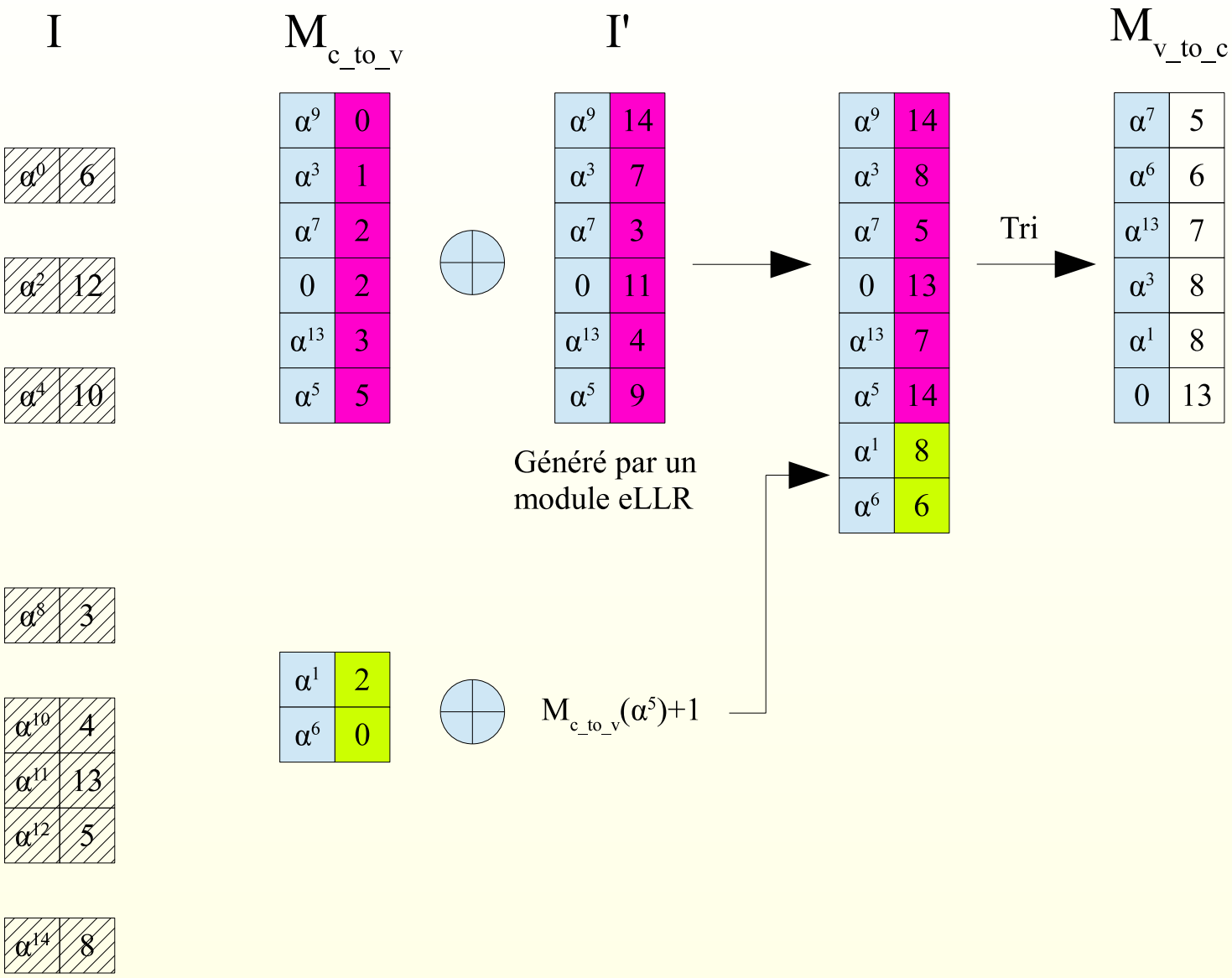
Traitement sous-optimal du nœud de variable



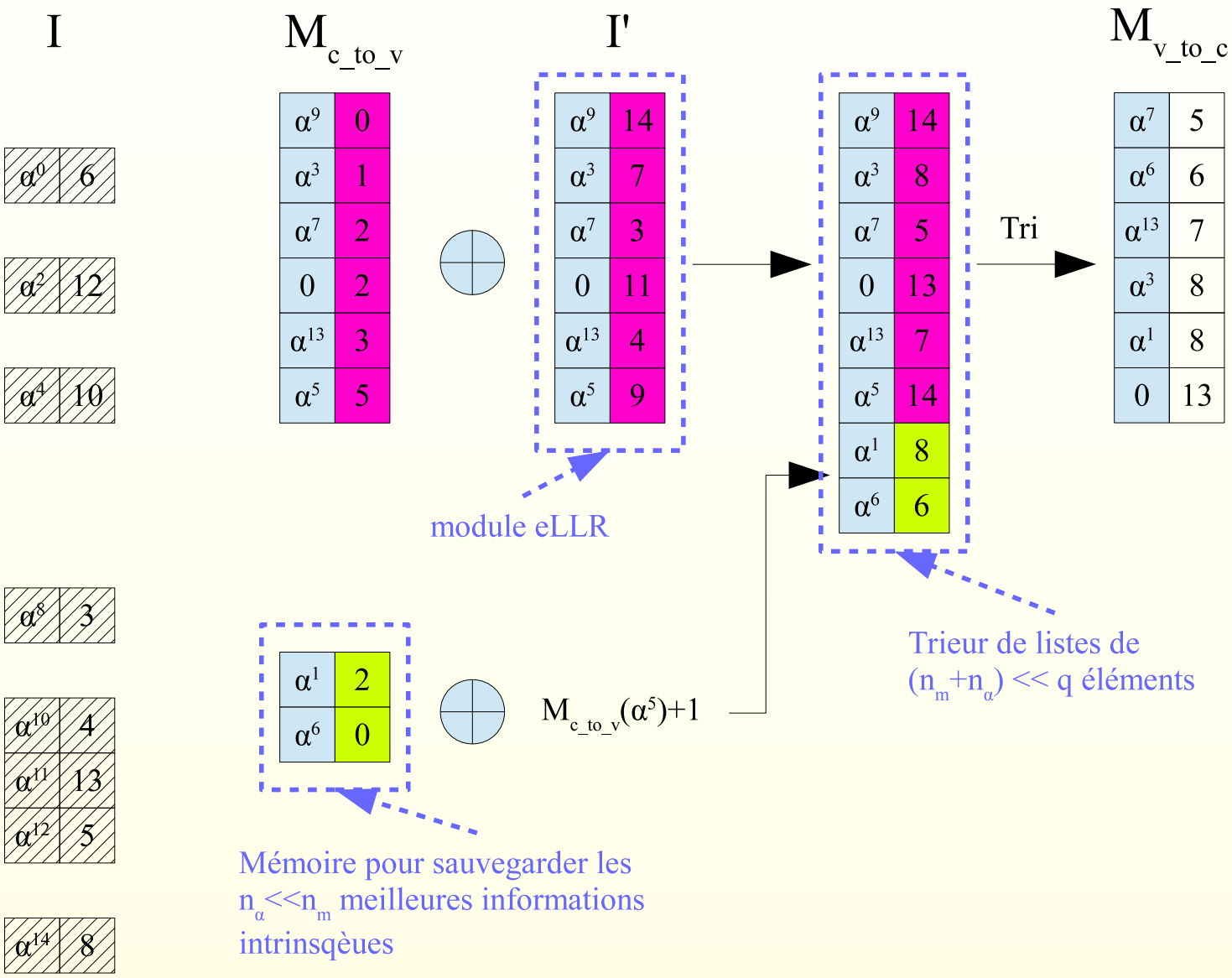
Traitement sous-optimal du nœud de variable



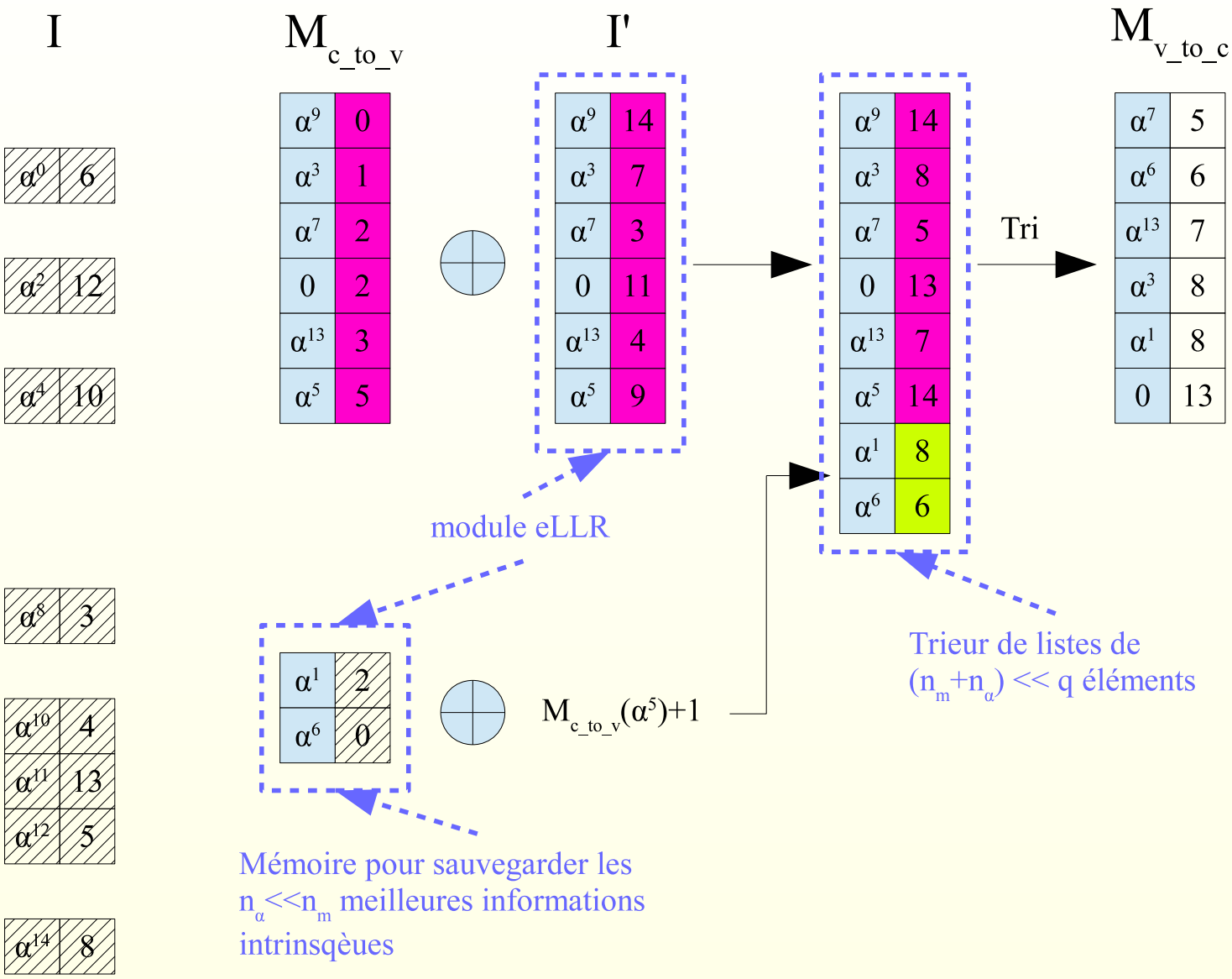
Traitement sous-optimal du nœud de variable



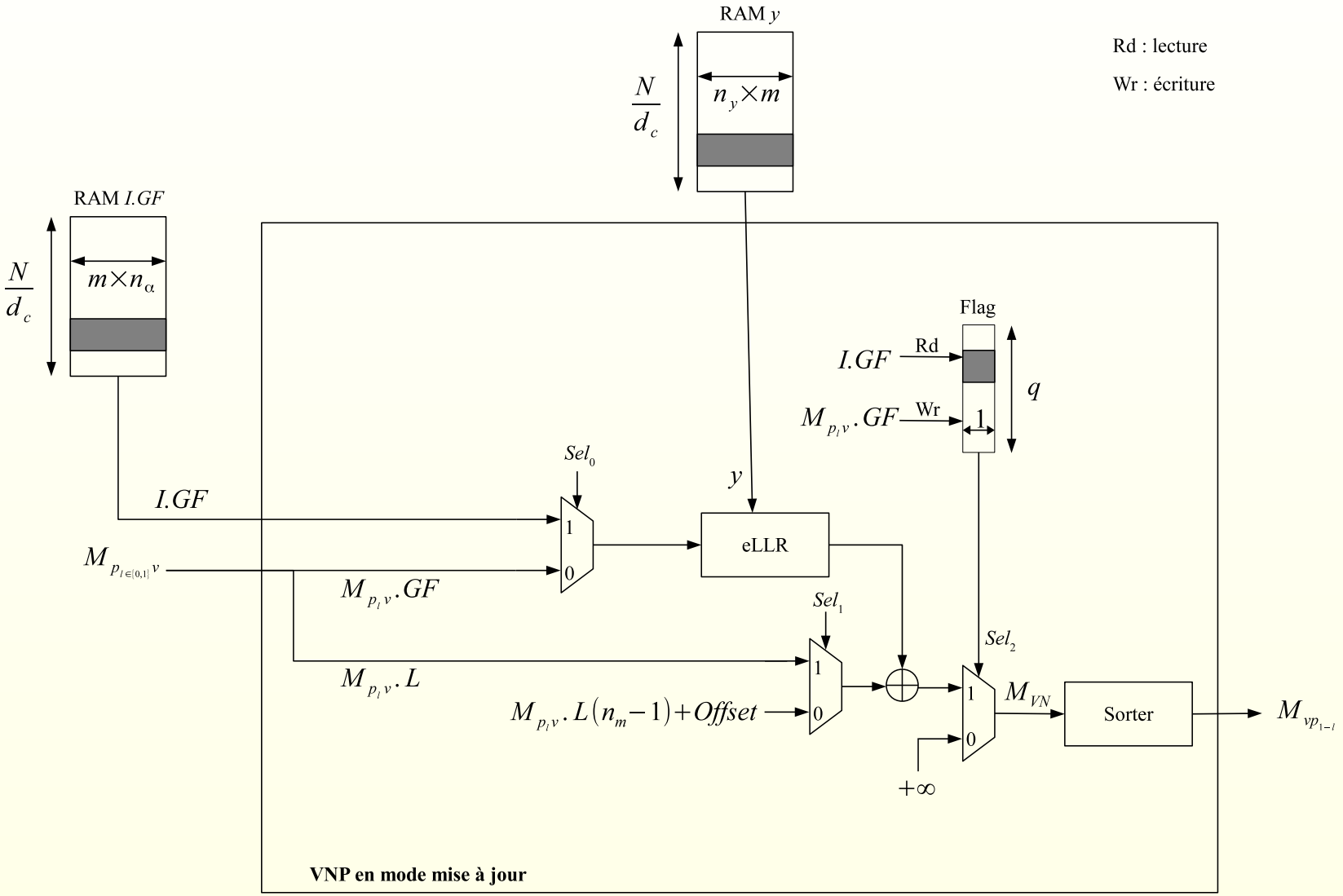
Traitement sous-optimal du nœud de variable



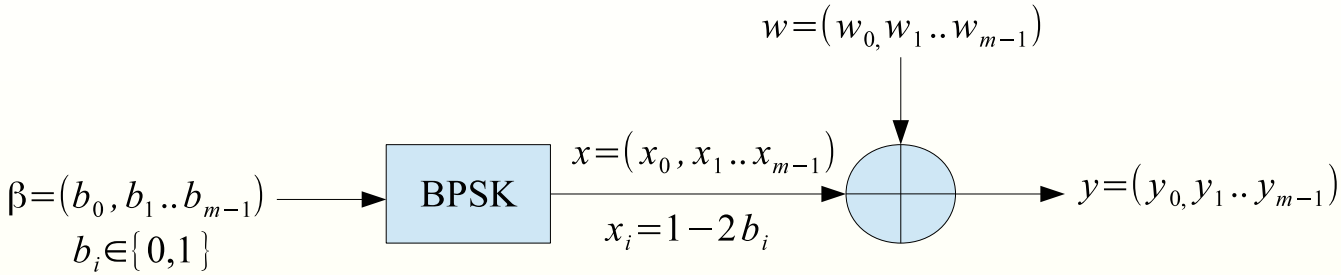
Traitement sous-optimal du nœud de variable



Architecture d'un VNP en mode mise à jour



Génération des LLRs intrinsèques pour une modulation BPSK



En considérant :

$$s_i = \begin{cases} 0, & \text{si } y_i > 0 \\ 1, & \text{sinon} \end{cases} \quad i = 0, 1, \cdots, m-1$$

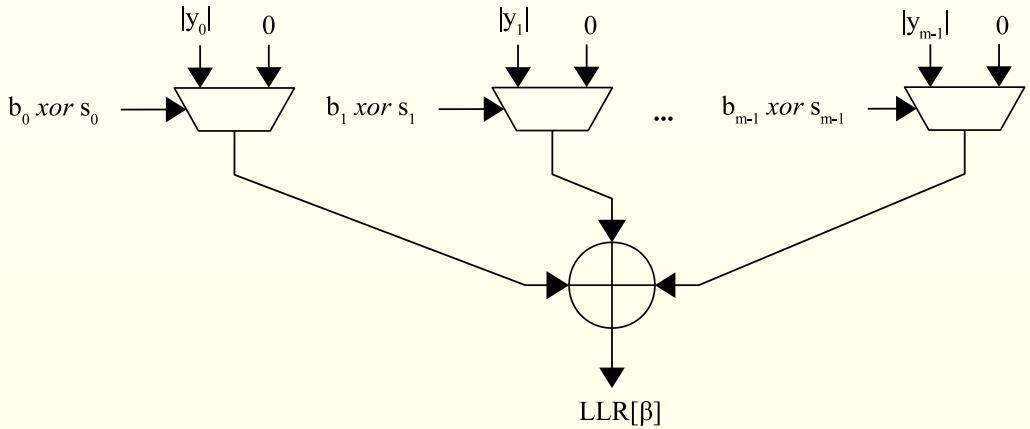
le LLR intrinsèque d'un symbole β modulé en BPSK s'exprime par :

$$LLR(\beta) = \frac{1}{\sigma^2} \sum_{i=0}^{m-1} |y_i| \Delta_i$$

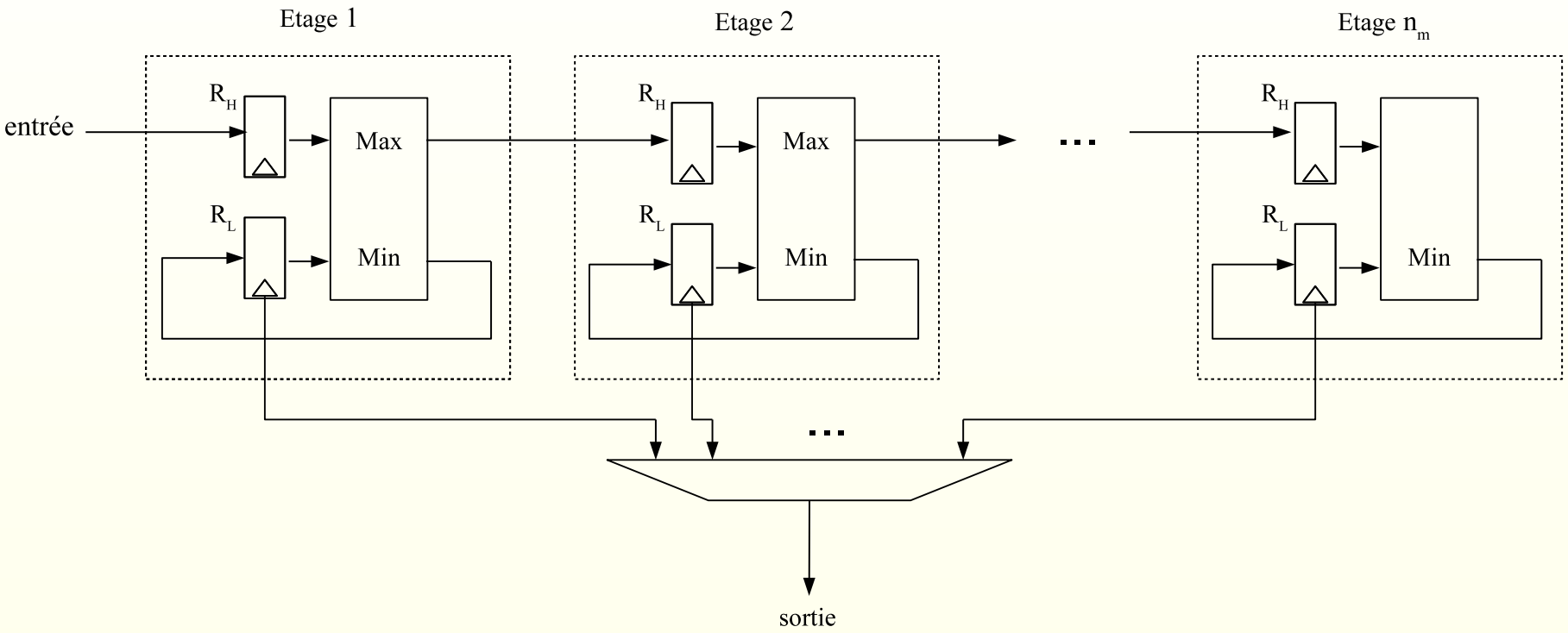
où

$$\Delta_i = \begin{cases} 0, & \text{si } s_i \neq b_i \\ 1, & \text{sinon} \end{cases} \quad i = 0, 1, \cdots, m-1$$

Architecture du module eLLR pour une modulation BPSK



Architecture du module de tri



Cette architecture permet de sélectionner les n_m symboles les plus fiables parmi une liste de taille $n \geq n_m$, dans notre cas $n = n_m + n_\alpha$.

Algorithme décision au niveau d'un Nœud de Variable

I

0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

$M^1_{c\_to\_v}$

α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

$M^2_{c\_to\_v}$

α^{12}	0
α^4	1
α^3	3
α^3	4
α^8	4
α^7	6

Algorithme décision au niveau d'un Nœud de Variable

I

0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

$M^1_{c\_to\_v}$

α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

$M^2_{c\_to\_v}$

α^{12}	0
α^4	1
α^3	3
α^6	4
α^9	4
α^7	6

$I + M^1_{c\_to\_v} + M^2_{c\_to\_v}$ →

APP

α^3	11
α^7	11
α^9	18

Algorithme décision au niveau d'un Nœud de Variable

I

0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

$M^1_{c\_to\_v}$

α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

$$I + M^1_{c\_to\_v} + offset_1$$

$$offset_1 = M^2_{c\_to\_v}(\alpha^7) + 1$$

APP

α^3	11
α^7	11
α^9	18
0	20
α^5	21
α^{13}	14

$M^2_{c\_to\_v}$

α^{12}	0
α^4	1
α^3	3
α^6	4
α^9	4
α^7	6

Algorithme décision au niveau d'un Nœud de Variable

I

0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

$M^1_{c\_to\_v}$

α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

$M^2_{c\_to\_v}$

α^{12}	0
α^4	1
α^3	3
α^6	4
α^9	4
α^7	6

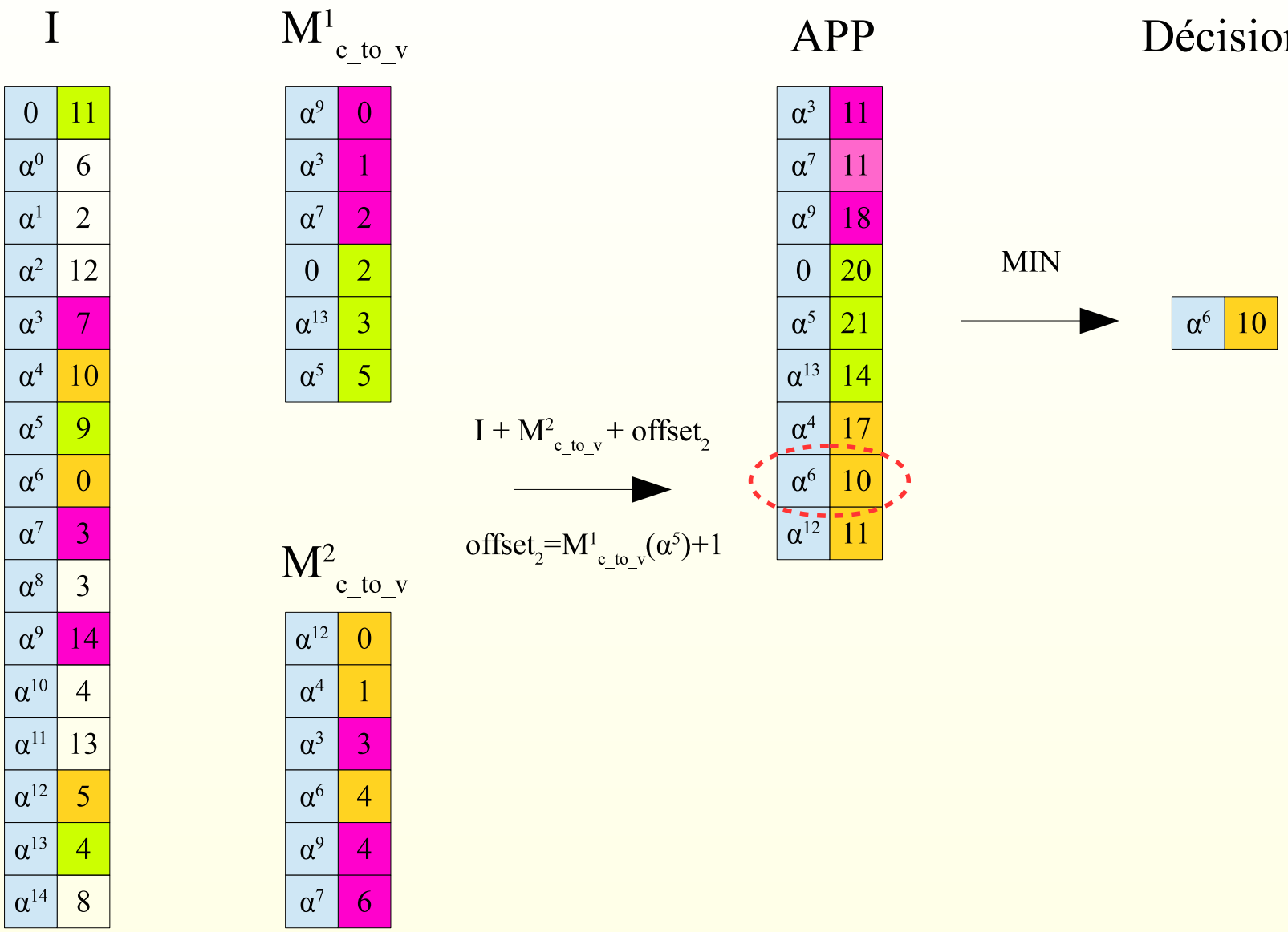
$$I + M^2_{c\_to\_v} + offset_2 \longrightarrow$$

$$offset_2 = M^1_{c\_to\_v}(\alpha^5) + 1$$

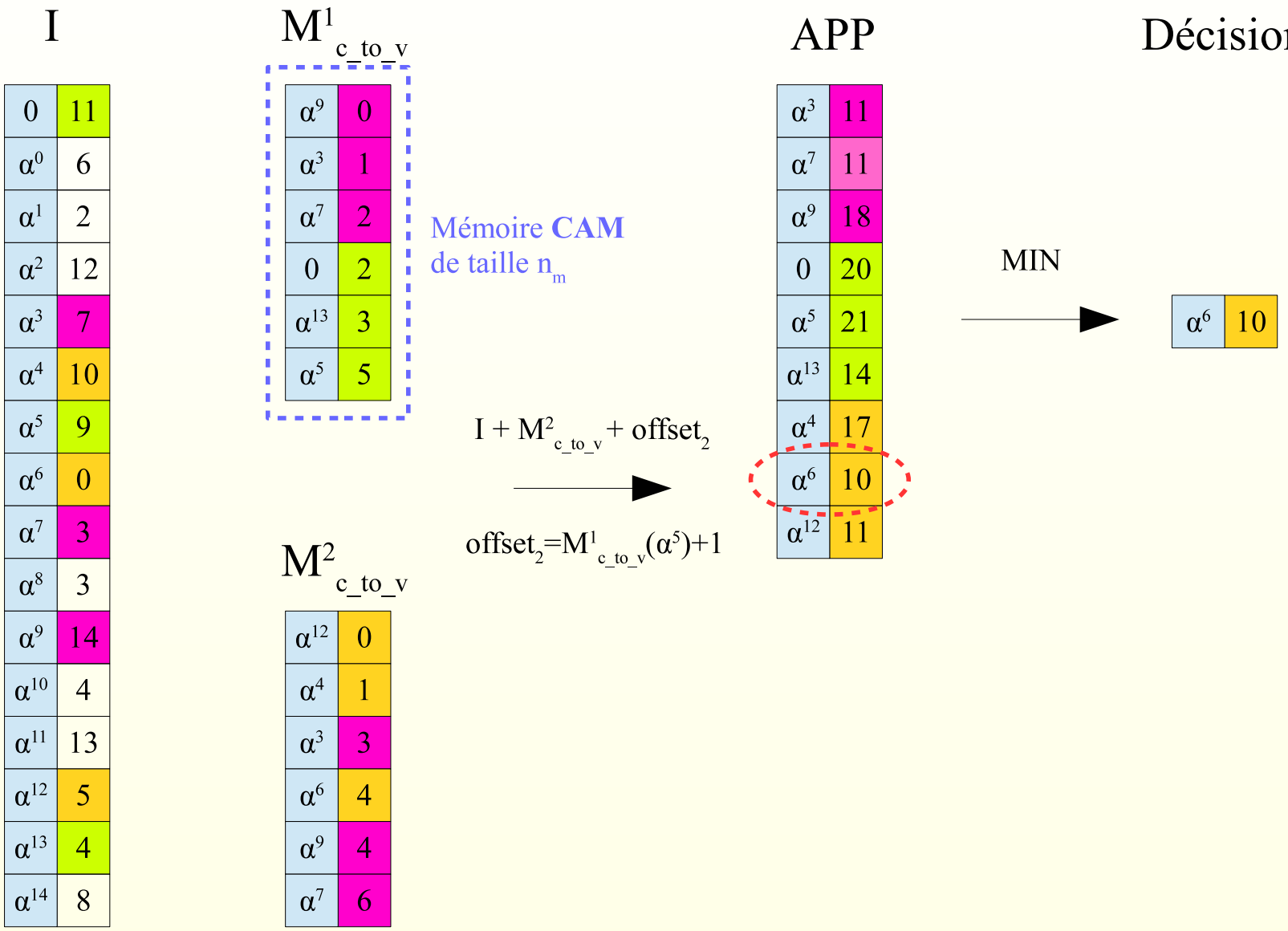
APP

α^3	11
α^7	11
α^9	18
0	20
α^5	21
α^{13}	14
α^4	17
α^6	10
α^{12}	11

Algorithme décision au niveau d'un Nœud de Variable



Algorithme décision au niveau d'un Nœud de Variable



I	
0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

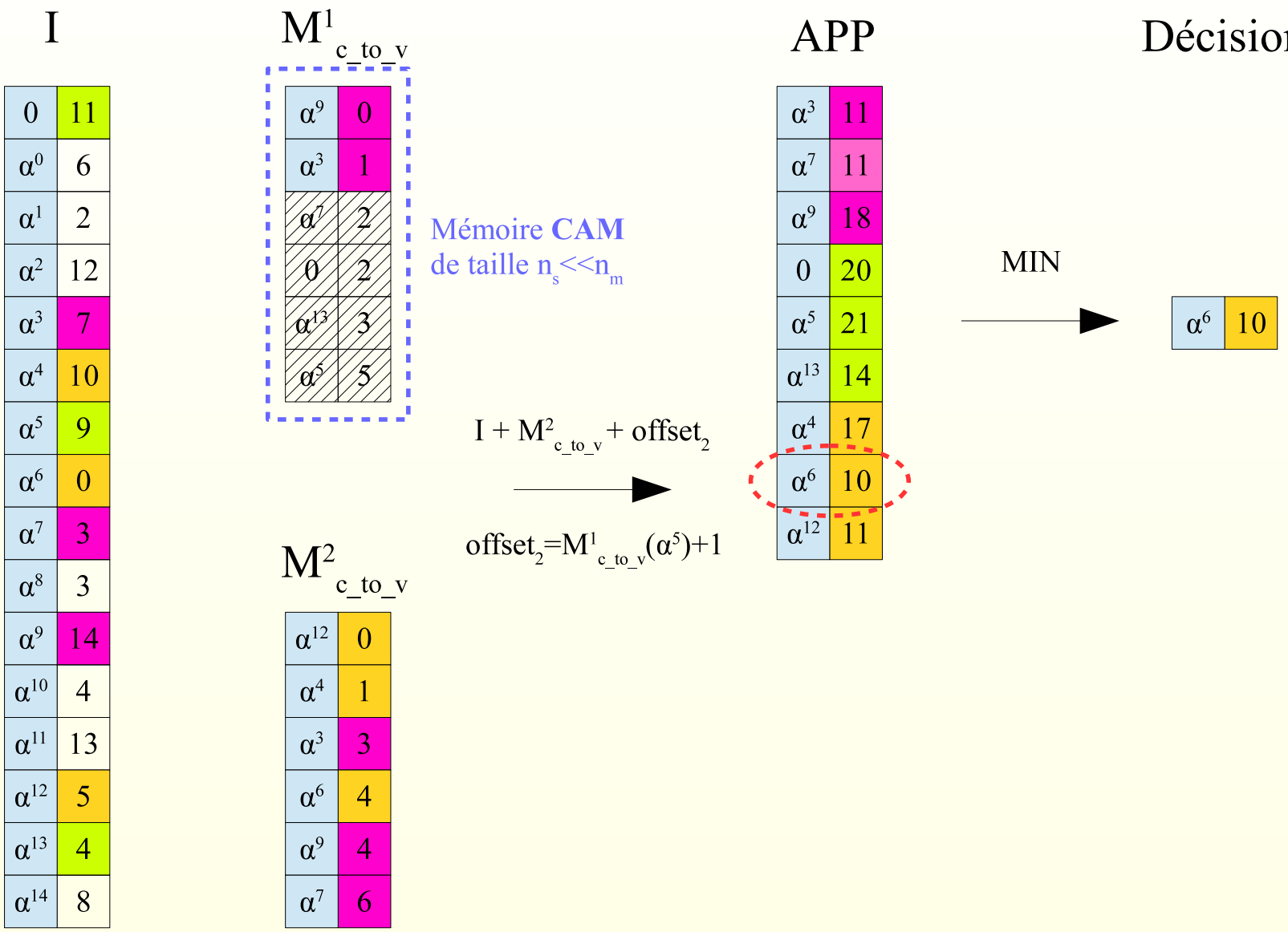
$M^1_{c\_to\_v}$	
α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

APP	
α^3	11
α^7	11
α^9	18
0	20
α^5	21
α^{13}	14
α^4	17
α^6	10
α^{12}	11

Décision	
α^6	10

$M^2_{c\_to\_v}$	
α^{12}	0
α^4	1
α^3	3
α^6	4
α^9	4
α^7	6

Algorithme décision au niveau d'un Nœud de Variable



0	11
α^0	6
α^1	2
α^2	12
α^3	7
α^4	10
α^5	9
α^6	0
α^7	3
α^8	3
α^9	14
α^{10}	4
α^{11}	13
α^{12}	5
α^{13}	4
α^{14}	8

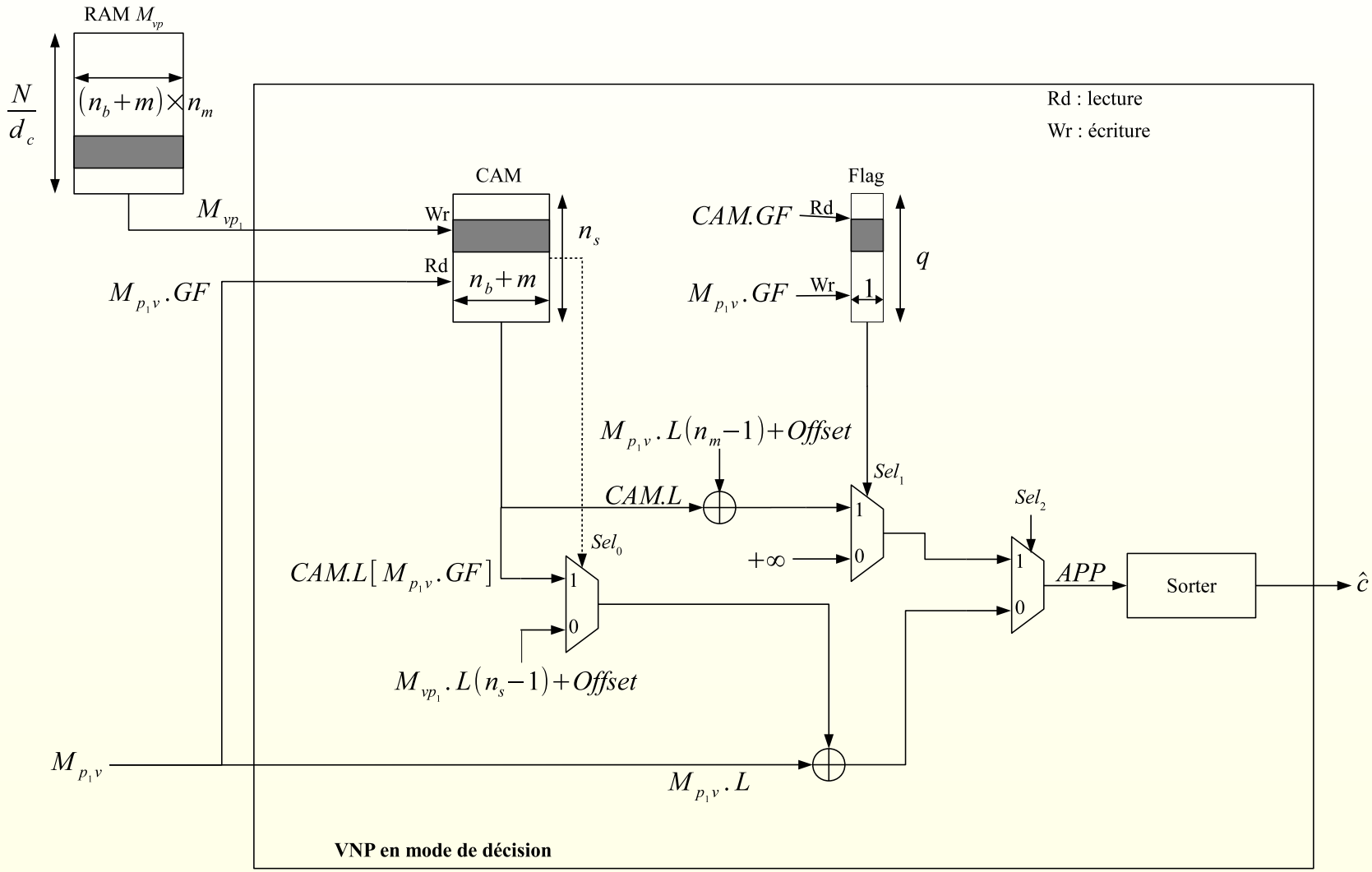
α^9	0
α^3	1
α^7	2
0	2
α^{13}	3
α^5	5

α^{12}	0
α^4	1
α^3	3
α^6	4
α^9	4
α^7	6

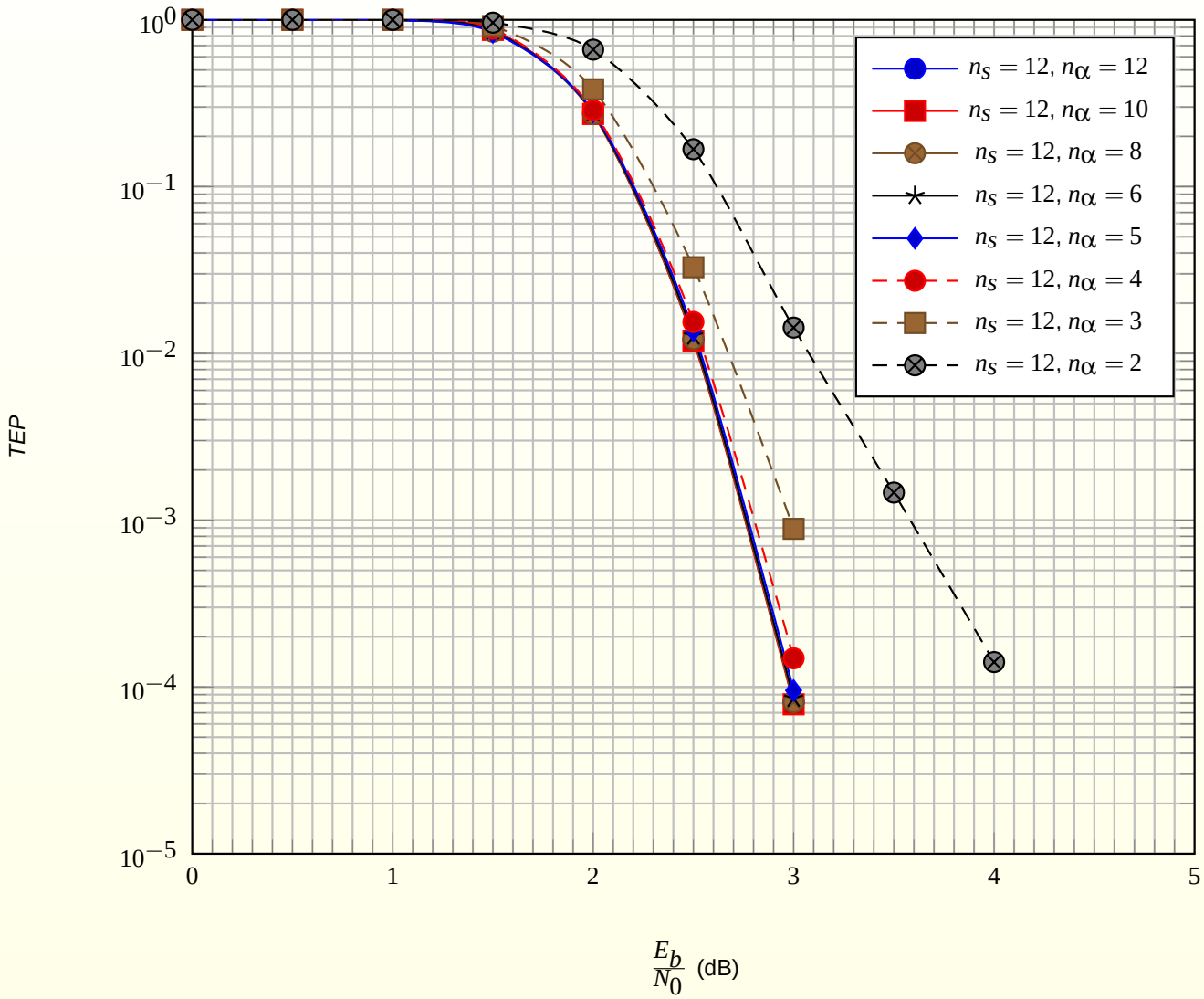
α^3	11
α^7	11
α^9	18
0	20
α^5	21
α^{13}	14
α^4	17
α^6	10
α^{12}	11

α^6	10

Architecture d'un VNP en mode de décision

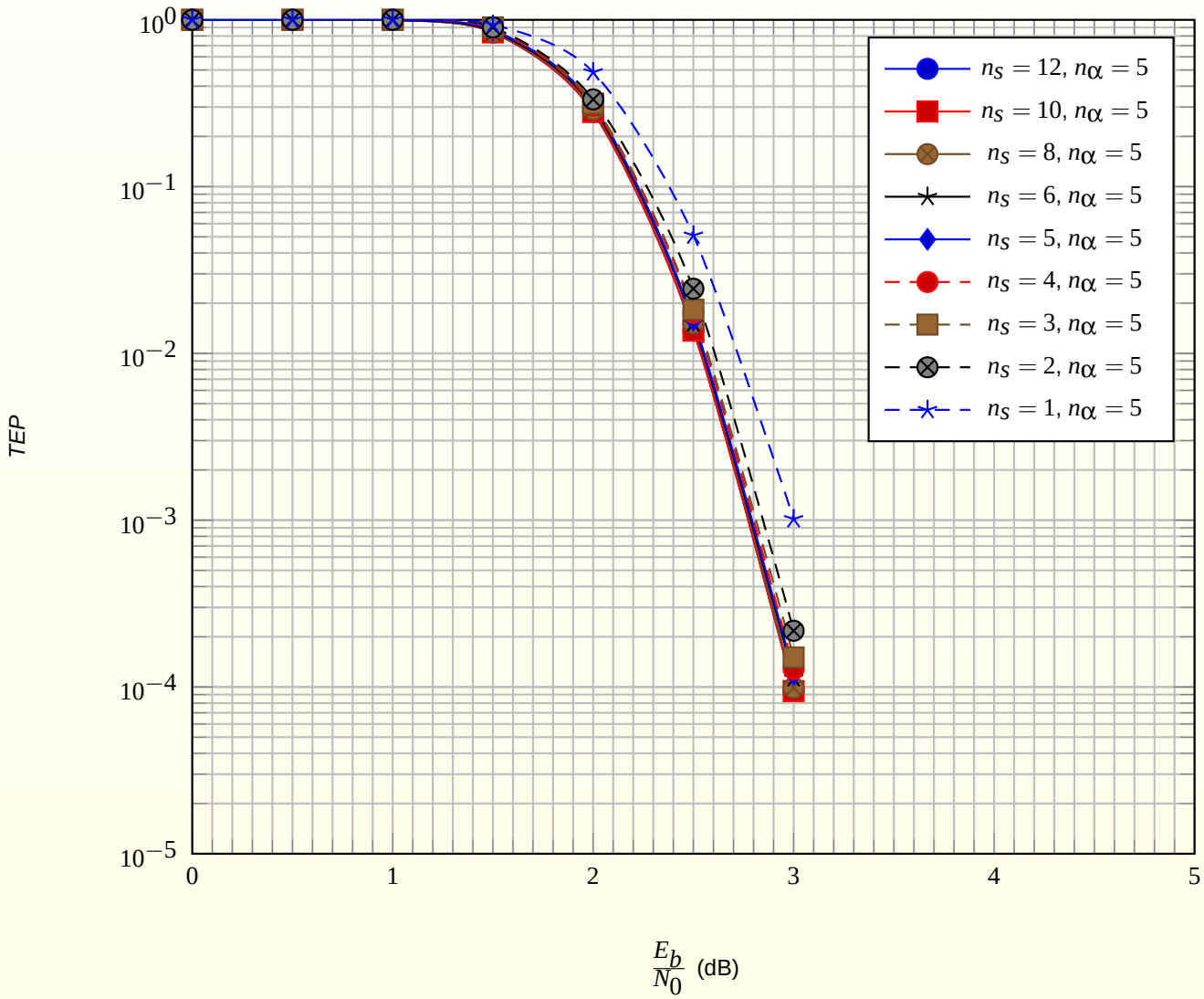


Détermination de la valeur de n_α



Canal AWGN, code $\mathbb{GF}(64)$-LDPC ($N = 1152$ bits, $R = \frac{2}{3}$), EMS ($n_m = 12$, $n_{iter} = 8$)

Détermination de la valeur de n_s



Canal AWGN, code $\mathbb{GF}(64)$-LDPC ($N = 1152$ bits, $R = \frac{2}{3}$), EMS ($n_m = 12$, $n_{iter} = 8$)

Traitement d'un ECN : État de l'art

Nous commençons par comparer les n_m éléments de la première colonne.

Étape d'initialisation (n_m cycles)

$\oplus$	$(u_0, 0)$	$(u_1, 1)$	$(u_2, 2)$	$(u_3, 3)$	$(u_4, 3)$	$(u_5, 4)$	$(u_6, 5)$	$(u_7, 5)$
$(v_0, 0)$	0	1	2	3	3	4	5	5
$(v_1, 0)$	0	1	2	3	3	4	5	5
$(v_2, 1)$	1	2	3	4	4	5	6	6
$(v_3, 2)$	2	3	4	5	5	6	7	7
$(v_4, 4)$	4	5	6	7	7	8	9	9
$(v_5, 4)$	4	5	6	7	7	8	9	9
$(v_6, 5)$	5	6	7	8	8	9	10	10
$(v_7, 6)$	6	7	8	9	9	10	11	11

Traitement d'un ECN : État de l'art

Extraction de l'élément $(u_0 + v_0)$ qui sera remplacé par le nouveau candidat $(u_1 + v_0)$ dans le comparateur

Extraction du premier élément

$\oplus$	$(u_0, 0)$	$(u_1, 1)$	$(u_2, 2)$	$(u_3, 3)$	$(u_4, 3)$	$(u_5, 4)$	$(u_6, 5)$	$(u_7, 5)$
$(v_0, 0)$	0	1	2	3	3	4	5	5
$(v_1, 0)$	0	1	2	3	3	4	5	5
$(v_2, 1)$	1	2	3	4	4	5	6	6
$(v_3, 2)$	2	3	4	5	5	6	7	7
$(v_4, 4)$	4	5	6	7	7	8	9	9
$(v_5, 4)$	4	5	6	7	7	8	9	9
$(v_6, 5)$	5	6	7	8	8	9	10	10
$(v_7, 6)$	6	7	8	9	9	10	11	11

Traitement d'un ECN : État de l'art

Extraction de l'élément $(u_0 + v_1)$ qui sera remplacé par le nouveau candidat $(u_1 + v_1)$ dans le comparateur

Extraction du deuxième élément

$\oplus$	$(u_0, 0)$	$(u_1, 1)$	$(u_2, 2)$	$(u_3, 3)$	$(u_4, 3)$	$(u_5, 4)$	$(u_6, 5)$	$(u_7, 5)$
$(v_0, 0)$	0	1	2	3	3	4	5	5
$(v_1, 0)$	0	1	2	3	3	4	5	5
$(v_2, 1)$	1	2	3	4	4	5	6	6
$(v_3, 2)$	2	3	4	5	5	6	7	7
$(v_4, 4)$	4	5	6	7	7	8	9	9
$(v_5, 4)$	4	5	6	7	7	8	9	9
$(v_6, 5)$	5	6	7	8	8	9	10	10
$(v_7, 6)$	6	7	8	9	9	10	11	11

Traitement d'un ECN : État de l'art

Extraction de l'élément $(u_1 + v_0)$ qui sera remplacé par le nouveau candidat $(u_2 + v_0)$ dans le comparateur

Extraction du troisième élément

$\oplus$	$(u_0, 0)$	$(u_1, 1)$	$(u_2, 2)$	$(u_3, 3)$	$(u_4, 3)$	$(u_5, 4)$	$(u_6, 5)$	$(u_7, 5)$
$(v_0, 0)$	0	1 →	2	3	3	4	5	5
$(v_1, 0)$	0	1	2	3	3	4	5	5
$(v_2, 1)$	1	2	3	4	4	5	6	6
$(v_3, 2)$	2	3	4	5	5	6	7	7
$(v_4, 4)$	4	5	6	7	7	8	9	9
$(v_5, 4)$	4	5	6	7	7	8	9	9
$(v_6, 5)$	5	6	7	8	8	9	10	10
$(v_7, 6)$	6	7	8	9	9	10	11	11

Traitement d'un ECN : État de l'art

Inconvénients de cette méthode :

1. Obligation d'attendre n_m cycles avant de sortir la première valeur
   $\Rightarrow$ Pénalise fortement le débit de décodage
2. Réalisation de n_m comparaisons par cycle d'horloge pour maintenir la liste triée
   $\Rightarrow$ Complexité matérielle importante

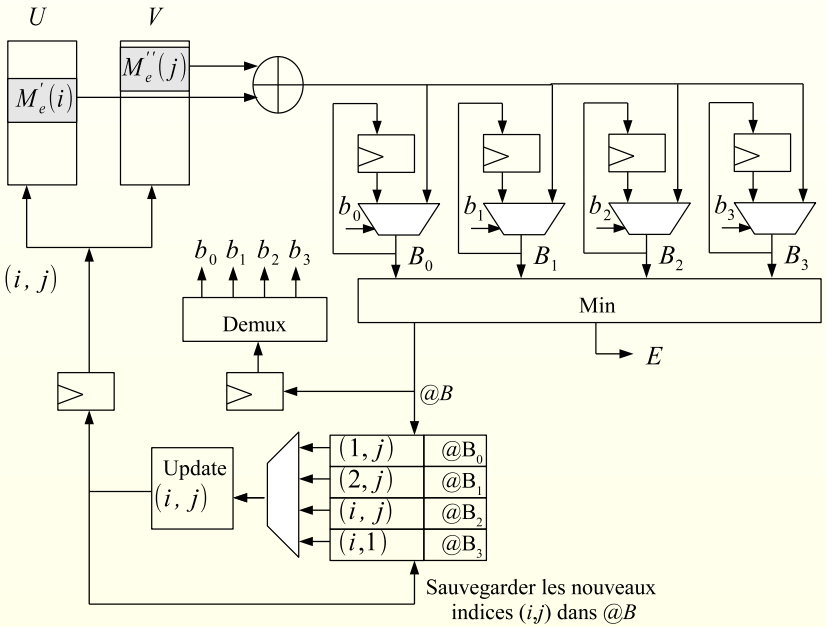
Extraction du troisième élément

$\oplus$	$(u_0, 0)$	$(u_1, 1)$	$(u_2, 2)$	$(u_3, 3)$	$(u_4, 3)$	$(u_5, 4)$	$(u_6, 5)$	$(u_7, 5)$
$(v_0, 0)$	0	1 →	2	3	3	4	5	5
$(v_1, 0)$	0	1	2	3	3	4	5	5
$(v_2, 1)$	1	2	3	4	4	5	6	6
$(v_3, 2)$	2	3	4	5	5	6	7	7
$(v_4, 4)$	4	5	6	7	7	8	9	9
$(v_5, 4)$	4	5	6	7	7	8	9	9
$(v_6, 5)$	5	6	7	8	8	9	10	10
$(v_7, 6)$	6	7	8	9	9	10	11	11

Traitement d'un ECN : État de l'art

Méthode L-Bubble

1. $4n_m$ additions au lieu de n_m^2.
2. 4 comparaisons par cycle au lieu de n_m
3. Un seul cycle pour sortir le premier élément (nécessairement $(u_0 + v_0)$)

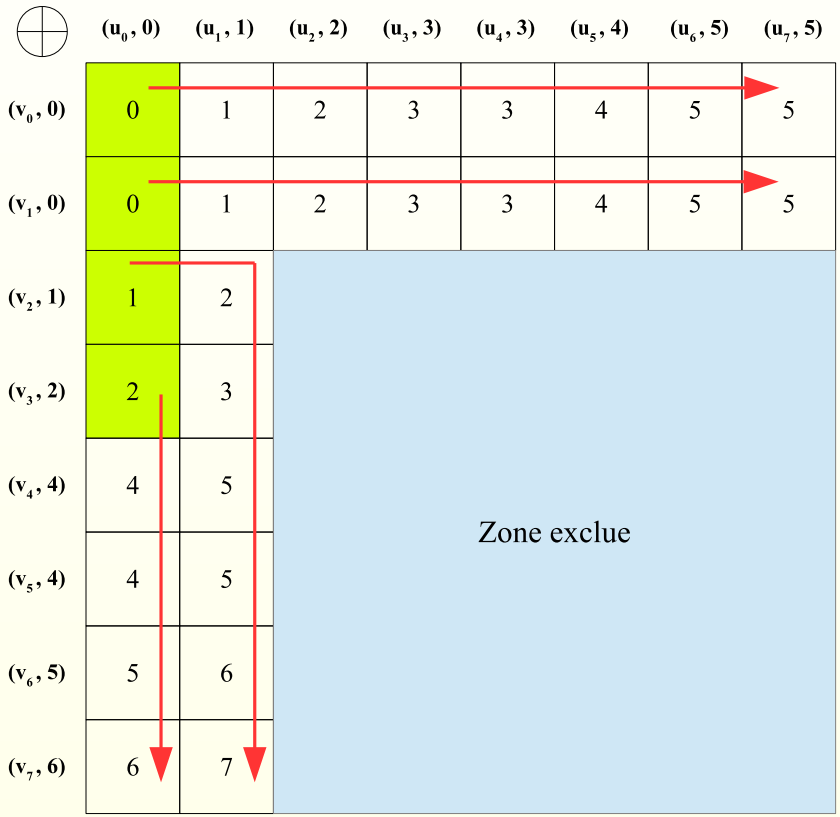


	$(u_0, 0)$	$(u_1, 1)$	$(u_2, 2)$	$(u_3, 3)$	$(u_4, 3)$	$(u_5, 4)$	$(u_6, 5)$	$(u_7, 5)$
$(v_0, 0)$	0	1	2	3	3	4	5	5
$(v_1, 0)$	0	1	2	3	3	4	5	5
$(v_2, 1)$	1	2						
$(v_3, 2)$	2	3						
$(v_4, 4)$	4	5						
$(v_5, 4)$	4	5						
$(v_6, 5)$	5	6						
$(v_7, 6)$	6	7						

Zone exclue

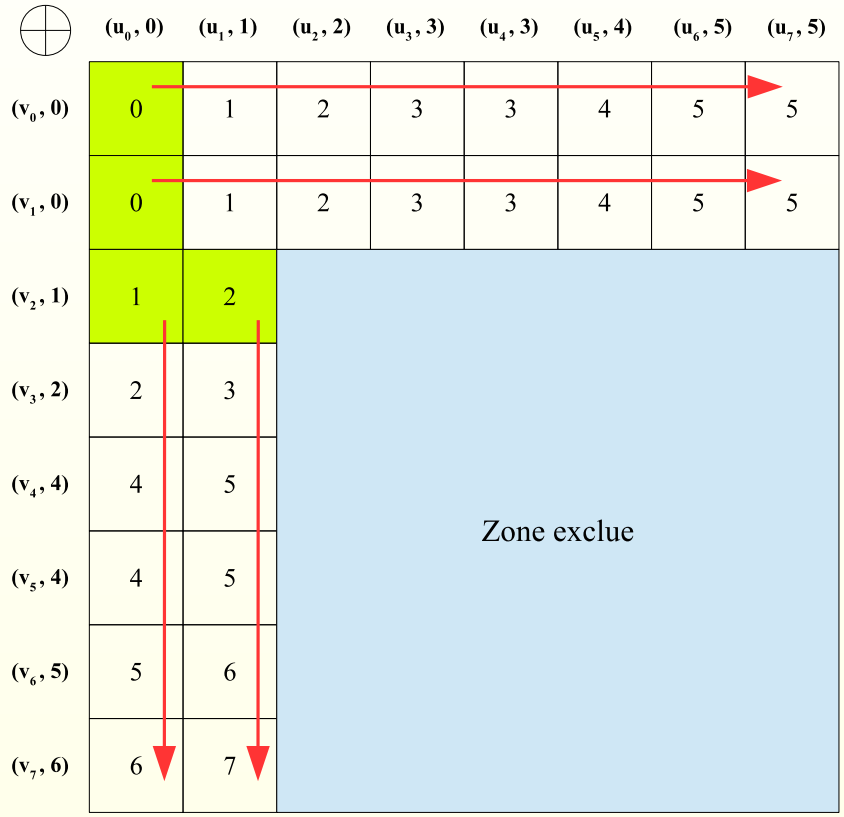
Traitement d'un ECN : Notre contribution

Méthode L-Bubble



1 trajectoire en forme de L $\Rightarrow$ L-Bubble

Méthode S-Bubble



4 trajectoires rectilignes $\Rightarrow$ Straight-Bubble

Architecture S-Bubble

Architecture parallèle d'un Processeur de Nœud de Parité

Réalisation matérielle du décodeur

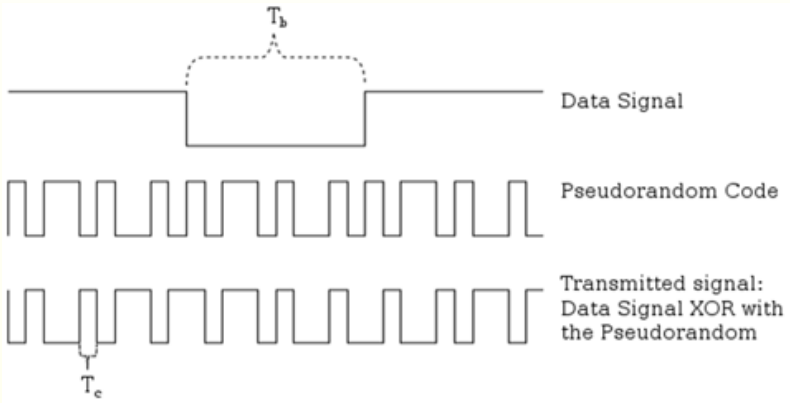
machine à état du décodeur parametre => latence vn, cn et décodeur

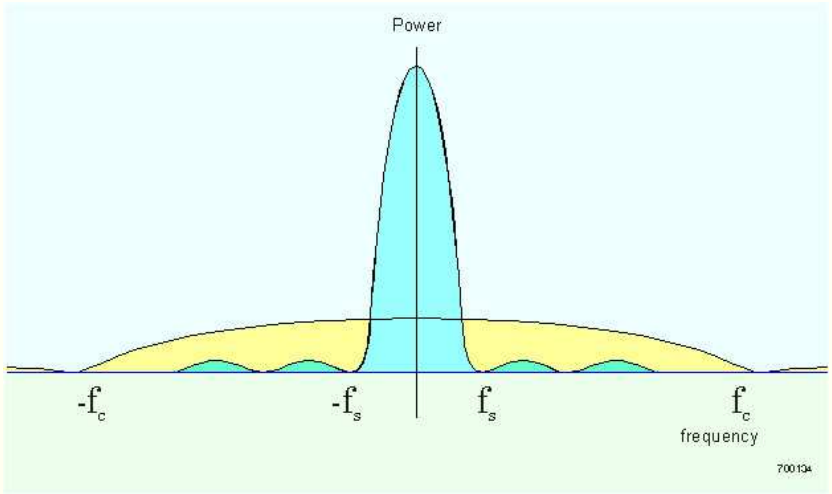
Résultat de synthèses

n'oublier pas la latence

Sommaire

Les modulations à étalement de spectre





Les applications haut débit utilisent de l'étalement de spectre M-aire, c'est à dire que la modulation est effectuée par groupes de $\log_2 M$ bits.

La modulation de Walsh-Hadamard

- La modulation de Walsh-Hadamard est une modulation à étalement de spectre orthogonale qui utilise $M = 2^m$ séquences de Walsh-Hadamard pour moduler des mots de m bits.
- Les séquences sont obtenues à partir d'une matrice de Hadamard de dimension $M \times M$.
- Chaque séquence contient M chips.

$$H_1 = 1$$

$$H_{2^n} = \begin{pmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{pmatrix} \quad n \geq 1$$

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

$$H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

Application : La norme IS-95 définie par Qualcomm (CDMAone) utilise une modulation orthogonale 64-aire dans le *Reverse Link* (mobile-to-base).

La modulation CCSK

La modulation par décalage cyclique de code (CCSK) est une modulation non-orthogonale M-aire qui utilise M séquences obtenues en décalant circulairement une séquence pseudo-aléatoire de M chips.

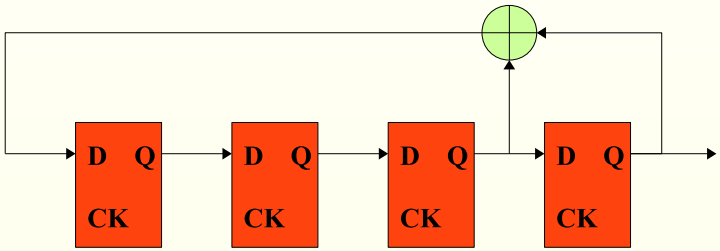
Modulation CCSK d'ordre 8

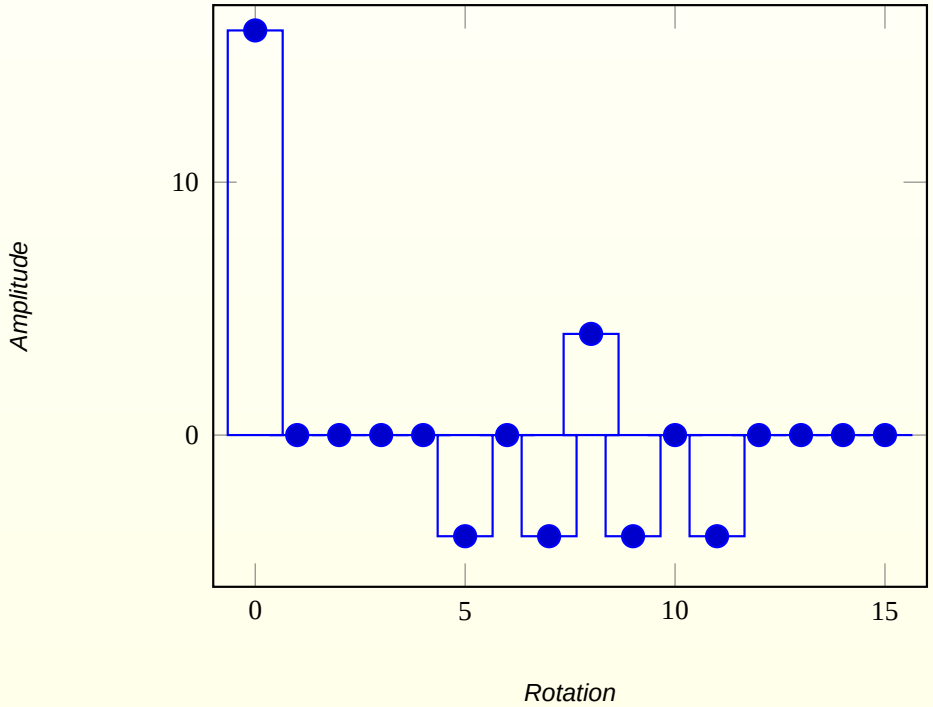
Symbole	Séquence CCSK
000	10000000
001	01000000
010	00100000
011	00010000
100	00001000
101	00000100
110	00000010
111	00000001

Génération de la séquence pseudo-aléatoire

- Une bonne séquence pseudo-aléatoire possède un maximum d'écart entre le pic à l'origine et les pics secondaires de sa fonction d'auto-corrélation.
- La séquence pseudo-aléatoire peut être générée par un registre à décalage à rétroaction linéaire.

$$P(x) = x^4 + x^3 + 1$$



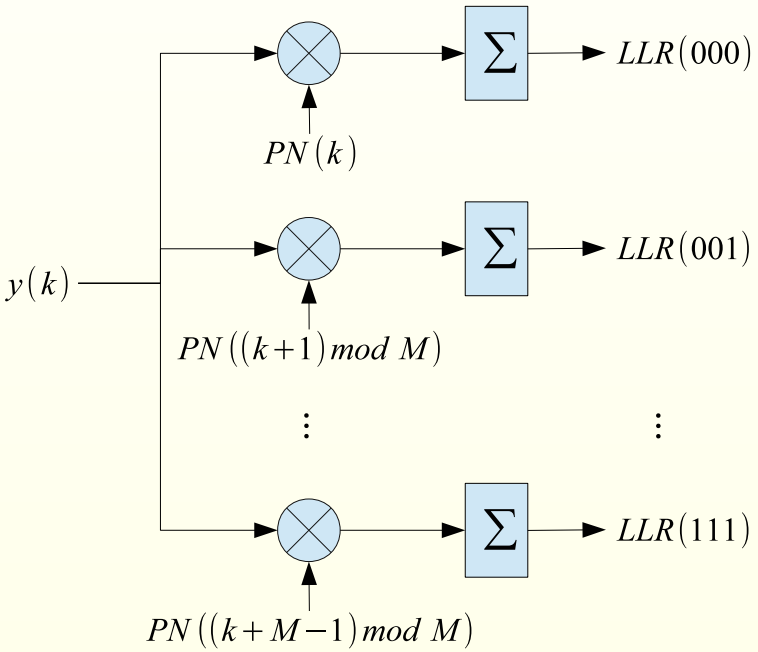


Ce registre à décalage génère une séquence périodique de 15 chips à laquelle nous ajoutons un chip supplémentaire :

$$PN = 1000100110101111$$

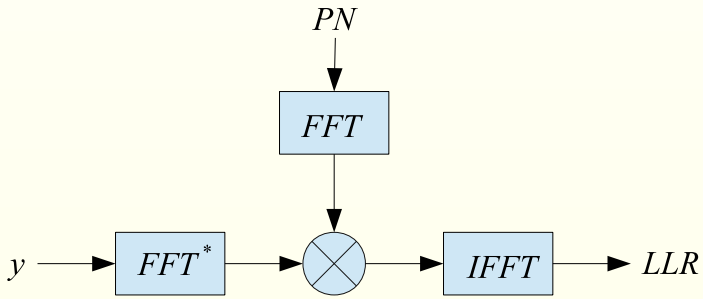
Démodulation CCSK

Le démodulateur détermine les fiabilités des symboles en calculant le produit d'inter-corrélation de la séquence pseudo-aléatoire et du signal reçu.



Complexité de l'ordre de q^2

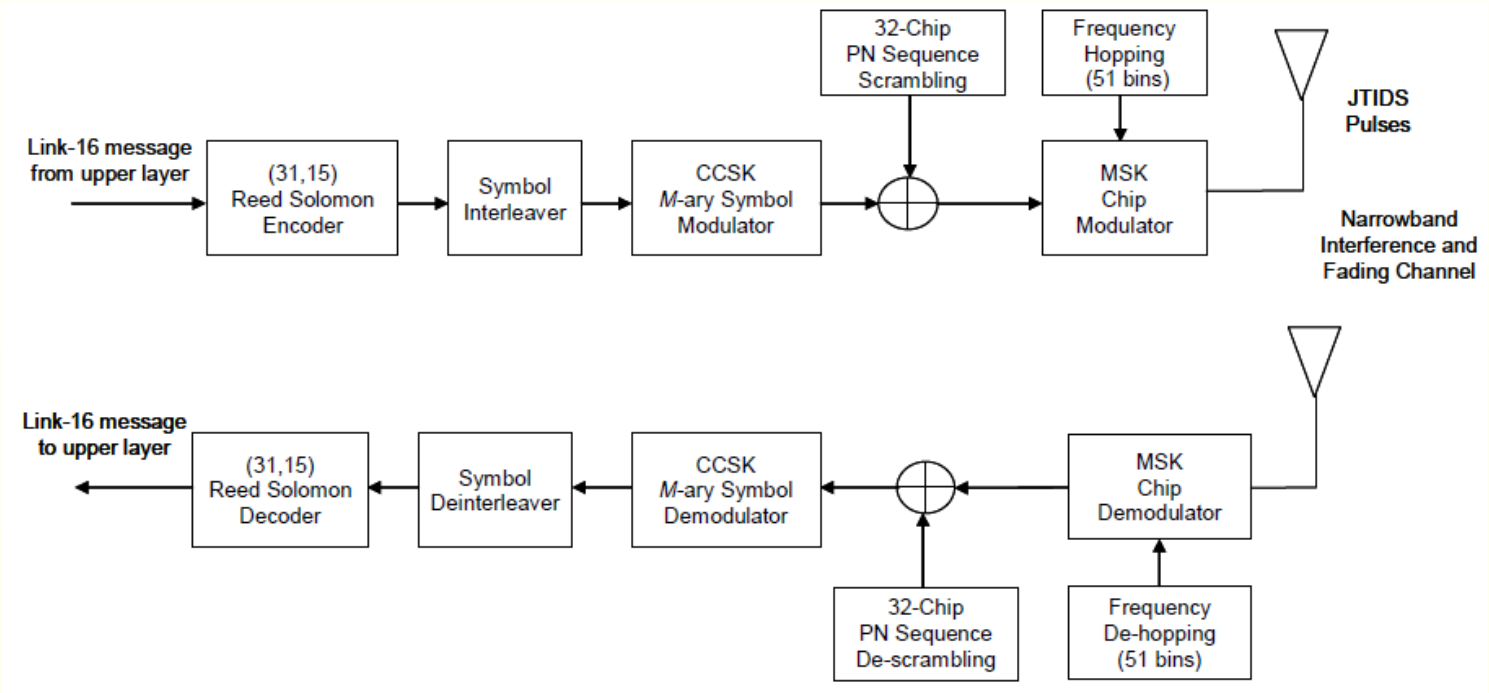
$\Rightarrow$



Complexité de l'ordre de $q \times \log_2 q$

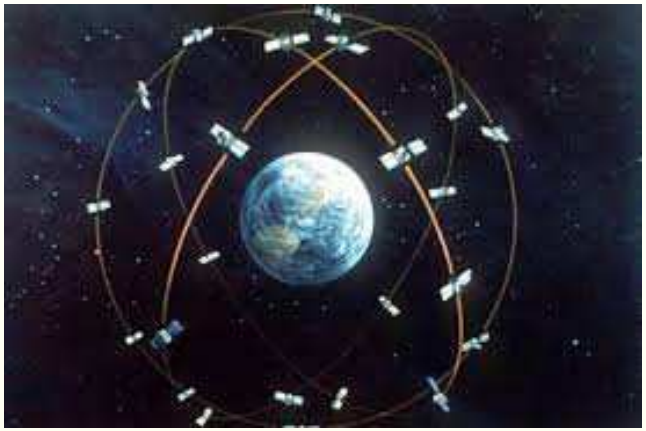
Applications de la modulation CCSK

Le système militaire Link-16/JTIDS utilise une modulation CCSK d'ordre 32 concaténée avec un code de Reed-Solomon.



La Liaison-16 (Link-16) est un standard de liaison de données tactiques de l'OTAN pour l'échange d'informations tactiques entre des unités militaires.

Applications de la modulation CCSK



Possibilité d'utiliser la modulation pour améliorer l'efficacité spectrale des futures signaux des systèmes de positionnement par satellite (GPS, Galileo. . .).

A. Garcia-Pena, D. Salos, O. Julien, L. Ries and T. Grelier, "Analysis of the use of CSK for future GNSS Signals," published in ION GNSS 2013, 26th International Technical Meeting of the Satellite Division of the Institute of Navigation, United Sates, 2013.

Notre contribution

- Le système JTIDS associe la démodulation CCSK à un décodeur de Reed-Solomon à décision dure.
- Les études considérant une démodulation CCSK souple se focalisent sur les décodeurs binaires (LDPC binaire, Viterbi).

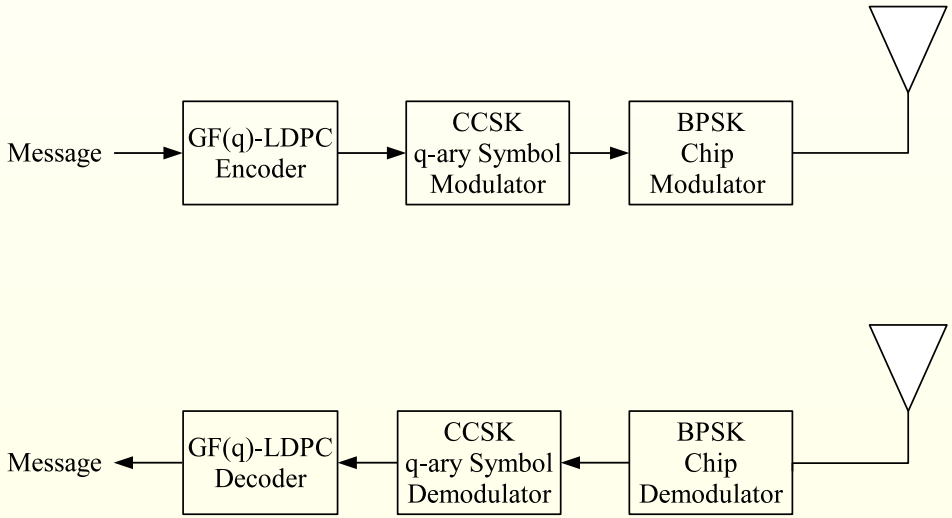
Nous proposons de :

- étudier les performances de la modulation CCSK associée à un code LDPC non-binaire en considérant un décodage itératif souple.

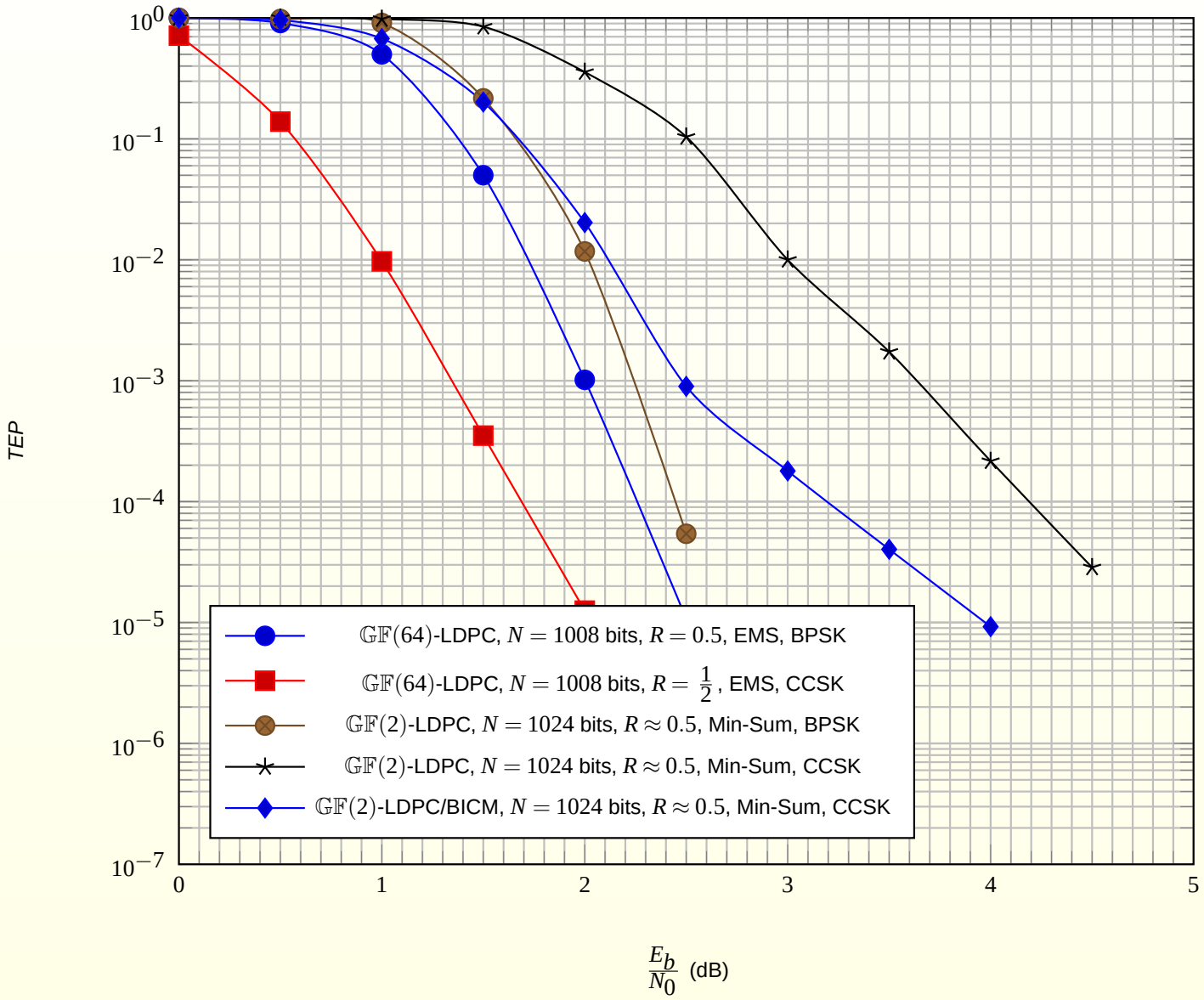
- mettre en valeur la disposition naturelle de la démodulation CCSK à être fusionnée avec une égalisation fréquentielle.

Concaténation d'un code LDPC non-binaire et une modulation CCSK

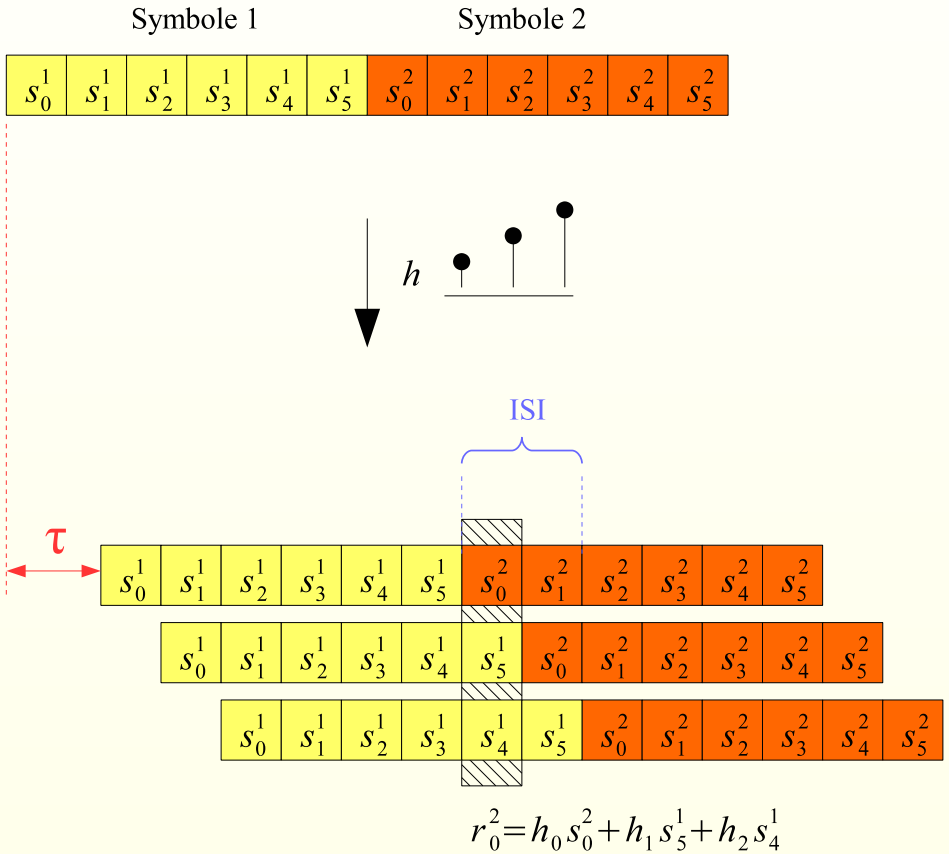
- La concaténation est directe et n'ajoute aucune complexité à l'émetteur.
- Le calcul des LLRs au récepteur peut s'effectuer efficacement par des opérations de FFT et FFT inverse.
- Intuitivement : aucune perte d'information entre le démodulateur et le décodeur $\rightarrow$ le décodeur bénéficie pleinement de la diversité temporelle des séquences CCSK $\rightarrow$ des performances optimales.



Performance dans un canal AWGN

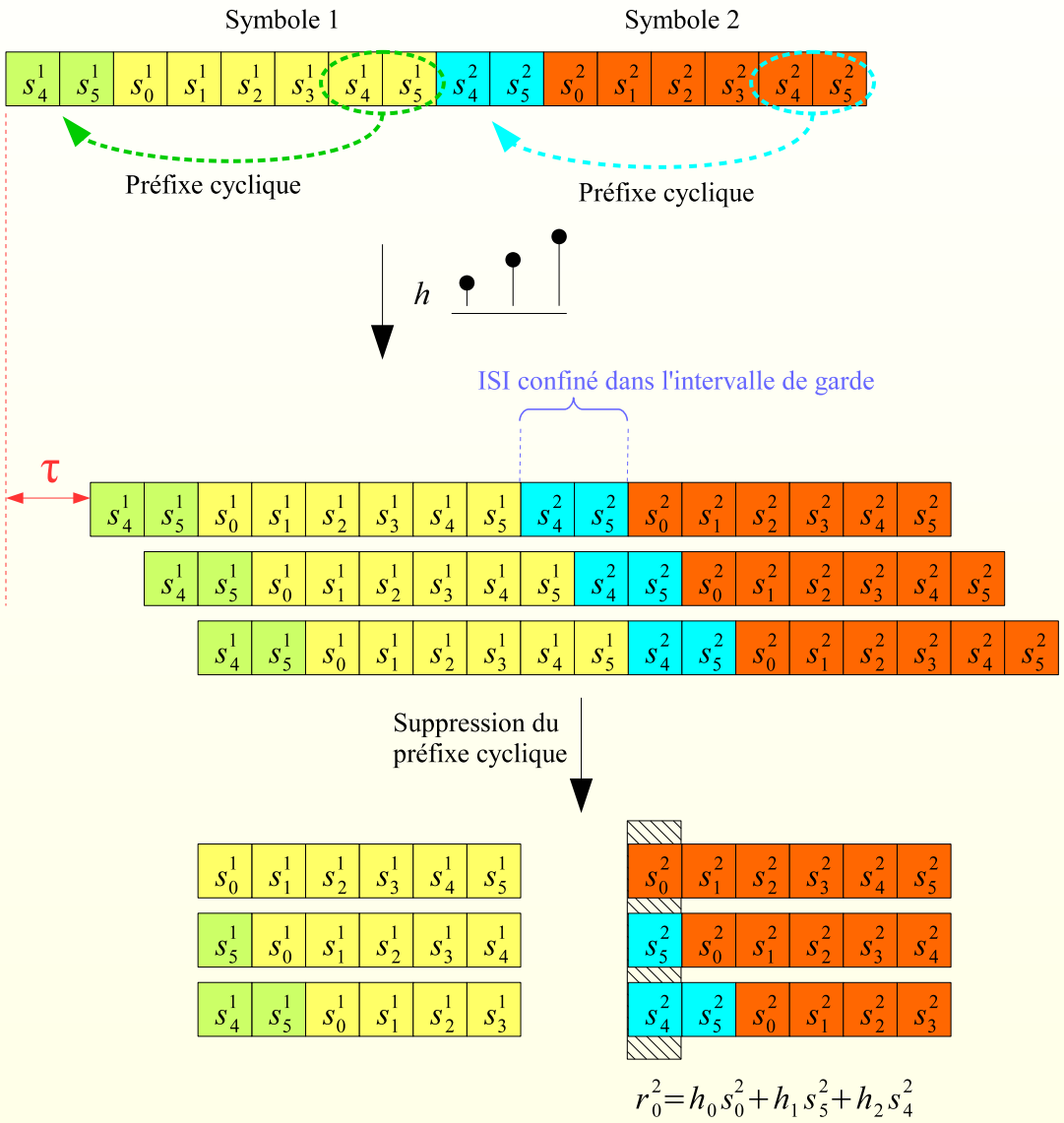


Interférence InterSymbole



Le signal reçu est convolué linéairement avec la réponse impulsionnelle du canal.

Préfixe cyclique



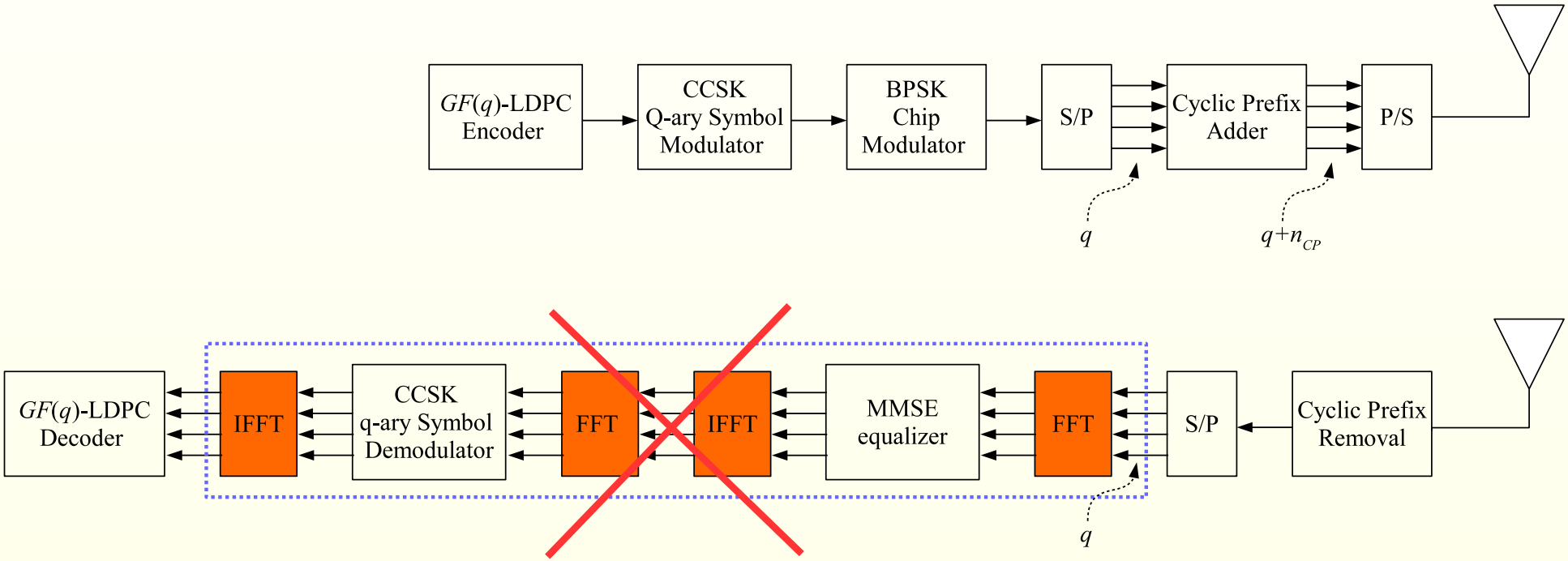
Le signal reçu est convolué circulairement avec la réponse impulsionnelle du canal :

$$\begin{aligned} r^i &= s^i \odot_c h \\ &= IFFT\left(FFT(s^i) \cdot FFT(h)\right) \end{aligned}$$

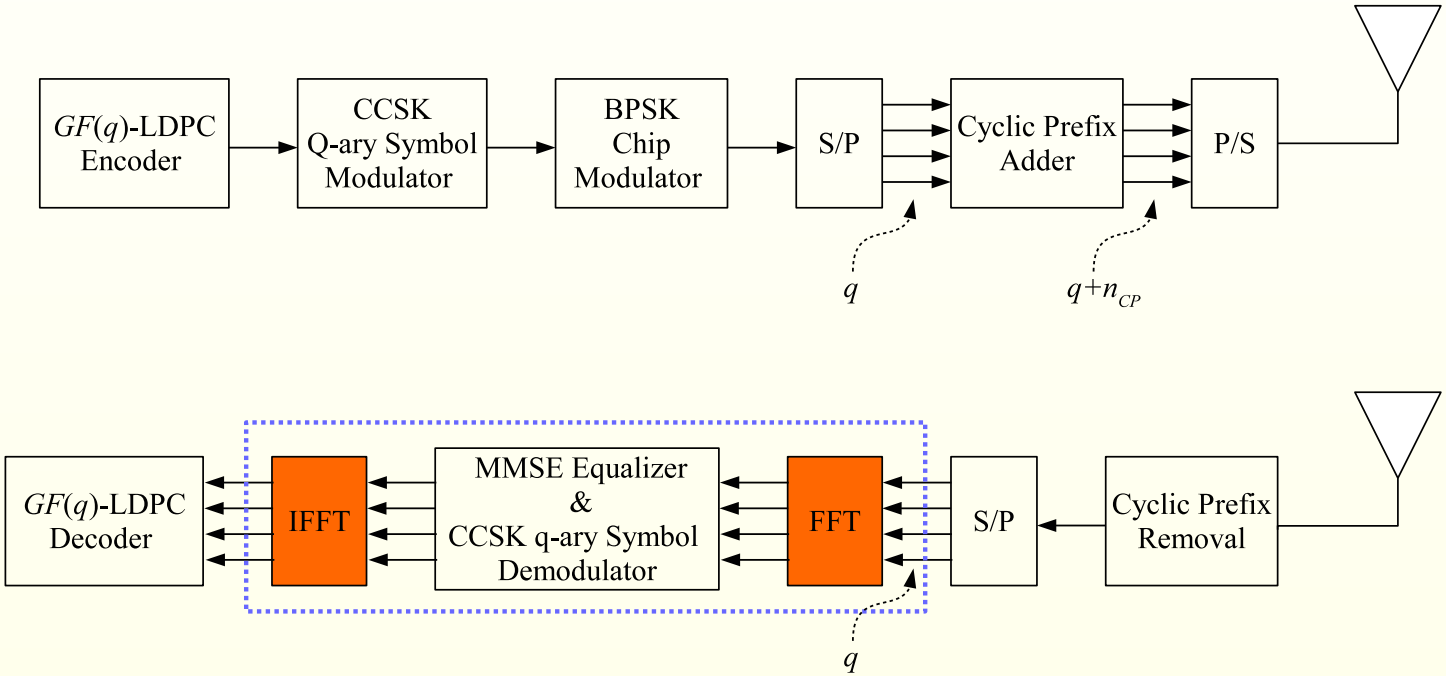
L'égalisation se traduit dans le domaine fréquentiel par :

$$\hat{s}^i = IFFT\left(FFT(r^i) \cdot \frac{1}{FFT(h) + \sigma^2}\right)$$

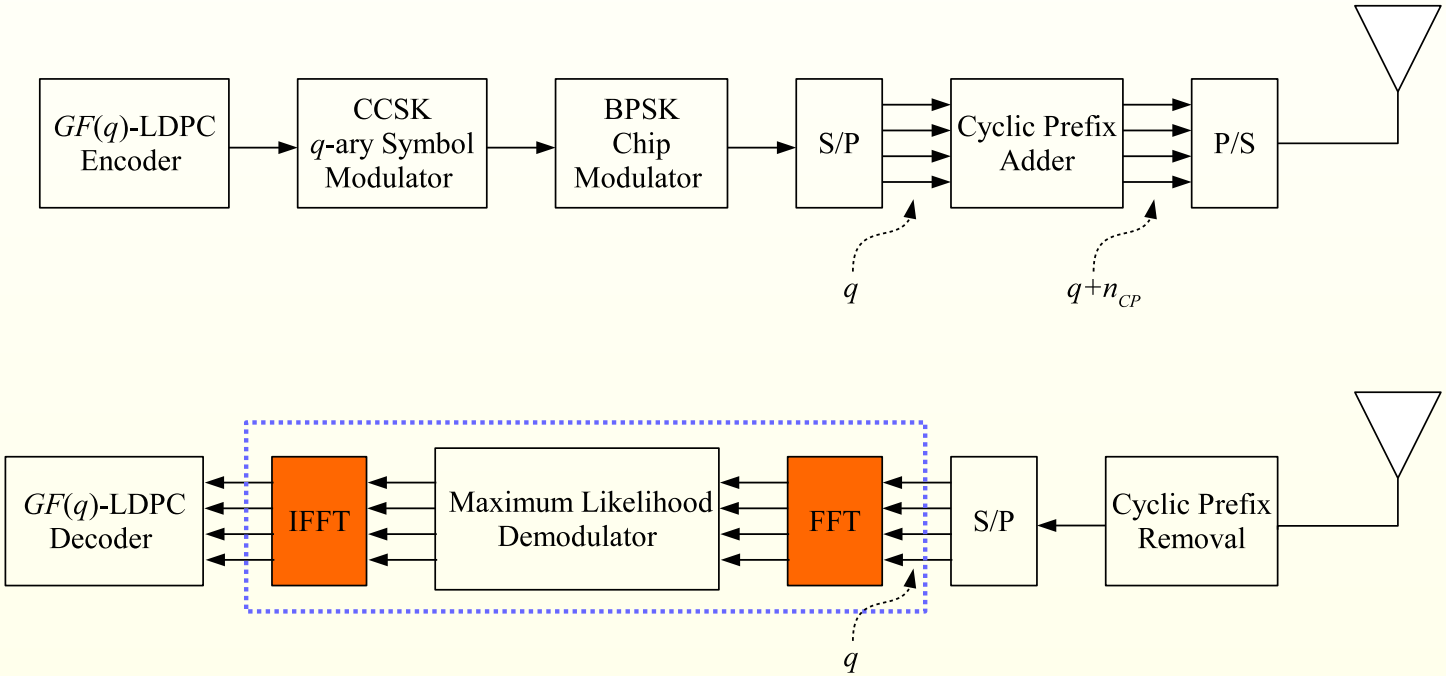
Système de transmission mono-porteuse avec préfixe cyclique et égalisation fréquentielle (SC-FDE)



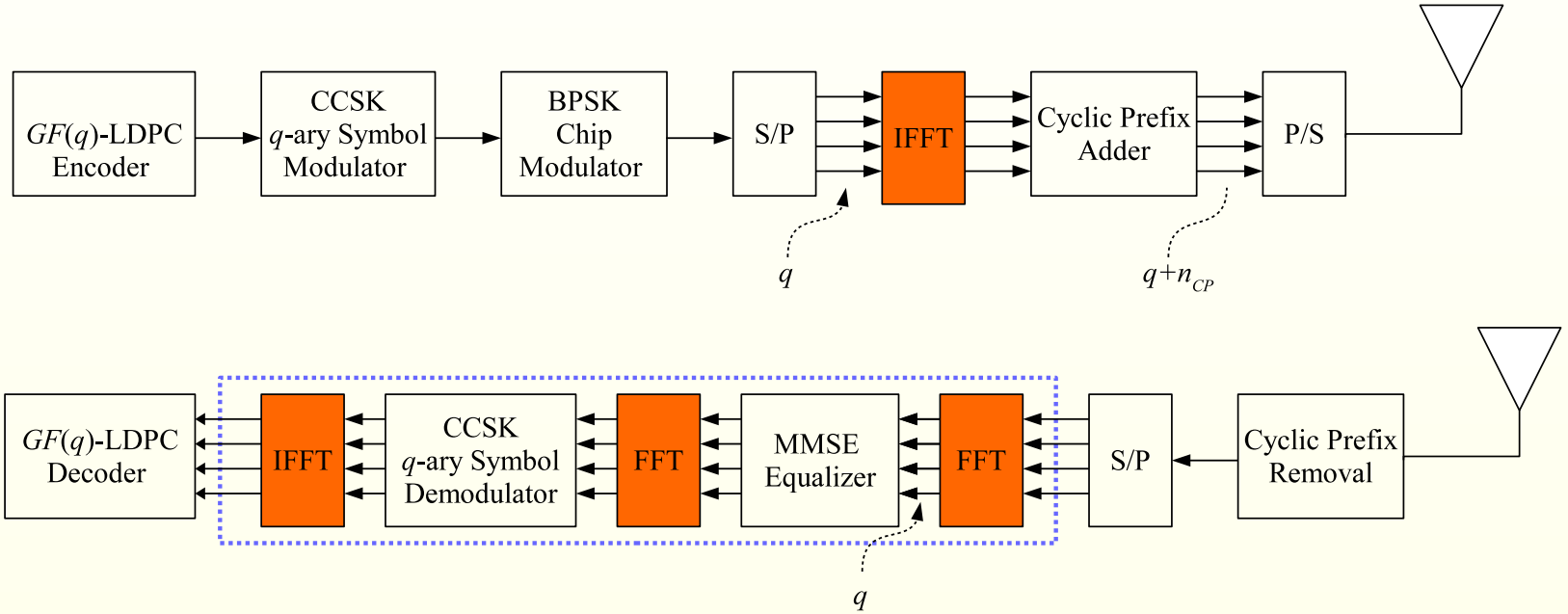
Système de transmission mono-porteuse avec préfixe cyclique et égalisation fréquentielle (SC-FDE)



Système de transmission mono-porteuse avec préfixe cyclique et détection à maximum de vraisemblance (SC-ML)



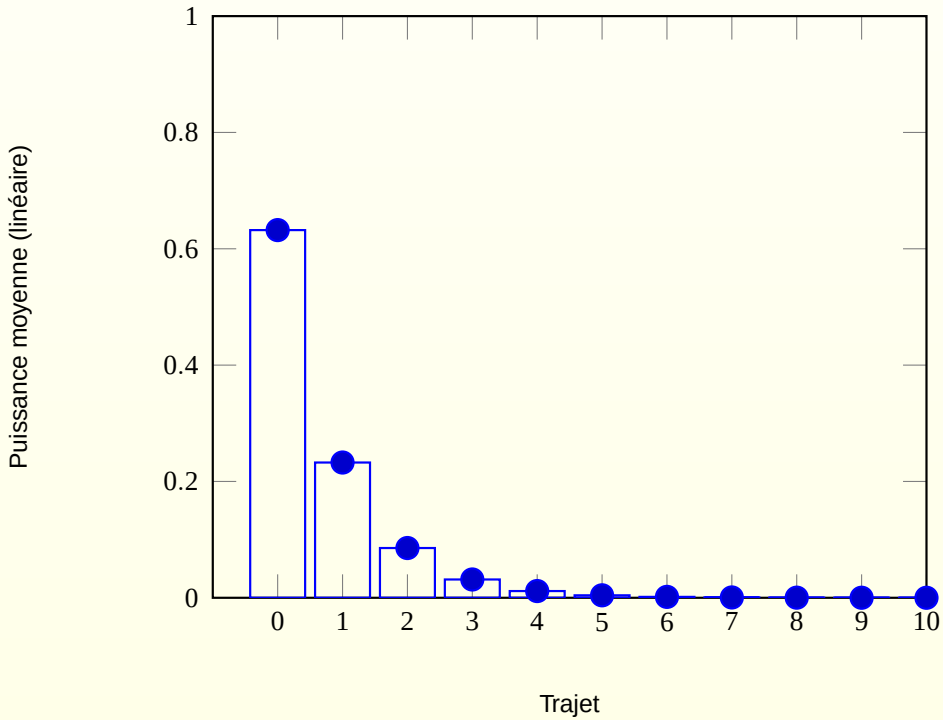
Système de transmission OFDM



Le modèle de canal utilisé dans les simulations

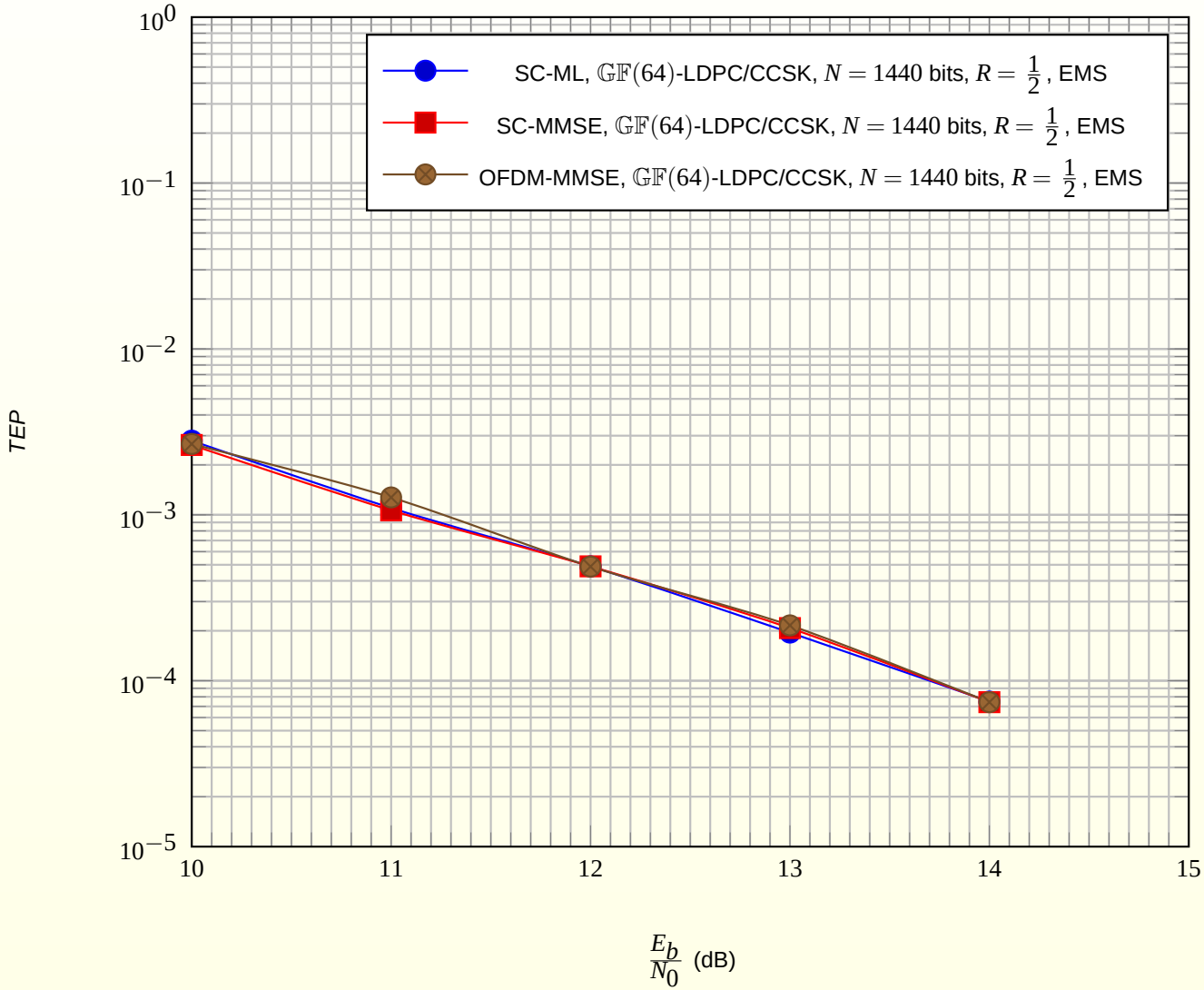
Les modèles de canaux standardisés pour le système HiperLAN/2

Channel model	r.m.s delay spread	Rice factor on first tap	Environment
A	50 ns	-	Office NLOS (no light of sight)
B	100 ns	-	Open space / Office NLOS
C	150 ns	-	Large open space NLOS
D	140 ns	10 dB	Large open space LOS
E	250 ns	-	Large open space NLOS

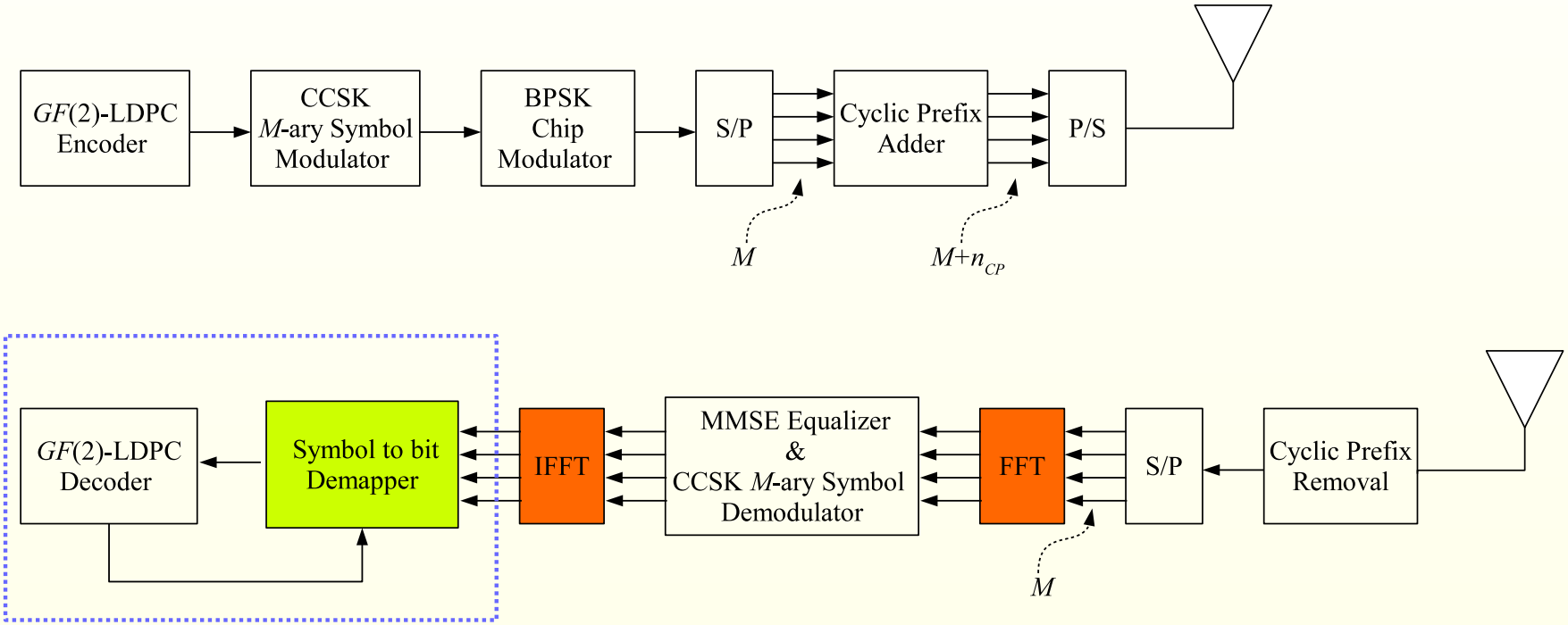


Profil de la puissance moyenne des retards de notre canal

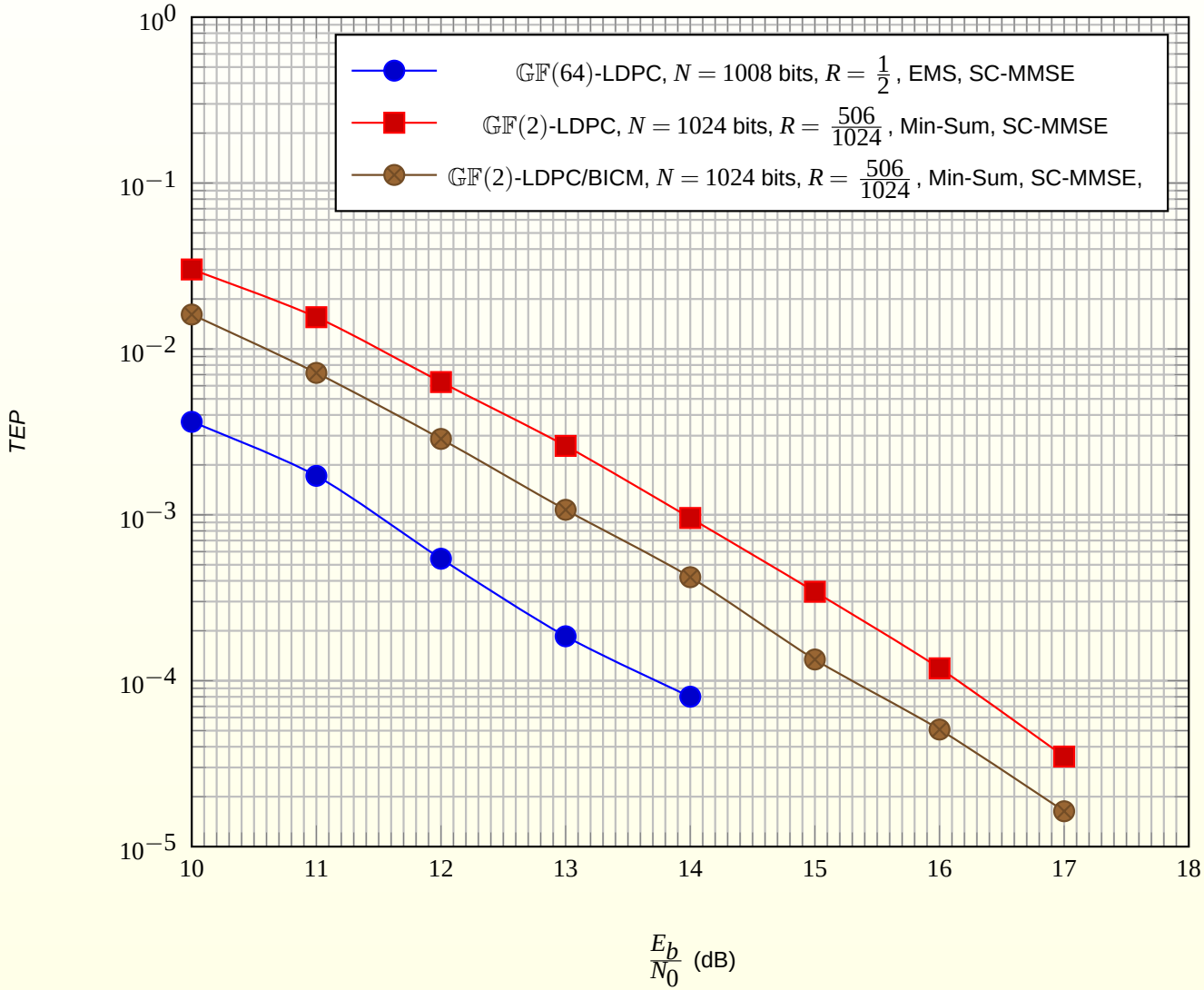
Performances de la concaténation d'un code LDPC non-binaire et d'une modulation CCSK dans les systèmes SC-MMSE, SC-ML et OFDM



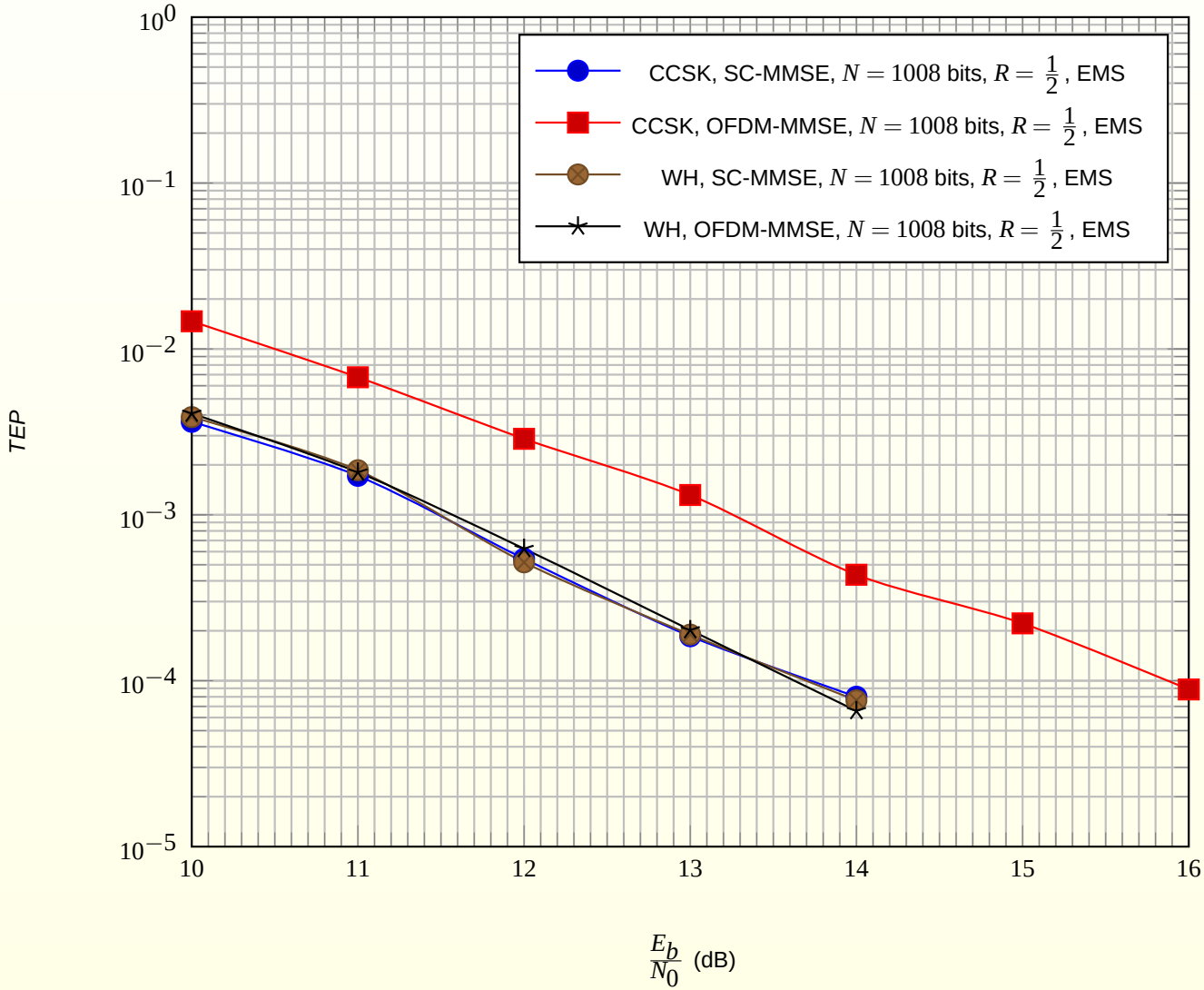
Concaténation d'un code LDPC binaire et d'une modulation CCSK dans un système SC-MMSE



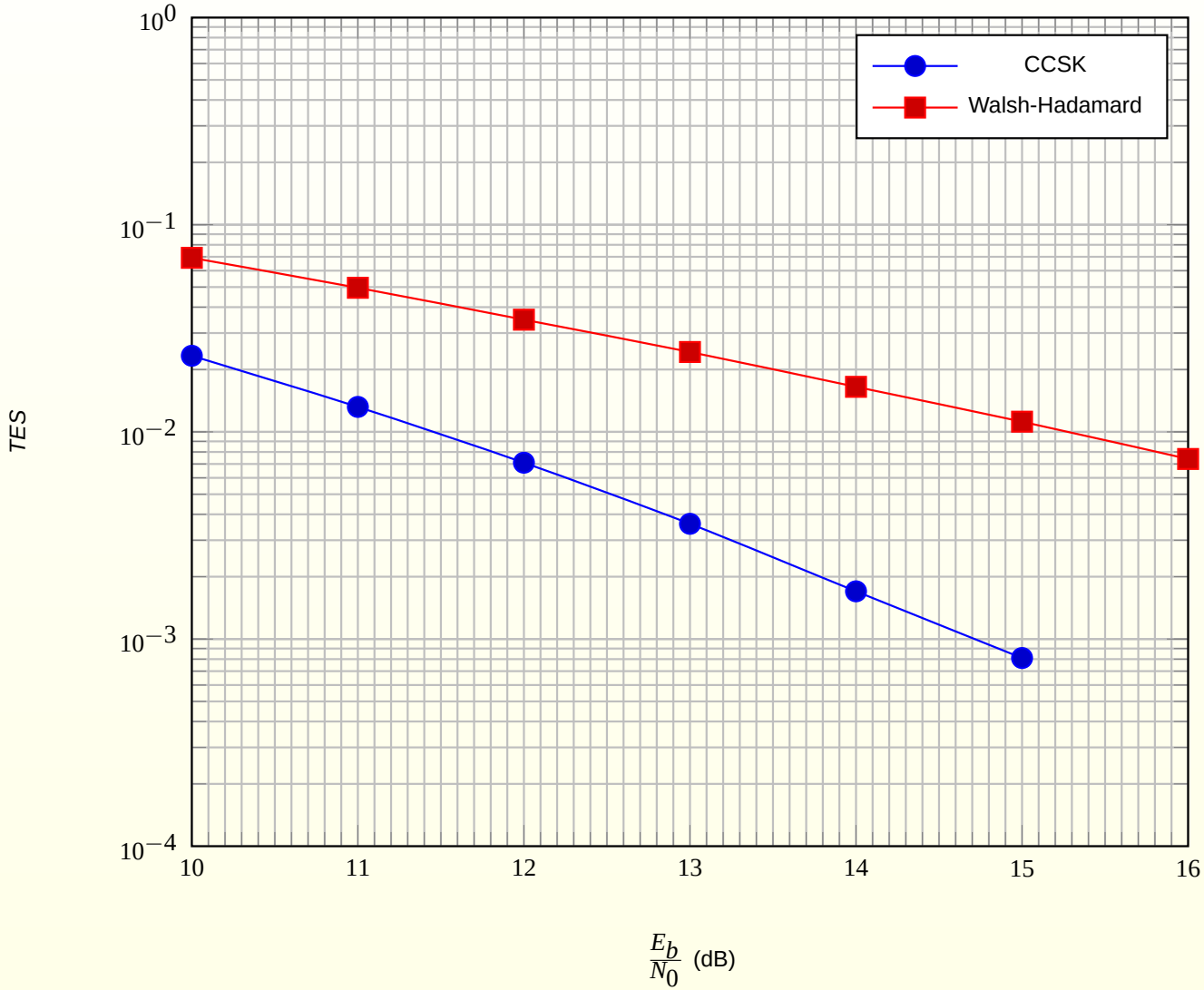
Performance de l'association d'un code LDPC binaire et d'une modulation CCSK dans un système SC-MMSE



Performance de la modulation de Walsh-Hadamard dans les systèmes SC-MMSE et OFDM



Performances des démodulations CCSK et Walsh-Hadamard à décision dure

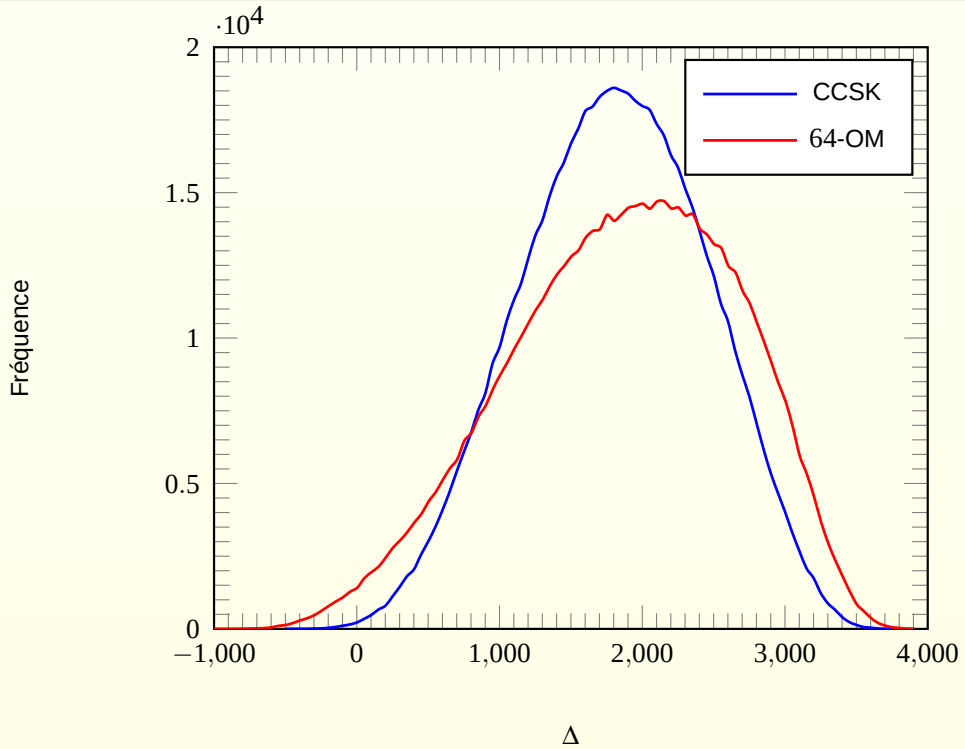


Dynamique des LLRs obtenus par démodulations CCSK et Walsh-Hadamard

Répéter les étapes suivantes n fois :

1. Générer uniformément un symbole $\beta \in \mathbb{GF}(64)$.
2. Transmettre la séquence CCSK associée à ce symbole à travers une réalisation aléatoire du canal.
3. Déterminer après égalisation MMSE la valeur : $\delta = LLR(\beta) - \max_{\beta' \in \mathbb{GF}(64)-\beta} LLR(\beta')$.

Refaire la même expérience avec la modulation de Walsh-Hadamard



Sommaire

Conclusions

Récepteur SC-FDE moins complexe que l'OFDM tout en ayant des performances comparable => éviter le problème de non-linéarité (PAPR) typique au au système OFDM.
Robustesse de l'association CCSK + NB-LDPC pour les liaisons militaires.

Perspective

ML versus MMSE
Robustesse de l'association CCSK + NB-LDPC pour les liaisons militaires.
Comparer CCSK et Walsh-Hadamard dans d'autre canaux sans fil.