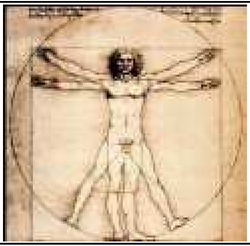


INFSCO-ICT-216203 DA VINCI

D4.5

"Report on Robust and Hardware Compliant Design of Non-Binary Protographs"

Contractual Date of Delivery to the CEC:	**31 12 2009** *(as specified in the contract)*
Actual Date of Delivery to the CEC:	**04 01 2010**
Author(s):	**David Declercq, Charly Poulliat and Emmanuel Boutillon**
Participant(s):	**ENSEA/UBS**
Workpackage:	***WP4 Advanced Channel Coding***
Estimated person months:	**24** *MM*
Security:	**PU** *(PU/PP/RE/CO)*
Nature:	**R** *(R/P/D/O)*
Version:	**1.0**
Total number of pages:	**16**

Abstract: This deliverable serves as technical annex of the final DAVINCI codes. We have developped in the DAVINCI project an improved design of Non-binary cycle codes compared to the existing state-of-the-art at the beginning of the project. Following ideas borrowed from existing designs for binary LDPC codes, we extended the construction of LDPC codes based on small well chosen protographs to the non binary case. The new codes that we propose are well suited for parallel hardware implementation, and show good robusteness to practical transmission constraints, like block-fading modes or rate-adaptibility via puncturing. Moreover, the proposed design lead to an improvement in the error floor region, due to a better minimum distance, while no SNR loss is witnessed in the convergence region.

The codes have been delivered in deliverable D4.4 at Mth 21, and this report explains the rationality and the new structure of the DAVINCI codes.

Keyword list: DA VINCI, deliverable, internal report.

Disclaimer:

Document history	Ver.	Date
Document created.	30/12/09	30/12/09

Report on Robust and Hardware Compliant Design of Non-Binary Protographs

David Declercq, Charly Poulliat and Emmanuel Boutillon

Abstract

This deliverable serves as technical annex of the final DAVINCI codes. We have developped in the DAVINCI project an improved design of Non-binary cycle codes compared to the existing state-of-the-art at the beginning of the project. Following ideas borrowed from existing designs for binary LDPC codes, we extended the construction of LDPC codes based on small well chosen protographs to the non binary case. The new codes that we propose are well suited for parallel hardware implementation, and show good robusteness to practical transmission constraints, like block-fading modes or rate-adaptibility via puncturing. Moreover, the proposed design lead to an improvement in the error floor region, due to a better minimum distance, while no SNR loss is witnessed in the convergence region.

The codes have been delivered in deliverable D4.4 at Mth 21, and this report explains the rationality and the new structure of the DAVINCI codes.

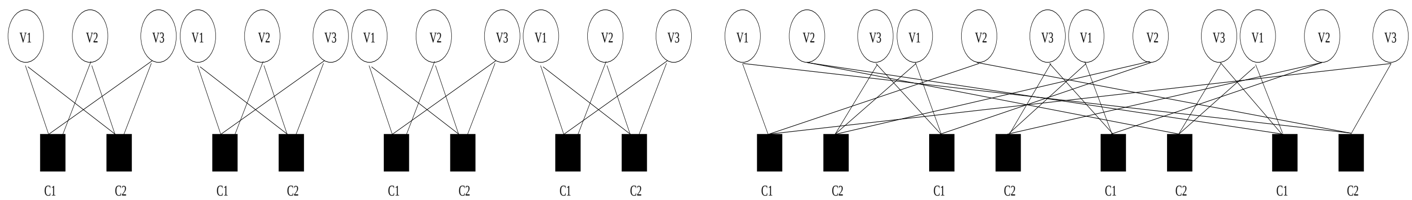
Contents

I. From Binary to Non-binary protograph codes

A. Binary case: definitions and notations

First introduced by [1], a binary protograph is defined as a small bipartite graph from which a larger graph is obtained by a so-called "copy-and-permute" procedure as described in figure 1. The procedure can be summarized as follows:

- *"Copy step":* first, the protograph is copied L times to obtain L replicas. L is arbitrary defined to achieve the targeted codeword length.
- *"Permutation step":* then, the edges of the individual replicas are permuted among the replicas to obtain a single but larger graph, but under some restrictions. Indeed, the edge permutations cannot be arbitrary due to the inherent structure of the protograph: the nodes of the protograph are labelled so that if variable node V_j is connected to check node C_i in the protograph, then variable node V_j in a replica can only connect to one of the L replicated C_i check nodes.



(a) Copy step (b) Permute step

Figure 1. "Copy-and-Permute" procedure applied to a rate 1/3 protograph

Finally, the permuted edge connections specify the non-zeros entries of the parity-check matrix associated with the resulting graph. The protograph itself is generally described using its *adjacency matrix* H_B also called base matrix [2], [3], where the coefficients $H_B(i,j)$ represent the number of edges between the $i-th$ check node C_i of the protograph and the $j-th$ variable node V_j. Note that using this representation enables to consider parallel edges, i.e. two nodes (a variable node and a check node) can be connected with more than one edge. These parallel edges however must be eliminated when building the larger graph using the copy-and-permute procedure to yield to a suitable representation of the code using a parity-check matrix.

In fact, the code ensemble defined by the protograph can be viewed as a structured sub-ensemble of the Low-Density Parity-Check (LDPC) code ensemble defined using the equivalent edge distributions $(\lambda(x), \rho(x))$ that can be also associated with the protograph. As an example, we will consider the regular $(2,4)$ - LDPC code ensemble (Variable nodes with degree 2 and check nodes with degree four), whose associated edge distributions are defined by $(\lambda(x) = x, \rho(x) = x^3)$. The following adjacency matrix is associated with a particular structured sub-ensemble of the regular $(2,4)$ - LDPC code ensemble

$$H_{B_1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{1}$$

The associated protograph is given by the following bipartite graph:

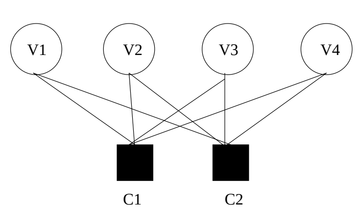


Figure 2. A protograph for a rate 1/2 (2,4)-LDPC code

In the binary case, the adjacency matrix itself is issued from some previous optimization procedures based on density evolution [1], [3] or on a multidimensional EXIT chart analysis [2].

B. Extension to the non-binary case

The definition of a non binary protograph results in the direct generalization of the binary case: a protograph is defined as a small bipartite graph from which a larger graph is obtained by a "copy-and-permute" procedure. The main difference relies on the fact that the constraint nodes (ie. parity check nodes) are defined over the Galois field GF(q). These constraints are also supposed to be randomly chosen. This is equivalent to consider that the protograph describes a topological structure whose edges are carrying randomly chosen non-binary values. So, the procedure to build a parity-check matrix associated with a non-binary protograph can be summarized as follows:

- *"Copy step":* first, the protograph is copied L times to obtain L replicas. L is arbitrary defined to achieve the targeted codeword length.
- *"Permutation step":* then, the edges of the individual replicas are permuted among the replicas to obtain a single but larger graph, but under some restrictions as in the binary case. In this step, only the topological structure is considered and so edges are connected according to the topological structure of the protograph.
- *"Edge assignment step":* Finally, non binary values are assigned to each and every edge. A priori, the assignment should be done randomly according to the previous definition, but one can also consider specific assignments if it is required (as done for Quasi-cyclic NB codes).

C. From a base matrix to the parity-check matrix

Once the protograph (or equivalently the base matrix) has been selected, one aims at building a larger graph using the copy-and permute procedure. The remaining question is how to select the constrained permutations among the different replicas of the protograph.

- *PEG-like algorithm* The first method is to consider random looking permutations by assigning edges using pseudo-random permutations. This can be done effectively by using an instance of the PEG algorithm [17] that aims at maximizing the girth of overall graph but with some additional constraints due to the protograh structures. Then, the coefficients are randomly assigned for a NB codes or can be optimized as done in [11].
- *Lifting using circulants and quasi-cyclic graphs* To reduce the descriptational complexity induced by the description (ie. storage) of all permutations used to build the overall graph, we can prefer to use permutations that can be efficiently described algorithmically. To this end, it is commonly convenient to choose circulant permutations: the parity-check matrix H is finally described an $M \times N$ array of $L \times L$ (weight-one) circulant permutation matrices (some of which may be the $L \times L$ zero matrix). The assignment of the circulants however should be done carefully to avoid short cycles and it can be done using an instance of the PEG sometimes referred to circulant PEG.

  Definition

  Lifted graphs: the operation which consists in replacing each non-zero entry with value d in the base-matrix by a set of d non-overlapping circulant matrices is called "lifting". The Order of the lift L corresponds to the size of the circulant matrices. ■

  In the binary case, when H is an array of circulants (ie. an array of circulant permutation matrices), the LDPC code will be quasi-cyclic, enabling easy encoding procedures as well as parallel hardware architecture design for the decoder. Note that in most cases, multiple successive liftings can be considered to enhance properties of the final graph.

  In the non-binary case, due to the size of the targeted final graph, it will be difficult to consider multiple lifts of the base matrix, therefore using circulant lift should limit the achievable girth of the final graph. More importantly, contrary to the binary case, the LDPC code will not be quasi-cyclic if the assignment (random or optimized) of the coefficients is considered independently for each and every edge in the lifted graph: indeed, as explained below, additional constraints have to be considered for the permutations to ensure quasi-cyclic property for the overall code.

- *quasi-cyclic NB protographs* To ensure the quasi-cyclic property, some additional constraints are required: we have to consider q-ary circulant permutations obtained using the circulant permutation matrices as used in [4], [5] or [6]. Thus, the coefficients assigned to a permutation matrix are not chosen independently. This additional constraint results in a more limited choice of the non zero values during the finite length optimization, leading to codes that can have lower minimum distance. This is mainly why we consider in the following optimized non-binary protograph-based codes with quasi-cyclic graphs that are not non-binary quasi-cyclic codes.

II. Code design

In this section, we describe how the codes for the DAVINCI project have been designed. The core of the design method is the design of a Tanner graph with good properties and then, the optimization of non-zero values that label the edges of the non-binary Tanner graph. Those two steps are explained in details in the following.

A. Different types of protographs

For the design of protographs for irregular binary LDPC codes, the criterion used to optimize the structure of the protograph is the minimization of the decoding threshold of the ensemble, with maximum degree constraint on the bitnode connectivity. For ultra-sparse $d_v = 2$ non-binary LDPC codes, this approach cannot be applied. First, $d_v = 2$ codes are not stable, and therefore using the decoding threshold as a mesure of performance does not make sense. Additionnaly, in the DAVINCI project, we aim at designing codes which have good performance for small to moderate lengths only, and finite length optimization constraints are more suited to our needs.
For these reasons, we will use the following optimization criterion for the protograph based code design.

- First choose a very small mother code filled with integers, the sum of integers represents the degree of each row/column.
- replace each integer by the same number of non-equal circulant matrix (shifted diagonal) with size L. Choose the circulants such that the girth is maximized. We obtain a very structured graph with number of symbol nodes $N_s = d_c.L$
- Choose the non-zeros values assigned to the edges of the Tanner graph with a two steps optimization algorithm (described in section II-B).

Step 1: choice of protograph
In step 1, we have to choose a proper protograph. Two solutions haver been investigated in the DAVINCI project. In figure 3, we give the two types of protographs that have been investigated. We have drawn the cases of target rates $R = 1/2$, $R = 2/3$ and $R = 3/4$. Type-I protograph contain only ”1”, while type-II protographs contain two columns with a ”2” and a ”0”. While type-I protograph are easier to study and to characterize [4], it can be shown that type-I protograph can have only girth multiple of 4 after the lifting operation. Since we want to avoid girth-4 codes, using type-I protographs leaves us with codes with girth equal to $g = 8$ or $g = 12$ only. This is a too strong limitation for the design of NB-LDPC codes with $d_v = 2$. For these codes, the minimum distance is directly related to the girth of the graph, regardeless of the choice of the non-zero values [11]. Therefore an increase of 2 in the girth of a given graph leads to a great performance improvement, especially in the error floor region. Type-I protographs are then not well suited to the design of $d_v = 2$ NB-LDPC codes.

On the contrary, type-II protographs do not have such limitations, and girths $g = 6$, $g = 10$ and $g = 14$ are reachable by an efficient lifting algorithm. Type-II protograph have another advantage due to the first 2 columns depicted in figure 3. Indeed, during the lifting operation, one can choose particular values of the circulants for the values ”2” without decreasing the girth. In particular, one can fix the 2 circulants in these first 2 columns to be the main diagonal and the first upper diagonal, which implements the zig-zag structure of practical LDPC codes proposed in standarts. The zig-zag structure ensures a linear time encoding with minimal effort at the transmitter, and using only the information of the parity check matrix, therefore preventing the storage of a generator matrix. In the DAVINCI project, we used this strategy and fixed the zig-zag structure as a constraint for the code design.

Step 2: optimized lifting
For $d_v = 2$ NB-LDPC codes, the Tanner graph should have the best possible topological properties in

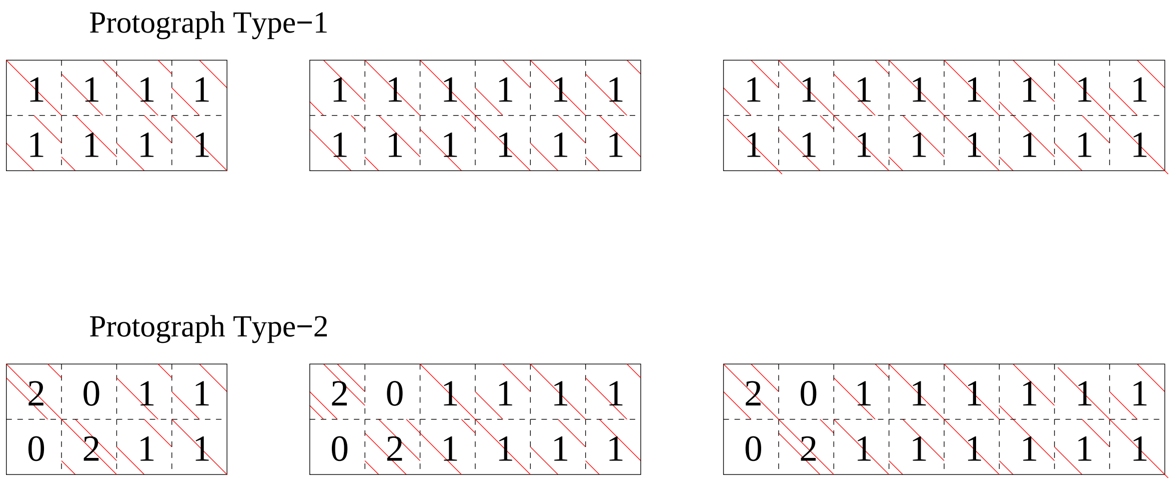


Figure 3. Simplest regular protograph with linear encoding constraints

terms of cycles. That is one should aim at the maximum possible girth, and minimum multiplicity of the number of cycles with minimum length. By adapting the PEG algorithm [17], one could easily take into account the multiplicity of the girth, together with a better local selection of the created edges as explained in [18]. In the DAVINCI project, we have adapted the algorithm proposed in [18] in order to perform the lifting operation. This will be called in the following *Lift-PEG* algorithm. It can be briefly described like this:

Lift-PEG algorithm

For each non-zero entry in the protograph from colum 1 to column d_c, choose the circulant which maximizes the 'local' girth of the graph and minimizes the number of small cycles created. The computation of the 'local' girth and multiplicity is done with the help of the computationnal tree. ■

The principle of the PEG algorithm is kept, but applied block by block, for each $L \times L$ circulant. Note also that we applied all the optimization tricks of the algorithm presented in [18] for the construction of the DAVINCI codes.

Lifting type-II protographs seemed however not optimal for various cases of codeword lengths and coderates. We have therefore proposed in the DAVINCI project to proceed with a ”double-lifting” method. This procedure consists in the following two steps:

- First expand the small type-II protograph with a small expansion L_1 to get a larger type-I protograph,
- Expand the type-I protograph with expansion L_2 such that $L_1 \times L_2 = L$.

This ”double-lifting” procedure gives an extra degree of freedom in the code design, while keeping the good property that all girths $g = 6$, $g = 8$, $g = 10$, $g = 12$, $g = 14$ and $g = 16$ can be reached. We found that using this strategy gave better graphs for various cases, especially rate $R = 1/2$ codes, which are the more important cases for wireless transmissions. The technique of double-lifting is illustrated on figure 4.

For all the codeword lengths from $N_s = 48$ to $N_s = 480$ with a step of 24 symbols, and for all considered coderates $R \in \{1/2, 2/3, 3/4, 5/6\}$, we have considered all the possible constructions and kept the best code in each case. The codes obtained with this approach form the set of final DAVINCI codes, which combine good error correction performance, and compliance with hardware implementation, both in terms of memory storage and level of parallelism (see section III-A). The parameters for graph construction are (L_1, L_2) lift orders from a type-II protograph. In all cases, the Lift-PEG algorithm is emloyed. If $L_1 = 1$ then single-lifting is considered with $L = L_2$, and otherwise, double-lifting is considered with

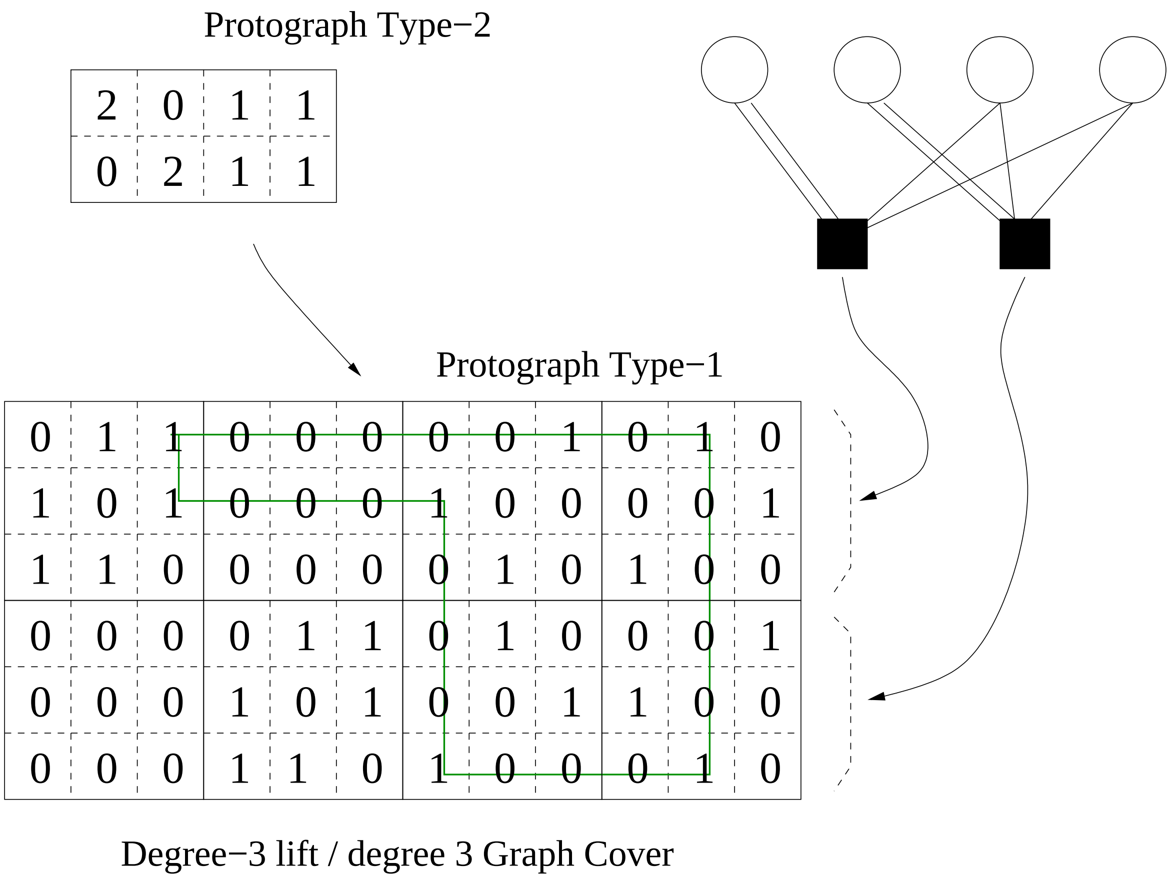


Figure 4. Method to obtain a type-I large protograph from a type-II small protograph. The large type-I protograph is then used with a second extension to get the final graph.

$L = L_1 \times L_2$. We have reported in table I the obtained girths for various cases of DAVINCI parameters. All girths reported here are either equal or better than the state-of-the-art codes that have been provided to the consortium at the beginning of the project. This means in particular that we expect no performance loss compared to the original codes, despite the constraints that have been introduced in the code design to simplify the hardware implementation of both the encoder and the decoder.

	$d_c = 4$	$d_c = 6$	$d_c = 8$	$d_c = 12$
$N_s = 120$	(1,30) g=12	(1,20) g=10	(1,15)g=8	(1,10) g=6
$N_s = 240$	(4,15) g=16	(1,40) g=10	(1,30) g=8	(1,20) g=8
$N_s = 360$	(3,30) g=16	(1,60) g=12	(1,45) g=10	(1,30) g=8
$N_s = 480$	(3,40) g=16	(1,80) g=12	(1,60) g=10	(1,40) g=8

Table I

OBTAINED GIRTH AFTER OPTIMIZED LIFTING. IN THE TABLE IS INDICATED THE CHOSEN LIFT ORDERS (L_1, L_2) TO REACH THE GIVEN GIRTH. IN RED ARE INDICATED THE GIRTHS WHICH COULD NOT BE REACHED WITH A REGULAR TYPE-I PROTOGRAPH.

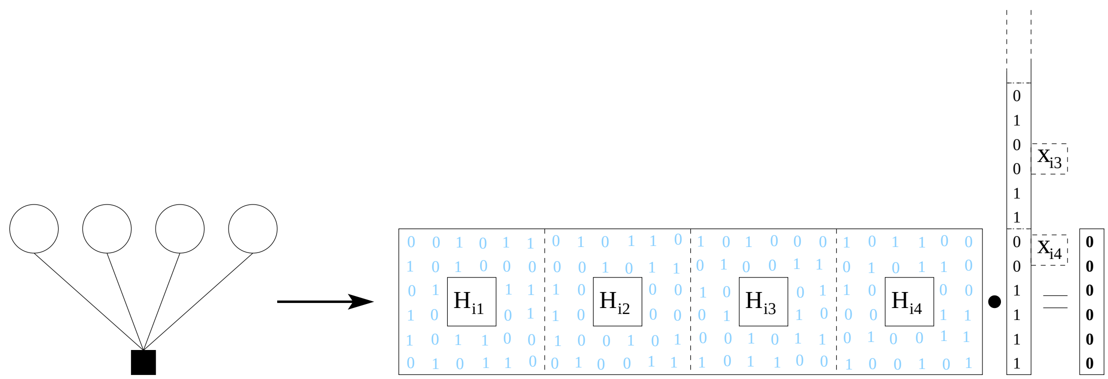
B. A new algorithm to chose non-zero values

For codes defined over GF(q), when addressing finite length design, it has been shown in [20] and [11] that selecting carefully the non binary entries of the parity-check matrix can improve the overall performance of the code when compared to randomly chosen coefficients. The selection of the non zero values can impact both on the waterfall and the on error floor. The observed performance gains are dependent of both the field order and the code rate.

In the waterfall region, selecting the edges label row-wise is critical. It is shown in [11] that “best” rows are selected according to their equivalent binary minimum distance and multiplicity of the minimum distance. The Binary Component Code of a non-binary parity check is build from the tranpose of the companion matrices of the non-zero values composing the parity check. Using binary matrix images for the non-zero values of the check and binary vector images for the codeword symbols, one get the following parity-check equation in a vector form:

$$\sum_{j=1}^{d_c} H_{ij}.\underline{x}_j = \underline{0}_p \qquad \text{in } (\mathbb{F}_2)^p$$

The binary image is explained in the following picture:



In this deliverable, and as a result of reserach in DAVINCI, we propose a new criterion selection for the non-zero values of a single parity check. In existing litterature (see [11] and references within), it is adviced to maximize the strength of the component code, and then choose the field values such that the binary image has the maximum “minimum Hamming distance” ($Dmin$), together with the minimum multiplicity of $Dmin$. Although locally optimal, this strategy is not optimal when used in a message passing iterative decoder. A better strategy, which is especially performant when the code is a regular cycle-code, is to optimize the balance between sub-codes of the component code.

Let us describe here this new idea. Since the code will propagate d_c extrinsic messages from the incomming message at each iteration, it is better to build extrinsic messages which statistically behave equally. In other words, the extrinsic messages should have its quantity of mutual information as close as possible to their average, since it is obvious that increasing the mutual information of one particular extrinsic output will result in decreasing the other mutual information, therefore propagating worse messages to the rest of the graph. In order to prove this statement, we need to measure non-binary EXIT charts. However, non-binary EXIT charts are not known for other types of channels than the non-binary erasure channel, and then we leave the proof of our statement to future developments.

New Optimization criterion for component code selection

Let a non-binary parity check of degree d_c *with non-zero values* $\{h_1 \dots h_{d_c}\}$. *Consider the* d_c *binary subcodes* $\mathcal{S}_{cc}(k)$ *formed from the combination of the* $d_c - 1$ *values in* $\{h_1 \dots h_{d_c}\}$ *except* h_k. *We choose for* h_k *the field values in* $GF(q)$ *such that:*

$$\min_{h_k}\left(\sum_k Dmin(\mathcal{S}_{cc}(k))\right)$$

$$\textit{constrained to } \min_k (Dmin(\mathcal{S}_{cc}(k)) > \alpha$$

■

This criterion ensures that both the component code and all the sub-codes of the components codes have good and close error correction capability. This new optimization criterion is indeed interesting since we saw slight improvement in the waterfall region compared to codes that use existing sets of non-zero values. Altogether, these optimized non-zeros values compensate the performance loss in the waterfall that is due to the hardware constraints that have been introduced. Moreover, the new sets of values do not reduce the global minimum distance of the codes, as shown in the sequel of this deliverable. We give below the best sets of coefficients that have been optimized with the new criterion, and that we used for the DAVINCI code design.

- best rows for $d_c = 4$

$$(\alpha^0, \alpha^9, \alpha^{26}, \alpha^{46}) - (\alpha^0, \alpha^{17}, \alpha^{26}, \alpha^{43})$$
$$(\alpha^0, \alpha^{17}, \alpha^{37}, \alpha^{54}) - (\alpha^0, \alpha^{20}, \alpha^{37}, \alpha^{46})$$

- best rows for $d_c = 6$

$$(\alpha^0, \alpha^7, \alpha^{18}, \alpha^{26}, \alpha^{44}, \alpha^{52}) - (\alpha^0, \alpha^8, \alpha^{19}, \alpha^{26}, \alpha^{37}, \alpha^{45})$$
$$(\alpha^0, \alpha^{11}, \alpha^{18}, \alpha^{29}, \alpha^{37}, \alpha^{55}) - (\alpha^0, \alpha^{18}, \alpha^{26}, \alpha^{37}, \alpha^{44}, \alpha^{55})$$

- best rows for $d_c = 8$

$$(\alpha^0, \alpha^7, \alpha^{13}, \alpha^{19}, \alpha^{28}, \alpha^{37}, \alpha^{45}, \alpha^{54})$$

- best rows for $d_c = 12$

$$(\alpha^0, \alpha^6, \alpha^{12}, \alpha^{18}, \alpha^{24}, \alpha^{30}, \alpha^{36}, \alpha^{42}, \alpha^{46}, \alpha^{52}, \alpha^{53}, \alpha^{59})$$

C. Global Optimization

The optimization consists in iteratively selecting the randomly permuted rows and optimize the local minimum distance on the small topologies like cycles or stopping sets, which are build from combination of cycles. Permutation of rows mean any ordering of the sets that have been a priori chosen (cf. preceeding section). For graphs with $d_v = 2$, the mentioned topological structures are closed structures which are isolated. Consequently, the distance spectrum of those isolated structures contribute directly to the low weight spectrum of the overall code. On figure 5, we have sketched on figure 5 the cases of a length-6 cycle and a stopping set composed of 2 imbricated length-6 cycles.

The optimisation procedure for the global nonzero values assignement follows:

- Build a cycle Tanner graph with $d_v = 2$ with maximum girth g and minimum number of short cycles.
- List all Cycles of length $\in \{g/2, g/2+1, \ldots, g/2+\gamma_1\}$ and put them in the set $\mathcal{S}_c$. List all Stopping Sets of length $\in \{\lfloor 3g/4 \rfloor, \lfloor 3g/4 \rfloor + 1, \ldots, \lfloor 3g/4 \rfloor + \gamma_2\}$ and put them in the set $\mathcal{S}_{ss}$.
- Consider a single set of d_c Field values as component code $\left[\alpha^1, \ldots, \alpha^{d_c}\right]$
- *Random walk* Optimization (one iteration):
  for each and every row: replace $\left[\alpha^1, \ldots, \alpha^{d_c}\right]$ by $\left[\alpha^{\pi_1}, \ldots, \alpha^{\pi_{d_c}}\right]$ *iff*

$$Cost\left([\alpha^{\pi_1}, \ldots, \alpha^{\pi_{d_c}}]\right) < Cost\left(\left[\alpha^1, \ldots, \alpha^{d_c}\right]\right)$$

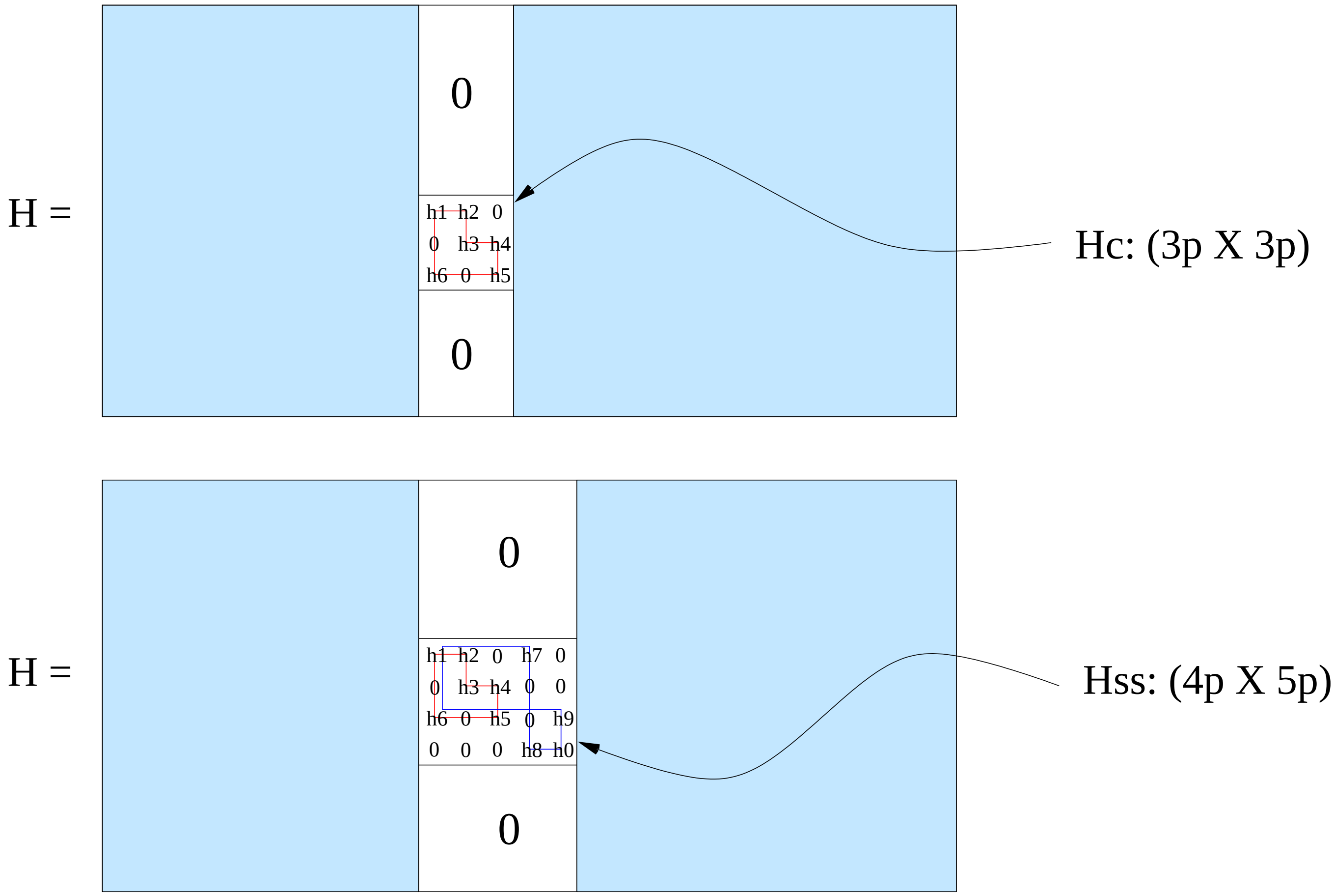


Figure 5. Binary image of a 6-cycle and a stopping set for a $d_v = 2$ cycle code. The stopping set structure is composed of two imbricated 6-cycles and a length-8 cycle. Both structures are isolated and define sub-codes, which distance spectra are closely related to the overall distance spectrum of the code. This figure considers the case of square $(p \times p)$ clusters.

where $Cost(.)$ is a weighted objective function of the low-weight spectrum computed locally on $\mathcal{S}_c \cup \mathcal{S}_{ss}$. The objective function could vary from one code to another, but in principle $Cost(.)$ decreases if the low-weight spectrum improves, aiming at an iterative improvement of the global minimum distance. Note that this optimization strategy differs from the one that has been proposed in [11] and has been found to have a better global convergence to an optimum.

III. Properties of the Optimized codes

A. Well adapted to parallel hardware decoding

The proposed LDPC matrix structure is similar to the structure used in most binary LDPC standard (DVB-S2, DVB-T2, WiMax,...). The motivation to use such a structure is double. First, it is sufficently regular to allow a carefull code design optimization without an exponential number of configuration to explore, second, but not least, it allows an hardware architecture with a maximum degree L of parallelism without any memory conflict.

This organisation allows also different type of scheduling for the check node processor. It can be either a parallel scheduling, as the solution proposed by UBS in D6.2 or a serial check node, as proposed by ST microelectronics in D6.3. Figure 6 shows two type of conflict free memory organization for a serial and a parallel check node scheduling (note that, for shake of clarity, the Variable Node processors are not shown).

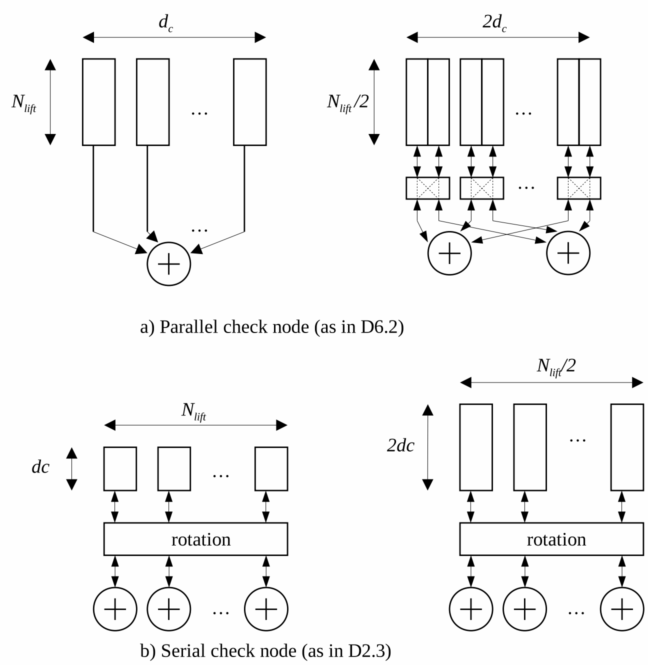


Figure 6. Different types of parallelism for type-I code.

The use of type-II code complexifies the hardware desing. Nevetheless, they should be considereted because of there higher decoding performance. Type-II code contains double diagonal, as for the binary LDPC used in DVB-S2, DVB-T2 standards. For the binary case, there is many publication dealing of this problem, see for example [25], [26], [27], [28], [29], [30], [31]. The situation for the proposed non

	$d_c = 4$	$d_c = 6$	$d_c = 8$	$d_c = 12$
$ceil(log_2(dc)$	2	3	3	4
Reduction	66.7%	50%	50%	33.3%

Table II
MEMORY REDUCTION TO STORED THE NONZERO VALUE USING A REDUCED VALUE SET AS A FUNCTION OF dc

binary type-II code is more simple. In fact, since the degree of variable node $dv = 2$, a double diagonal is involved only once in the parity check matrix.

The second advantage of the proposed codes in terms of hardware simplification is due to the specific choice of nonzero values. Having a limited number of nonzero values offers an hardware advantage in terms of memory. In fact, instead of storing a symbol GF(64) per edge of a partity check (i.e. 6 bits), it is sufficient to store the index of the corresponding symbol (i.e. $log_2(dc)$ bits). Table II give the memory reduction according to dc.

B. Minimum distance properties

We have reported on tables III to VI the minimum distance properties of new DAVINCI codes. The spectra have been estimated with the improved impulse method presented in [24]. These results need to be compared to the minimum distance of the benchmark codes given in deliverable D4.1. As expected, the minimum distances are almost always similar, with a slight improvement in terms of multiplicity for the code design. This is not a great improvement, but at least, we have shown that no error correction performance is lost from using strong constraints for hardware implementation. Additionnaly, this lower multiplicity of low-weight codewords will have an impact on the error floor and the probability of undetected errors.

Code Rate (d_v, d_c)	N_s coded Symbols	Girth(#mult)	MinDist(#mult)
$R = \frac{1}{2}$	48	g5(48)	17(23)
$(2, 4)$	72	g6(192)	19(16)
	96	g6(96)	20(2)
	144	g6(36)	21(4)
	192	g7(128)	22(3)
	240	g8(1275)	24(53)
	288	g8(984)	24(9)
	384	g8(832)	24(5)
	480	g8(500)	26(31)

Table III
MINIMUM DISTANCE PROPERTIES OF DESIGNED NB-LDPC CODES, RATE $R = 1/2$ CODES

Code Rate (d_v, d_c)	N_s coded Symbols	Girth(#mult)	MinDist(#mult)
$R = \frac{2}{3}$ $(2, 6)$	48	g4(172)	10(2)
	72	g4(117)	11(6)
	96	g4(60)	12(12)
	144	g4(48)	12(5)
	192	g5(320)	14(5)
	240	g5(128)	14(2)
	288	g6(3280)	14(1)
	384	g6(2240)	15(3)
	480	g6(1840)	15(7)

Table IV
MINIMUM DISTANCE PROPERTIES OF DESIGNED NB-LDPC CODES, RATE $R = 2/3$ CODES

Code Rate (d_v, d_c)	N_s coded Symbols	Girth(#mult)	MinDist(#mult)
$R = \frac{3}{4}$			
$(2, 8)$	72	g3(54)	9(59)
	96	g4(552)	10(36)
	144	g4(369)	10(6)
	192	g4(240)	11(4)
	240	g4(120)	11(5)
	288	g4(144)	12(36)
	384	g5(2064)	12(11)
	480	g5(1440)	12(9)

Table V
MINIMUM DISTANCE PROPERTIES OF DESIGNED NB-LDPC CODES, RATE $R = 3/4$ CODES

IV. CONCLUSION

To conclude this deliverable, we recall the main achievements that have been made during the DAVINCI project with respect to practical code design of non-binary LDPC codes.

We have considered the following constraints that aim at simplifying the encoder/decoder architecture, and then which render the codes *"practical"*:

- Quasi-cyclic protograph based: allows full parallelism of the decoder, and then improves latency,
- Consider double-diagonal or zig-zag structure for the first part of the code: allows simple linear encoding,
- Consider a limited number of non-zeros values: limits the storage requirements.

The new and original developments compared to existing work are the following:

- *double-lifting graph design with PEG-lifting algorithm.* Although this method is highly inspired from the litterature, this combination of different techniques has not been used to the best of our knowledge,

Code Rate (d_v, d_c)	N_s coded Symbols	Girth(#mult)	MinDist(#mult)
$R = \frac{5}{6}$ $(2, 12)$			
	96	g3(272)	6(4)
	144	g3(192)	7(5)
	192	g4(3760)	8(14)
	240	g4(2940)	9(233)
	288	g4(2916)	9(137)
	384	g4(2184)	9(45)
	480	g4(1800)	9(13)

Table VI
MINIMUM DISTANCE PROPERTIES OF DESIGNED NB-LDPC CODES, RATE $R = 5/6$ CODES

- *new criterion for the choice of non-zeros values sets*. This is the main original improvement in our opinion. Plans are made to publish these results in 2010,
- *new global constraint and global optimization algorithm*. This is not very original, but we have designed a global optimization algorithm with better global convergence.

As a result, the advantages of this new code design are:

- The constraints that have been added combined with the new developments allowed to design codes which could be integrated easily in chips for standart application, *without* sacrifying performance in the waterfall region or in the error floor region (showed buy the minimum distance properties).
- Robustness to puncturing: The quasi-cyclic structure of our design allows to support *regular* puncturing, both bit-wise and symbol-wise, while preserving good correction performance. This will be demonstrated in deliverable D4.7.

REFERENCES

[1] J. Thorpe, “Low-Density Parity-Check (LDPC) Codes Constructed from Protographs,” *JPL INP, Tech. Rep.*, Aug. 2003.

[2] Gianluigi Liva, Shumei Song, Lan Lan, Yifei Zhang, Shu Lin, William Ryan, “Design of LDPC Codes: A Survey and New Results,” *Journal of Communications Software and Systems*, 2006.

[3] T. Richardson and R. Urbanke, “Modern Coding Theory”, *Cambridge University Press*, 2008.

[4] S. Lin, S. Song, L. Lan, L. Zeng, and Y. Y. Tai, “Constructions of nonbinary quasi-cyclic LDPC codes: a finite field approach,” in Proc. Information Theory and Applications (ITA) Workshop, UCSD, 2006.

[5] Rong-Hui Peng and Rong-Rong Chen,“Design of Nonbinary Quasi-Cyclic LDPC Cycle Codes”, Information Theory Workshop, Tahoe City, CA,, USA, Sept. 2007.

[6] Sung-Eun Park, Chiwoo Lim, Thierry Lestable, Jaeyoel Kim, Kyeongcheol Yang, “A Class of Structured LDPC Codes Over GF(q) for Efficient Encoding,” in the proc. VTC Spring, pp. 2218–2222, 2007.

[7] V. Rathi and R. Urbanke, “Density evolution, thresholds and the stability condition for non-binary LDPC codes,” *IEE Proceedings-Communications,* Vol. 152, No. 6, pp. 1069–1074, Dec. 2005.

[8] A. Goupil, M. Colas, G. Gelle and D. Declercq, “FFT-based BP Decoding of General LDPC Codes over Abelian Groups,” *IEEE Trans. on Commun.*, vol. 55, no. 4, pp. 644–649, April 2007.

[9] L. Sassatelli and D. Declercq, “Non-binary Hybrid LDPC Codes: structure, decoding and optimization”, in the proc. of ITW’06, Chengdu, China, October 2006.

[10] L. Sassatelli and D. Declercq, “Analysis of Non-binary Hybrid LDPC Codes”, in the proc. of ISIT’07, Nice, France, June 2007.

[11] C. Poulliat, M. Fossorier and D. Declercq, “Design of regular $(2, d_c)$-LDPC codes over GF(q) using their binary images”, *IEEE Trans. on Commun.*, vol. 56, no. 10, pp. 1626–1635, October 2008.

[12] A. Voicila, D. Declercq, F. Verdier M. Fossorier and P. Urard, ”Split Non-binary LDPC Codes”, in the proc. of ISIT’08, Toronto, Canada, July 2008.

[13] W. Chen, C. Poulliat, D. Declercq, L. Conde-Canencia, A. Al-Ghouwayel and E. Boutillon, ”Non-Binary LDPC Codes defined over the General Linear Group: Finite Length Design and Practical Implementation Issues”, in the proc. of VTC’09 (Special Session FP7-ICT-RAS-Cluster), Barcelona, Spain, April 2009.

[14] A. Voicila, D. Declercq, F. Verdier, M. Fossorier and P. Urard, ”Low-Complexity Decoding for non-binary LDPC Codes in High Order Fields”, accepted in IEEE Trans. Commun., 2009

[15] L. Barnault and D. Declercq, “Fast Decoding Algorithm for LDPC over GF(2^q)”, in the proc. of ITW’03, Paris, France, 2003.

[16] X.Y. Hu and E. Eleftheriou, “Binary Representation of Cycle Tanner-Graph GF(2^q) codes”,*EEE Int. Conf. on Commun.*, Paris, France, June 2004.

[17] X.-Y. Hu, E. Eleftheriou and D.M. Arnold, “Regular and Irregular Progressive Edge-Growth Tanner Graphs“, IEEE Trans. on Inf. Theory, Vol. 51, No. 1, pp. 386–398, January 2005.

[18] A. Venkiah, D. Declercq and C. Poulliat, “Design of Cages with a Randomized Progressive Edge Growth Algorithm”, *IEEE Commun. Letters*, vol. 12(4), pp. 301-303, April 2008.

[19] T. Verhoeff, “An updated table of minimum-distance bounds for binary linear codes,” *IEEE Trans. Inform. Theory*, vol. IT-33, no. 5, pp. 665-680, Sept. 1987.

[20] M.C. Davey and D. MacKay, “Low-Density Parity-Check Codes over GF(q)”, *IEEE Commun. letters*, vol. 2, pp. 165-167, June 1998.

[21] S. Pfletschinger, A. Mourad, E. Lopez, D. Declercq and G. Bacci, “Performance Evaluation of Non-Binary LDPC Codes on Wireless Channels”, in the proc. of ICT Mobile Summit, Santander, Spain, June 2009.

[22] J.J. Boutros, O. Pothier and G. Zémor, “Generalized low density (Tanner) codes”, IEEE International Conference on Communications, vol. 1, pp. 441-445, Vancouver, Canada, June 1999.

[23] D. Declercq and M. Fossorier, “Decoding Algorithms for Nonbinary LDPC Codes over GF(q)”, IEEE Trans. on Commun., vol. 55(4), pp. 633-643, April 2007.

[24] D. Declercq and M. Fossorier, “Improved Impulse Method to Evaluate the Low Weight Profile of Sparse Binary Linear Codes”, in the proc. of ISIT’08, Toronto, Canada, July 2008.

[25] C. Marchand, J.-B. Dor?, L. Conde-Canencia, E. Boutillon, ”Conflict Resolution by Matrix Reordering for DVB-T2 LDPC Decoders”, Globecom 2009, Haiwai, Dec. 2009.

[26] Y. Sun, M. Karkooti, and J. Cavallaro, ?High throughput, parallel, scalable LDPC encoder/decoder architecture for OFDM systems,? in Design, Applications, Integration and Software, 2006 IEEE Dallas/CAS Workshop on, (Richarson, USA), pp. 39?42, Oct. 2006.

[27] F. Kienle, T. Brack, and N. Wehn, ?A synthesizable IP core for DVB-S2 LDPC code decoding,? in DATE ?05: Proceedings of the conference on Design, Automation and Test in Europe, (Munich, Germany), pp. 100? 105, IEEE Computer Society, Mar. 2005.

[28] M. Gomes, G. Falcao, V. Silva, V. Ferreira, A. Sengo, and M. Falcao, ?Flexible parallel architecture for DVB-S2 LDPC decoders,? in Global Telecommunications Conference, 2007. GLOBECOM ?07. IEEE, (Washington, USA), pp. 3265?3269, Nov. 2007.

[29] J. Dielissen, A. Hekstra, and V. Berg, ?Low cost LDPC decoder for DVB-S2,? in Design, Automation and Test in Europe, 2006. DATE ?06. Proceedings, vol. 2, (Munich, Germany), pp. 1?6, Mar. 2006.

[30] M. Rovini, G. Gentile, F. Rossi, and L. Fanucci, ?A minimum-latency block-serial architecture of a decoder for IEEE 802.11n LDPC codes,? in Very Large Scale Integration, 2007. VLSI - SoC 2007. IFIP International Conference on, (Atlanta, USA), pp. 236?241, Oct. 2007.

[31] T. Bhatt, V. Sundaramurthy, V. Stolpman, and D. McCain, ?Pipelined block-serial decoder architecture for structured LDPC codes,? in Acous- tics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, vol. 4, (Toulouse, France), p. IV, May 2006.