NB-LPDC Check node Architecture

Emmanuel Boutillon, Hassan Harb, Cédric Marchand, Ali Al Ghouwayel*, Laura Conde-Canencia

Lab-STICC Laboratory, Université de Bretagne Sud, Lorient, France
* Lebanese International University (LIU), Beirut, Lebanon

Outline

What is a NB-LDPC?

What are there advantages over binary codes?

Check Node processing (Extended Min Sum)

Pre-sorting of input vector

Conclusion

Outline

What is a NB-LDPC?

What are there advantages over binary codes?

Check Node processing (Extended Min Sum)

Pre-sorting of input vector

Conclusion

What is a NB-LDPC?

Is it a LDPC… except that parity check equations are done on a Galois Field GF(q) of cardinality $q>2$.

Let us remind what is a Galois Field.

Let P[X] be a irreducible polynomial of degree $\log_2(q)$.

The set of polynomial GF(q) = GF(2)[X] modulo P[X] has a Galois Field structure, i.e.:

addition: (GF(q),+)

multiplication (GF(q), x)

…and all associated nice properties

Operation in GF(4)

Example of addition: $\alpha^1 + \alpha^0 = (0, 1) + (1 , 0) = (1 , 1) = \alpha^2$

+	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

(GF(4),+) is a group.

Multiplication: $\alpha^i \times \alpha^j = \alpha^{i+j \bmod 3}$

x	0	α^0	α^1	α^2
0	0	0	0	0
α^0	0	α^0	α^1	α^2
α^1	0	α^1	α^2	α^0
α^2	0	α^2	α^0	α^1

(GF(4)*,x) is a group.

NB-LDPC

LDPC : Defined by a parity check matrix H where element are on GF(2) (either 0 or 1).

NB-LDPC : Defined by a parity check matrix H where coefficients $h_{i,j}$ are on GF(q) (either 0 or α^i).

Exemple on GF(4):
$$H = \begin{pmatrix} \alpha^0 & \alpha^2 & 0 & \alpha^2 & 0 & \alpha^1 \\ 0 & \alpha^1 & \alpha^2 & 0 & \alpha^0 & \alpha^1 \\ \alpha^2 & 0 & \alpha^0 & \alpha^1 & \alpha^0 & 0 \end{pmatrix}$$

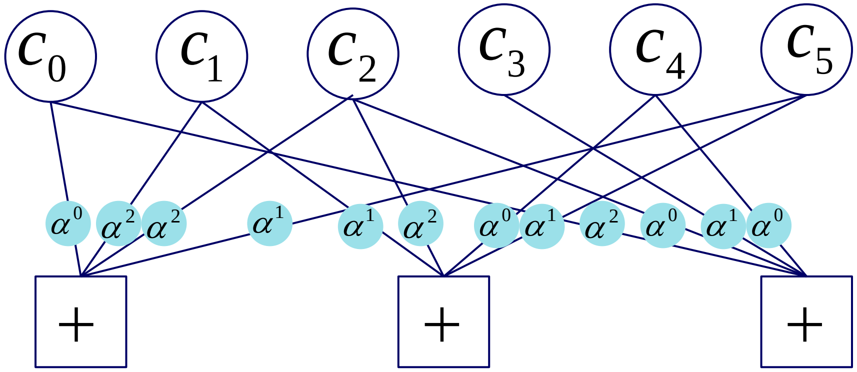
c a vector of 6 GF(4) symbols (i.e. 12 bits) belongs to the code iff H.**c** = 0

NB-LDPC

Exemple on GF(4):

$$H = \begin{pmatrix} \alpha^0 & \alpha^2 & 0 & \alpha^2 & 0 & \alpha^1 \\ 0 & \alpha^1 & \alpha^2 & 0 & \alpha^0 & \alpha^1 \\ \alpha^2 & 0 & \alpha^0 & \alpha^1 & \alpha^0 & 0 \end{pmatrix}$$

$$\begin{cases} \alpha^0 c_0 + \alpha^2 c_1 + \alpha^2 c_3 + \alpha^1 c_5 = 0 \\ \alpha^1 c_1 + \alpha^2 c_2 + \alpha^0 c_4 + \alpha^1 c_5 = 0 \\ \alpha^2 c_0 + \alpha^0 c_2 + \alpha^1 c_3 + \alpha^0 c_4 = 0 \end{cases}$$



NB-LDPC

A NB-LDPC of size (n,m) in GF(q) correspond to a binary code of size ($n \times \log_2(q)$, $m \times \log_2(q)$)

Good decoding performance for $d_v = 2$ (variable degree)

=> girth is high (higher than the binary counterpart)

=> Believe Propagation is close to the MAP decoding algorithm

High spectral efficiency:
Coded modulation outperform BICM modulation (information theory argument).

Not a free lunch: high decoding complexity

Outline

What is a NB-LDPC?

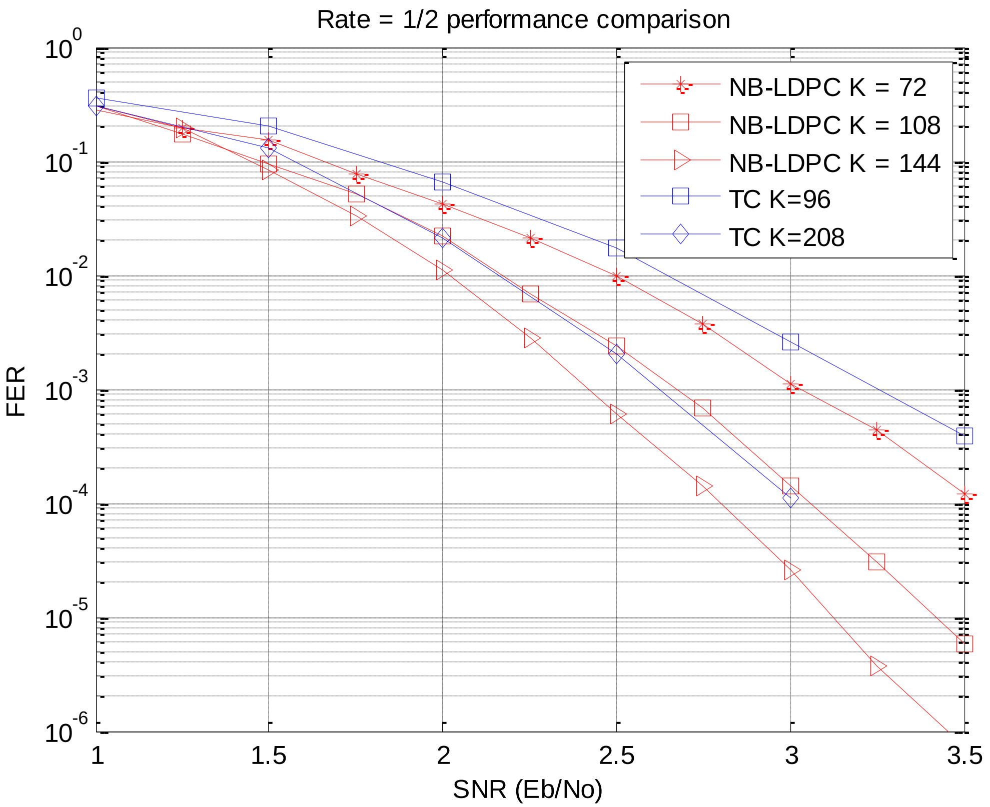
What are there advantages over binary codes?

Check Node processing (Extended Min Sum)

Pre-sorting of input vector

Conclusion

Very good performance for small code length.



Comparison between rate 1/2

Turbo-Code proposed for 5G [1]

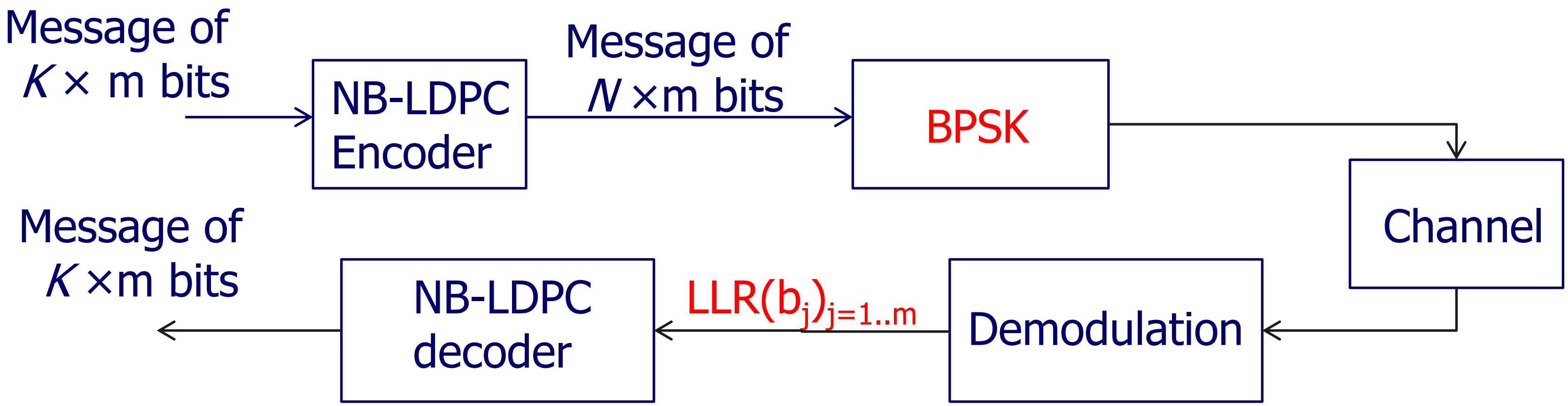
and

QC-NB-LDPC over GF(64) [2]

[1] http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_87/Docs/R1-1612938.zip

[2] UBS's construction.

LLR computation for BPSK



Message of $K \times$ m bits

NB-LDPC Encoder

Message of $N \times$m bits

BPSK

Channel

Demodulation

$\text{LLR}(b_j)_{j=1..m}$

NB-LDPC decoder

Message of $K \times$m bits

$$LLR(s=\alpha^i) = \sum_{j=0}^{m-1} ((\alpha^i(j)) \oplus HD(b_j)) \times \left| LLR(b_j) \right|$$

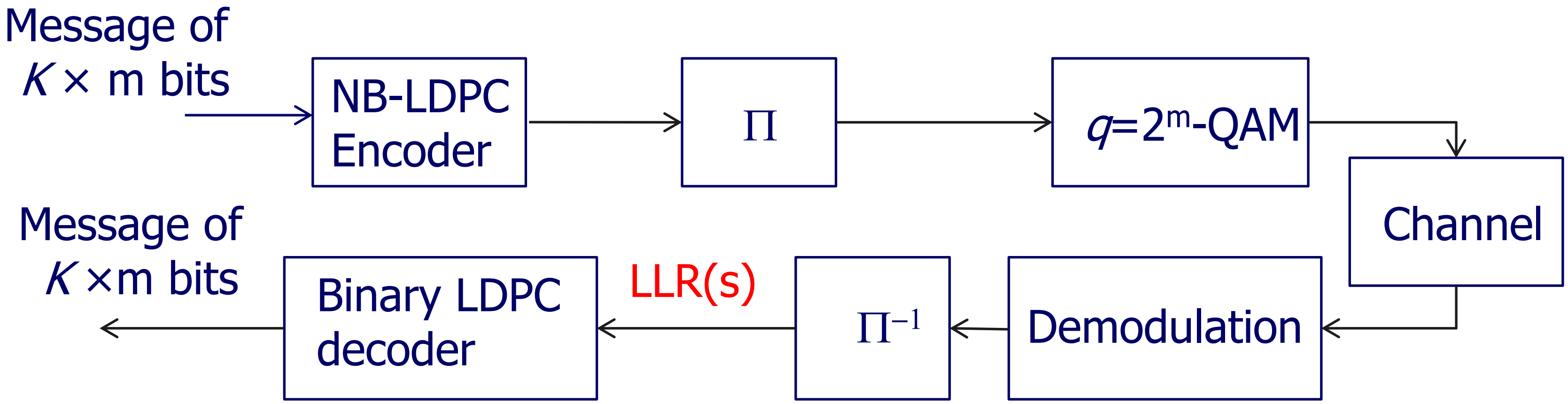
Example:

$\text{LLR}(b_1)$=-2 ; $\text{LLR}(b_0)$ = 4 => Hard Decision : (1,0) => α^1

LLR(s=0) = 2
$\text{LLR}(s=\alpha^0)$ = 2 + 4 = 6
$\text{LLR}(s=\alpha^1)$ = 0
$\text{LLR}(s=\alpha^2)$ = 4

0 = (0,0)
α^0 = (0,1)
α^1 = (1,0)
α^2 = (1,1)

LLR computation for $q=2^m$ QAM using Bit-Interleaved Coded Modulation (BICM)



Message of $K \times$ m bits

NB-LDPC Encoder

Π

$q=2^m$-QAM

Channel

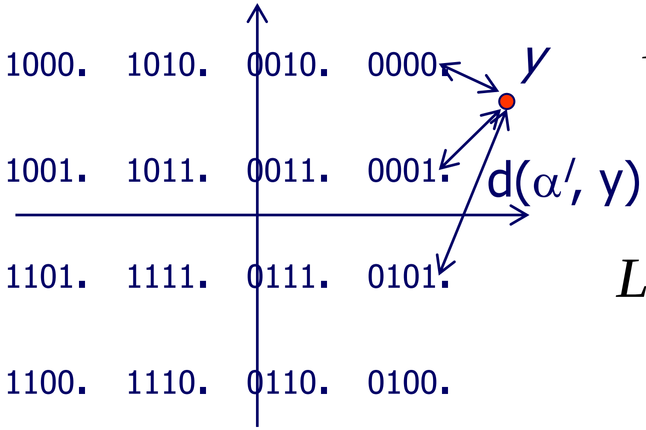
Demodulation

Π^{-1}

LLR(s)

Binary LDPC decoder

Message of $K \times$m bits

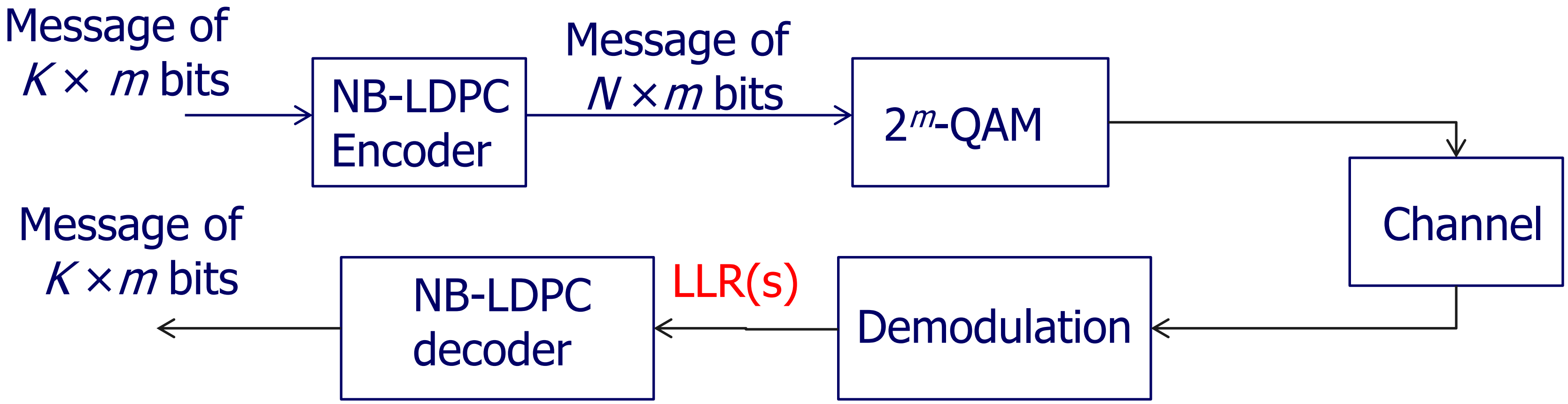


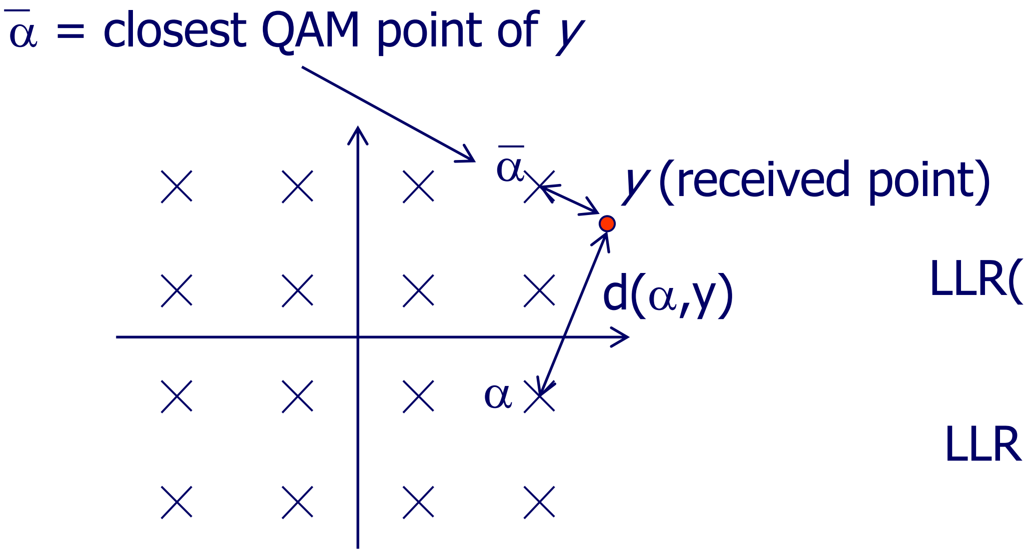
1000.	1010.	0010.	0000.
1001.	1011.	0011.	0001.
1101.	1111.	0111.	0101.
1100.	1110.	0110.	0100.

$$LLR(b_j) \approx \frac{2}{\sigma^2}\left(\underset{x_0^j}{Min}\left(d(x_0^j, y)^2\right) - \underset{x_1^j}{Min}\left(d(x_1^j, y)^2\right)\right)$$

$$LLR(s=\alpha^i) = \sum_{j=0}^{m-1}((\alpha^i(j)) \oplus HD(b_j)) \times \left|LLR(b_j)\right|$$

LLR computation for $q=2^m$ QAM using Coded Modulation

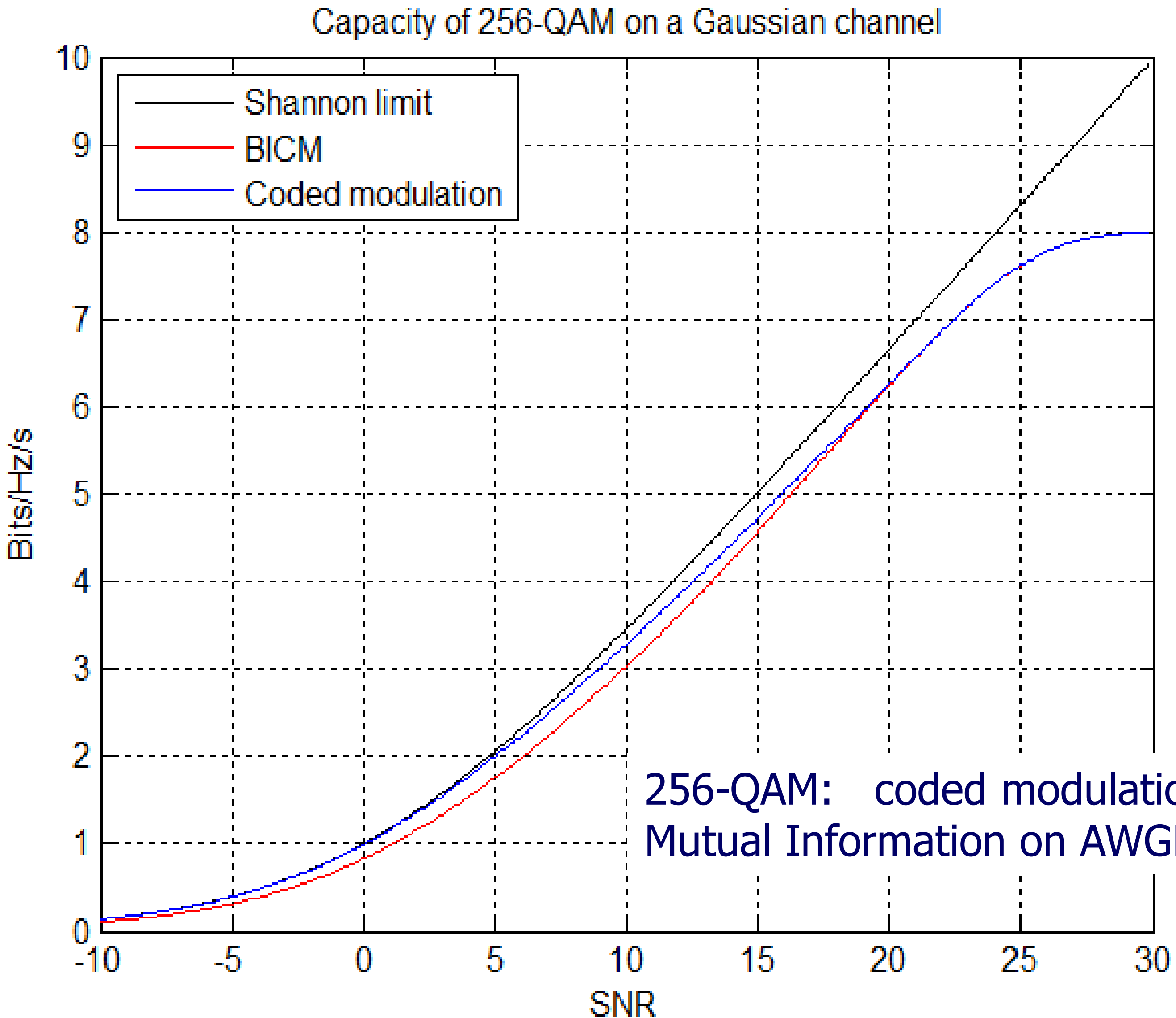




$$\mathrm{LLR}(\alpha) = 2/\sigma^2(d(y,\alpha)^2 - d(y,\bar{\alpha})^2)$$

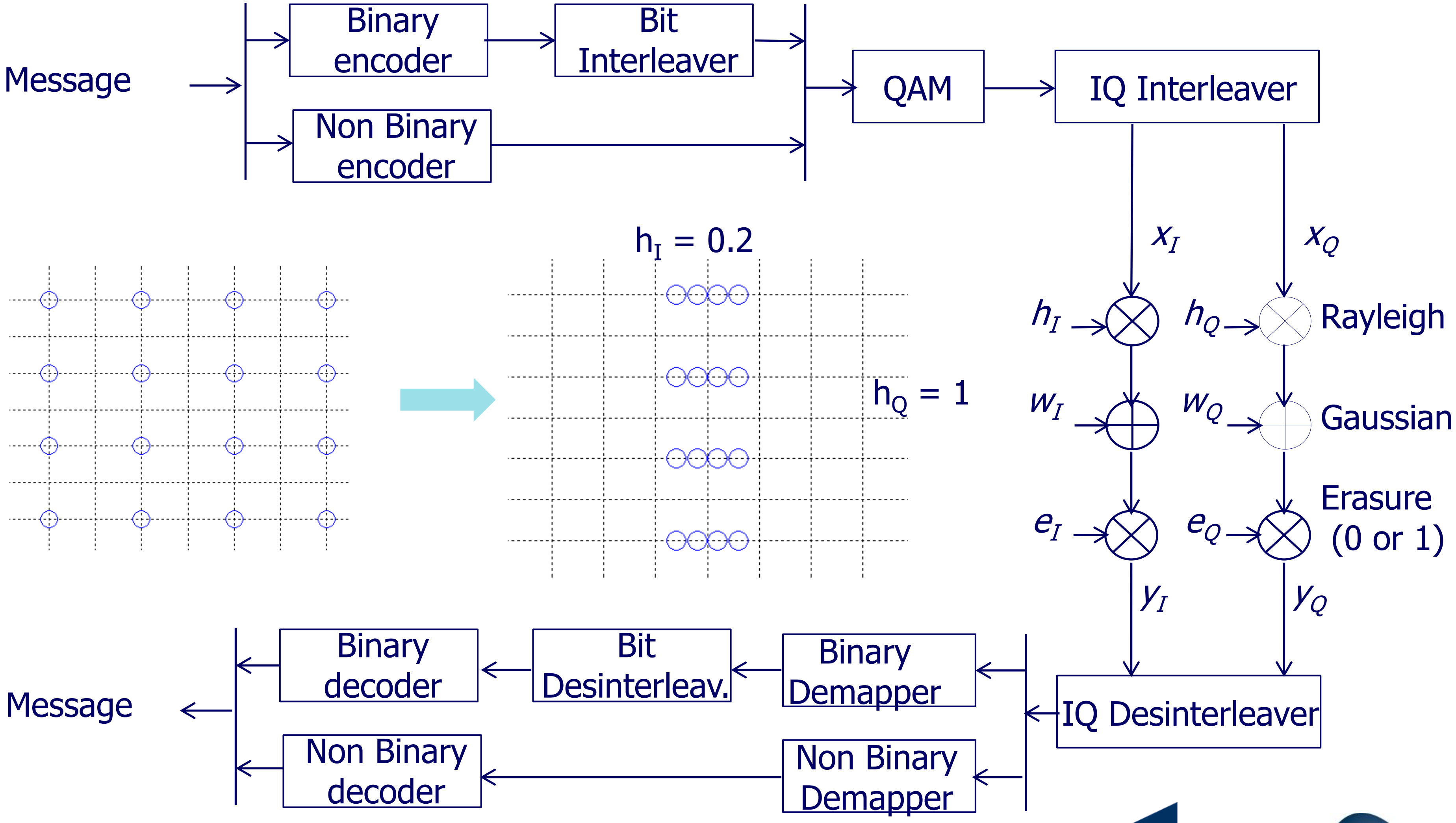
$$\mathrm{LLR}(\alpha) \geq 0 \text{ and } \mathrm{LLR}(\bar{\alpha}) = 0$$

Mutual information in AWGN Channel



256-QAM: coded modulation vs. BICM
Mutual Information on AWGN

Signal Space Diversity in Rayleigh Channel



Message

Binary encoder

Bit Interleaver

Non Binary encoder

QAM

IQ Interleaver

x_I

x_Q

$h_I = 0.2$

$h_Q = 1$

h_I

h_Q

Rayleigh

w_I

w_Q

Gaussian

e_I

e_Q

Erasure (0 or 1)

y_I

y_Q

IQ Desinterleaver

Binary Demapper

Non Binary Demapper

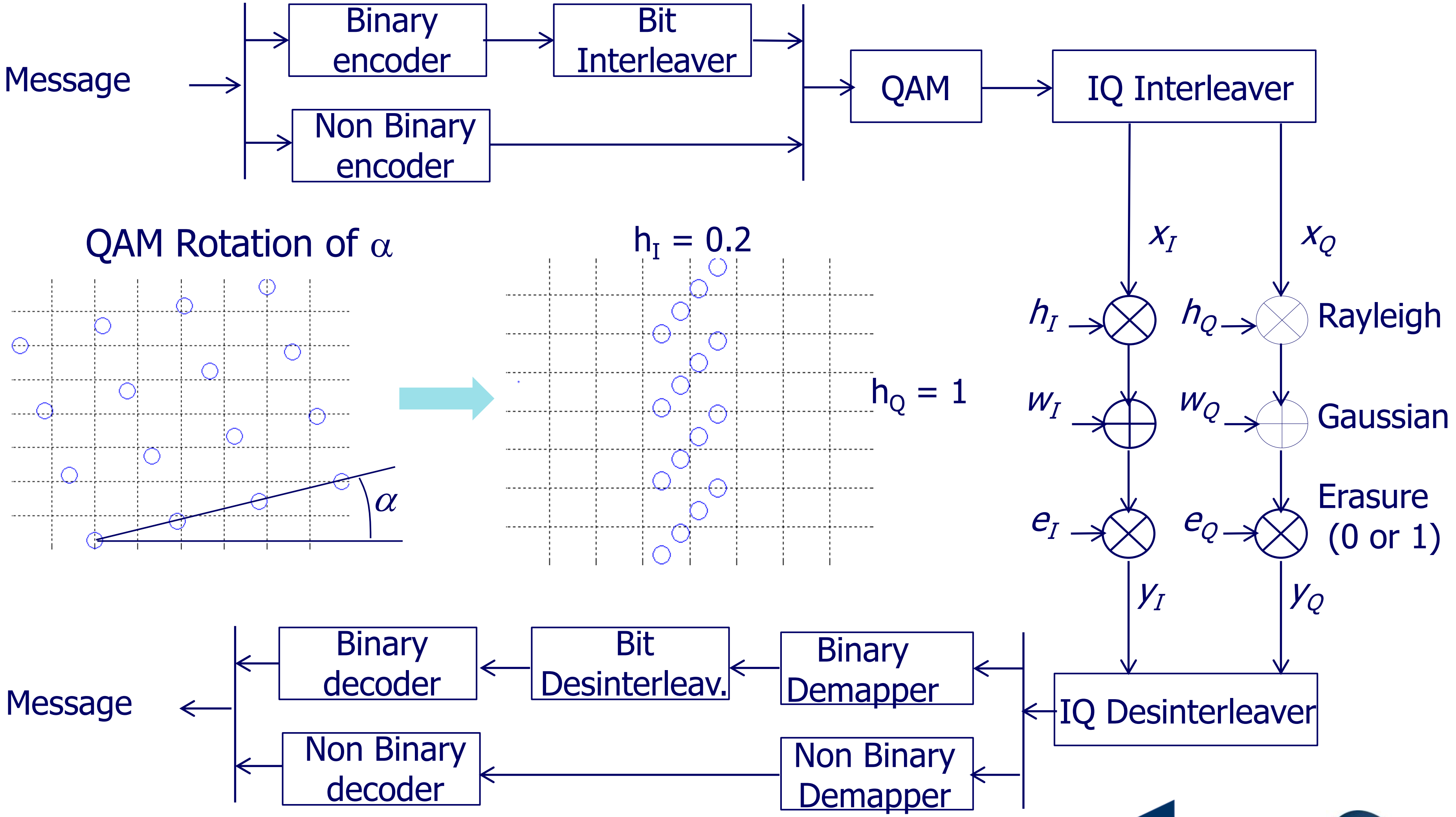
Bit Desinterleav.

Binary decoder

Non Binary decoder

Message

Signal Space Diversity in Rayleigh Channel



Message

Binary encoder

Bit Interleaver

Non Binary encoder

QAM

IQ Interleaver

QAM Rotation of α

α

$h_I = 0.2$

$h_Q = 1$

x_I

x_Q

h_I

h_Q

Rayleigh

w_I

w_Q

Gaussian

e_I

e_Q

Erasure (0 or 1)

y_I

y_Q

IQ Desinterleaver

Binary Demapper

Non Binary Demapper

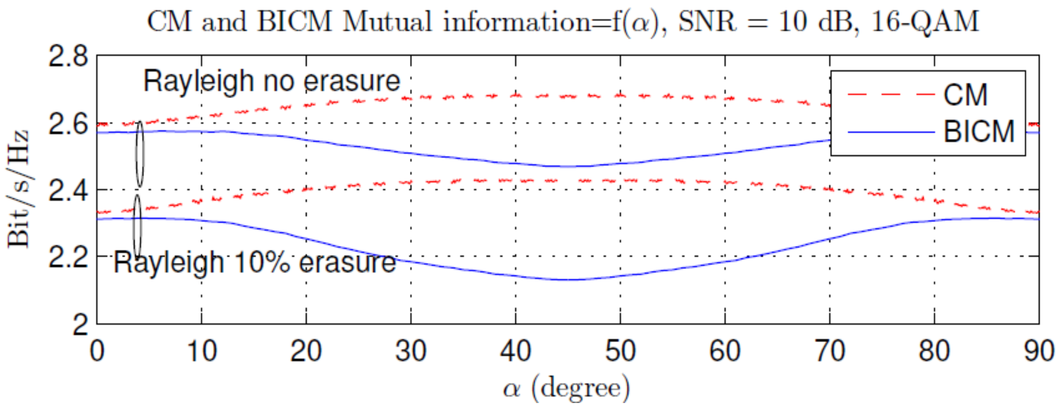
Bit Desinterleav.

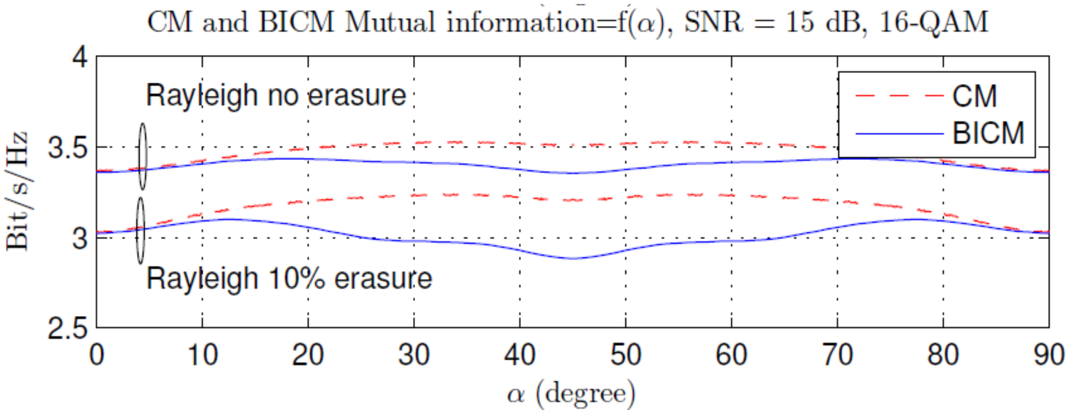
Binary decoder

Non Binary decoder

Message

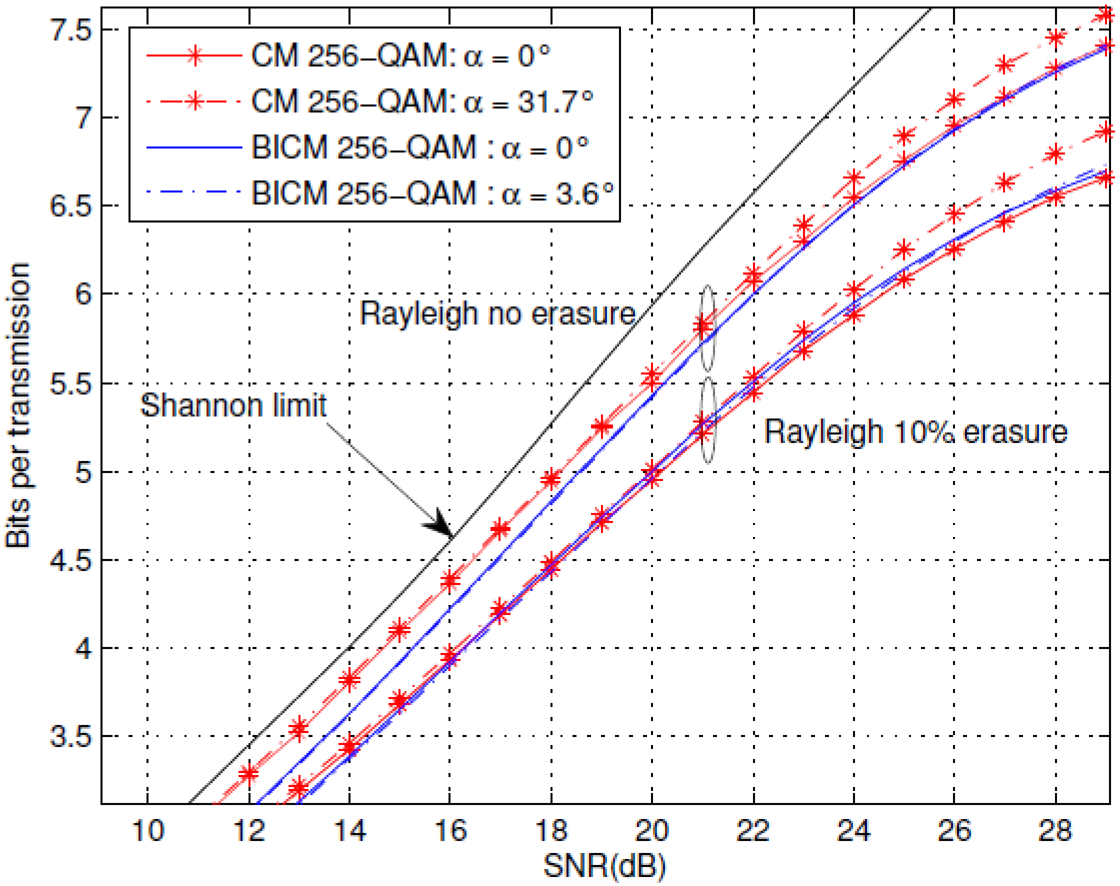
Mutual Information as a function of α for 16-QAM modulation





- CM presents for all rotation angles higher MI than BICM scheme.
- Rotation always provides gain for CM unlike for BICM.
- For Rayleigh channel with erasure SSD technique provide more significant gain

Mutual Information as a function of α for 256-QAM modulation



CM with SSD outperforms the BICM with SSD scheme for:

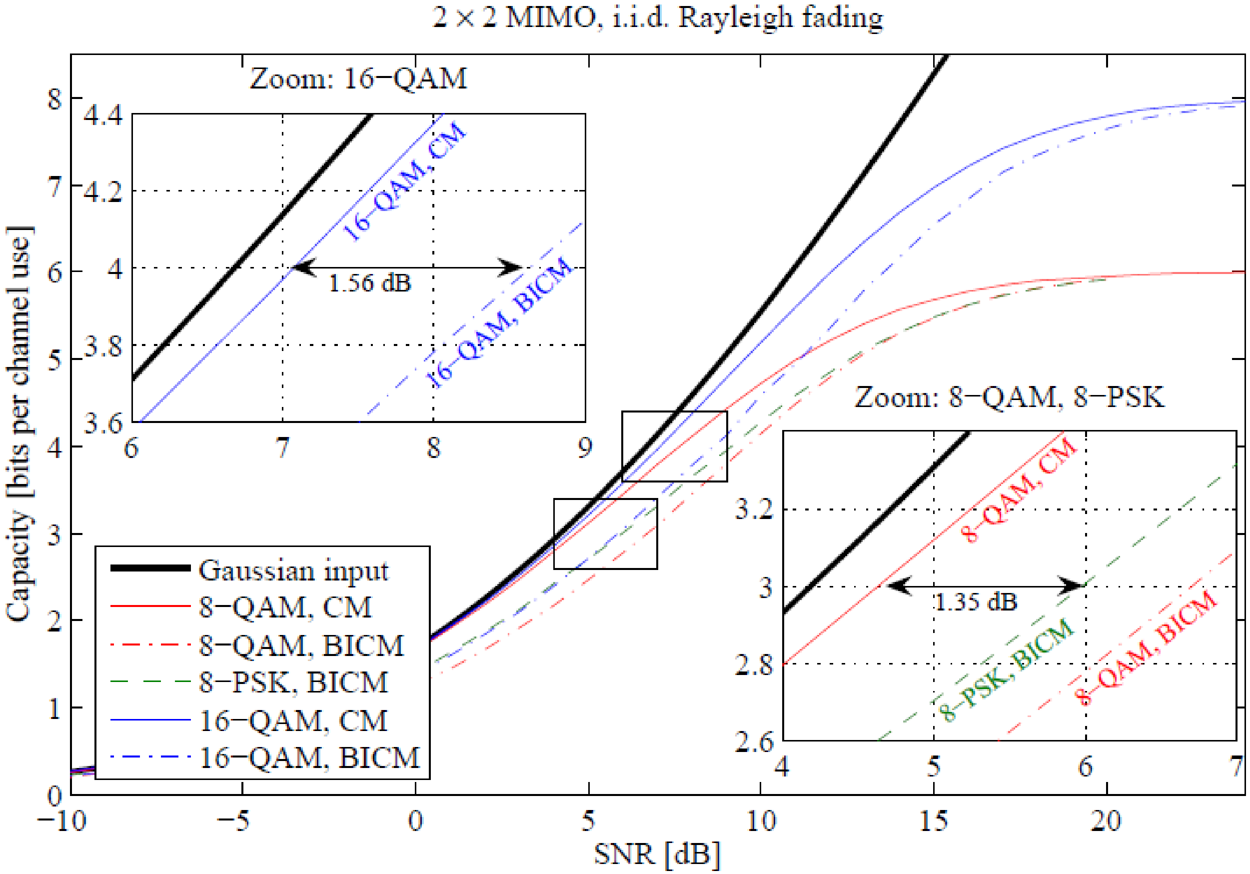
- All SNRs.
- Rates.
- Rayleigh Channel with and without erasures.

Gain is more important for Rayleigh channel with erasure.

A. Adbmouleh et al., «On Signal Space Diversity for Non Binary Coded Modulation Schemes", ICT 2016 Thessalonique (Greece) 18/29

Mutual Information in MIMO Channel

Higher capacity of NB coded modulation vs. BICM in MIMO channels



[D. Declercq, IEEE SSC SCV Tutorial, Santa Clara, Oct. 21st, 2010]

Outline

What is a NB-LDPC?

What are they advantages over binary codes?

Check Node processing (Extended Min Sum)

Pre-sorting of input vector

Conclusion

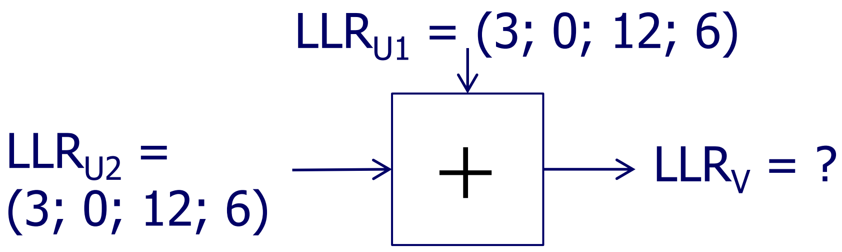
Main NB-LDPC Decoding algorithms

- Several “message based” decoding algorithms
  - Exchange of symbols: stochastic, symbol-flipping.
  - Believe Propagation and its suboptimal versions
    - FFT domain (q log q)
    - Min-Max [1]
    - Trellis-based EMS [2]
    - Extended Min-Sum (EMS) [3] (can approach BP algorithm)

- Principle of EMS: exchanged messages keep only the first n_m LLR and their associated symbols ($n_m << q$)

[1] V. Savin, “Min-max decoding for non binary LDPC codes,” in Proc. IEEE Int. Symp. Information Theory, ISIT’2008. Toronto, Canada, July 2008.
[2] D. Declercq and M. Fossorier, “Decoding algorithms for nonbinary LDPC codes over GF(q),” IEEE Trans. Comm., vol. 55, no. 4, pp. 633– 643, April 2007
[3] E. Li, D. Declercq, K. Gunnam, “Trellis-Based Extended Min-Sum Algorithm for Non-Binary LDPC Codes and its Hardware Structure”, IEEE Trans. on Communications 61(7):2600-2611 · July 2013

Example



LLR_{U1} = (3; 0; 12; 6)

LLR_{U2} =
(3; 0; 12; 6)

+

LLR_V = ?

a) Elementary Check Node

+	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

b) GF(4) addition table

U\V	18	7	9	0
3	21;0	10;α^0	12;α^1	3;α^2
0	18;α^0	7;0	9;α^2	0;α^1
12	30;α^1	19;α^2	21;0	12;α^0
6	24,α^2	13;α^1	15;α^0	6;0

c) (U+V) table

$$LLR_V(\alpha^k) \approx \underset{\alpha^i,\alpha^j \in GF(q)^2 / \alpha^i+\alpha^j=\alpha^k}{MIN} LLR_{U1}(\alpha^i) + LLR_{U2}(\alpha^j)$$

Example

$U_1\backslash U_2$	18	7	9	0
3	21;0	10;α^0	12;α^1	3;α^2
0	18;α^0	7;0	9;α^2	0;α^1
12	30;α^1	19;α^2	21;0	12;α^0
6	24,α^2	13;α^1	15;α^0	6;0

a) LLR_V (0) = min{21, 7, 21, 6} = 6.

$U_1\backslash U_2$	18	7	9	0
3	21;0	10;α^0	12;α^1	3;α^2
0	18;α^0	7;0	9;α^2	0;α^1
12	30;α^1	19;α^2	21;0	12;α^0
6	24,α^2	13;α^1	15;α^0	6;0

b) LLR_V (α^0)= min{10, 18, 12, 15} = 10.

$U_1\backslash U_2$	18	7	9	0
3	21;0	10;α^0	12;α^1	3;α^2
0	18;α^0	7;0	9;α^2	0;α^1
12	30;α^1	19;α^2	21;0	12;α^0
6	24,α^2	13;α^1	15;α^0	6;0

c) LLR_V (α^1)= min{12, 0, 30, 13} = 0.

$U_1\backslash U_2$	18	7	9	0
3	21;0	10;α^0	12;α^1	3;α^2
0	18;α^0	7;0	9;α^2	0;α^1
12	30;α^1	19;α^2	21;0	12;α^0
6	24,α^2	13;α^1	15;α^0	6;0

d) LLR_V (α^2)= min{3, 9, 19, 23} = 3.

$$LLR_V(\alpha^k) \approx \underset{\alpha^i,\alpha^j \in GF(q)^2 / \alpha^i+\alpha^j=\alpha^k}{MIN} LLR_{U1}(\alpha^i)+LLR_{U2}(\alpha^j)$$

Extended Min-Sum algorithm on a toy example

ECN Processing:

$$LLR_E(\alpha^k) \approx \underset{\alpha^i,\alpha^j \in GF(q)^2 / \alpha^i+\alpha^j=\alpha^k}{MIN} LLR_U(\alpha^i) + LLR_V(\alpha^j)$$

Example : (0; α^0; α^1; α^2) is the GF implicit order.

$LLR(U_1)$ = (3; 0; 12; 6) keep best $n_m = 2$ => ((0, α^0), (3, 0))

$LLR(U_2)$ = (18; 7; 9; 0) keep best $n_m = 2$ => ((0, α^2), (7, α^0))

	18;0	7; α^0	9; α^1	0; α^2
3;0	21;0	10;α^0	12;α^1	3;α^2
0; α^0	18;α^0	7;0	9;α^2	0;α^1
12;α^1	30;α^1	19;α^2	21;0	12;α^0
6; α^2	24,α^2	13;α^1	15;α^0	6;0

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

Extract the n_m smallest values among the n_m^2 values

Complexity: $2q^2$ => 4 ×n_m additions (Bubble algorithm)

Extended Min-Sum (EMS) Check Node (CN) processing

CN equation

$d_{c} = 4$

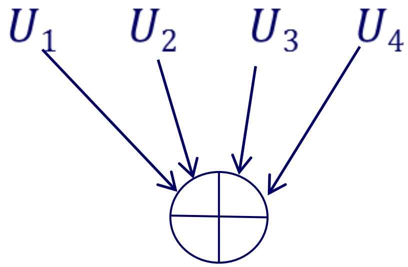
$U_1 \oplus U_2 \oplus U_3 \oplus U_4 = 0$

$V_1 = U_2 \oplus U_3 \oplus U_4$

$V_2 = U_1 \oplus U_3 \oplus U_4$

$V_3 = U_1 \oplus U_2 \oplus U_4$

$V_4 = U_1 \oplus U_2 \oplus U_3$



NB parity-check equation with d_c messages of n_m couples each

$$v_i^+(x) = \min\left\{ \sum_{i'=1, i'\neq i}^{d_c} U_{i'}^+[j_{i'}] \mid \bigoplus_{i'=1, i'\neq i}^{d_c} U_{i'}^{\oplus}[j_{i'}] = x \right\} j_{i'} \in \{0, 1, \ldots n_m - 1\}$$

Sort $v_i^+(x)$, $x \in GF(q)$, to generate output message V_i that contains the n_m couples ($v_i^+(x)$, $v_i^{\oplus}(x)$) with the smallest $v_i^+(x)$ values.

Extended Min-Sum (EMS)
Check Node (CN) processing, $d_c = 3$

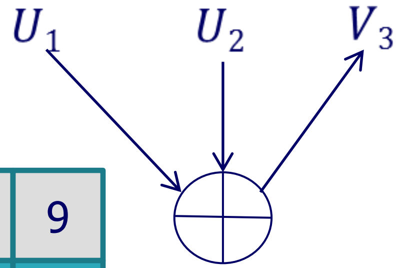
- Rather simple computation.

U_1, n_m=4

n_m=4

+	0	3	5	9
0	0	3	5	9
2	2	5	7	11
4	4	7	9	14
11	11	14	16	20

+	0	3	5	9
0	0	3	5	9
2	2	5	7	11
4	4	7	9	14
11	11	14	16	20



0	2	3	4

Output V_2 of the
n_m = 4 smallest values.

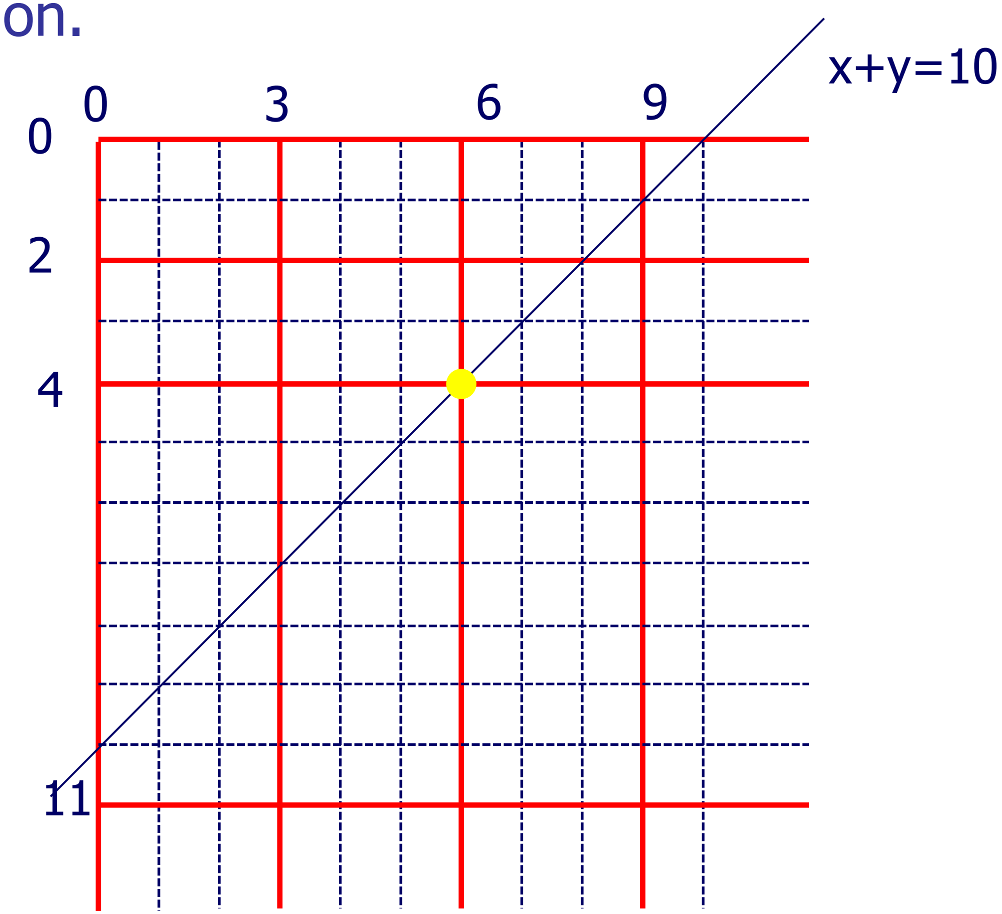
Geometrical interpretation

- Rather simple computation.

$n_m=4$

$n_m=4$

+	0	3	6	9
0	0	3	6	9
2	2	5	8	11
4	4	7	10	14
11	11	14	17	20



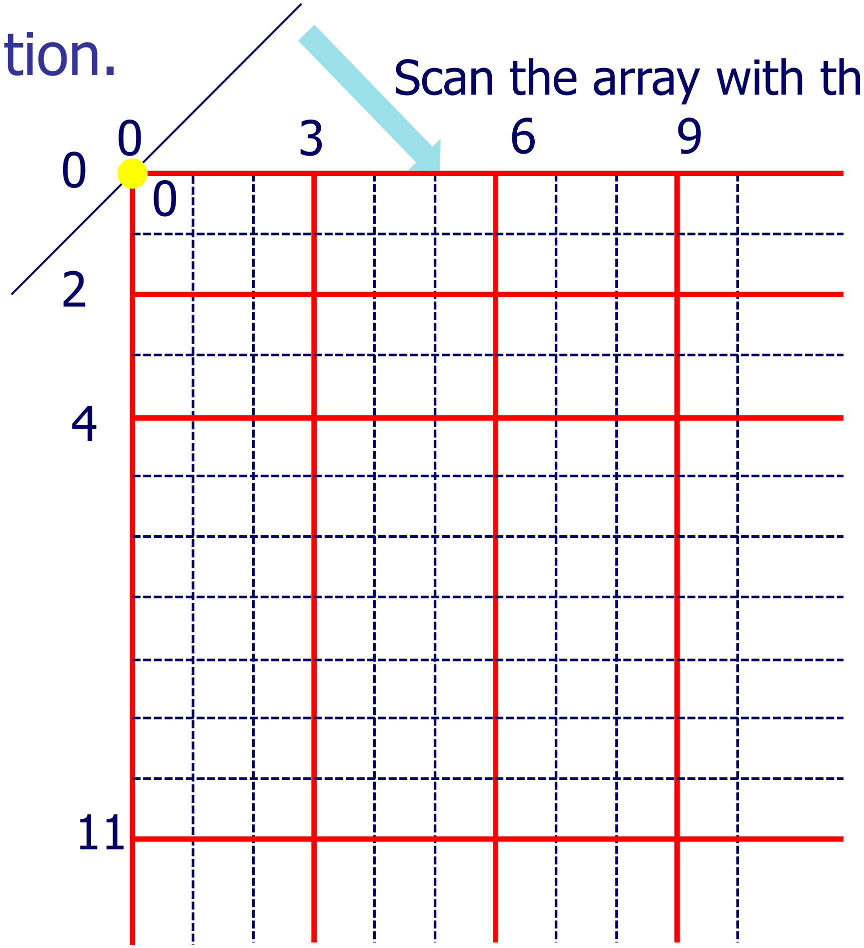
Geometrical interpretation for $d_c = 3$.

- Rather simple computation.

$n_m=4$

$n_m=4$

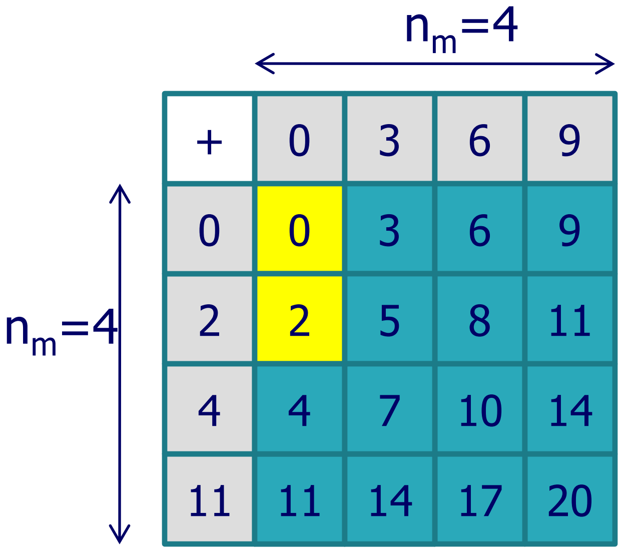
+	0	3	6	9
0	0	3	6	9
2	2	5	8	11
4	4	7	10	14
11	11	14	17	20



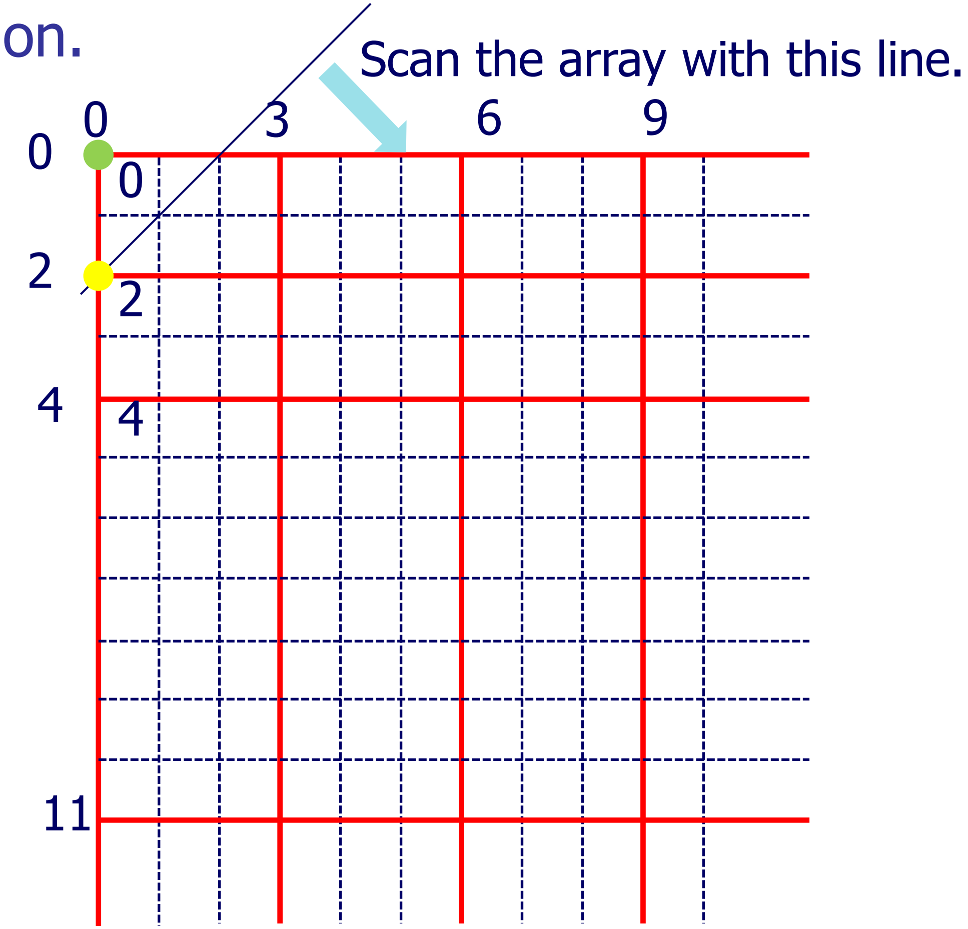
Scan the array with this line.

Geometrical interpretation for $d_c = 3$.

- Rather simple computation.

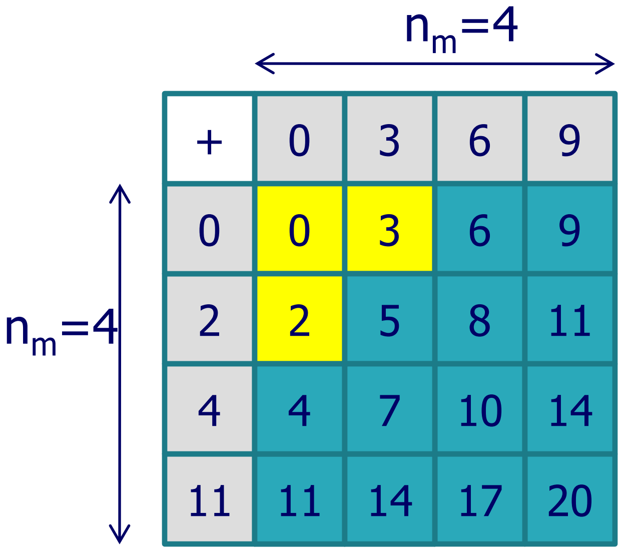


+	0	3	6	9
0	0	3	6	9
2	2	5	8	11
4	4	7	10	14
11	11	14	17	20

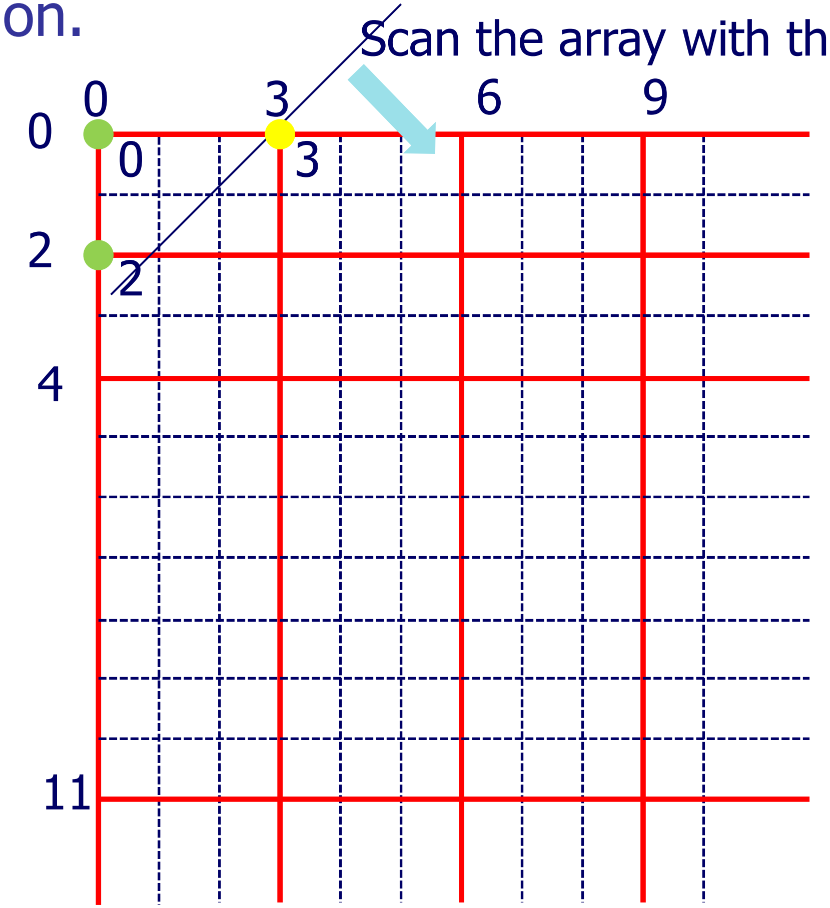

Scan the array with this line.

Geometrical interpretation for $d_c = 3$.

- Rather simple computation.



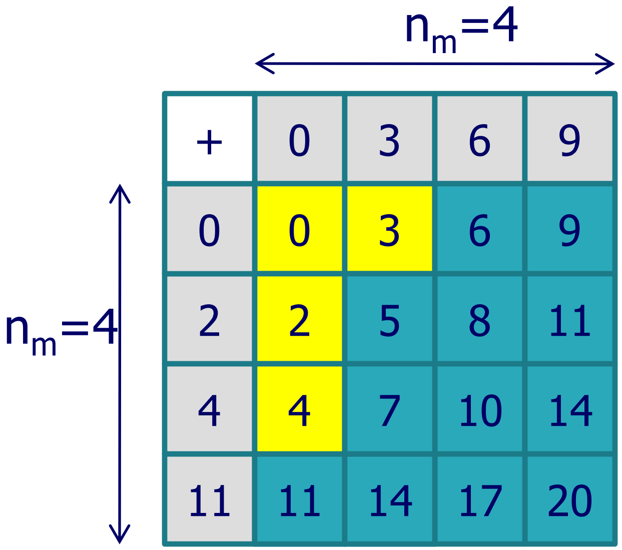
+	0	3	6	9
0	0	3	6	9
2	2	5	8	11
4	4	7	10	14
11	11	14	17	20



Scan the array with this line.

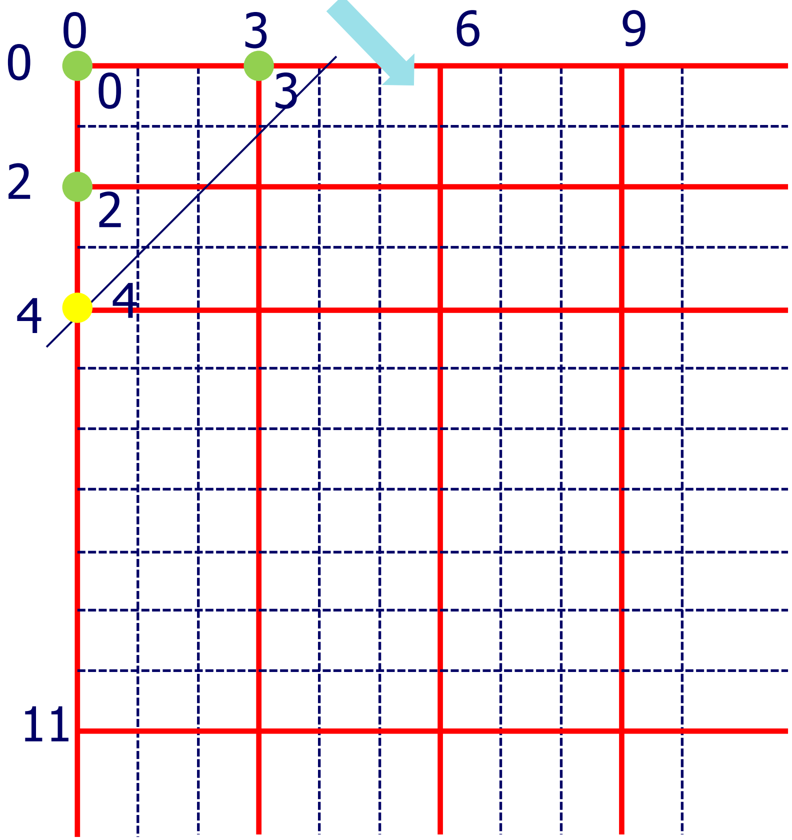
Geometrical interpretation for $d_c = 3$.

- Rather simple computation.

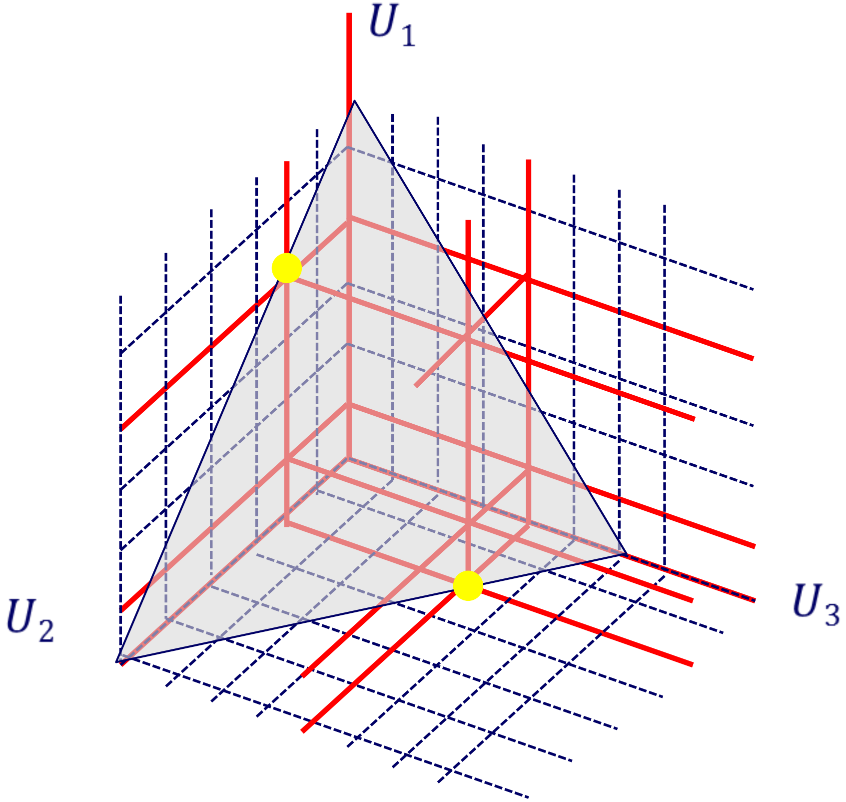


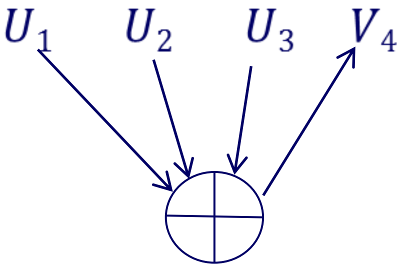
+	0	3	6	9
0	0	3	6	9
2	2	5	8	11
4	4	7	10	14
11	11	14	17	20

Scan the array with this line.



Geometrical interpretation for $d_c = 3$.





Generate V_4 by scanning the d_c-1 = 3 dimension space with plane x + y + z = Cte.

(same process for V_1, V_2, V_3).

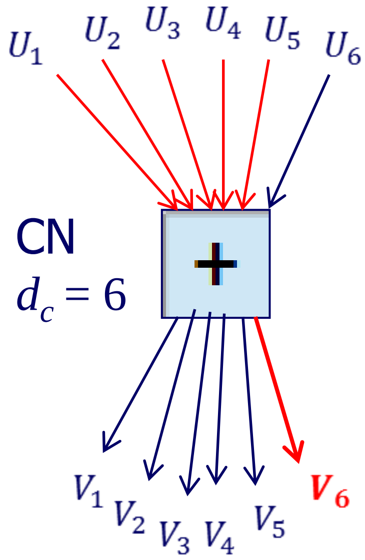
Direct approach: complexity of $d_c \times (n_m)^{dc\text{-}1}$

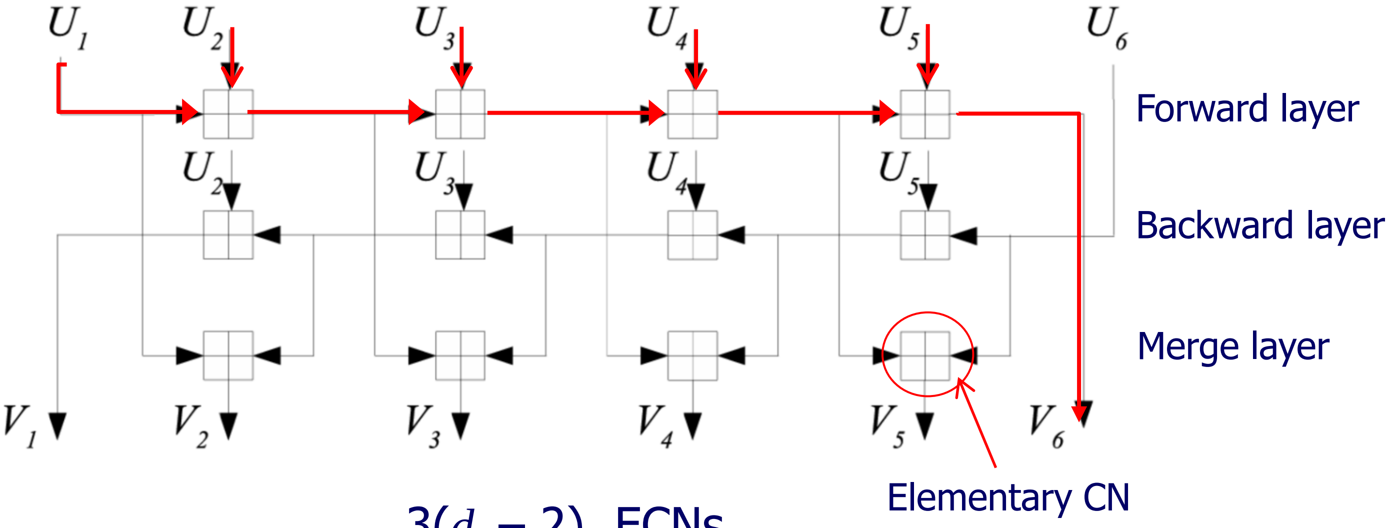
State of the art solutions

Forward-Backward: divide and conquer method.

Syndrome based: global method

Forward-Backward (FB) implementation of EMS CN decoder



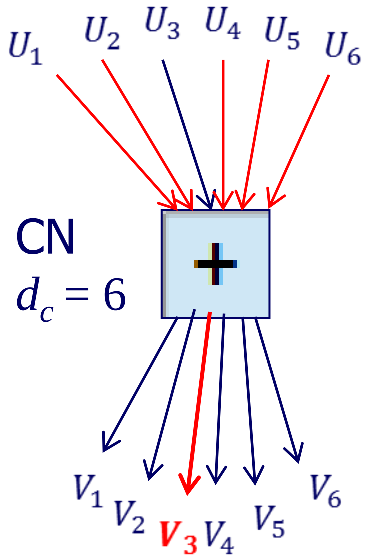


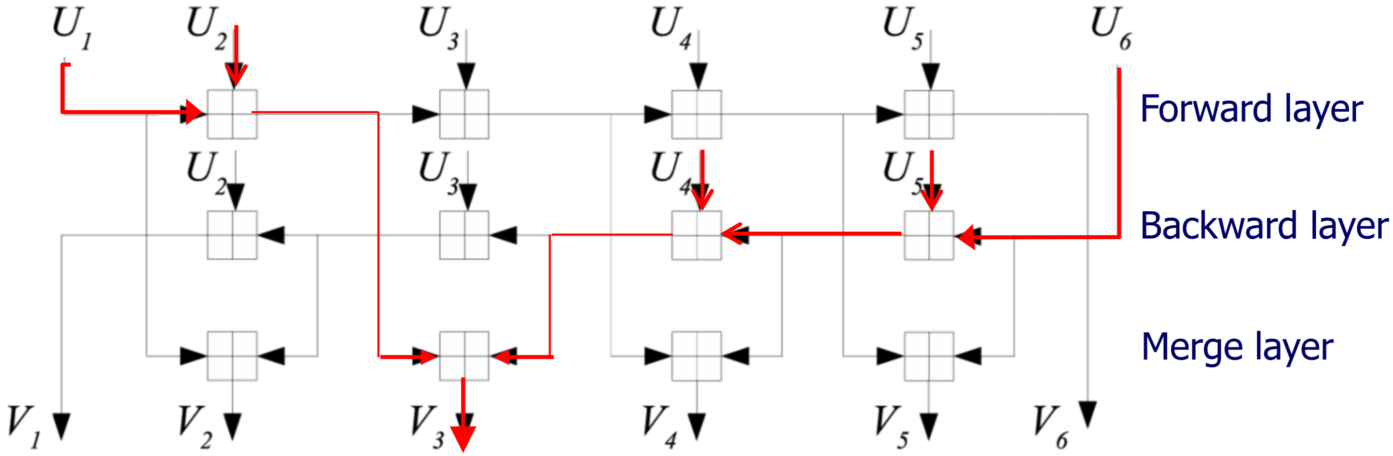
$3(d_c - 2)$ ECNs

$$v_i^+(x) = \min\left\{ \sum_{i'=1,i'\neq i}^{d_c} U_{i'}^+[j_{i'}] \mid \bigoplus_{i'=1,i'\neq i}^{d_c} U_{i'}^{\oplus}[j_{i'}] = x \right\}$$

➡ Reducing complexity at the ECN level to reduce the overall CN complexity

Forward-Backward (FB) implementation of EMS CN decoder



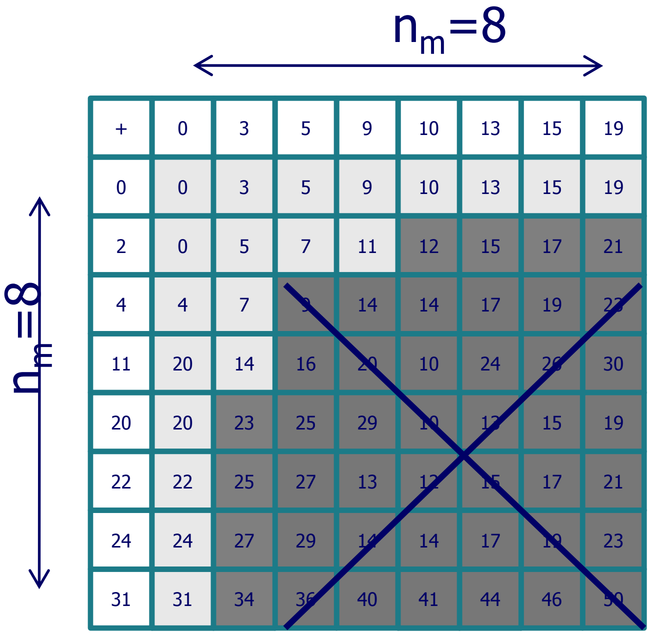


$3(d_c - 2)$ ECNs

$$v_i^+(x) = \min\left\{ \sum_{i'=1, i'\neq i}^{d_c} U_{i'}^+[j_{i'}] \mid \bigoplus_{i'=1, i'\neq i}^{d_c} U_{i'}^{\oplus}[j_{i'}] = x \right\}$$

Reducing complexity at the ECN level to reduce the overall CN complexity

Hardware simplification of ECN



+	0	3	5	9	10	13	15	19
0	0	3	5	9	10	13	15	19
2	0	5	7	11	12	15	17	21
4	4	7	9	14	14	17	19	23
11	20	14	16	20	10	24	26	30
20	20	23	25	29	10	13	15	19
22	22	25	27	13	12	15	17	21
24	24	27	29	14	14	17	19	23
31	31	34	36	40	41	44	46	50

Restric the research of candidate along the axis.

Working load: extract n_m smalest values among ($3n_m - 4$) (n_m^2 initially).

=>Small performance degradation

$\Rightarrow$Significant hardware saving

[1] Y. S. Park et al., “A fully parallel nonbinary LDPC decoder with fine-grained dynamic clock gating,” IEEE Journal of Solid-State Circuits, Feb 2015.

[2] E. Boutillon et al. Design of a GF(64)-LDPC Decoder Based on the EMS Algorithm IEEE Transactions on Circuits and Systems I, 2013

Syndrome based: direct research in the space of dimension d_c.

Idea: Among the T = $(n_m)^{dc}$ configuations,

1) Select a subset $\Delta \subset$ T of vector δ of dimension d_c.

2) For each δ, compute syndrom $S(\delta) = (S(\delta)^+, S(\delta)^\oplus)$

$$S(\delta)^+ = \sum_{i=1}^{dc} U_i[\delta(i)]^+ \qquad S(\delta)^\oplus = \oplus_{i=1}^{dc} U_i[\delta(i)]^\oplus$$

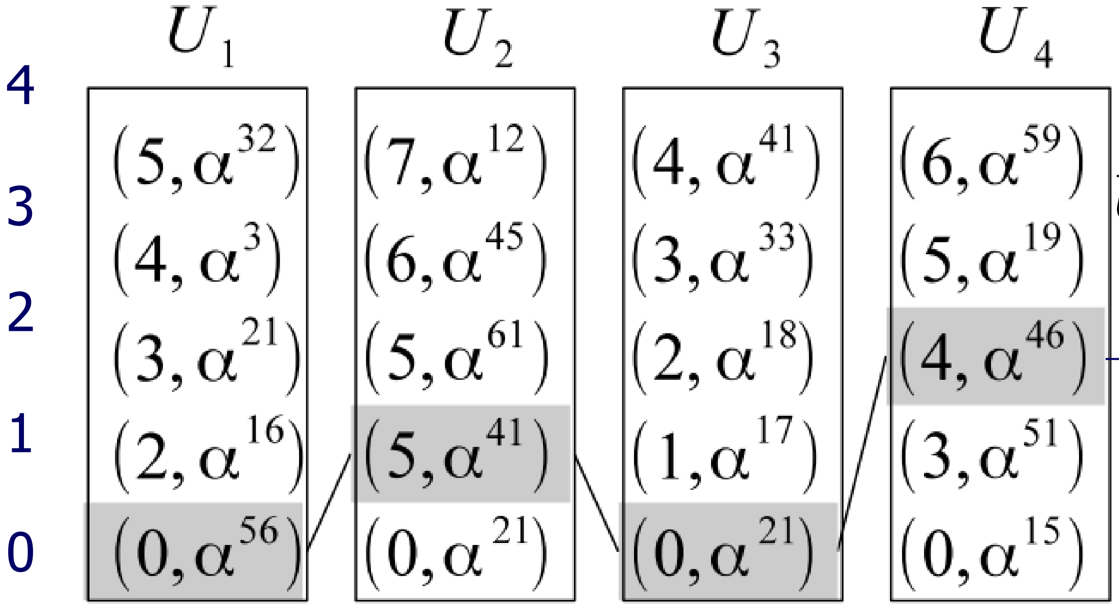
3) Sort the $S(\delta)$ in increading order of $S(\delta)^+$

4) Generate the i th output message V_i by suppressing the contribution of the i th dimension in $S(\delta)$ (Decorrelation Unit)

If $\delta(k) = 0$, then S(d) contribute to output V_k as:

$$S(\delta)\big|_{\rightarrow k} = (S(\delta)^+, S(\delta)^\oplus \oplus U_k[\delta(k)]^\oplus)$$

Syndrome based: direct research in the space of dimension d_c.



	U_1	U_2	U_3	U_4
4	$(5,\alpha^{32})$	$(7,\alpha^{12})$	$(4,\alpha^{41})$	$(6,\alpha^{59})$
3	$(4,\alpha^{3})$	$(6,\alpha^{45})$	$(3,\alpha^{33})$	$(5,\alpha^{19})$
2	$(3,\alpha^{21})$	$(5,\alpha^{61})$	$(2,\alpha^{18})$	$(4,\alpha^{46})$
1	$(2,\alpha^{16})$	$(5,\alpha^{41})$	$(1,\alpha^{17})$	$(3,\alpha^{51})$
0	$(0,\alpha^{56})$	$(0,\alpha^{21})$	$(0,\alpha^{21})$	$(0,\alpha^{15})$

$$U_4[2] = (4, a^{46})$$

$$\delta = (0,1,0,2)$$

$$S(\delta)^{+} = \sum_{i=1}^{dc} U_i[\delta(i)]^{+} = 9 \qquad S(\delta)^{\oplus} = \oplus_{i=1}^{dc} U_i[\delta(i)]^{\oplus} = \alpha^{46}$$

$$S(\delta)\Big|_{\rightarrow 1} = (9, \quad \alpha^{56} + \alpha^{46}) = (9, \quad \alpha^{28})$$

Method to select Δ

Set decoding condition of min-sum algorithm
(GF size, Parity check matrix, SNR, nb of iterations).

Perform a Monte-Carlo simulation:
Each time a vector δ is used to generate a CN output, set +1 to its score.

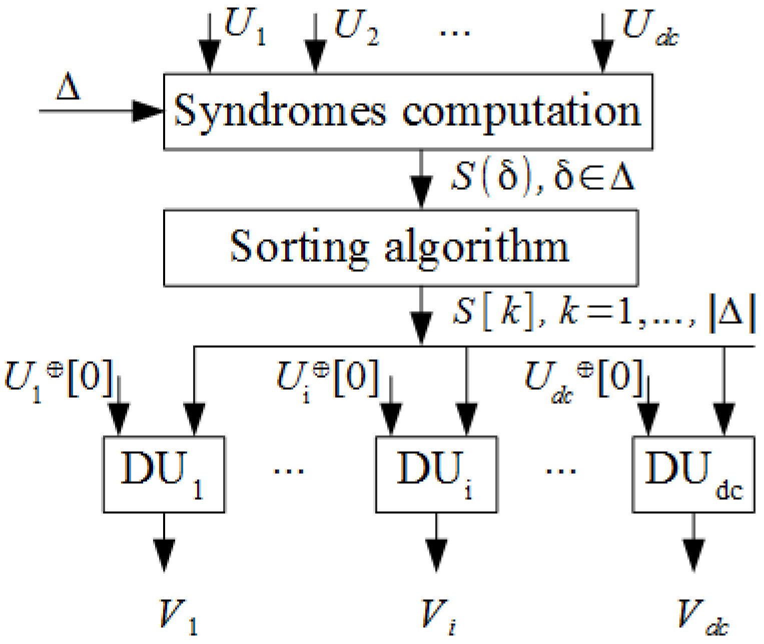
Sort the vector δ in decreasing order of their score.

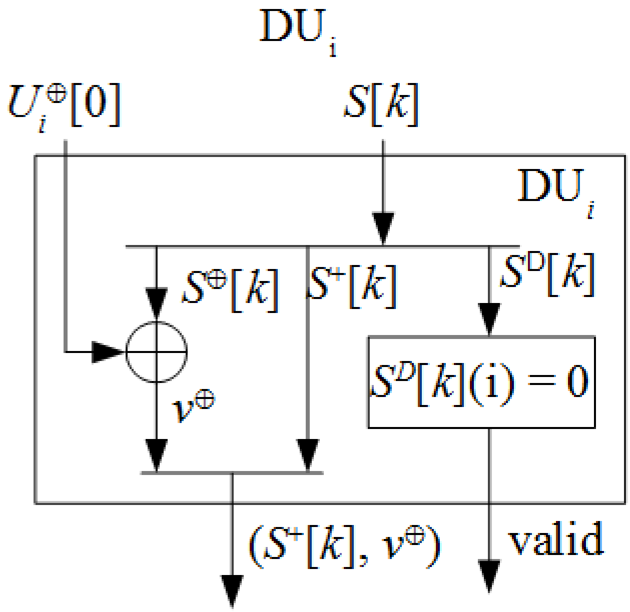
Set Δ as the set of the first z vectors with higher score.

The choice of $z = |\Delta|$ is a trade-off performance complexity.

In practice, z is of the order of $n_m d_c^2$.

Architecture of a syndrome based Check Node





[1] gives a proof of concept of syndrome based check node for $d_c = 4$, GF(256) NB-LDPC code.

[1] V. Rybalkin et al. "A new architecture for high speed, low latency NB-LDPC check node processing for GF(256)," IEEE VTC Spring, 2016.

Outline

What is a NB-LDPC?

What are they advantages over binary codes?

Check Node processing (Extended Min Sum)

Pre-sorting of input vector

Conclusion

Pre-sorting technique and its application to EMS FB implementation

Pre-sorting: sort input messages to further favorise ECN simplification…

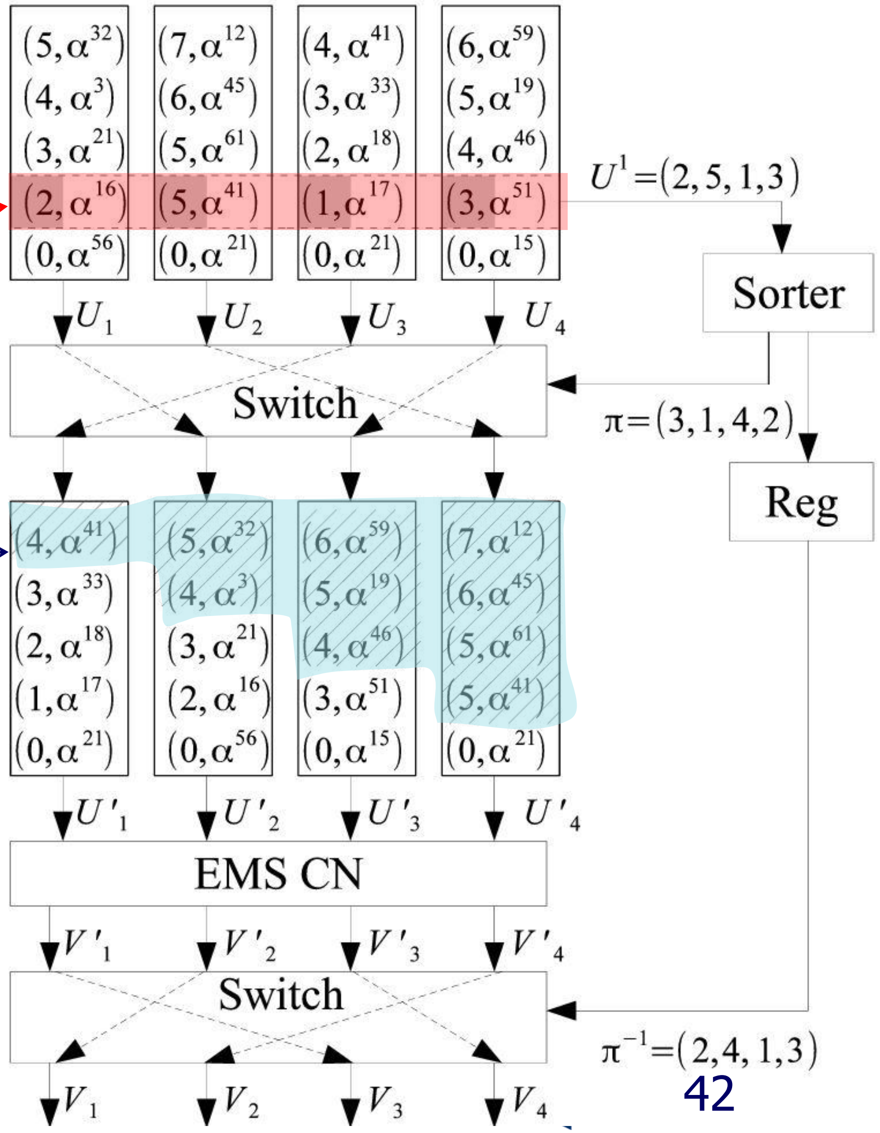
as a function of the second LLR values, because they mainly determine the reliability of the first symbol

Blue values not considered in CN processing as they will not contribute to output messages

➡ Hardware simplification

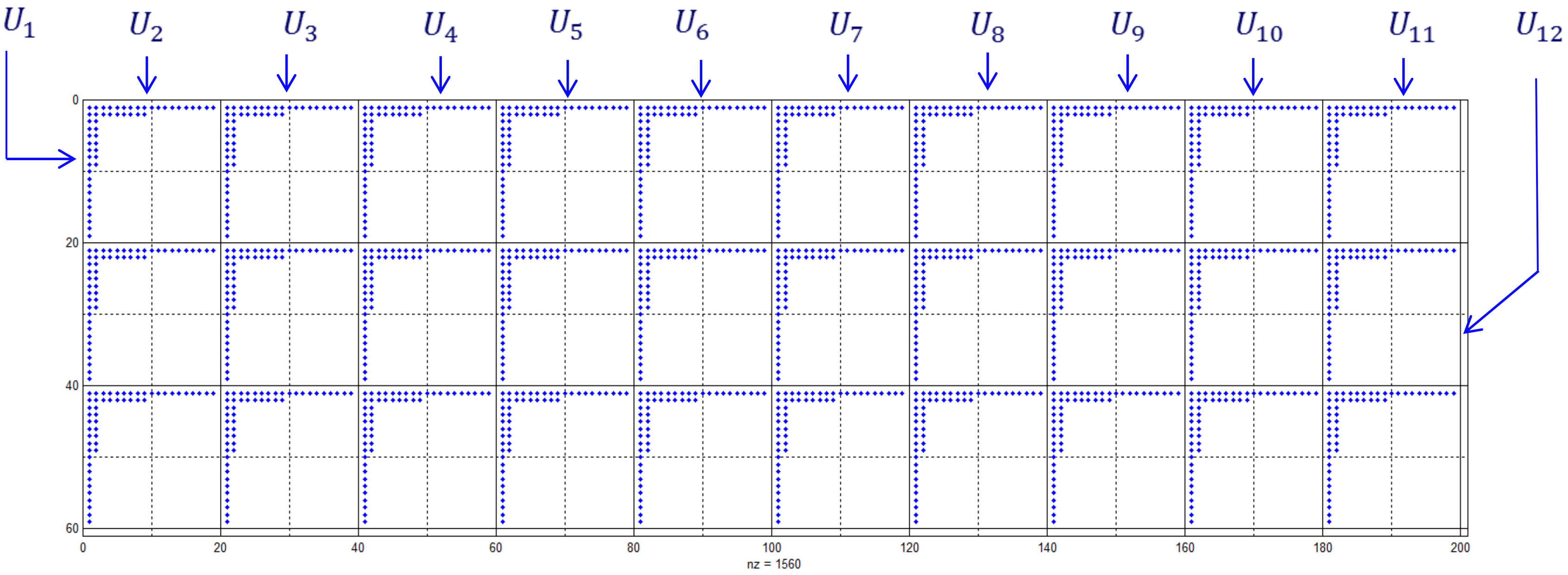
New blocks in architecture:

- ○ 2 switches
- ○ 1 sorter



Forward-Backward CN with simplified ECNs

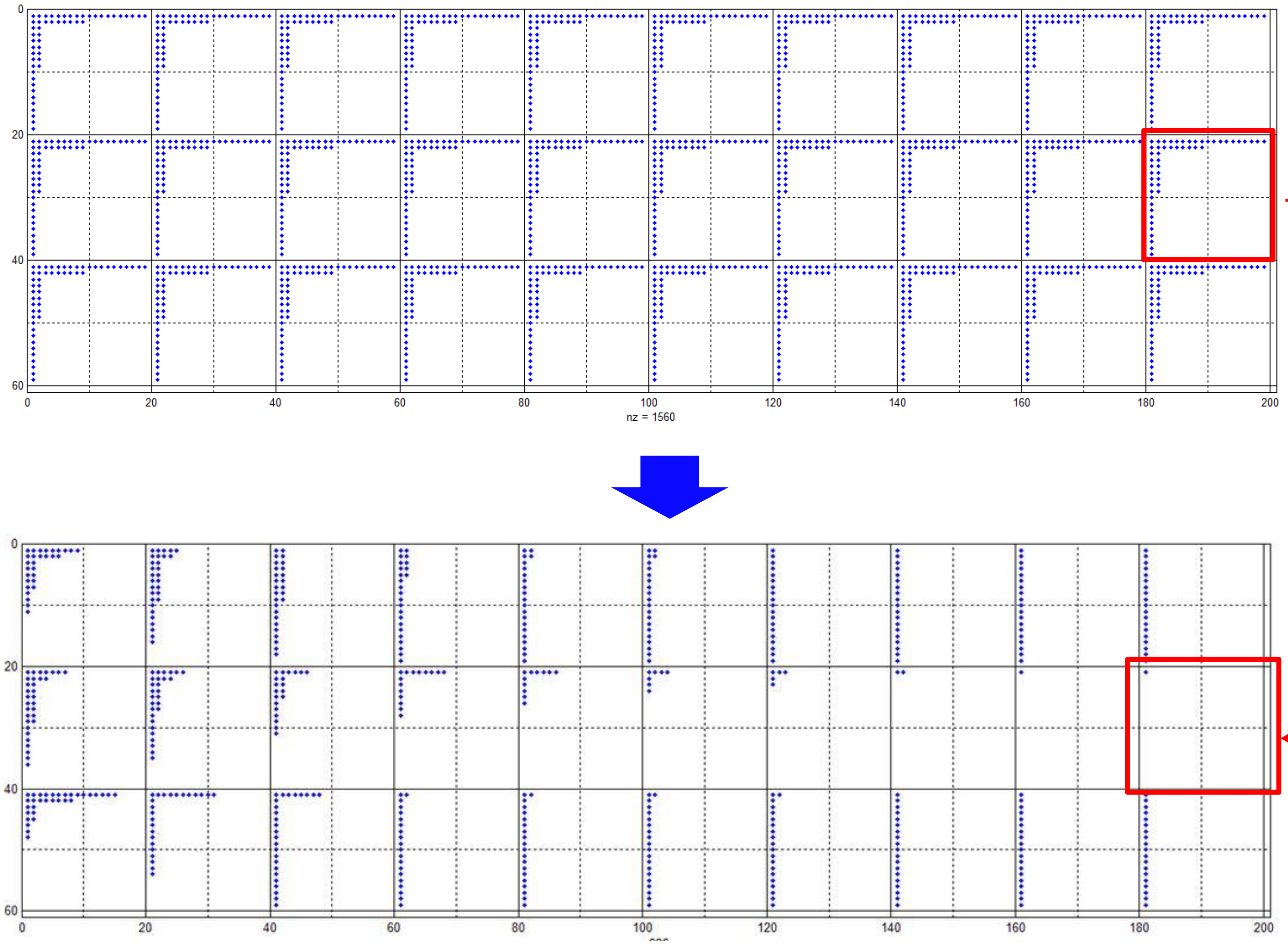
- S-Bubble architecture for $d_c = 12$



Points represent the potentiel selected values in each ECN

➡ Simplification thanks to the **pre-sorting technique** …

How pre-sorting can simplify the architecture ?



Without pre-sorting

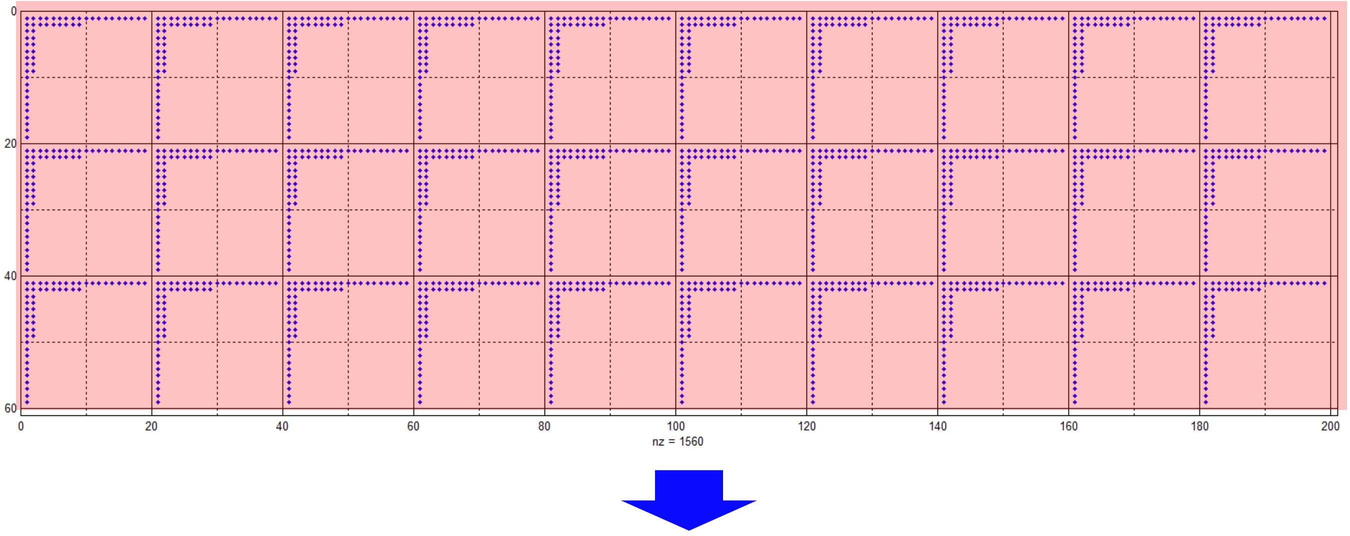
S-Bubble architecture
(4 FIFOs + control)

Example of ECN simplification

1 XOR architecture

With pre-sorting

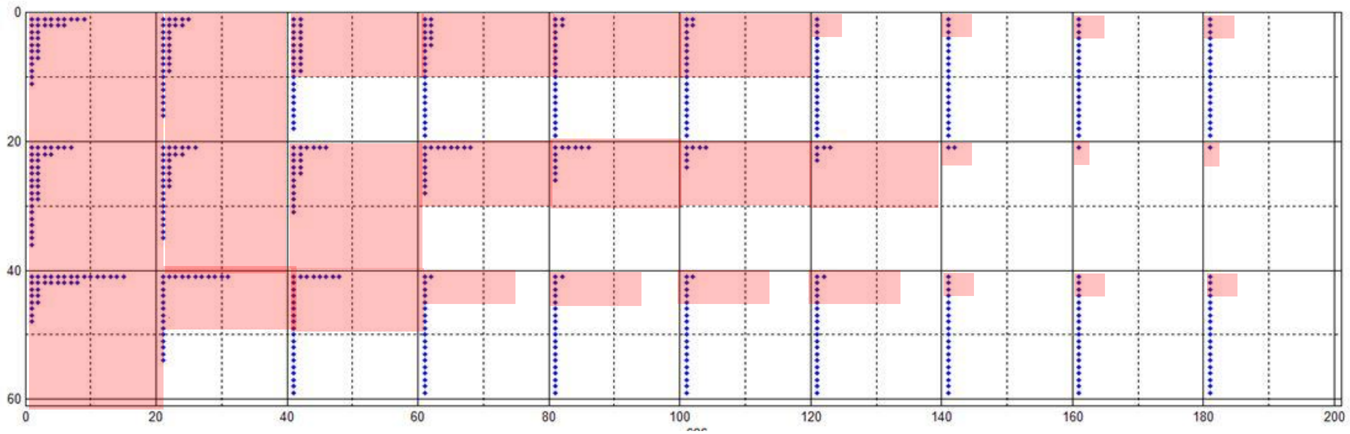
How pre-sorting can simplify the architecture ?



Without pre-sorting

~ 4100 slices
FPGA Xilinx Virtex 6

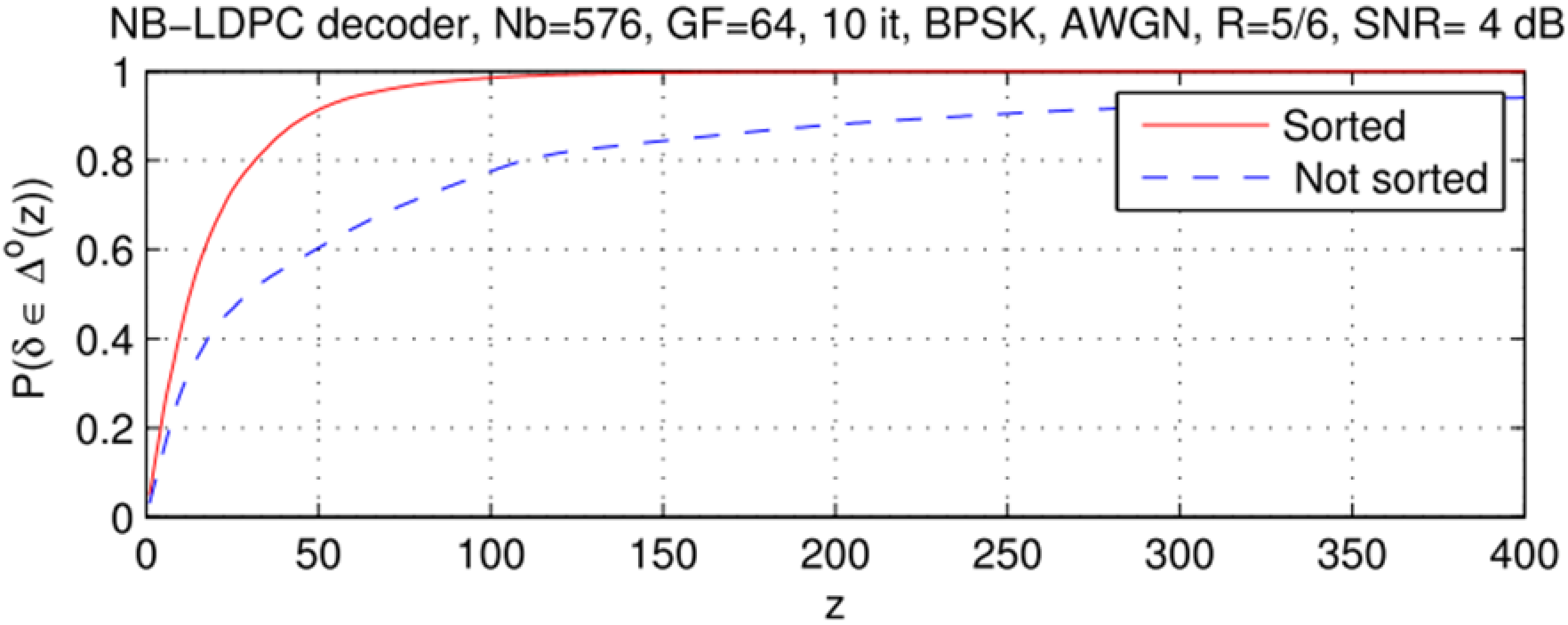
GF(64), n_m = 20



With pre-sorting

~ 1900 slices
FPGA Xilinx Virtex 6

Pre-sorting with Syndrome based CN



With pre-sorting, a smaller value of *z* is required to have an almost perfect CN processing => hardware saving

Sorted curve (red) is almost independant of the degree d_c => Path to a flexible NB-LDPC decoder.

Outline

What is a NB-LDPC?

What are they advantages over binary codes?

Check Node processing (Extended Min Sum)

Pre-sorting of input vector

Conclusion

Conclusion on NB-LDPC code

- Coded modulation over perform BICM modulation.

- Reduced-complexity Check Node implementation of NB-LDPC decoders based on the Extended Min-Sum

- Application of the **pre-sorting** technique to process the check node reduces the complexity gap between binary and Non-Binary decoder.

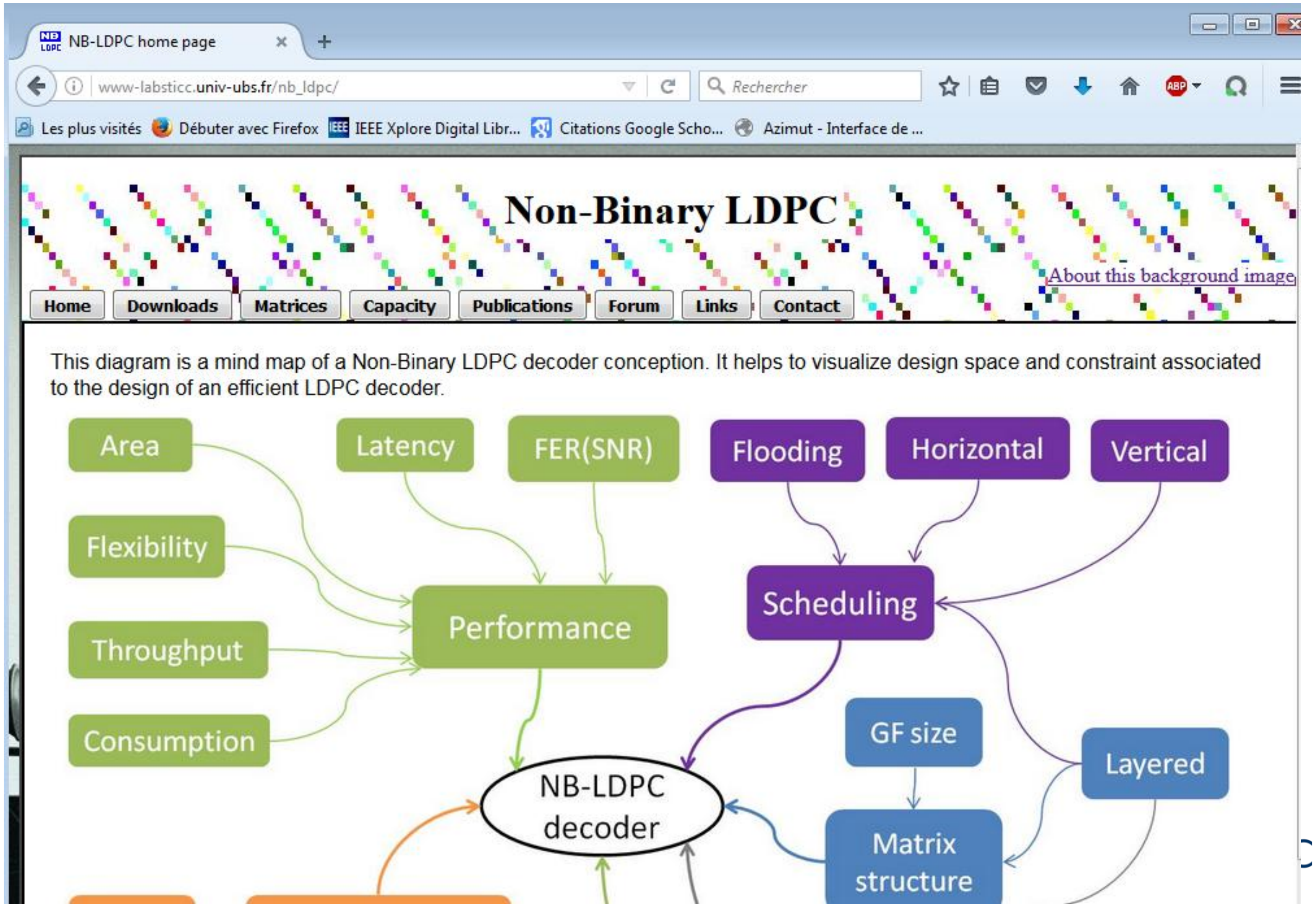
- **Future work:** global decoder implementation (on doing work)

- Visit “NB-LDPC codes” webpage: http://www-labsticc.univ-ubs.fr/nb_ldpc/

(Source code available under request)

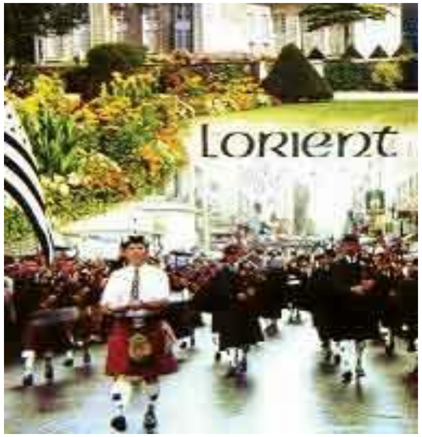
NB-LDPC web page

- Web page: http://www-labsticc.univ-ubs.fr/nb_ldpc/



This diagram is a mind map of a Non-Binary LDPC decoder conception. It helps to visualize design space and constraint associated to the design of an efficient LDPC decoder.

Thank you. Questions ?



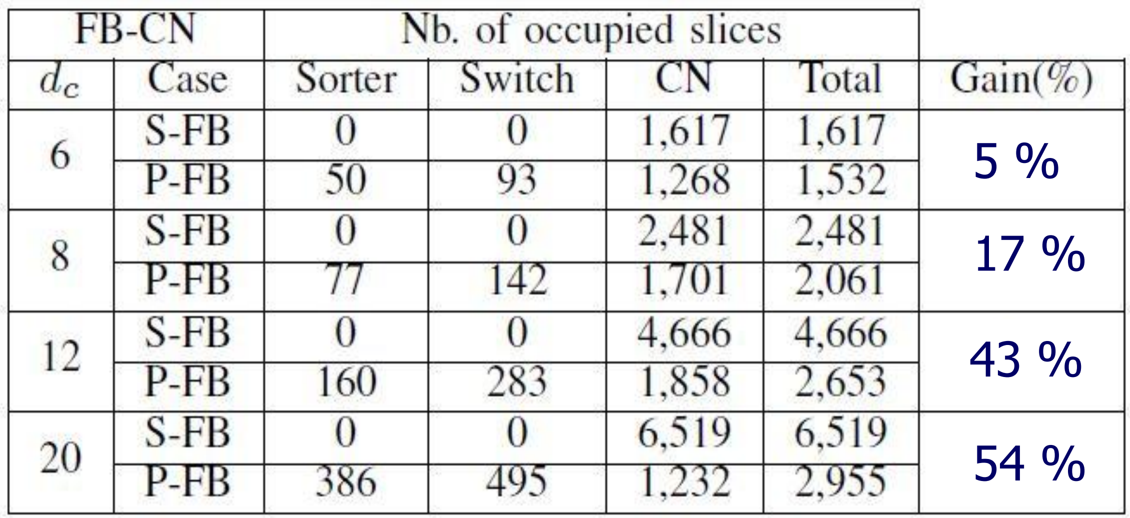
Contact: Prof. Emmanuel Boutillon
Email: **emmanuel. boutillon@univ-ubs.fr**

Lorient, France

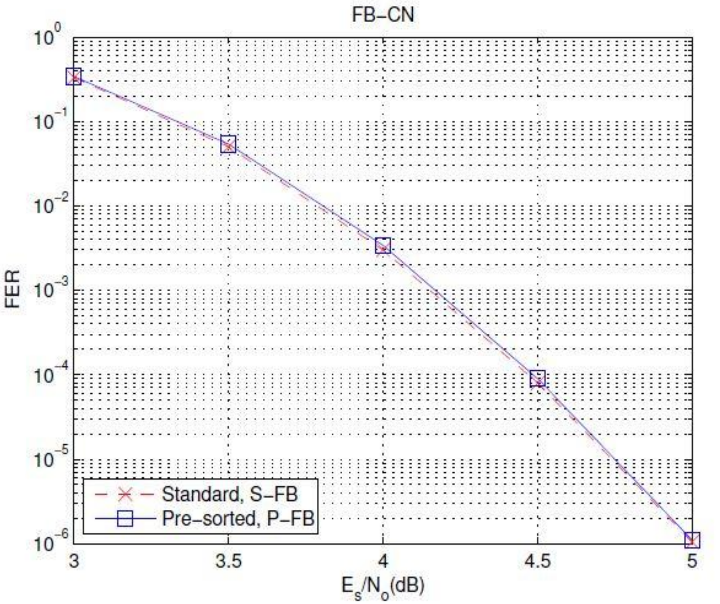
Application to serial architecture: implementation and simulation results

S-FB: S-bubble architecture

P-FB: with pre-sorting, simplified architecture



FB-CN		Nb. of occupied slices				
d_c	Case	Sorter	Switch	CN	Total	Gain(%)
6	S-FB	0	0	1,617	1,617	5 %
	P-FB	50	93	1,268	1,532	
8	S-FB	0	0	2,481	2,481	17 %
	P-FB	77	142	1,701	2,061	
12	S-FB	0	0	4,666	4,666	43 %
	P-FB	160	283	1,858	2,653	
20	S-FB	0	0	6,519	6,519	54 %
	P-FB	386	495	1,232	2,955	



Simulation results GF-(64)-LDPC
N=576 bits, d_v = 2, d_c = 12, R = 5/6

- Additionnal sorter and switch blocks in P-FB architecture
- Significant area reduction at the CN level in P-FB architecture
- Global area gain, especially for high d_c orders
- No performance loss

Extended Min–Sum (EMS) Check Node (CN) processing

The reliability of the first value is strongly determined by the second LLR value $U[1]$:

$$\boldsymbol{P(U[0]^{\oplus}) \leq \frac{1}{1+e^{-U[1]}}}$$

$$\boldsymbol{P(U[i]^{\oplus}) \leq \frac{e^{-U[i]}}{1+e^{-U[1]}}}$$

$$\mathbf{0,7416 \leq 0,8791 \leq 0,9526}$$

$$\mathrm{LLR}(x) = -\log \frac{\boldsymbol{P(x)}}{\boldsymbol{P(\bar{x})}}$$

- ☐ EMS: only the n_m most reliable couples

LLR value	GF(8) symbol
$U[0] = 0$	α_7
$U[1] = 3$	α_6
$U[2] = 3$	α_4
$U[3] = 4$	α_2

$n_m = 4$, sorted in increasing order of LLR values