

Architecture FPGA Générique pour décodeur LDPC

F. Guilloud – J.-L. Danger – E. Boutillon

GDR ISIS

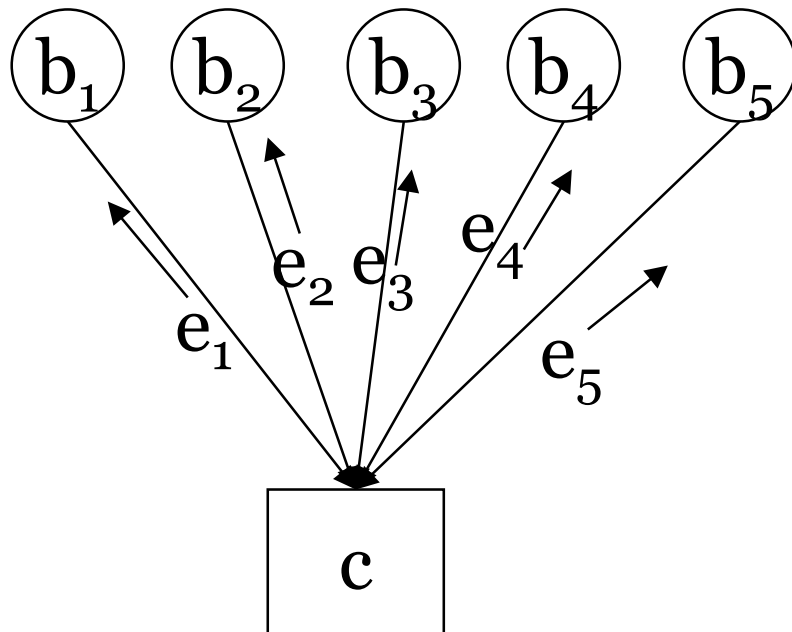
Journée LDPC du 19 décembre 2002

Telecom Paris

Plan de la Présentation

- Une méthode sous-optimale du calcul des valeurs extrinsèques.
- L'influence de la quantification (entière) sur le décodage.
- Architecture : introduction
- Présentation d'une architecture « générique ».

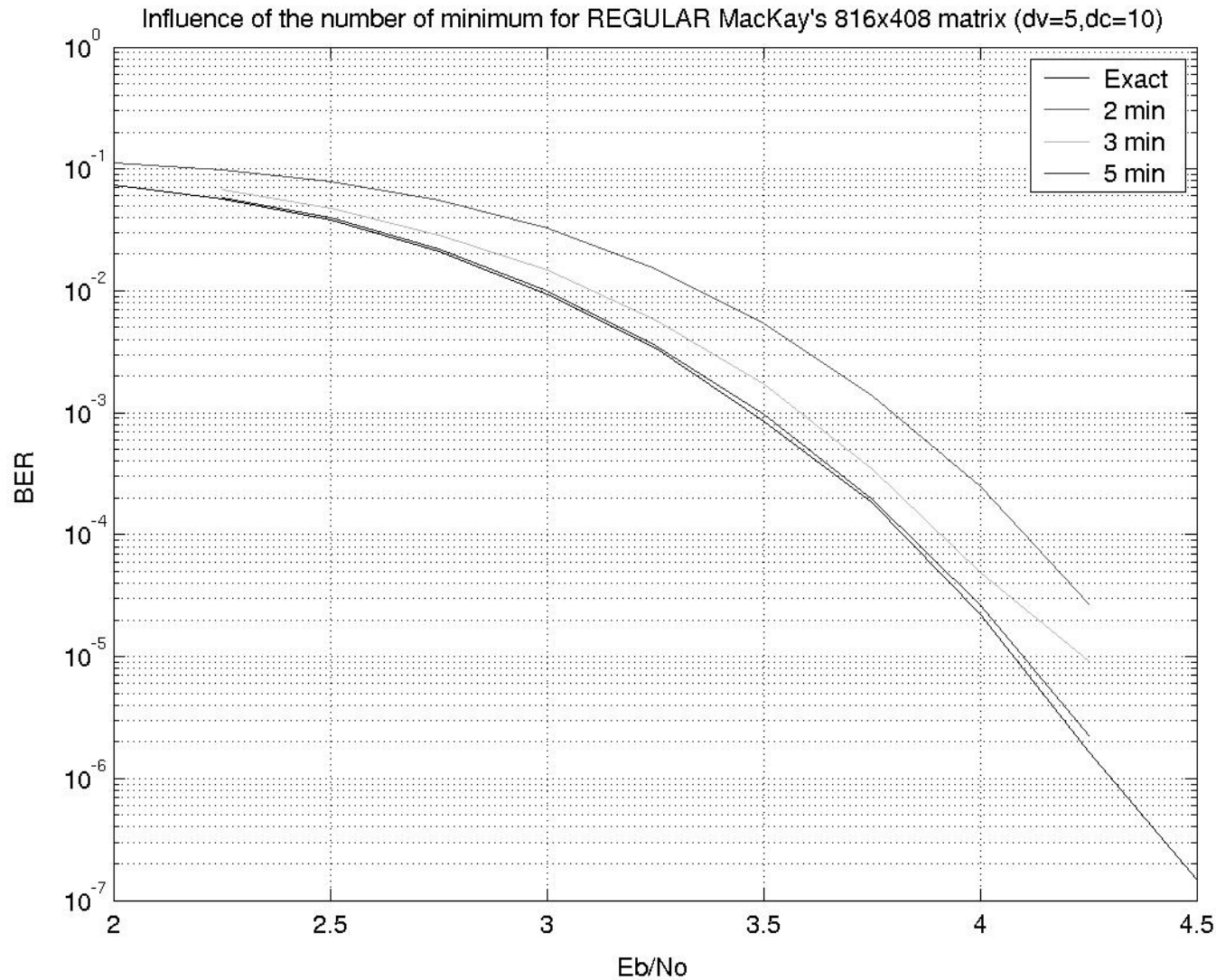
Calcul sous-optimal des LLR extrinsèques



- Calcul des e_i : $e_i = \bigoplus_{\substack{j \neq i \\ 1 < j < 5}} (b_j)$
- Observation :
les b_j de faible amplitude ont plus de poids.
- Idée :
On ne calcule les e_i qu'avec les N (ou $N-1$) b_i minimum.

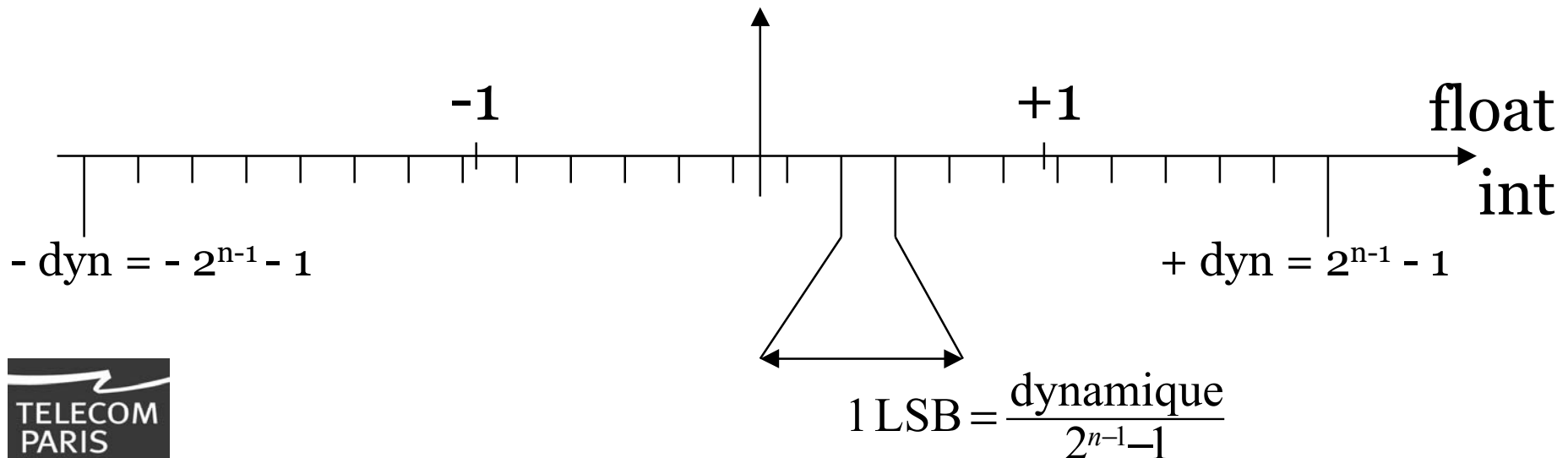
Exemple : $N = 3$: et $b_1 < b_2 < b_3 < \text{autres } b_i$: $e_i = \bigoplus_{\substack{j \neq i \\ 1 < j < 3}} (b_j)$

Simulations



Influence de la Quantification Entière

- 2 choix à faire :
 - ⇒ Nombre de bit : n
 - ⇒ Valeur d'un LSB, ou de la dynamique (i.e. la pleine échelle)



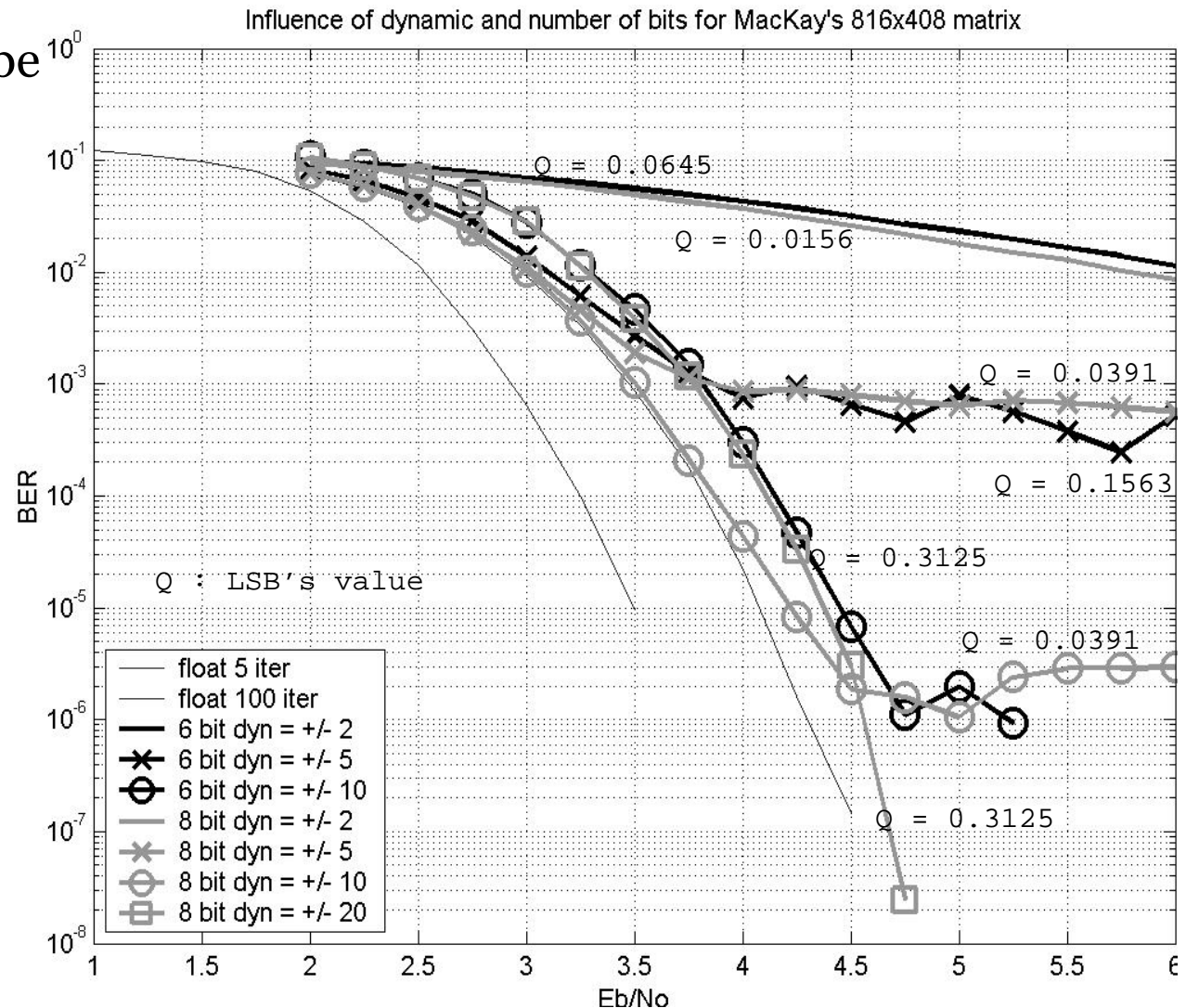
Simulations

- Éloignement de la courbe réelle : il diminue si la valeur du LSB diminue.
- Effet de « seuil » : il est d'autant plus bas que la dynamique est grande.

LLR intrinsèque = $2y/\sigma^2$

$\text{var}(y) = \sigma^2 \Rightarrow$

$\text{var}(\text{LLR intrinsèque}) = 4/\sigma^2$



Solutions

- On ne connaît pas le canal :

⇒ $\text{LLR}(\text{intrinsèque}) = \alpha \cdot y$ où $\alpha = \text{cste}$

⇒ $\text{var}(\text{LLR intrinsèque}) = \alpha^2 \sigma^2$

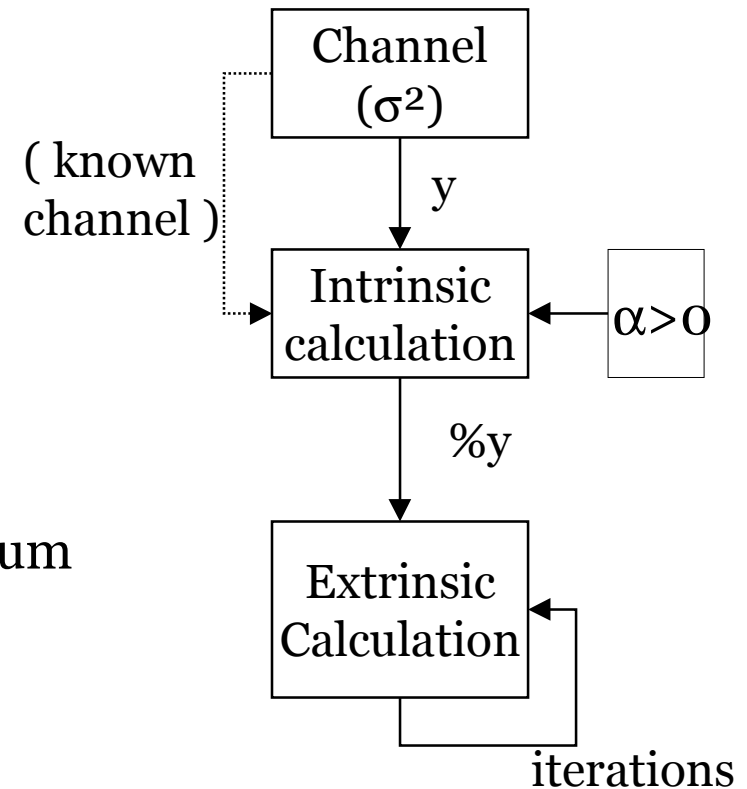
- On connaît le canal

⇒ Mettre une grande dynamique

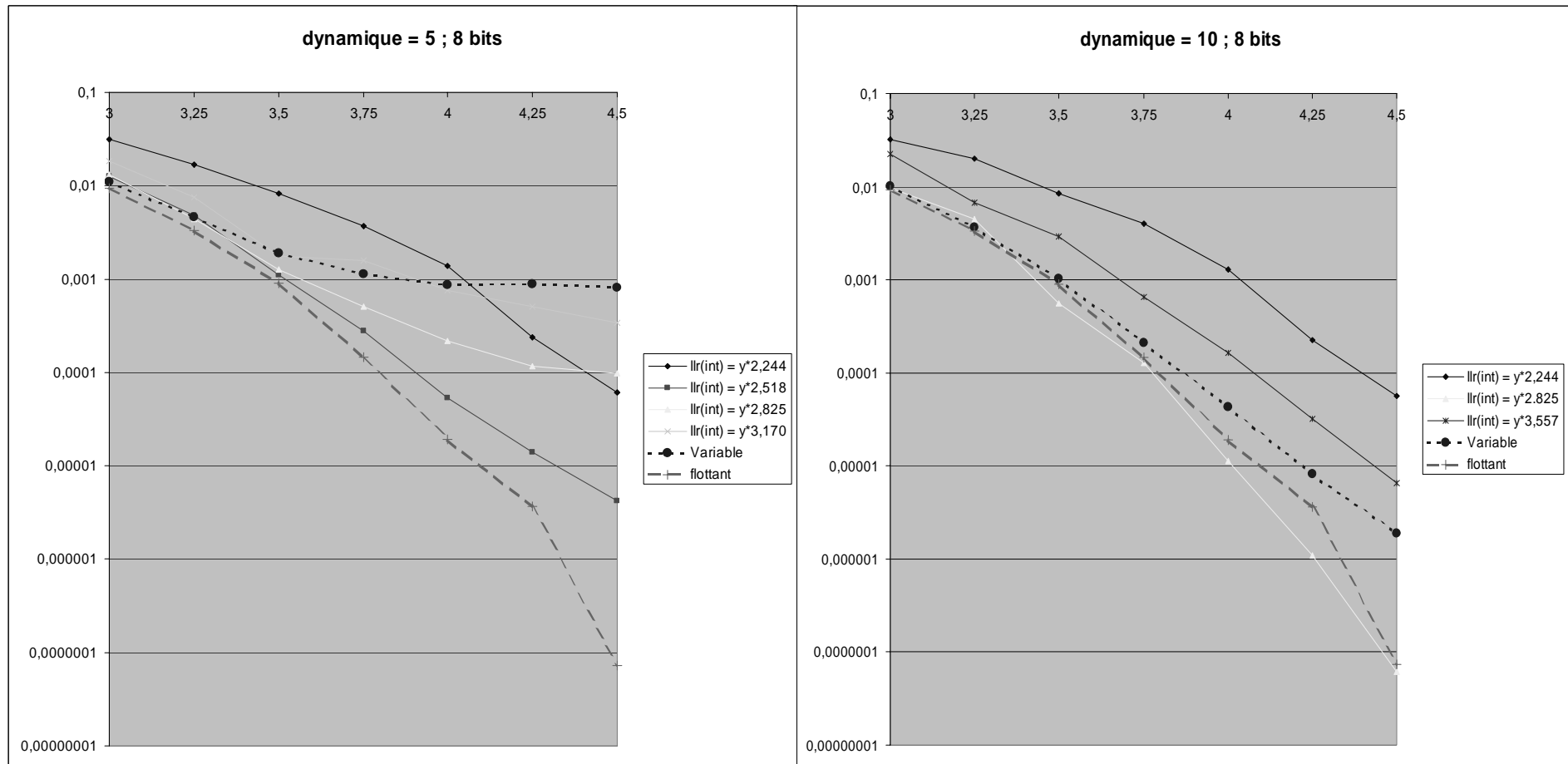
- Il faut alors un nombre de bit élevé pour ne pas trop diminuer le quantum
- $\text{LLR}(\text{intrinsèque}) = \alpha \cdot y$ où $\alpha = 2 / \sigma^2$
- ⇒ $\text{var}(\text{LLR intrinsèque}) = 4 / \sigma^2$

⇒ Rendre $\text{var}(\text{LLR intrinsèque})$ constante

- $\text{LLR}(\text{intrinsèque}) = \alpha \cdot y / \sigma$
- ⇒ $\text{var}(\text{LLR intrinsèque}) = \alpha^2$



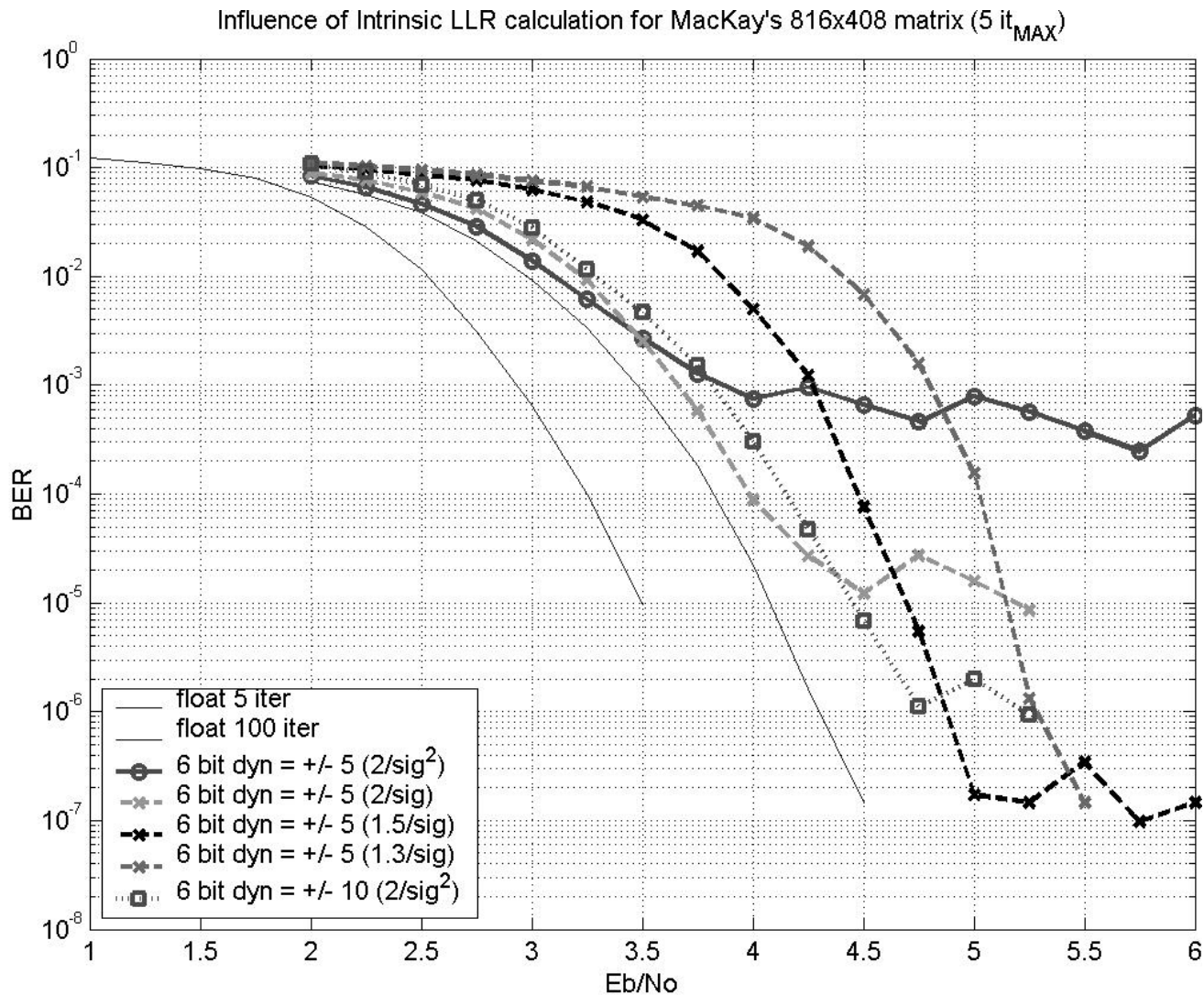
Si on ne connaît pas le canal (Simulations)



Constant : $llr(y) = \alpha.y$

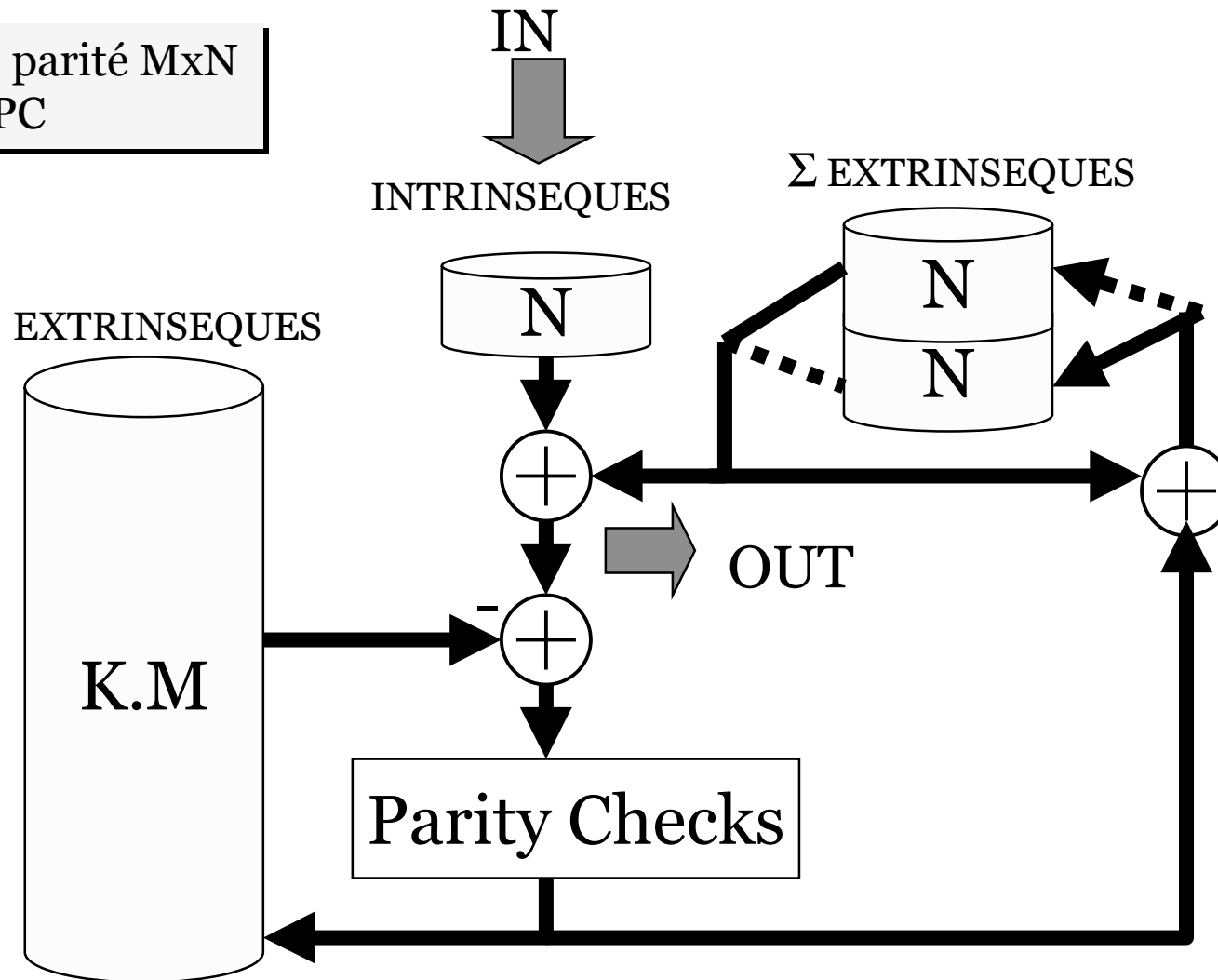
Variable : $llr(y) = 2y/\sigma^2$

Si on divise par σ (Simulations)

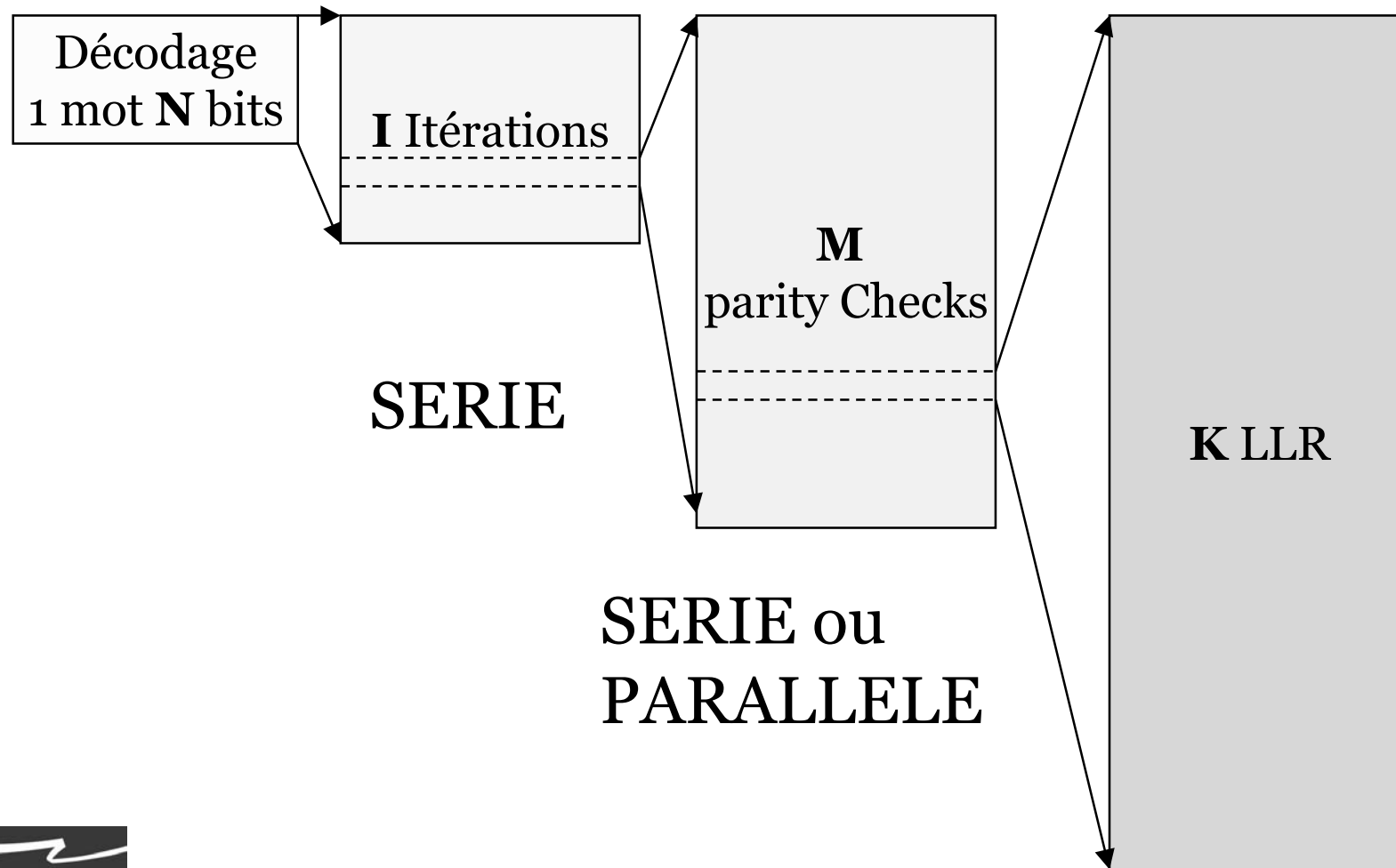


Architecture : Introduction

Matrice de parité $M \times N$
 $K = \text{poids PC}$



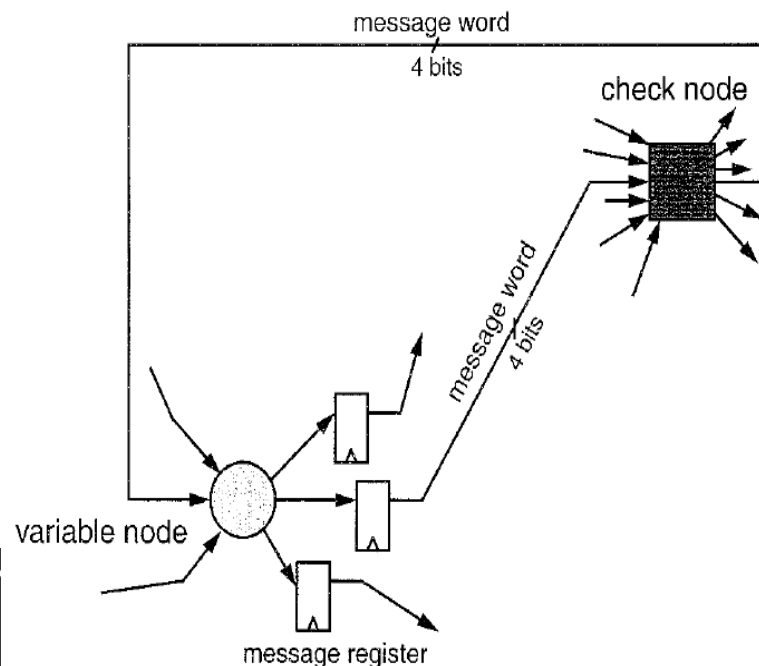
Graphe des calculs



Architecture parallèle

● Exemple

⇒ Blanksby, Howland, « A 690-mW 1-Gb/s 1024-b Rate-1/2 Low-Density Parity-Check Code Decoder » (IEEE Trans. on Solid-State Circuits, 2002.)



● Avantages

⇒ performances : 1Gb/s 64 itérations
 ⇒ consommation : 690mW
 ⇒ PER=2 10⁻⁴ @ 2,5 dB

● Inconvénients

⇒ Routage (2.k.M.B= 26624 equipotentiels, développement d'un outil de CAO ad hoc)
 ⇒ code figé
 ⇒ taille : 52.5mm² en techno 0.16μ

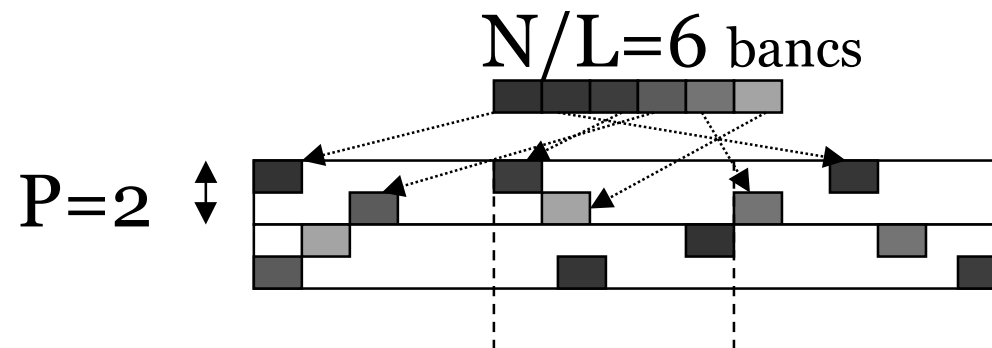
Architecture « Decoder-first »(2)

- Avantages :

- ⇒ facilité d'implémentation
- ⇒ programmabilité des permutations

- Inconvénients :

- ⇒ contraintes sur la matrice
- ⇒ poids moyen k des PC = $N/(L.P)$

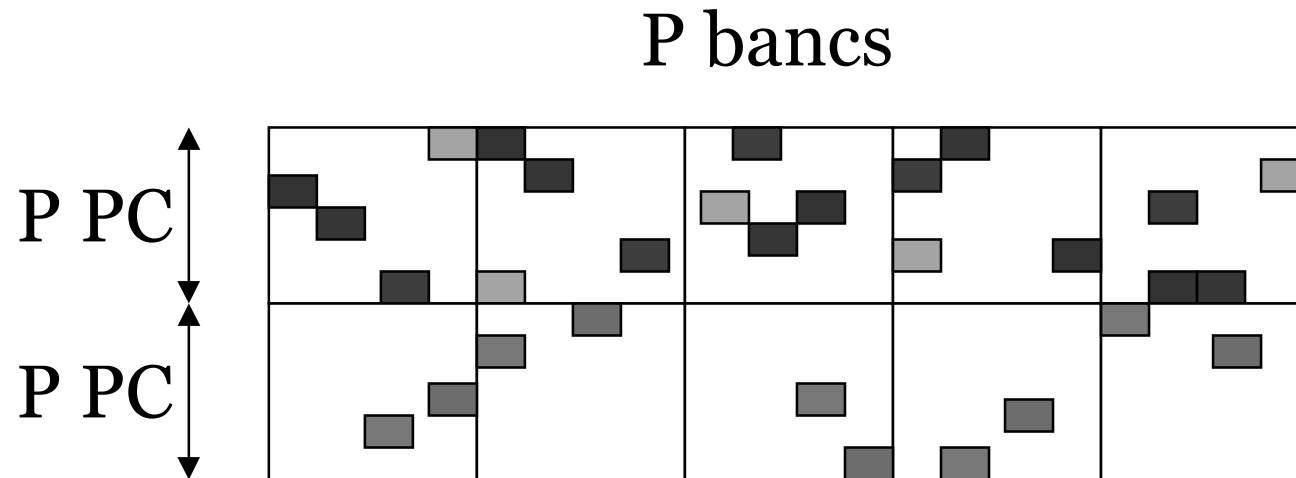


Architecture ENST

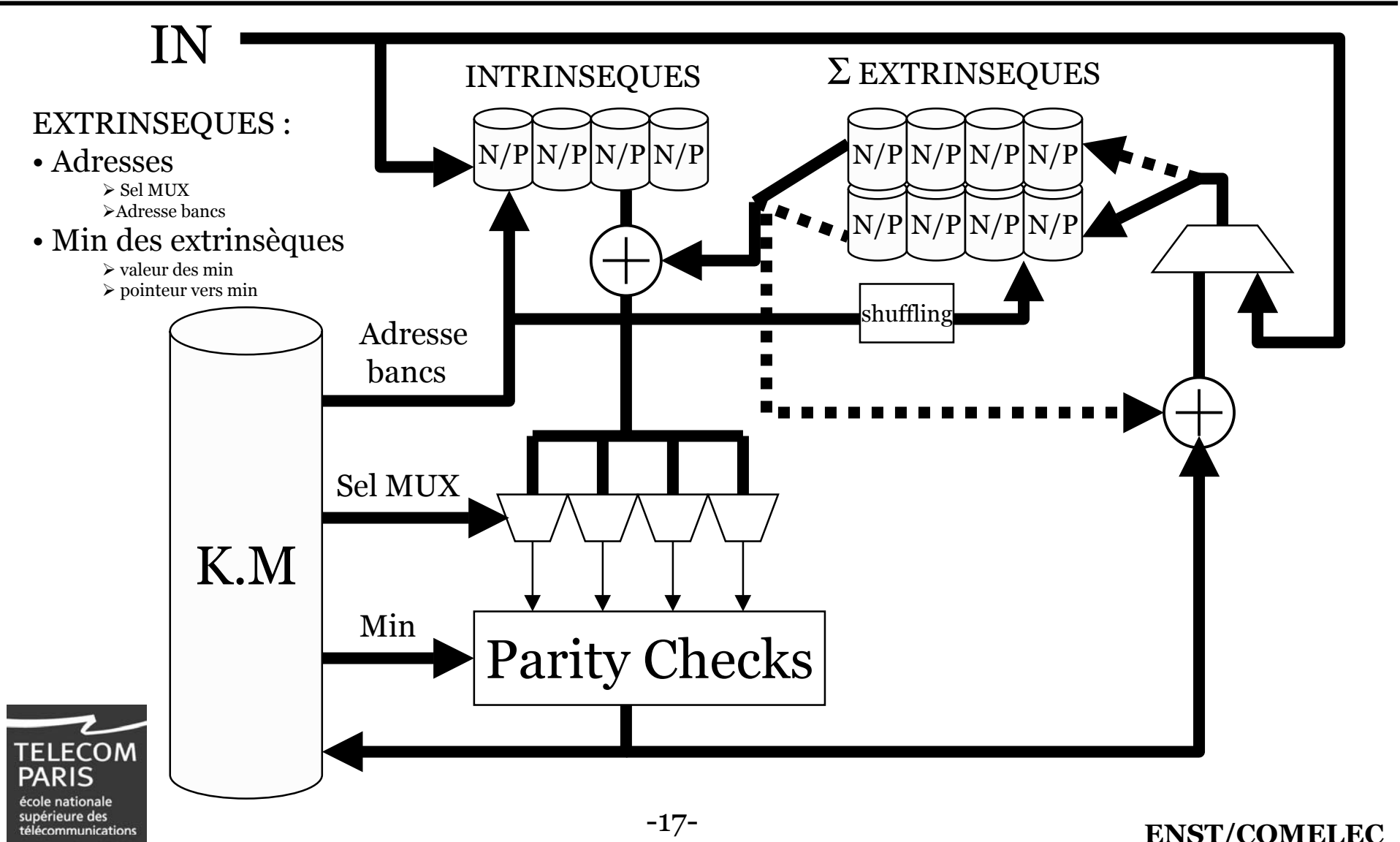
- Architecture « decoder-first » => semi parallèle
- Flexibilité accrue pour tester rapidement les codes sur FPGA :
 - ⇒ emploi de mémoires décrivant les Parity Checks.
 - ⇒ Degré k quelconque
 - Décodage moins contraint des matrices irrégulières
 - Calcul sériel des P parity checks => performances légèrement dégradées, en partie compensées avec P élevé
- Calcul des LLR avec une méthode sous optimale
 - ⇒ Utilisation de degrés élevés
 - ⇒ Réduction des calculs et de la taille de la mémoire des extrinsèques

Séquencement

$K=4$
Cycle 1
Cycle 2
Cycle 3
Cycle 4
 $K'=2$
Cycle 5
Cycle 6



Architecture



Performances Attendues

- Utilisation de la plateforme d'accélération FPGA
 - ⇒ 2*XILINX XCV1000E
 - 96 blocs mémoires 256x16 bits, 6144 CLBs
 - ⇒ Téléchargement et simulation rapides de nouveaux codes

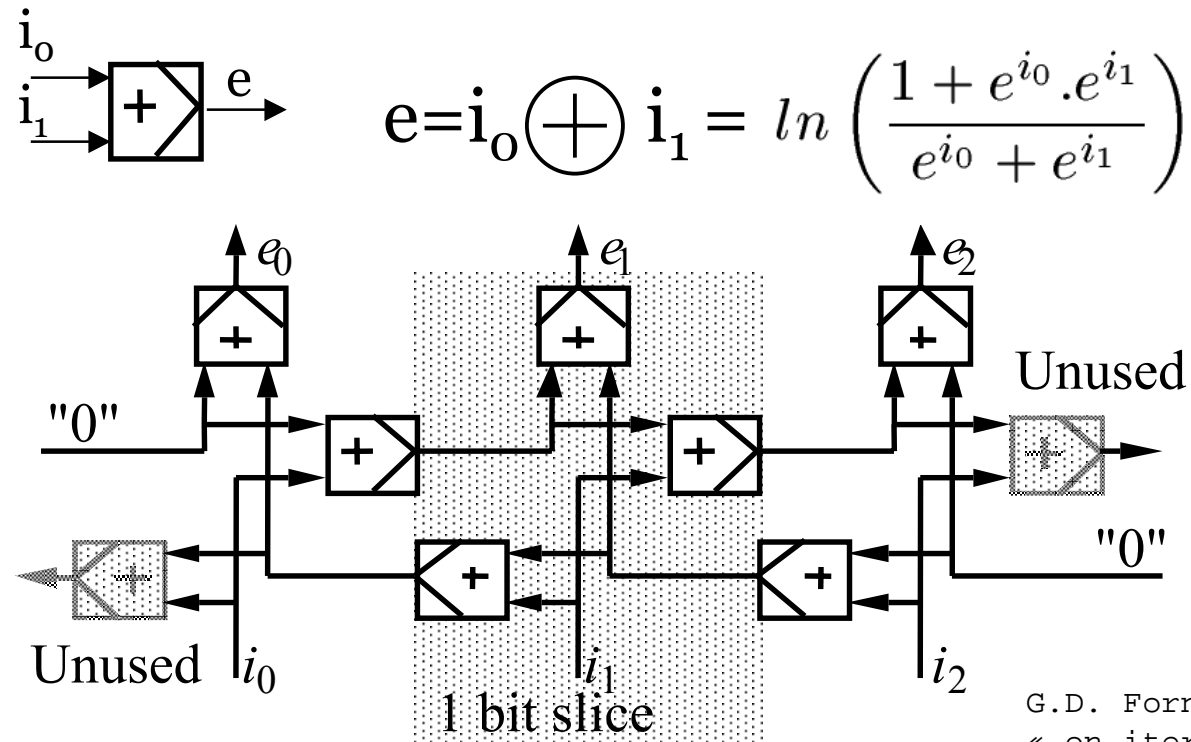
- Débit = $F_h \cdot P \cdot N / k \cdot M \cdot I$
 - ⇒ données sur 8 bits
 - ⇒ 10Mbits/s avec $F_h=50\text{MHz}, P=16, N=1024, M=512, k=8, I=20$

Perspectives

- Implémentation FPGA
- Étude des « error floor »
- Validation des résultats sur la quantification
- Recherche :
 - ⇒ Algorithme
 - ⇒ Architecture

MERCI ! ☺

Calcul des LLR



G.D. Forney
« on iterative decoding
and the two-way
algorithm »
Proc. of the int. Symp. On
Turbo-Codes, Brest 1997

La complexité croît en K

Electronic Notes in Theoretical Computer Science 74 (2002)

URL: <http://www.elsevier.nl/locate/entcs/volume74.html>

Performance of Low Density Parity Check Codes as a Function of Actual and Assumed Noise Levels. David J.C. MacKay and Christopher P. Hesketh