

# Extended-Forward Architecture for Simplified Check Node Processing in NB-LDPC Decoders

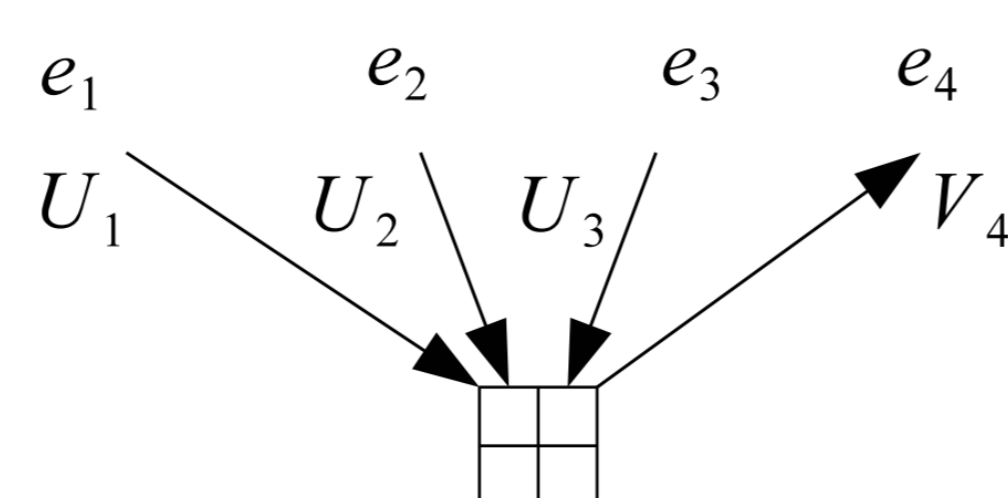
Cédric Marchand, Emmanuel Boutillon,  
Hassan Harb, Laura-Conde-Canencia and  
Ali Al Ghouwayel  
Université de Bretagne Sud, Lab-STICC UMR 6285  
[http://www-labsticc.univ-ubs.fr/nb\\_ldpc/](http://www-labsticc.univ-ubs.fr/nb_ldpc/)



## Introduction

Non-Binary Low-Density Parity-Check codes have better communication performance than their binary counterparts but they suffer from higher complexity, especially for the check node processing.

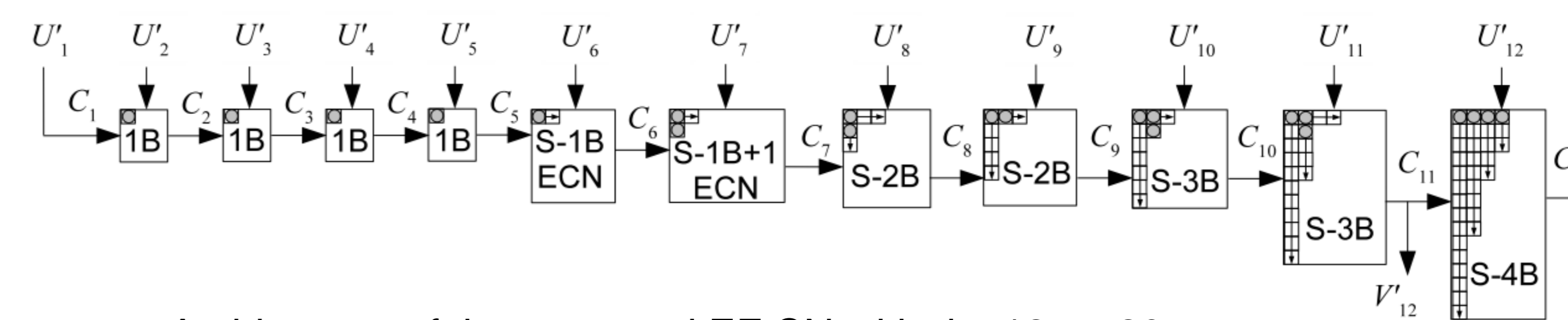
$$e^+(x) = -\log \left( \frac{P(e=x)}{P(e=\bar{x})} \right)$$



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

A sorting of the input vectors based on a reliability criteria is performed prior to the check node processing. This presorting process allows the Extended Min-Sum (EMS) [1] check node process to focus its effort mainly on the weakest inputs. A simplification of the syndrome based architecture [3] is here combined with the presorting process to further reduce check node complexity.

## Extended-Forward with presorting



Architecture of the proposed EF CN with  $d_c=12$ ,  $z=20$ .

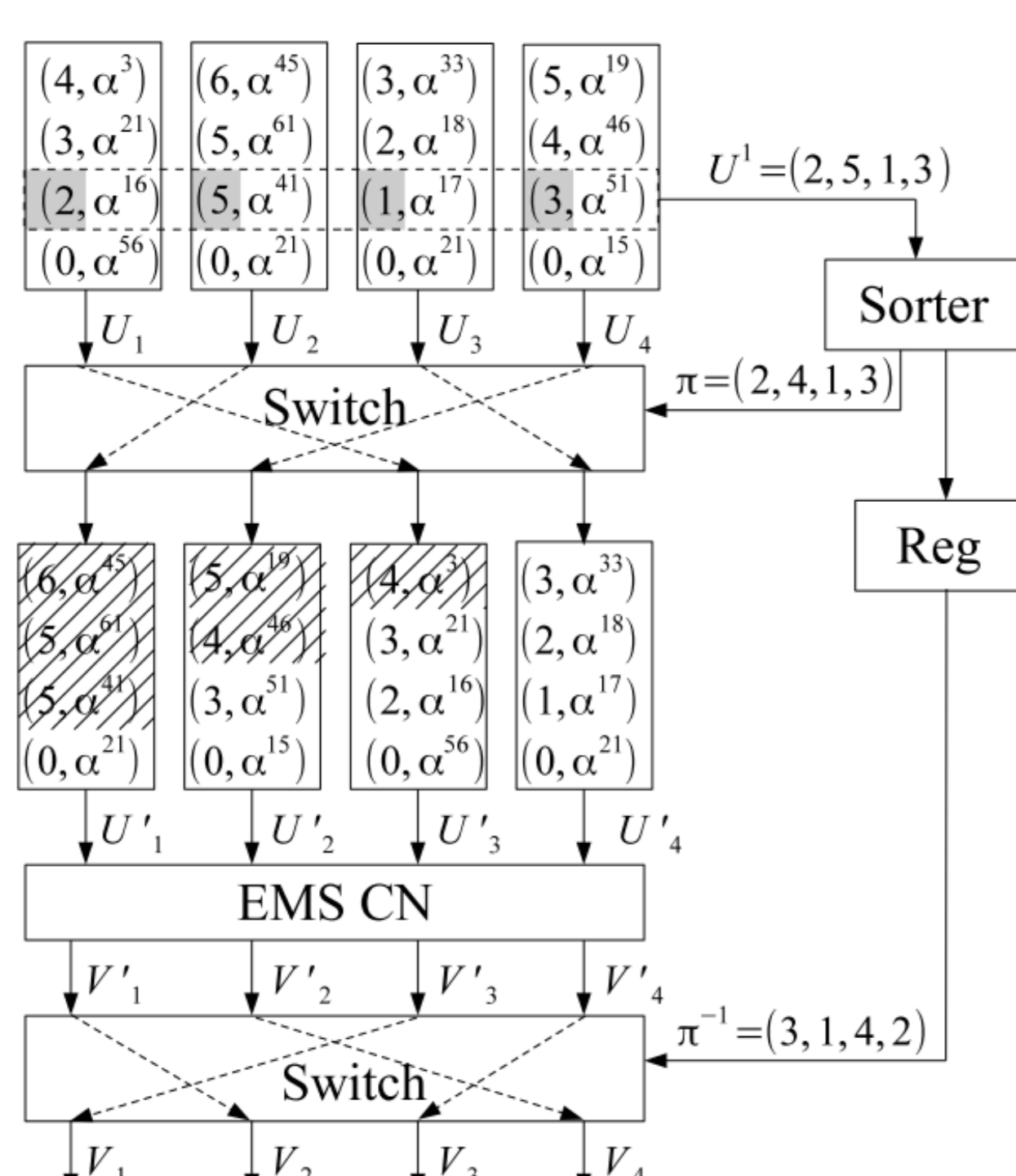
CN GF(64)	Number of slices			$F_{max}$ (MHz)	$L_{CN}$ (cycles)
	Total	REG	LUT		
FB RE	3904	6019	10314	227	22
PS FB RE	2729	4119	6733	229	20
EF RE	2633	4628	6768	217	24
PS EF RE	1987	3151	4814	222	19
PS EF	1495	2383	3616	227	19

Z : Number of processed syndromes  
FB : Forward-Backward  
PS : PreSorted  
EF : Extended Forward  
RE : Redundancy Elimination

Post synthesis results on Virtex 6, for GF(64) CN architectures.

The **throughput** can reach 534 Mbps with the FB architecture and 544 Mbps with the PS EF architecture

## Presorting principle [2]



The inputs are presorted based on the reliability of the second most reliable GF of each input.

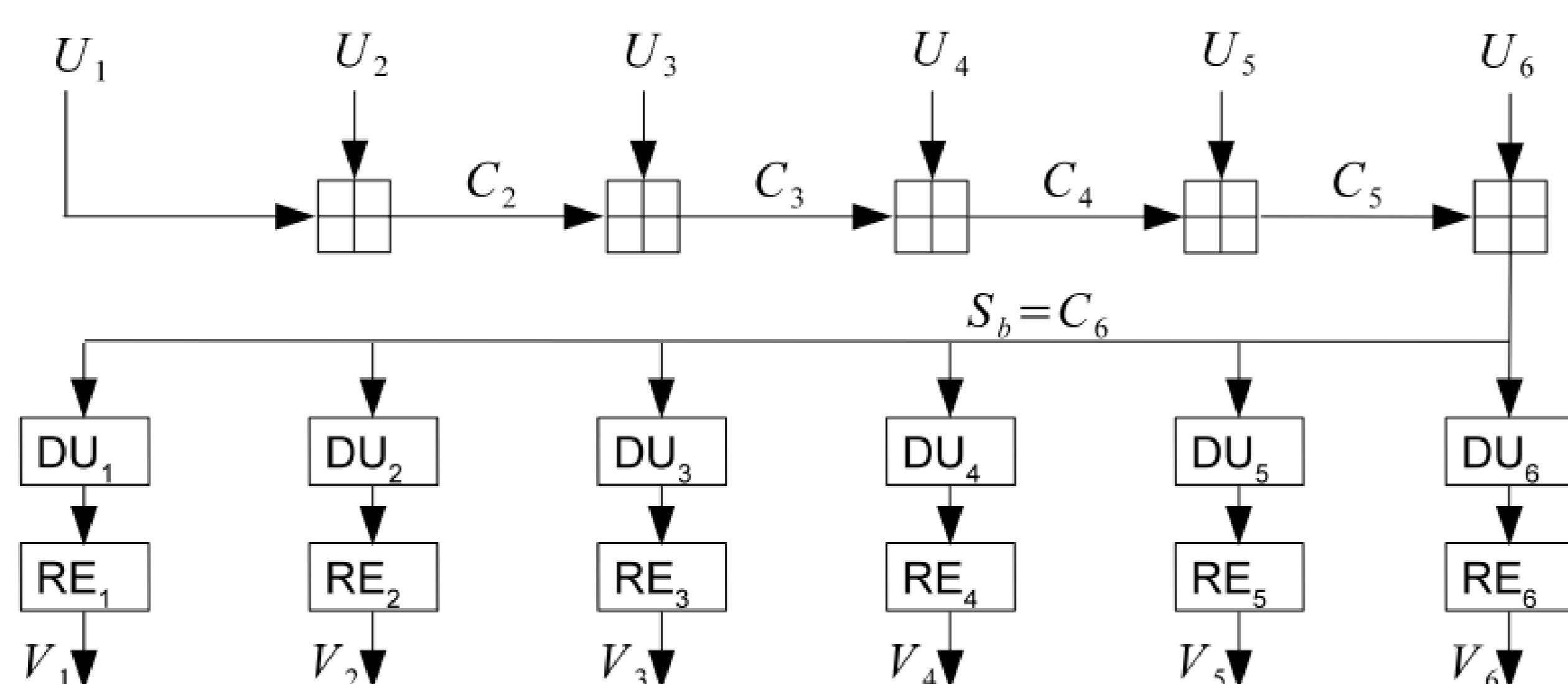
The presorting process reduces statistically the number of deviation paths required to generate the most reliable CN outputs.

In this particular example, the hatched tuples are not involved in the computation of the first 8 syndromes.

Compared to the standard CN, the presorting process requires extra hardware:

a  $d_c$ -input vector sorter and two permutation networks (or switches). However, it allows some simplification in the CN itself, globally leading to a complexity reduction of the whole CN processing.

## Extended-Forward architecture



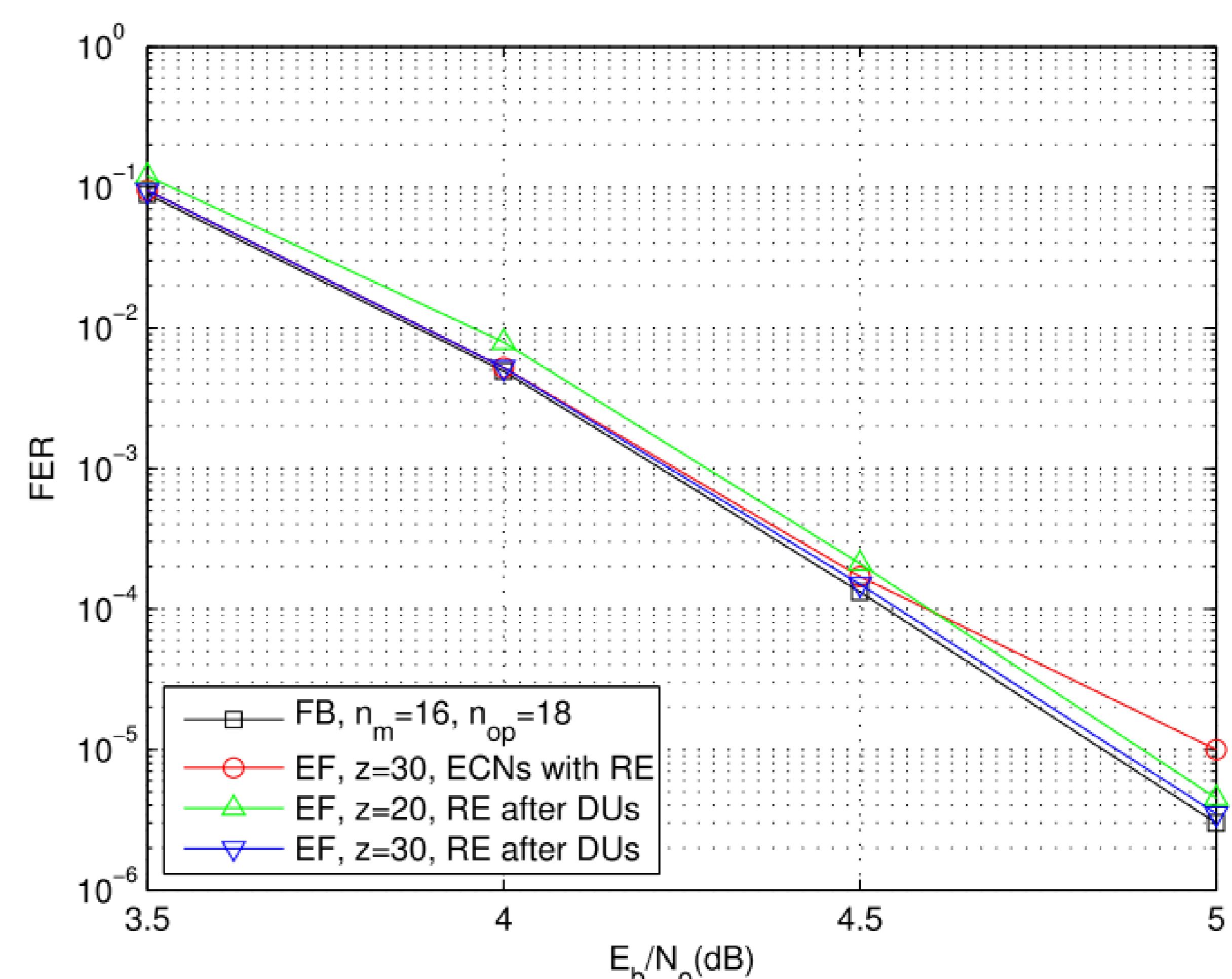
The Extended-forward architecture is a simplification of the syndrome based architecture [3]. The sorted syndromes are computed using Elementary Check Nodes.

This architecture dynamically generates a set of syndromes, keeping the complexity linear with  $d_c$ . The output of a forward layer in a FB decoder processes all input except the last one  $U_{dc}$ . The idea is to add an ECN to merge the forward layer with  $U_{dc}$ . The output of this Extended Forward (EF) generates a set of sorted syndromes  $S_b$ . The syndrome computation is equivalent to the equation

$$S_b = \boxplus_{i=1}^{d_c} U_i.$$

Note that  $S_b$  can also be computed in  $\log_2(d_c)$  layers of ECNs using a tree structure. To avoid loss of information, the Redundancy Elimination must not be done in ECN and is performed after de-correlation.

## Simulation results



Simulation results of NB-LDPC decoding algorithms for (576, 480) code over GF(64) under AWGN channel and BPSK modulation.

Extrinsic and intrinsic LLR messages are quantized on 6 bits and a posteriori LLRs on 8 bits. The maximum number of iterations is set to 10.

The red curve shows that Redundancy Elimination should not be done in ECN but after decorrelation.

## Conclusion

The presented solutions efficiently incorporate Elementary Check Nodes into a Syndrome-based architecture. Moreover the CN inputs are presorted leading to drastic simplifications of most of the ECNs. The resulting architecture benefits from flexibility and linear complexity of the Extended Forward solution and complexity reduction due to presorting.

- [1] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity decoding for non-binary ldpc codes in high order fields," *ICT 2010*  
[2] C. Marchand and E. Boutillon, "NB-LDPC check node with pre-sorted input," *ISTC 2016*  
[3] P. Schläfer, N. Wehn, M. Alles, T. Lehnigk-Emden, and E. Boutillon, "Syndrome based check node processing of high order NB-LDPC decoders," *ICT 2015*