FPGA Implementation of Advanced FEC Schemes for Intelligent Aggregation Networks

Ding Zou* and Ivan B. Djordjevic
University of Arizona, Department of Electrical and Computer Engineering
1230 E. Speedway Blvd., Tucson, AZ 85719, USA

ABSTRACT

In state-of-the-art fiber-optics communication systems the fixed forward error correction (FEC) and constellation size are employed. While it is important to closely approach the Shannon limit by using turbo product codes (TPC) and low-density parity-check (LDPC) codes with soft-decision decoding (SDD) algorithm; rate-adaptive techniques, which enable increased information rates over short links and reliable transmission over long links, are likely to become more important with ever-increasing network traffic demands. In this invited paper, we describe a rate adaptive non-binary LDPC coding technique, and demonstrate its flexibility and good performance exhibiting no error floor at BER down to 10^{-15} in entire code rate range, by FPGA-based emulation, making it a viable solution in the next-generation high-speed intelligent aggregation networks.

Keywords: Fiber-optics communications, low-density parity-check (LDPC) codes, adaptive coding, FPGA implementation, intelligent aggregation networks

1. INTRODUCTION

Software defined network (SDN) is currently emerging as a key technology to enable the next generation of optical transport networks and access networks supporting ever increasing traffic demands due to its dynamic, manageable, and cost effective nature as well as adaptability through abstraction of high-level functionalities [1]. In order to meet those requirements, such as agility and programmable configurability; physical layer demands dynamic wavelength, bandwidth allocation, and rate adaptive forward error correction (FEC) codes with high flexibility and unified architecture. In the past decades, a number of FECs have been intensively studied and extensively investigated in many communication systems such as space communication links, digital subscriber lines, as well as wireless systems. To be specific, Reed Solomon (RS) codes, concatenated codes, product codes, and low-density parity-check (LDPC) codes, are recommended by various ITU-T standards [2-6]. They are different in terms of transmission overhead (redundancy), implementation complexity, net coding gain (NCG), and burst error correction capability, to mention few.

In this invited paper, we describe and compare various LDPC coding schemes, including binary LDPC codes and non-binary LDPC codes, as well as corresponding rate adaptation techniques for both cases. By computer simulation, we first optimize the bit-width of input log-likelihood ratios (LLRs), check-to-variable messages, and variable-to-check messages. Then we show via FPGA emulation that the large-girth LDPC coding schemes can achieve large net coding gain (NCG) exhibiting no error floor at BERs down to 10^{-15}, when layered decoding algorithm is employed. Meanwhile, we demonstrate a flexible rate adaptation technique with the proposed unified decoder architecture. Finally, by comparing the implementation complexity of RS codes and concatenated codes, we conclude that the proposed flexible LDPC coding schemes with reconfigurable unified architecture represent one of the promising candidates for the next-generation intelligent optical aggregation networks.

The remainder of the paper is organized as follows. In Section 2, general construction method of non-binary LDPC codes is discussed and its hardware friendly decoding algorithm is overviewed. The proposed software defined rate adaptive FPGA-based decoder architecture is introduced in Section 3, while the emulation results are presented and compared in Section 4. Section 5 concludes the paper.

* dingzou@email.arizona.edu; phone +1520 621-4768

Optical Metro Networks and Short-Haul Systems VIII, edited by Atul K. Srivastava, Werner Weiershausen,
Benjamin B. Dingel, Achyut K. Dutta, Proc. of SPIE Vol. 9773, 977309 · © 2016 SPIE
CCC code: 0277-786X/16/$18 · doi: 10.1117/12.2214884

2. RATE ADAPTIVE LDPC CODES

2.1 Construction of non-binary LDPC codes

In this section, we discuss a method of constructing a high-rate non-binary LDPC codes with low error floors. This process consists of two steps: we first construct a large-girth binary LDPC codes, then replace the 1's in binary parity check matrix with non-zero element in Galois field GF(q) by random selection [7]. It is well known that density evolution (DE) [8] or extrinsic information transfer (EXIT) chart analysis [9] can be employed to derive a channel capacity approaching signal-to-noise ratio threshold for a given code rate. However, in this paper, we choose a quasi-cyclic regular LDPC code design based on permutation matrices due to its implementation efficiency. Following the guidelines in [10], the parity check matrix of a (c, r)-regular non-binary QC-LDPC code can be represented by

$$H_{c\times B,r\times B}^{NB} = H_{c\times B,r\times B}^{NB} * H_{c\times B,r\times B}^{NB} = \begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,r-1} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,r-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{c-1,0} & w_{c-1,1} & \cdots & w_{c-1,r-1} \end{bmatrix} * \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,r-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,r-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{c-1,0} & h_{c-1,1} & \cdots & h_{c-1,r-1} \end{bmatrix} \quad (1)$$

where $w_{i,j}$, and $w_{0,0}$ are both $B\times B$ circulant permutation matrix with the same offset over GF(2), and GF(q), respectively. Operation $*$ is element-wise multiplication over GF(q). For the sake of efficient implementation, we choose $w_{i,j}$ such that every element in the circulant permutation matrix is equal.

2.2 Decoding of non-binary LDPC codes

There are several algorithms proposed for decoding of the non-binary LDPC codes including Q-ary sum-product algorithm (QSPA), log-domain FFT-QSPA (Log-FFT), mixed-domain FFT-QSPA, max-log QSPA, extended min-sum algorithm, and min-max algorithm (MMA) [11-15]. Let $R_{cv}^{k,l}$, $L_{vc}^{k,l}$, and L_v represent the check c to variable v message, the variable v to check c message at k-th iteration and l-th layer, and the log-likelihood ratio (LLR) from the channel, respectively; where $k = 1, \dots, I_{max}$ and $l = 1, \dots, c$. The layered min-max algorithm (LMMA) is adopted in this paper; and the data flow can be summarized as follows:

- Initialization step:

$$R_{cv}^{k,l}(a) = 0 \text{ for } k = 1, l = 1, \dots, c \quad (2)$$

- Bit decision step:

$$L_v^{k,l}(a) = L_v(a) + \textstyle\sum_{l'} R_{cv}^{k,l'}(a) \quad (3)$$

$$\hat{c}_v = \arg\min_{a\in GF(q)} L_v^{k,l}(a) \quad (4)$$

- Variable node processing rule:

$$L_{vc}^{k,l}(a) = L_v(a) + \textstyle\sum_{l'\neq l} R_{cv}^{k,l'}(a) \quad (5)$$

$$L_{vc}^{k,l}(a) = L_{vc}^{k,l}(a) - \min_{a'\in GF(q)} L_v^{k,l}(a) \quad (6)$$

- Check node processing rule:

$$R_{cv}^{k,l}(a) = \min_{a_{v'}\in L(c|a_v=a)}(\max_{v'\in N(c)/v} L_{v'c}^{k,l}(a_{v'})) \quad (7)$$

the index k' in Eq. (5) is set to $k-1$ when $l < l'$ while set to k otherwise and Eq. (6) is necessary for numerical reasons to ensure the convergence of the algorithm. In Eq. (7), the $L(c|a_v = a)$ is the set of sequences of finite field elements $a_{v'}$ $v' \in N(c)/v$ [N(c) denotes the neighborhood of check-node c] such that $\sum_{v'\in N(c)/v} h_{cv'}a_{v'} = h_{cv}a_v$, while it is realized by trellis-based recursive approach.

2.3 Rate adaptation via shortening

The basic rate adaptation via either shortening or puncturing is widely used everywhere in communication systems, and can be introduced in both block and convolutional codes [16]. In this paper, we use shortening to achieve rate adaptive LDPC coding since it can allows a wide range of rate adjustment with unified decoder architecture through a set of reconfigurable registers in FPGA. Because of the quasi-cyclic structure of our non-binary LDPC codes, we shorten entire sub-block by adding the least number of logics' blocks. For example, we start from a (3,15)-regular non-binary LDPC codes with rate of 0.8, and we can obtain a class of shortened regular non-binary LDPC codes with column weight and row weight of {(3,14), (3,13), (3,12), (3,11), (3,10)}, which corresponding to code rates of {0.786, 0.77, 0.75, 0.727, 0.7}. It is straightforward to obtain lower rate by increasing shortening length, however, this will becomes more and more inefficient and the rate below 0.7 is not recommend in optical communication systems.

3. ARCHITECTURE OF LDPC DECODER

3.1 Overview architecture

In order to verify the performance of the proposed rate adaptive LDPC codes, we use a field programmable gate array (FPGA) platform, whose high-level diagram is illustrated in Fig. 1. This platform is similar to other platforms reported in the literature [4, 5, 17] and consists of three parts: a Gaussian noise generator, an LDPC decoder, and an error counter circuit. The Gaussian noise generator generates Gaussian distributed log-likelihood ratios (LLRs), originating from BPSK transmission over an AWGN channel, using two uniform random number generators and the Box-Muller transform. Such obtained sequences are then fed to the LDPC decoder after quantization. The software configuration interface is implemented by a lightweight microblaze processor used to configure the emulation setup, such as noise variance, shortening length, and read errors; and so on. Without loss of generality, all-zero codeword is assumed to be transmitted. The LDPC decoder is based on the layered decoding algorithm [18] and uses a scaled min-sum check-node computation rule with constant scaling factor and a min-max check node computation rule for binary and non-binary LDPC codes, respectively. This architecture is duplicated D times in order to increase the throughput.

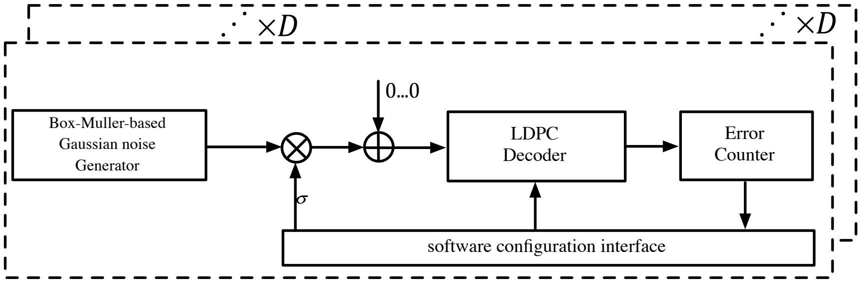
⋰ ×D
⋰ ×D
0...0
Box-Muller-based Gaussian noise Generator
LDPC Decoder
Error Counter
σ
software configuration interface

Figure 1. High-level schematic of the FPGA evaluation platform.

3.2 Rate adaptive binary LDPC

As shown in Fig. 2 (a), the binary LDPC decoder consists of two major memory blocks (one stores channel LLR and another stores *a posteriori* probability (APP) messages), two processing blocks (variable node unit (VNU) and check node unit (CNU)), an early termination unit (ETU), and a number of mux blocks, wherein its selection of output signal can be software reconfigurable to adjust the shortening length [7]. The memory consumption is dominated by LLR message with size of $n\times W_L$ APP messages of size $c\times n\times W_R$, where W_L and W_R denote the precisions used to represent LLR and APP messages. The logic consumption is dominated by CNU, as shown in Fig. 2 (b). The ABS-block first takes the absolute value of the inputs and the sign XOR array produces the output sign. In the two least minimums' finder block, we find the first minimum value via binary tree and trace back the survivors to find the second minimum value as well as the position of the first minimum value. This implies that we can write 3 values and r sign bits back to the APP memories instead of r values. However, we will not take advantage of memory reduction techniques for comparison in the following sections.

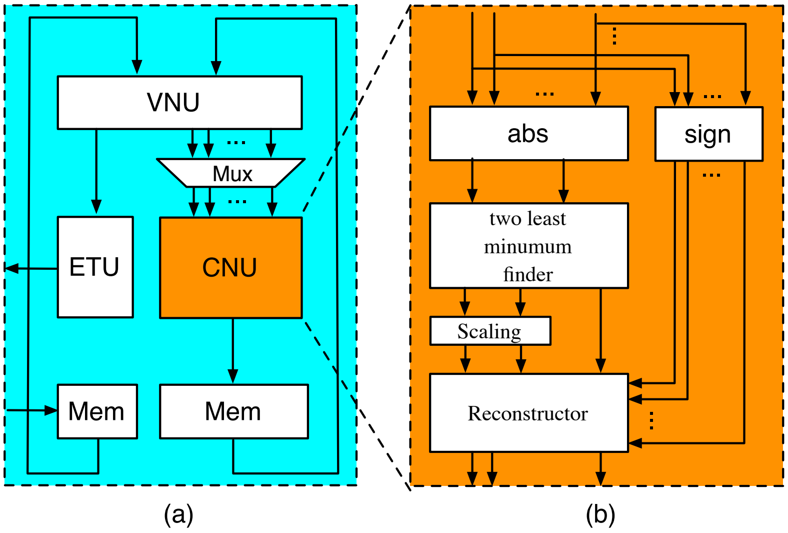

Figure 2. Architecture of rate adaptive binary LDPC decoder: (a) overall schematic diagram, (b) schematic diagram of CNU.

3.3 Rate adaptive non-binary LDPC decoder architecture

Similarly to the rate adaptive binary LDPC decoder architecture discussed above, the architecture of the LMMA-based non-binary LDPC decoder is presented in Fig. 3(a). There are four types of memories used in implementation: memory R_{cv} with size of $c \times n \times q \times W_R$ stores the information from check nodes to variable nodes, memory for L_v with size of $n \times q \times W_L$ stores the initial log-likelihood ratios, memory for $\hat{c}$ with size $n \times \log_2 q$ stores the decoded bits, and memories inside each CNU store the intermediate values. The same notations are borrowed from previous subsection except that q denotes the size of Galois field. As shown in Fig. 3(b), it is obvious that CNU is the most complex part of the decoding algorithm, which consists of r inverse permutators, r BCJR-based min-max processors and r permutators, and two types of the first-in-first-out (FIFO) registers. The inverse permutator block shifts the incoming message vector cyclically. The first FIFO is used to perform the parallel-to-serial conversion. After the min-max processor, which is implemented by low-latency bidirectional BCJR algorithm, the processed data is fed into another FIFO block performing serial-to-parallel conversion, followed by the permutator block. Because of high complexity of CNU design and high memory requirements of non-binary decoder than that of binary decoder, reduced-complexity architectures and selective version of MMA can be further exploited [19-20].

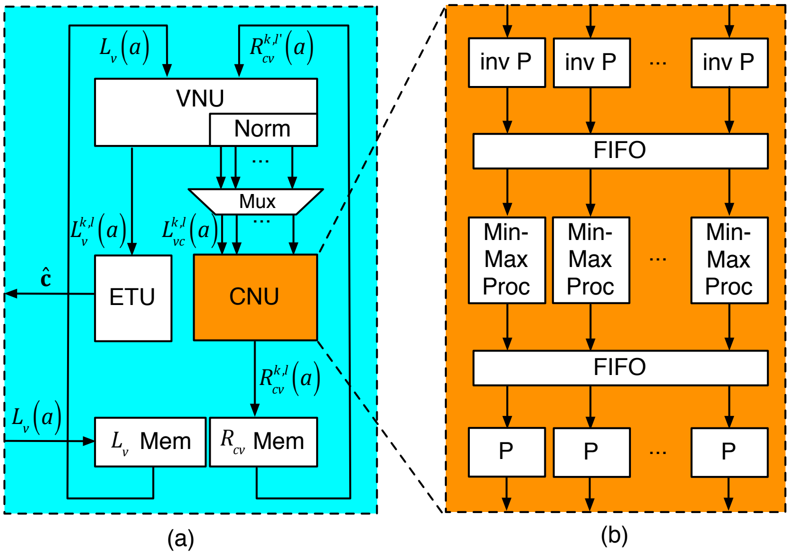

Figure 3. Architecture of rate adaptive non-binary LDPC decoder: (a) overall schematic diagram, (b) schematic diagram of CNU.

4. EMULATION RESULTS AND ANALYSIS

4.1 Comparison of three rate adaptive schemes in terms of resources utilization

We compare three rate adaptive schemes based on binary LDPC codes, non-binary LDPC codes, and binary generalized LDPC codes (GLDPC) [21]. These three architectures can be software-defined by initializing configurable registers in FPGA. In order to make a fair comparison, we first design a (3, 15)-regular quasi-cyclic binary and non-binary LDPC codes with length of 34,635 and select (15, 11) Hamming codes as local code in generalized LDPC codes. The required precision is 6-bit for binary LDPC decoder and binary GLDPC decoder while 8-bits are required for non-binary LDPC decoder. At the same time, we implement 15 VNUs and 1 CNU in order to keep the same throughput in three cases. The resource utilization is summarized in Table 1. One can clearly notice that the binary GLDPC decoder has a slightly increased number of occupied slices and memories since one more APP-based CNU is required to process the simple linear block code. On the other hand, LMMA-based non-binary LDPC codes consumes 3.6 times larger memory size than the binary one because of large field size and high quantization precision, while the occupied number of slices is five times larger than that in binary case because of higher complexity of CNU.

Table 1. Logic resources utilization.

Resources	Binary LDPC decoder	Binary GLDPC decoder	Non-binary LDPC decoder
Occupied Slices	2,969 out of 74,400 (3%)	7,180 out of 74,400 (9%)	16,842 out of 74,400 (22%)
RAMB36E1	113 out of 1,064 (10%)	113 out of 1,064 (10%)	338 out of 1,064 (31%)
RAMB18E1	14 out of 2,128 (1%)	127 out of 2,128 (5%)	221 out of 2,128 (9%)

4.2 BER performance analysis

The BER vs. Q-factor performances of the rate adaptive binary and non-binary LDPC code are presented in Fig. 4 and Fig. 5. The FPGA-based emulation was conducted over binary (BI)-AWGN channel and 6 and 8 bits precision are used in binary and non-binary LDPC decoder, respectively. A set of column weight and row weight configurations of {(3, 15), (3, 14), (3, 13), (3, 12), (3, 11), (3, 10)}, which corresponds to the code rates of {0.8, 0.786, 0.77, 0.75, 0.727, 0.7}, can be achieved by software-based reconfiguration of specific register in FPGA. The girth-10 regular (34635, 27710, 0.8) binary and non-binary mother code can achieve a Q-limit of 5.2 dB and 5.14 dB at BER of 10^{-15}, which corresponds to NCG of 11.83 dB and 11.89 dB. The rate adaptive non-binary LDPC codes outperforms the binary LDPC codes by approximated 0.06 dB in all range of rate from 0.7~0.8. In addition, we believe this gap will be larger when combined with higher modulation schemes enabling 100Gbits/s (with QPSK) 400Gbits/s (with 16QAM) optical communication systems.

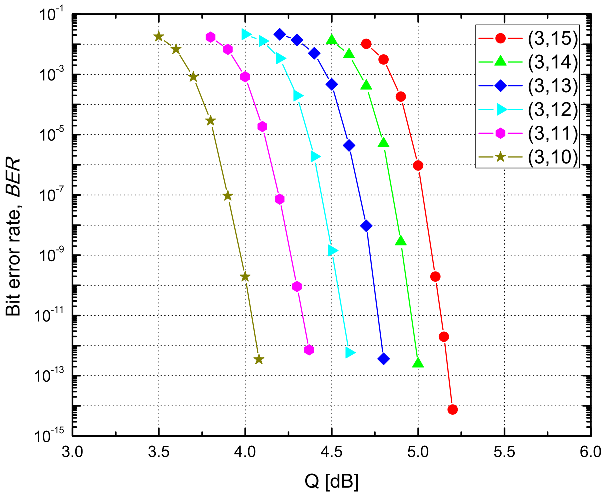

Figure 4. BER performance of rate-adaptive binary LDPC decoder implemented in FPGA.

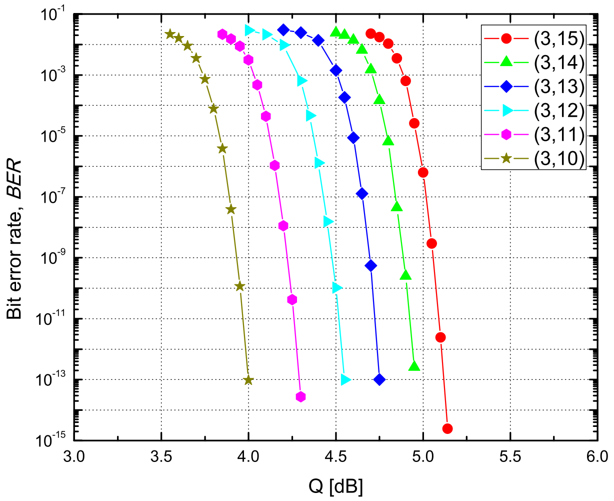

Figure 5. BER performance of rate-adaptive nonbinary LDPC decoder implemented in FPGA.

5. CONCLUSION

In this invited paper, we have proposed a rate adaptive non-binary LDPC scheme and demonstrated with a unified FPGA-based decoder architecture that it provides excellent rate flexibility. Compared to binary LDPC codes, the non-binary LDPC codes provide additional ~0.06dB gain and we believe a larger gain can be achieved when combined with high order modulations. Thus the proposed rate adaptive non-binary LDPC codes are suitable to intelligent aggregation networks and beyond.

ACKNOWLEDGMENT

This work was supported in part by NSF ERC Center for Integrated Access Networks (CIAN) under grant EEC-0812072 as well as by MURI ONR program.

REFERENCES

[1] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Vilioen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," IEEE Commun. Mag., 51(7), 36-43, (2013).

[2] ITU-T, G.975.1, "Transmission systems and media, digital systems and networks," (2004).

[3] G. Cho, L. Klak, and J. M. Kahn, "Rate-adaptive coding for optical fiber transmission systems," J. Lightw. Technol., 29(2), 222-233, (2011).

[4] D. Chang, et al., "FPGA verification of a single QC-LDPC code for 100Gb/s optical systems without error floor down to BER of 10-15," Proc. OFC/NFOEC 2011, OTuN2 (2011).

[5] A. Leven and L. Schmalen, "Status and recent advances on forward error correction technologies for lightwave systems," J. Lightw. Technol., 32(16), 2735-2750, (2014).

[6] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, "Staircase codes: FEC for 100 Gb/s OTN," J. Lightw. Technol., 30(1), 110–117, (2012).

[7] D. Zou, and I. B. Djordjevic, "FPGA implementation of concatenated non-binary QC-LDPC codes for high-speed optical transport," Optics Express, 23(11), 14501-14509, (2015).

[8] S. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," IEEE Trans. Inf. Theory, 47(2), 657-670, (2001).

[9] S. Ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," IEEE Trans. Commun.," 54(4), 670-678, (2004).

[10] M. P. C. Fossorier, “Quasi-cyclic low-density parity-check codes from circulant permutation matrices,” IEEE Trans. Inf. Theory, 50(8), 1788-1793 (2004).

[11] L. Barnault and D. Declercq, “Fast decoding algorithm for LDPC over GF(2^q),” Proc. IEEE Information Theory Workshop, 70-73, (2003).

[12] H. Wymeersch, H. Steendam, and M. Moeneclaey, “Decoding algorithm for nonbinary LDPC codes over GF(q),” IEEE Trans. Commun. 55(4), 633-643 (2007).

[13] C. Spagnol, E. Popovici, and W. Marnane, “Hardware implementation of GF(2m) LDPC decoders,” IEEE Trans. Circuits Syst., 56(12), 2609-2620 (2009).

[14] D. Declercq and M. Fossorier, “Decoding algorithm for nonbinary LDPC codes over GF(q),” IEEE Trans. Commun. 55(4), 633-643 (2007).

[15] V. Savin, “Min-max decoding for non-binary LDPC codes,” Proc. IEEE International Symposium on Information Theory, 960-964, (2008).

[16] W. E. Ryan and S. Lin, Channel Codes: Classical and Modern. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[17] D. Zou, and I. B. Djordjevic. “FPGA implementation of high-performance QC-LDPC decoder for optical communications,” in Proc. SPIE 9388, 93880P (2015).

[18] D. E. Hocevar, “A reduced complexity decoder architecture via layered decoding of LDPC codes,” Proc. IEEE Signal Processing Systems, 107-112, (2004).

[19] Y. Ueng, K. Liao, H. Chou, and C. Yang, “A high-throughput trellis-based layered decoding architecture for non-binary LDPC codes using Max-Log-QSPA,” Proc. IEEE Trans. Signal, 61(11), 2940-2951 (2013).

[20] X. Zhang, F. Cai, “Reduced-complexity decoder architecture for non-binary LDPC codes,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 19(7), 1229-1238 (2011).

[21] D. Zou, and I. B. Djordjevic. “An FPGA design of generalized low-density parity-check codes for rate-adaptive optical transport networks,” Proc. SPIE 9773-22 (2016).