Next Generation FEC for High-Capacity Communication in Optical Transport Networks

Ivan B. Djordjevic, *Member, IEEE*, Murat Arabaci, *Student Member, IEEE*, and Lyubomir L. Minkov

(Invited Paper)

***Abstract*—Codes on graphs of interest for next generation forward error correction (FEC) in high-speed optical networks, namely turbo codes and low-density parity-check (LDPC) codes, are described in this invited paper. We describe both binary and nonbinary LDPC codes, their design, and decoding. We also discuss an FPGA implementation of decoders for binary LDPC codes. We then explain how to combine multilevel modulation and channel coding optimally by using coded modulation. Also, we describe an LDPC-coded turbo-equalizer as a candidate for dealing simultaneously with fiber nonlinearities, PMD, and residual chromatic dispersion.**

***Index Terms*—Coded modulation, codes on graphs, fiber-optics communications, low-density parity-check (LDPC) codes, turbo equalization.**

I. Introduction

THE transport capabilities of fiber-optic communication systems have increased tremendously in the past two decades, primarily due to advances in optical devices and technologies, and have enabled the Internet as we know it today with all its impacts on the modern society. In particular, dense wavelength division multiplexing (DWDM) became a viable, flexible, and cost-effective transport technology. Network operators already consider 100 Gb/s per DWDM channel transmission, yet the performance of fiber-optic communication systems operating at those data rates is degraded significantly due to several transmission impairments including intra- and interchannel nonlinearities, the nonlinear phase noise, and polarization-mode dispersion (PMD) [1], [2]. These effects constitute the current limiting factors in efforts to accommodate demands for higher capacities/speeds, longer link lengths, and more flexible wavelength switching and routing capabilities in optical networks. To deal with those channel impairments, novel advanced techniques in modulation and detection, coding and signal processing should be developed; and some important approaches will be described in this invited paper.

Manuscript received January 13, 2009; revised March 02, 2009. First published May 02, 2009; current version published July 24, 2009. This paper was supported in part by the National Science Foundation (NSF) under Grant IHCS-0725405, in part by NEC Laboratories America, and in part by Opnext, Inc.

The authors are with the Department of Electrical and Computer Engineering of University of Arizona, Tucson, AZ 85721 USA (e-mail: ivan@ece.arizona.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2009.2022044

Codes on graphs [3], such as turbo codes [4]–[9] and low-density parity-check (LDPC) codes [10]–[15] have revolutionized communications, and are becoming standard in many applications. LDPC codes, invented by Gallager [10] in 1960s, are linear block codes for which the parity check matrix has low density of ones. LDPC codes have generated great interests in the coding community recently, and this has resulted in a great deal of understanding of the different aspects of LDPC codes and their decoding process. An iterative LDPC decoder based on the *sum-product algorithm* (SPA) has been shown to achieve a performance as close as 0.0045 dB to the Shannon limit [13]. The inherent low-complexity [10]–[15] of this decoder opens up avenues for its use in different high-speed applications, including optical communications.

The purpose of this invited paper is threefold: (i) to describe different classes of codes on graphs of interest for optical communications, (ii) to describe how to combine multilevel modulation and channel coding optimally (Section IV) and (iii) to describe how to perform equalization and soft decoding jointly. We first describe briefly, in Section II, the codes on graphs proposed for use in optical communications, namely, turbo-product codes (TPCs) and LDPC codes. Due to the fact that LDPC codes can match and outperform TPCs in terms of bit-error ratio (BER) performance while having a lower complexity decoding algorithm, in this paper, we are mostly concerned with LDPC codes. We describe basic concepts (in Section III) of LDPC codes and describe how to design large girth quasi-cyclic LDPC codes. We also provide a log-domain decoding algorithm and its implementation on an FPGA. The main problem in decoder implementation for large girth binary LDPC codes is the excessive codeword length and fully parallel implementation on a single FPGA is quite a challenging problem. To solve this problem, we describe nonbinary LDPC codes over $\mathrm{GF}(2^m)$ of large girth. Then we describe, in Section IV, how to optimize multilevel modulation and coding process to achieve the best possible BER performance through the use of multilevel coding (MLC) and coded orthogonal frequency division multiplexing (OFDM). Finally, in Section V, we discuss how to combine the *maximum a posteriori probability* (MAP) equalizer in an optimal fashion with an LDPC decoder, in so-called turbo-equalization fashion.

II. Codes on Graphs

The codes on graphs of interest in optical communications include turbo codes, turbo-product codes, and LDPC codes. The turbo codes [4]–[9] can be considered as the generalization of

the concatenation of codes in which, during iterative decoding, the decoders interchange the soft messages for a certain number of times. Turbo codes can approach channel capacity closely in the region of interest for wireless communications. However, they exhibit strong error floors in the region of interest for fiber-optics communications (see [5]); therefore, alternative iterative soft decoding approaches are to be sought. As recently shown in [7]–[9], [12]–[15], turbo-product codes and LDPC codes can provide excellent coding gains, and when properly designed, do not exhibit error floor in the region of interest for fiber optics communications.

A turbo-product code (TPC) is an (n_1n_2, k_1k_2, d_1d_2) code in which codewords form an $n_1 \times n_2$ array such that each row is a codeword from an (n_1, k_1, d_1) code C_1, and each column is a codeword from an (n_2, k_2, d_2) code C_2. With n_i, k_i and d_i $(i = 1,2)$ we denote the codeword length, dimension and minimum distance, respectively, of the i^{th} component code. The soft bit reliabilities are iterated between decoders for C_1 and C_2. In fiber-optics communications, TPCs based on BCH component codes are intensively studied, e.g., [7]–[9].

A. LDPC Codes

If the parity-check matrix has a low density of ones and the number of 1's per row and per column are both constant, the code is said to be a *regular LDPC* code. To facilitate the implementation at high speed, we prefer the use of regular rather than irregular LDPC codes. The graphical representation of LDPC codes, known as bipartite (Tanner) graph representation, is helpful in efficient description of LDPC decoding algorithms. A *bipartite (Tanner) graph* is a graph whose nodes may be separated into two classes (*variable* and *check* nodes), and where *undirected edges* may only connect two nodes not residing in the same class. The Tanner graph of a code is drawn according to the following rule: check (function) node c is connected to variable (bit) node v whenever element h_{cv} in a parity-check matrix $\boldsymbol{H}$ is a 1. In an $m \times n$ parity-check matrix, there are $m = n - k$ check nodes and n variable nodes. As an illustrative example, consider the $\boldsymbol{H}$-matrix of the following LDPC code

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

For any valid codeword $\boldsymbol{v} = [v_0 v_1 \ldots v_{n-1}]$, the checks used to decode the codeword are written as:

- Equation (c_0): $v_0 + v_2 + v_4 = 0 \pmod 2$.
- Equation (c_1): $v_0 + v_3 + v_5 = 0 \pmod 2$.
- Equation (c_2): $v_1 + v_2 + v_5 = 0 \pmod 2$.
- Equation (c_3): $v_1 + v_3 + v_4 = 0 \pmod 2$.

The bipartite graph (Tanner graph) representation of this code is given in Fig. 1(a). The circles represent the bit (variable) nodes while squares represent the check (function) nodes. For example, the variable nodes v_0, v_2, and v_4 are involved in Eq. (c_0) and, therefore, connected to the check node c_0. A closed path in a bipartite graph comprising l edges that closes back on itself is called a *cycle* of length l. The shortest cycle in the bipartite graph is called the *girth*. The girth influences the minimum distance of LDPC codes, correlates the extrinsic log-likelihood ratios (LLRs) and, therefore, affects the decoding performance. The use of large girth LDPC codes is preferable because the large girth increases the minimum distance and de-correlates the extrinsic info in the decoding process. To improve the iterative decoding performance, we have to avoid cycles of length 4, and preferably 6 as well. To check for the existence of short cycles, one has to search over $\boldsymbol{H}$-matrix for the patterns shown in Fig. 1(b) and (c).

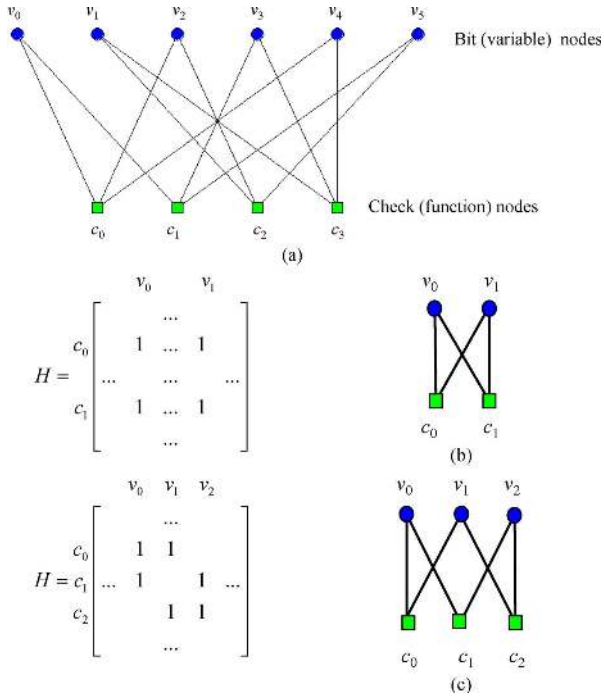

Fig. 1. (a) Bipartite graph of LDPC(6, 2) code described by $\boldsymbol{H}$ matrix below. Cycles in a Tanner graph: (b) cycle of length 4 and (c) cycle of length 6.

The code description can be done by the degree distribution polynomials $\lambda(x)$ and $\rho(x)$, for the variable-node (v-node) and the check-node (c-node) respectively [15]

$$\lambda(x) = \sum_{d=1}^{d_v} \lambda_d x^{d-1}, \qquad \rho(x) = \sum_{d=1}^{d_c} \rho_d x^{d-1} \tag{1}$$

where λ_d and ρ_d denote the fraction of the edges that are connected to degree-d v-nodes and c-nodes, respectively, and d_v and d_c denote the maximum v-node and c-node degrees, respectively.

III. Quasi-Cylic (QC) LDPC Codes

In this section, we describe a method for designing large girth QC LDPC codes; an efficient and simple variant of SPA suitable for use in optical communications, namely the min-sum-with-correction term algorithm; an FPGA implementation of their binary decoders; and nonbinary QC LDPC codes.

A. Design of Large Girth Quasi-Cyclic LDPC Codes

Based on Tanner's bound for the minimum distance of an LDPC code [11] [see (2), shown at the bottom of the next page, where g and w_c denote the girth of the code graph and the

column weight, respectively, and where d stands for the minimum distance of the code], it follows that large girth leads to an exponential increase in the minimum distance, provided that the column weight is at least 3. ($\lfloor\rfloor$ denotes the largest integer less than or equal to the enclosed quantity.) For example, the minimum distance of girth-10 codes with column weight $r = 3$ is at least 10. The parity-check matrix of regular QC LDPC codes [14], [16] can be represented by

$$\boldsymbol{H} = \begin{bmatrix} I & I & I & \cdots & I \\ I & P^{S[1]} & P^{S[2]} & \cdots & P^{S[c-1]} \\ I & P^{2S[1]} & P^{2S[2]} & \cdots & P^{2S[c-1]} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ I & P^{(r-1)S[1]} & P^{(r-1)S[2]} & \cdots & P^{(r-1)S[c-1]} \end{bmatrix} \tag{3}$$

where I is $p \times p$ (p is a prime number) identity matrix, P is $p \times p$ permutation matrix given by $P = (p_{ij})_{p\times p}$, $p_{i,i+1} = p_{p,1} = 1$ (zero otherwise), and where r and c represent the number of block-rows and block-columns in (3), respectively. The set of integers S are to be carefully chosen from the set $\{0, 1, \ldots, p-1\}$ so that the cycles of short length, in the corresponding Tanner (bipartite) graph representation of (3), are avoided. According to Theorem 2.1 in [16], we have to avoid the cycles of length $2k$ ($k = 3$ or 4) defined by the following equation:

$$\begin{aligned} S[i_1]j_1 + S[i_2]j_2 + \cdots + S[i_k]j_k \\ = S[i_1]j_2 + S[i_2]j_3 + \cdots + S[i_k]j_1 \bmod p \end{aligned} \tag{4}$$

where the closed path is defined by (i_1, j_1), (i_1, j_2), (i_2, j_2), $(i_2, j_3), \ldots, (i_k, j_k), (i_k, j_1)$ with the pair of indices denoting row-column indices of permutation-blocks in (3) such that $l_m \neq l_{m+1}, l_k \neq l_1$ ($m = 1, 2, \ldots, k; l \in \{i, j\}$). Therefore, we have to identify the sequence of integers $S[i] \in \{0, 1, \ldots, p-1\}$ ($i = 0, 1, \ldots, r-1$; $r < p$) not satisfying (4), which can be done either by computer search or in a combinatorial fashion. For example, to design the QC LDPC codes in [17], we introduced the concept of the cyclic-invariant difference set (CIDS). The CIDS-based codes come naturally as girth-6 codes, and to increase the girth we had to selectively remove certain elements from a CIDS. The design of LDPC codes of rate above 0.8, column weight 3 and girth-10 using the CIDS approach is a very challenging and is still an open problem. Instead, in our recent paper [14] , we solved this problem by developing an efficient computer search algorithm. We add an integer at a time from the set $\{0, 1, \ldots, p-1\}$ (not used before) to the initial set S and check if the (4) is satisfied. If the (4) is satisfied, we remove that integer from the set S and continue our search with another integer from set $\{0, 1, \ldots, p-1\}$ until we exploit all the elements from $\{0, 1, \ldots, p-1\}$. The code rate of these QC codes, R, is lower-bounded by

$$R \geq \frac{|S|\,p - rp}{|S|\,p} = 1 - \frac{r}{|S|} \tag{5}$$

and the codeword length is $|S|p$, where $|S|$ denotes the cardinality of set S. For a given code rate R_0, the number of elements from S to be used is $\lfloor r/(1-R_0)\rfloor$. With this algorithm, LDPC codes of arbitrary rate can be designed.

1) Example 1: By setting $p = 2311$, the set of integers to be used in (3) is obtained as

$$\begin{aligned} S = \{&1, 2, 7, 14, 30, 51, 78, 104, \\ &129, 212, 223, 318, 427, 600, 808\}. \end{aligned}$$

The corresponding LDPC code has rate $R_0 = 1 - 3/15 = 0.8$, column weight 3, girth-10 and length $|S|p = 15 \cdot 2311 = 34665$. In the example above, the initial set of integers was $S = \{1, 2, 7\}$, and the set of rows to be used in (3) is {1,3,6}. The use of a different initial set will result in a different set from that obtained above.

2) Example 2: By setting $p = 269$, the set S is obtained as

$$\begin{aligned} S = \{&0, 2, 3, 5, 9, 11, 12, 14, 27, 29, 30 \\ &32, 36, 38, 39, 41, 81, 83, 84, 86, 90, 92, 93, 95 \\ &108, 110, 111, 113, 117, 119, 120, 122\}. \end{aligned}$$

If 30 integers are used, the corresponding LDPC code has rate $R_0 = 1 - 3/30 = 0.9$, column weight 3, girth-8 and length $30 \cdot 269 = 8070$.

B. Decoding of LDPC Codes

In this sub-section, we describe the min-sum with correction term decoding algorithm [15], [18]. It is a simplified version of the original algorithm proposed by Gallager [10]. Gallager proposed a near optimal iterative decoding algorithm for LDPC codes that computes the distributions of the variables in order to calculate the *a posteriori probability* (APP) of a bit v_i of a codeword $\boldsymbol{v} = [v_0 v_1 \ldots v_{n-1}]$ to be equal to 1 given a received vector $\boldsymbol{y} = [y_0 y_1 \ldots y_{n-1}]$. This iterative decoding scheme engages passing the extrinsic info back and forth among the c-nodes and the v-nodes over the edges to update the distribution estimation. Each iteration in this scheme is composed of two half-iterations. In Fig. 2, we illustrate both the first and the second halves of an iteration of the algorithm. As an example, in Fig. 2(a), we show the message sent from v-node v_i to the c-node c_j. v_i-node collects the information from channel (y_i sample), in addition

$$d \geq \begin{cases} 1 + \frac{w_c}{w_c-2}\left((w_c-1)^{\lfloor (g-2)/4\rfloor} - 1\right), & \frac{g}{2} = 2m+1 \\ 1 + \frac{w_c}{w_c-2}\left((w_c-1)^{\lfloor (g-2)/4\rfloor} - 1\right) + (w_c-1)^{\lfloor (g-2)/4\rfloor}, & \frac{g}{2} = 2m \end{cases} \tag{2}$$

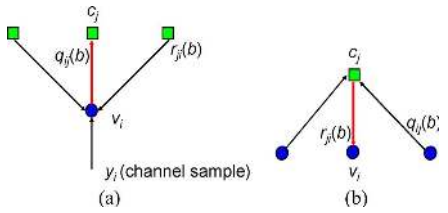

Fig. 2. Illustration of the half-iterations of the sum-product algorithm: (a) first half-iteration: extrinsic info sent from v-nodes to c-nodes, and (b) second half-iteration: extrinsic info sent from c-nodes to v-nodes.

to extrinsic info from other c-nodes connected to v_i-node, processes them and sends the extrinsic info (not already available info) to c_j. This extrinsic info contains the information about the probability $\Pr(v_i = b|y_i)$, where $b \in \{0,1\}$. This is performed in all c-nodes connected to v_i-node. On the other hand, Fig. 2(b) shows the extrinsic info sent from c-node c_j to the v-node v_i, which contains the information about Pr(c_j equation is satisfied $|\boldsymbol{y}$). This is done repeatedly to all the c-nodes connected to v_i-node.

After this intuitive description, we describe the min-sum-with-correction-term algorithm in more detail [15] because of its simplicity and suitability for high-speed implementation. Generally, we can either compute APP $\Pr(v_i|\boldsymbol{y})$ or the APP ratio $l(v_i) = \Pr(v_i = 0|\boldsymbol{y})/\Pr(v_i = 1|\boldsymbol{y})$, which is also referred to as the likelihood ratio. In log-domain version of the sum-product algorithm, we replace these likelihood ratios with log-likelihood ratios (LLRs) due to the fact that the probability domain includes many multiplications which leads to numerical instabilities, whereas the computation using LLRs involves addition only. Moreover, the log-domain representation is more suitable for finite precision representation. Thus, we compute the LLRs by $L(v_i) = \log[\Pr(v_i = 0|\boldsymbol{y})/\Pr(v_i = 1|\boldsymbol{y})]$. For the final decision, if $L(v_i) > 0$, we decide in favor of 0 and if $L(v_i) < 0$, we decide in favor of 1.

To further explain the algorithm, we introduce the following notations due to MacKay [12].

V_j	= {v-nodes connected to c-node c_j}.
$V_{j\backslash i}$	= {v-nodes connected to c-node c_j}\{v-node v_i}.
C_i	= {c-nodes connected to v-node v_i}.
$C_{i\backslash j}$	= {c-nodes connected to v-node v_i} \ {c-node c_j}.
$M_v(\sim i)$	= {messages from all v-nodes except node v_i}.
$M_c(\sim j)$	= {messages from all c-nodes except node c_j}.
P_i	= $\Pr(v_i = 1|y_i)$.
S_i	= event that the check equations involving c_i are satisfied.
$q_{ij}(b)$	= $\Pr(v_i = b|S_i, y_i, M_c(\sim j))$.
$r_{ji}(b)$	= $\Pr(\text{check equation } c_j \text{ is satisfied} \mid v_i = b, M_v(\sim i))$.

In the log-domain version of the sum-product algorithm, all the calculations are performed in the log-domain as follows:

$$L(v_i) = \log\left(\frac{\Pr(v_i = 0|y_i)}{\Pr(v_i = 1|y_i)}\right)$$
$$L(r_{ji}) = \log\left(\frac{r_{ji}(0)}{r_{ji}(1)}\right) \quad L(q_{ij}) = \log\left(\frac{q_{ij}(0)}{q_{ij}(1)}\right). \quad (6)$$

The algorithm starts with the initialization step where we set $L(v_i)$ as follows:

$$L(v_i) = (-1)^{y_i} \log\left(\frac{1-\varepsilon}{\varepsilon}\right), \text{ for BSC}$$
$$L(v_i) = \frac{2y_i}{\sigma^2}, \text{ for binary input AWGN}$$
$$L(v_i) = \log\left(\frac{\sigma_1}{\sigma_0}\right) - \frac{(y_i-\mu_0)^2}{2\sigma_0^2} + \frac{(y_i-\mu_1)^2}{2\sigma_1^2}$$
$$\text{for BA-AWGN}$$
$$L(v_i) = \log\left(\frac{\Pr(v_i = 0|y_i)}{\Pr(v_i = 1|y_i)}\right), \text{ for abritrary channel} \quad (7)$$

where ε is the probability of error in the binary symmetric channel (BSC), σ^2 is the variance of the Gaussian distribution of the AWGN, and μ_j and σ_j^2 ($j = 0{,}1$) represent the mean and the variance of Gaussian process corresponding to the bits $j = 0{,}1$ of a binary asymmetric (BA)-AWGN channel. After initialization of $L(q_{ij})$, we calculate $L(r_{ji})$ as follows:

$$L(r_{ji}) = L\left(\sum_{i' \in V_j \backslash i} b_{i'}\right) = L(\cdots \oplus b_k \oplus b_l \oplus b_m \oplus b_n \cdots)$$
$$= \ldots L_k \boxed{+} L_l \boxed{+} L_m \boxed{+} L_n \boxed{+} \cdots \quad (8)$$

where $\oplus$ denotes the modulo-2 addition, and $\boxed{+}$ denotes a pairwise computation defined by

$$L_a \boxed{+} L_b = \text{sign}(L_a)\text{sign}(L_b)\min(|L_a|, |L_b|) + s(L_a, L_b)$$
$$s(L_a, L_b) = \log\left(1 + e^{-|L_a + L_b|}\right) - \log\left(1 + e^{-|L_a - L_b|}\right). \quad (9)$$

The term $s(L_a, L_b)$ is the correction term and is implemented as a lookup table.

After we calculate $L(r_{ji})$, we update

$$L(q_{ij}) = L(v_i) + \sum_{j' \in C_{i\backslash j}} L(r_{j'i})$$
$$L(Q_i) = L(v_i) + \sum_{j \in C_i} L(r_{ji}). \quad (10)$$

Finally, the decision step is as follows:

$$\hat{v}_i = \begin{cases} 1, & L(Q_i) < 0 \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

If the syndrome equation $\hat{\boldsymbol{v}}\boldsymbol{H}^T = 0$ (where the superscript T denotes transposition) is satisfied or the maximum number of iterations is reached, we stop, otherwise, we recalculate $L(r_{ji})$ and update $L(q_{ij})$ and $L(Q_i)$ and check again. It is important to set the number of iterations high enough to ensure that most of

the codewords are decoded correctly and low enough not to affect the processing time. It is important to mention that decoder for good LDPC codes require less number of iterations to guarantee successful decoding.

C. BER Performance of LDPC Codes

The results of simulations for an additive white Gaussian noise (AWGN) channel model are given in Fig. 3, where we compare the large girth LDPC codes [Fig. 3(a)] against RS codes, concatenated RS codes, TPCs, and other classes of LDPC codes. In optical communications, it is a common practice to use the Q-factor as a figure of merit of binary modulation schemes instead of signal-to-noise ratio.[1] In all simulation results in this paper, we maintained the double precision. For the LDPC(16935,13550) code, we also provided 3- and 4-bit fixed-point simulation results [see Fig. 3(a)]. Our results indicate that the 4-bit representation performs comparable to the double-precision representation whereas the 3-bit representation performs 0.27 dB worse than the double-precision representation at the BER of $2 \cdot 10^{-8}$. The girth-10 LDPC(24015,19212) code of rate 0.8 outperforms the concatenation $\mathrm{RS}(255,239) + \mathrm{RS}(255,223)$ (of rate 0.82) by 3.35 dB and RS(255,239) by 4.75 dB both at BER of 10^{-7}. The same LDPC code outperforms projective geometry (PG) $(2, 2^6)$ based LDPC(4161,3431) (of rate 0.825) of girth-6 by 1.49 dB at BER of 10^{-7}, and outperforms CIDS based LDPC(4320,3242) of rate 0.75 and girth-8 LDPC codes by 0.25 dB. At BER of 10^{-10}, it outperforms lattice based LDPC(8547,6922) of rate 0.81 and girth-8 LDPC code by 0.44 dB, and $\mathrm{BCH}(128,113) \times \mathrm{BCH}(256,239)$ TPC of rate 0.82 by 0.95 dB. The net effective coding gain (NECG) at BER of 10^{-12} is 10.95 dB.

In Fig. 3(b), different LDPC codes are compared against RS (255,223) code, concatenated RS code of rate 0.82 and convolutional code (CC) (of constraint length 5). It can be seen that LDPC codes, both regular and irregular, offer much better performance than hard-decision codes. It should be noticed that pairwised balanced design (PBD) based irregular LDPC code of rate 0.75 is only 0.4 dB away from the concatenation of convolutional-RS codes [denoted in Fig. 3(b) as $\mathrm{RS} + \mathrm{CC}$] with significantly lower code rate $R = 0.44$ at BER of 10^{-6}. As expected, irregular LDPC codes (black colored curves), outperform regular LDPC codes (pink colored curves).

D. FPGA Implementation of Large Girth LDPC Codes

We use the min-sum algorithm which is a further simplified version of the min-sum-with-correction-term algorithm detailed in the previous subsection. The only difference is that the min-sum algorithm omits the correction term in (9). Among various alternatives, we adopted a partially parallel architecture in our implementation since it is a natural choice for quasi-cyclic codes. In this architecture, a processing element (PE) is assigned to a group of nodes of the same kind instead of a single node. A PE mapped to a group of bit nodes is called a bit-processing element (BPE), and a PE mapped to a group of check nodes is called a check-processing element (CPE). BPEs (CPEs) process

[1]The Q-factor is defined as $Q = (\mu_1 - \mu_0)/(\sigma_1 + \sigma_0)$, where μ_j and σ_j $(j = 0,1)$ represent the mean and the standard deviation corresponding to the bits $j = 0,1$.

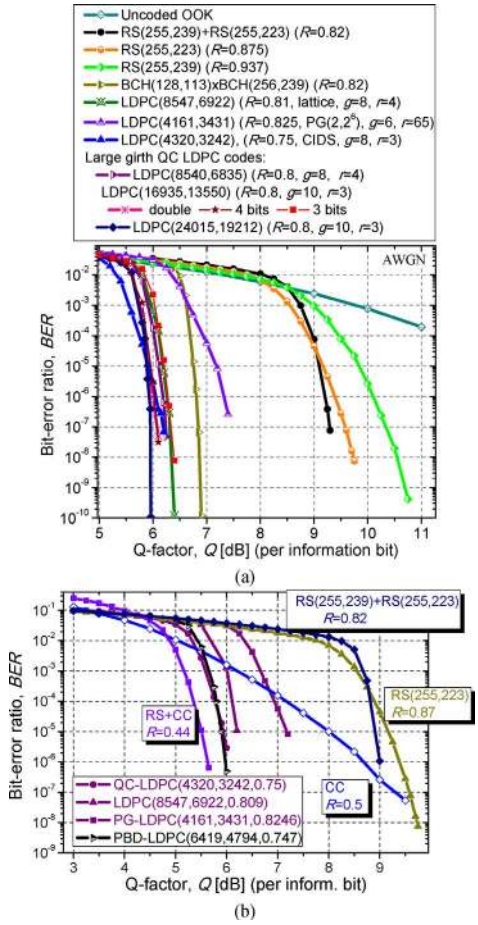

Fig. 3. (a) Large girth QC LDPC codes against RS codes, concatenated RS codes, TPCs, and previously proposed LDPC codes on an AWGN channel model, and (b) LDPC codes versus convolutional, concatenated RS, and concatenation of convolutional and RS codes on an AWGN channel. Number of iterations in sum-product-with-correction-term algorithm was set to 25.

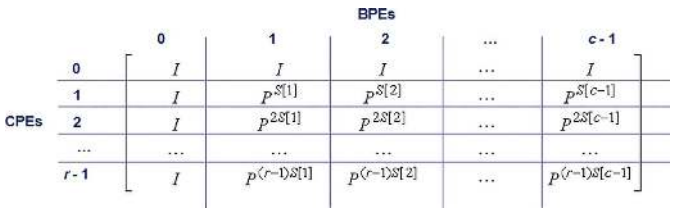

Fig. 4. Assignment of bit nodes and check nodes to BPEs and CPEs, respectively.

the nodes assigned to them in a serial fashion. However, all BPEs (CPEs) carry out their tasks simultaneously. Thus, by changing the number of elements assigned to a single BPE and CPE, one can control the level of parallelism in the hardware. In Fig. 4, we depict a convenient method for assigning BPEs and CPEs to the nodes in a QC-LDPC code. This method is not only easy to implement but also advantageous since it simplifies the memory addressing.

The messages between BPEs and CPEs are exchanged via memory banks. In Table I, we summarize the memory allocation in our implementation where we used the following notation: MEM B and MEM C denote the memories used to store bit node and check node edge values, respectively; MEM E

TABLE I
MEMORY ALLOCATION OF THE IMPLEMENTATION

MEM Name	MEM B	MEM C	MEM E	MEM I	MEM R
Data word (bits)	8	11	1	8	32
Address word (bits)	16	16	15	15	10
Memory block size (words)	50805	50805	16935	16935	625

```
for i = 0 to p-1 do
    for each b = 0 to c-1 do
        - Read r data values from MEM B located in the range
          [b * p * r + i * r, b * p * r + i * r + r - 1].
        - Sum them up and store the sum.
        - Update MEM E at location (b * p + i).
        for each k = 0 to r-1 do
          - Subtract from the computed sum the value
          located at (b * p * r + i * r + k - 1) in MEM B.
          - Use this value to update MEM C at location
          (k * p + ((p - k * b) mod p)) * c + b).
        end
    end
end
```

Fig. 5. Pseudo code describing assignment of bit nodes and check nodes to BPEs and CPEs.

stores the codeword estimate; MEM I stores the initial log-likelihood ratios; and finally, MEM R holds the state of the random number generator needed for AWGN source, which is based on Mersenne Twister algorithm.

In our initial design [20], we used the MitrionC hardware programming language, which is "an intrinsically parallel C-family language" developed by Mitrionics, Inc. [21]. Using MitrionC syntax, we provided a pseudo code in Fig. 5 showing how the data are transferred from MEM B to MEM C after being processed by BPEs. The code features three loop expressions of two types. The *for* loop sequentially executes its loop body for every bit node, i, in a BPE. On the contrary, the *for each* loop is a parallel loop, and, hence, the operations in the loop body are applied to all the elements in its declaration simultaneously. To expatiate, due to the first *for each* loop, all BPEs perform their operations on their ith bit nodes in parallel. Since we are using a single memory in our implementation to store the edge values of all check nodes, the second *for each* loop causes a BPE to update its connections in MEM C in a pipelined fashion. As also shown in Fig. 5, we compute the memory addresses to read/write data from/to "on-the-fly" using the bit node ID (i), BPE ID (b) and CPE ID (k). This convenient calculation of addresses is possible because of the quasi-cyclic nature of the code and the way we assigned BPEs and CPEs.

We tested our design on the FPGA Subsystem located at the High Performance Computing (HPC) Center at The University of Arizona. This FPGA Subsystem consists of SGI RASC RC1000 Blade having two Virtex 4 LX2000 FPGAs. In Fig. 6, we present BER performance comparison of FPGA and software implementations for a girth-10 quasi-cyclic LDPC (16935, 13550) code. We observe a close agreement between the two BER curves. Furthermore, the performance of the min-sum algorithm is only 0.2 dB worse than that of the min-sum-with-correction-term algorithm at the BER of 10^{-6} and the gap gets

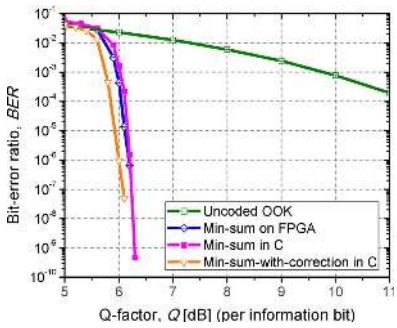

Fig. 6. BER performance comparison of FPGA and software implementations of the min-sum algorithm.

closer as the Q factor increases. The NECG of the min-sum algorithm for the same LDPC code at BER of 10^{-6} is found to be 10.3 dB.

The main problem in decoder implementation for large girth binary LDPC codes is the excessive codeword length, and a fully parallel implementation on a single FPGA is quite a challenging problem. To solve this problem, in the next subsection, we will consider large-girth nonbinary LDPC codes over $\mathrm{GF}(2^m)$ [22]–[25]. By designing codes over higher-order fields, we aim to achieve the coding gains comparable to binary LDPC codes but for shorter codeword lengths.

E. Nonbinary QC LDPC Codes

In this sub-section, we describe a two-stage design technique for constructing nonbinary regular, high-rate LDPC codes. We show that the complexity of the nonbinary decoding algorithm over $\mathrm{GF}(4)$ used to decode this code is 1.1 times less complex compared to the min-sum-with-correction-term algorithm, described in sub-section B, used for decoding a bit-length-matched binary LDPC code. Furthermore, we demonstrate that by enforcing the nonbinary LDPC codes to have the same nonzero field element in a given column in their parity-check matrices, we can reduce the hardware implementation complexity of their decoders without incurring any degradation in the error-correction performance.

A q-ary LDPC code is a linear block code defined as the null space of a sparse parity-check matrix $\boldsymbol{H}$ over a finite field of q elements that is denoted by $\mathrm{GF}(q)$ where q is a prime or prime-power. Davey and MacKay [23] devised a q-ary sum-product algorithm (QSPA) to decode q-ary LDPC codes, where $q = 2^p$ and p is an integer. They also proposed an efficient way of conducting QSPA via fast Fourier transform (FFT-QSPA). FFT-QSPA is further analyzed and improved in [24]. A mixed-domain version of the FFT-QSPA (MD-FFT-QSPA) that reduces the computational complexity by transforming the multiplications into additions with the help of logarithm and exponentiation operations is proposed in [26]. Due to the availability of efficient decoding algorithms, we consider q-ary LDPC codes where q is a power of two.

In the first step of our two-stage code design technique, we design binary QC LDPC codes of girth-6 using the algebraic construction method based on the multiplicative groups of finite

fields [26]. Let α be a primitive element of $\mathrm{GF}(q)$ and let $\boldsymbol{W} = [w_{i,j}]$ be a $(q-1)$-by-$(q-1)$ matrix given as follows:

$$\boldsymbol{W} = \begin{bmatrix} \alpha^0 - 1 & \alpha - 1 & \dots & \alpha^{q-2} - 1 \\ \alpha - 1 & \alpha^2 - 1 & \dots & \alpha^{q-1} - 1 \\ \dots & \dots & \dots & \dots \\ \alpha^{q-2} - 1 & \alpha^{q-1} - 1 & \dots & \alpha^{2(q-2)} - 1 \end{bmatrix}. \quad (12)$$

We can transform $\boldsymbol{W}$ into a quasi-cyclic parity-check matrix $\boldsymbol{H}^{(1)}$ of the following form:

$$\boldsymbol{H}^{(1)} = \begin{bmatrix} \boldsymbol{A}_{0,0} & \boldsymbol{A}_{0,1} & \dots & \boldsymbol{A}_{0,n-1} \\ \boldsymbol{A}_{1,0} & \boldsymbol{A}_{1,1} & \dots & \boldsymbol{A}_{1,n-1} \\ \dots & \dots & \dots & \dots \\ \boldsymbol{A}_{m-1,0} & \boldsymbol{A}_{m-1,1} & \dots & \boldsymbol{A}_{m-1,n-1} \end{bmatrix} \quad (13)$$

where every sub-matrix $\boldsymbol{A}_{i,j}$ is related to the field element $w_{i,j}$ by

$$\boldsymbol{A}_{i,j} = \left[\boldsymbol{z}(w_{i,j}) \;\; \boldsymbol{z}(\alpha w_{i,j}) \;\; \boldsymbol{z}\left(\alpha^2 w_{i,j}\right) \;\; \dots \;\; \boldsymbol{z}\left(\alpha^{q-2} w_{i,j}\right)\right]^T \quad (14)$$

where $\boldsymbol{z}(\alpha^i) = (z_0, z_1, \dots, z_{q-2})$ is a $(q-1)$-tuple over $\mathrm{GF}(2)$ whose ith component $z_i = 1$ and all other $q-2$ components are zero. Using Theorem 1 in [26], we can show that the parity-check matrix, $\boldsymbol{H}^{(1)}$, given in (13), which is a $(q-1)$-by-$(q-1)$ array of circulant permutation and zero matrices of size $(q-1)$-by-$(q-1)$, has a girth of at least six. We use this quasi-cyclic, girth-6 parity-check matrix $\boldsymbol{H}^{(1)}$ in the second stage.

If we simply choose γ block-rows and ρ block-columns from $\boldsymbol{H}^{(1)}$ while avoiding the zero matrices, we obtain a (γ, ρ)-regular parity-check matrix whose null space yields a (γ, ρ)-regular LDPC code with a rate of at least $(\rho - \gamma)/\rho$. Instead of a simple, random selection, however, if we choose rows and columns from $\boldsymbol{H}^{(1)}$ while avoiding performance-degrading short cycles, we can boost the performance of the resulting LDPC code. Hence, following the guidelines in [16], the first step in the second stage is to select γ rows and ρ columns from $\boldsymbol{H}^{(1)}$ in such a way that the resulting binary quasi-cyclic code has a girth of eight. In the second step, we replace the 1's in binary parity-check matrix with nonzero elements from $\mathrm{GF}(q)$ either by completely random selection or by enforcing each column to have the same nonzero element from $\mathrm{GF}(q)$ while letting the nonzero element of each column be determined again by a random selection. We denote the final q-ary (γ, ρ)-regular, girth-8 matrix by $\boldsymbol{H}^{(2)}$.

Following the two-stage design we had discussed above, we generated (3,15)-regular, girth-8 LDPC codes over the fields $\mathrm{GF}(2^p)$, where $0 \leq p \leq 7$. All the codes had a code rate (R) of at least 0.8 and, hence, an overhead $\mathrm{OH} = (1/R - 1)$ of 25% or less. We compared the BER performances of these codes against each other and against some other well-known codes, namely the ITU-standard RS(255,239), RS(255,223) and $\mathrm{RS}(255, 239) + \mathrm{RS}(255, 223)$ codes; and $\mathrm{BCH}(128, 113) \times \mathrm{BCH}(256, 239)$ TPC. We used the binary AWGN (BI-AWGN) channel model in our simulations and set the maximum number of iterations to 50. In Fig. 7, we present the BER performances of the set of nonbinary LDPC codes discussed above. Using the figure, we can conclude that when we fix the girth of a nonbinary regular, rate-0.8 LDPC code at eight, increasing the field order above eight exacerbates the BER performance. In addition to having better BER performance than codes over higher order fields, codes over $\mathrm{GF}(4)$ have smaller decoding complexities when decoded using MD-FFT-QSPA algorithm since the complexity of this algorithm is proportional to the field order. Thus,

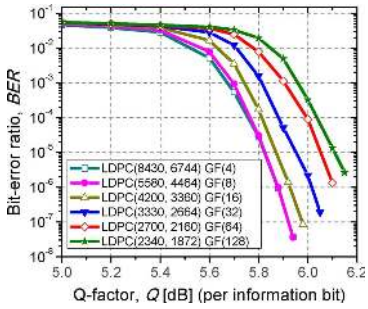

Fig. 7. Comparison of nonbinary, (3,15)-regular, girth-8 LDPC codes over BI-AWGN channel.

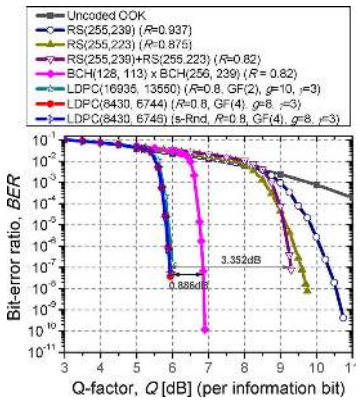

Fig. 8. Comparison of 4-ary (3,15)-regular, girth-8 LDPC codes; a binary, girth-10 LDPC code, three RS codes and a TPC code.

we focus our attention on nonbinary, regular, rate-0.8, girth-8 LDPC codes over $\mathrm{GF}(4)$ in the rest of the sub-section.

In Fig. 8, we compare the BER performance of the LDPC(8430,6744) code over $\mathrm{GF}(4)$ discussed in Fig. 7 against that of the RS(255,239) code, RS(255,223) code, $\mathrm{RS}(255, 239) + \mathrm{RS}(255, 223)$ concatenation code, and $\mathrm{BCH}(128, 113) \times \mathrm{BCH}(256, 239)$ TPC. We observe that the LDPC code over $\mathrm{GF}(4)$ outperform all of these codes with a significant margin. In particular, it provides an additional coding gain of 3.363 dB and 4.401 dB at BER of 10^{-7} when compared to the concatenation code $\mathrm{RS}(255, 239) + \mathrm{RS}(255, 223)$ and the RS(255,239) code, respectively. Its coding gain improvement over $\mathrm{BCH}(128, 113) \times \mathrm{BCH}(256, 239)$ TPC is 0.886 dB at BER of 4×10^{-8}. Finally, we computed the NECG of the 4-ary, regular, rate-0.8, girth-8 LDPC code over $\mathrm{GF}(4)$ to be 10.784 dB at BER of 10^{-12}. We also presented in Fig. 8 a competitive, binary, (3,15)-regular, LDPC(16935,13550) code proposed in [14]. We can see that the 4-ary, (3,15)-regular, girth-8 LDPC(8430,6744) code beats the bit-length-matched binary LDPC code with a margin of 0.089 dB at BER of 10^{-7}. More importantly, the complexity of the MD-FFT-QSPA used for decoding the nonbinary LDPC code is lower than the min-sum-with-correction-term algorithm [18], [27] used for decoding the corresponding binary LDPC code. The complexity of MD-FFT-QSPA for a q-ary, bit-length matched (γ, ρ)-regular nonbinary LDPC code with $M/\log q$ check nodes is given by $(M/\log q)2\rho q(\log q + 1 - 1/(2\rho))$ additions. On the other hand, to decode binary (γ, ρ)-regular LDPC codes using the min-sum-with-correction-term algorithm [18], [27] one needs $M\rho(\rho - 2)$ additions. Thus, a (3,15)-regular 4-ary nonbinary

LDPC code requires 91.28% of the computational resources required in decoding a bit-length matched (3,15)-regular LDPC code of the same rate and bit length.

IV. Coded Modulation

In this section, we describe how to optimally combine modulation with channel coding, and describe two coded-modulation schemes: (i) multilevel coding [28], [29], and (ii) coded-OFDM [30]. Using this approach, modulation, coding and multiplexing are performed in a unified fashion so that, effectively, the transmission, signal processing, detection and decoding are done at much lower symbol rates. At these lower rates, dealing with the nonlinear effects and PMD is more manageable, while the aggregate data rate per wavelength is maintained above 100 Gb/s.

A. Multilevel Coding

M-ary PSK, M-ary QAM and M-ary DPSK achieve the transmission of $\log_2 M$ $(=\ m)$ bits per symbol, providing bandwidth-efficient communication. In coherent detection, the data phasor $\phi_l \in \{0, 2\pi/M, \ldots, 2\pi(M-1)/M\}$ is sent at each lth transmission interval. In direct detection, the modulation is differential, the data phasor $\phi_l = \phi_{l-1} + \Delta\phi_l$ is sent instead, where $\Delta\phi_l \in \{0, 2\pi/M, \ldots, 2\pi(M-1)/M\}$ is determined by the sequence of $\log_2 M$ input bits using an appropriate mapping rule. Let us now introduce the transmitter architecture employing LDPC codes as channel codes. If component LDPC codes are of different code rates but of the same length, the corresponding scheme is commonly referred to as multilevel coding (MLC). If all component codes are of the same code rate, corresponding scheme is referred to as the bit-interleaved coded-modulation (BICM). The use of MLC allows us to adapt the code rates to the constellation mapper and channel. For example, for Gray mapping, 8-PSK and AWGN, it was found in [31] that optimum code rates of individual encoders are approximately 0.75, 0.5 and 0.75, meaning that 2 bits are carried per symbol. In MLC, the bit streams originating from m different information sources are encoded using different (n, k_i) LDPC codes of code rate $r_i = k_i/n$. k_i denotes the number of information bits of the ith $(i = 1, 2, \ldots, m)$ component LDPC code, and n denotes the codeword length, which is the same for all LDPC codes. The mapper accepts m bits, $\boldsymbol{c} = (c_1, c_2, \ldots, c_m)$, at time instance i from the $(m \times n)$ interleaver column-wise and determines the corresponding M-ary $(M = 2^m)$ constellation point $\boldsymbol{s}_i = (I_i, Q_i) = |\boldsymbol{s}_i| \exp(j\phi_i)$ [see Fig. 9(a)].

The receiver input electrical field at time instance i for an optical M-ary differential phase-shift keying (DPSK) receiver configuration from Fig. 9(b) is denoted by $E_i = |E_i| \exp(j\varphi_i)$. The outputs of I- and Q-branches [upper and lower-branches in Fig. 9(b)] are proportional to $\mathrm{Re}\{E_i E_{i-1}^*\}$ and $\mathrm{Im}\{E_i E_{i-1}^*\}$, respectively. The corresponding coherent detector receiver architecture is shown in Fig. 9(c), where

$$S_i = |S|\, e^{j\varphi_{S,i}} (\varphi_{S,i} = \omega_S t + \varphi_i + \varphi_{S,PN})$$

is coherent receiver input electrical field at time instance i and

$$L = |L|\, e^{j\varphi_L} (\varphi_L = \omega_L t + \varphi_i + \varphi_{L,PN})$$

is the local laser electrical field. For homodyne coherent detection, the frequency of the local laser (ω_L) is the same as that

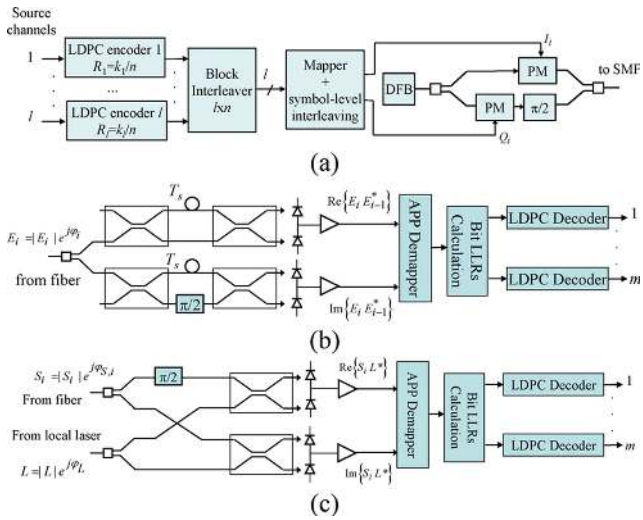

Fig. 9. Bit-interleaved LDPC-coded modulation scheme: (a) transmitter architecture, (b) direct detection architecture, and (c) coherent detection receiver architecture. $T_s = 1/R_s$, R_s is the symbol rate.

of the incoming optical signal (ω_L), so the balanced outputs of I- and Q-channel branches [upper- and lower-branches of Fig. 9(c)] can be written as

$$\begin{aligned} v_I(t) &= R\,|S_k|\,|L| \cos(\varphi_i + \varphi_{S,PN} - \varphi_{L,PN}) \\ (i-1)\,T_s &\le t < iT_s \\ v_Q(t) &= R\,|S_k|\,|L| \sin(\varphi_i + \varphi_{S,PN} - \varphi_{L,PN}) \\ (i-1)\,T_s &\le t < iT_s \end{aligned} \tag{15}$$

where R is photodiode responsivity while $\varphi_{S,PN}$ and $\varphi_{L,PN}$ represent the laser phase noise of transmitting and receiving (local) laser, respectively.

The outputs at I- and Q-branches (in either coherent or direct detection case), are sampled at the symbol rate (we assume perfect synchronization), and the symbol LLRs are calculated in an APP demapper block as follows:

$$\lambda(\boldsymbol{s}) = \log \frac{P(\boldsymbol{s} = \boldsymbol{s}_0 | \boldsymbol{r})}{P(\boldsymbol{s} \ne \boldsymbol{s}_0 | \boldsymbol{r})} \tag{16}$$

where $P(\boldsymbol{s}|\boldsymbol{r})$ is determined by using Bayes' rule

$$P(\boldsymbol{s}|\boldsymbol{r}) = \frac{P(\boldsymbol{r}|\boldsymbol{s})\,P(\boldsymbol{s})}{\sum\limits_{\boldsymbol{s}'} P(\boldsymbol{r}|\boldsymbol{s}')\,P(\boldsymbol{s}')}. \tag{17}$$

Notice that $\boldsymbol{s}_i = (I_i, Q_i)$ is the transmitted signal constellation point at time instance i, while $\boldsymbol{r}_i = (r_{I,i}, r_{Q,i})$, $r_{I,i} = v_I(t = iT_s)$, and $r_{Q,i} = v_Q(t = iT_s)$ are the samples of I- and Q-detection branches from Fig. 9(b) and (c). In the presence of fiber nonlinearities, $P(\boldsymbol{r}_i|\boldsymbol{s}_i)$ from (17) is estimated by evaluation of histograms, employing sufficiently long training sequence. Notice that for direct detection, even in the absence of nonlinearities we have to use the histogram method because the distribution functions are not Gaussian. With $P(\boldsymbol{s}_i)$ we denote the *a priori* probability of symbol $\boldsymbol{s}_i$, while $\boldsymbol{s}_0$ is a referent symbol. The normalization in (16) is introduced to eliminate the

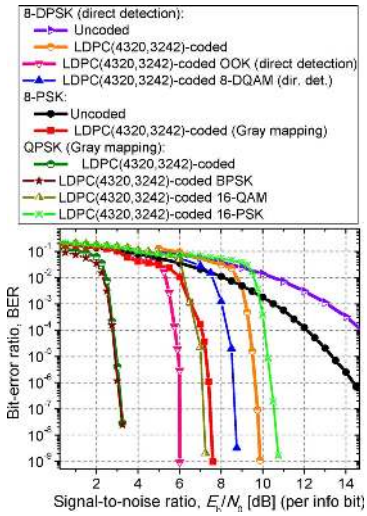

Fig. 10. BER performance comparison between bit-interleaved LDPC-coded modulation with coherent detection schemes and direct detection schemes over the AWGN channel. E_b represents the average bit energy, and N_0 is the power spectral density.

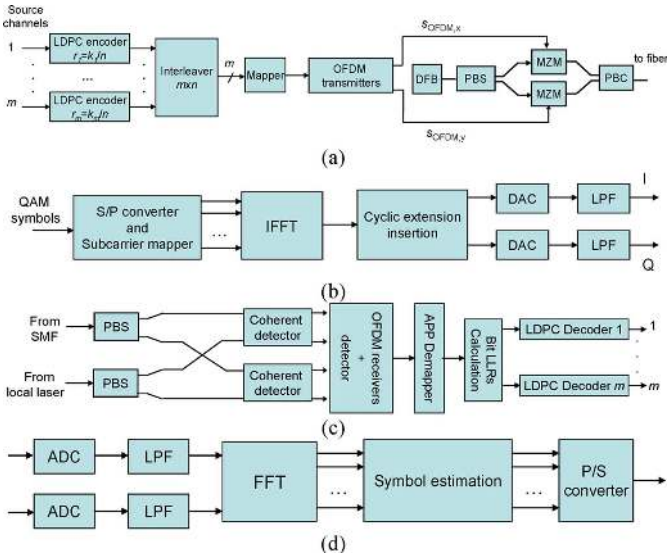

Fig. 11. Polarization-multiplexed LDPC-coded OFDM employing both polarizations: (a) transmitter architecture, (b) OFDM transmitter configuration, (c) receiver architecture, and (d) OFDM receiver configuration. DFB: distributed feedback laser, PBS(C): polarization beam splitter (combiner), MZM: dual-drive Mach–Zehnder modulator, APP: *a posteriori* probability, LLRs: log-likelihood ratios.

denominator from (17). The bit LLRs c_j $(j = 1, 2, \ldots, m)$ are determined from symbol LLRs of (16) as

$$L(\hat{c}_j) = \log \frac{\sum_{\boldsymbol{s}_i : c_j = 0} \exp[\lambda(\boldsymbol{s}_i)]}{\sum_{\boldsymbol{s}_i : c_j = 1} \exp[\lambda(\boldsymbol{s}_i)]}. \tag{18}$$

The jth bit LLR in (18) is obtained as the logarithm of the ratio of a probability that $c_j = 0$ and probability that $c_j = 1$. In the nominator (denominator), the summation is done over all symbols $\boldsymbol{s}_i$ having 0 (1) at the position j. The APP demapper extrinsic LLRs (the difference of demapper bit LLRs and LDPC decoder LLRs from previous step) for LDPC decoders become

$$L_{M,e}(\hat{c}_j) = L(\hat{c}_j) - L_{D,e}(c_j). \tag{19}$$

With $L_{D,e}(c)$ we denoted LDPC decoder extrinsic LLRs which are initially set to zero. The LDPC decoder extrinsic LLRs (the difference between LDPC decoder output and the input LLRs), $L_{D,e}$, are forwarded to the APP demapper as *a priori* bit LLRs $(L_{M,a})$ so that the symbol *a priori* LLRs are calculated as

$$\lambda_a(\boldsymbol{s}) = \log P(\boldsymbol{s}) = \sum_{j=0}^{m-1} (1 - c_j) L_{D,e}(c_j). \tag{20}$$

By substituting (20) into (17) and then (16), we are able to calculate the symbol LLRs for the subsequent iteration. The iteration between the APP demapper and LDPC decoder is performed until the maximum number of iterations is reached, or the valid code-words are obtained.

The results of the simulations, which use 30 iterations in the sum-product algorithm and 10 iterations between the APP demapper and the LDPC decoder, and employ only BICM and Gray mapping, are shown in Fig. 10. Although the actual noise in the repeated systems is dominated by the ASE noise, in this calculation we observed the thermal noise dominated scenario, to be consistent with digital communication literature [39]. The coding gain for 8-PSK at the BER of 10^{-9} is about 9.5 dB and a much larger coding gain is expected at BERs below 10^{-12}.

Bit-interleaved LDPC-coded 8-PSK with coherent detection outperforms LDPC-coded 8-DPSK with direct detection by 2.23 dB at the BER of 10^{-9}. 8-DQAM outperforms 8-DPSK by 1.15 dB at the same BER. LDPC-coded 16-QAM slightly outperforms LDPC-coded 8-PSK, and significantly outperforms LDPC-coded 16-PSK. As expected, LDPC-coded BPSK and LDPC-coded QPSK (with Gray mapping) perform very closely, and they both outperform LDPC-coded OOK by almost 3 dB.

B. Polarization-Multiplexed Coded-OFDM

In this sub-section we describe how to combine coded modulation with OFDM, which is illustrated in Fig. 11. The transmitter configuration up to the mapper is identical to that already described in Fig. 9. The 2-D signal constellation points [see Fig. 11(b)] are split into two streams for OFDM transmitters corresponding to the x- and y-polarizations. The QAM constellation points are considered to be the values of the fast Fourier transform (FFT) of a multicarrier OFDM signal. The OFDM symbol is generated as follows: N_{QAM} input QAM symbols are zero-padded to obtain N_{FFT} input samples for inverse FFT (IFFT), N_G nonzero samples are inserted to create the guard interval, and the OFDM symbol is multiplied by the Blackman-Harris window function. For efficient chromatic dispersion and PMD compensation, the length of cyclically extended guard interval should be longer than the total spread due to chromatic dispersion and DGD.

The cyclic extension is accomplished by repeating the last $N_G/2$ samples of the effective OFDM symbol part (N_{FFT} samples) as a prefix, and repeating the first $N_G/2$ samples as a suffix. After D/A conversion (DAC), the RF OFDM signal is converted into the optical domain using the dual-drive Mach-Zehnder modulator (MZM). Two MZMs are needed, one for

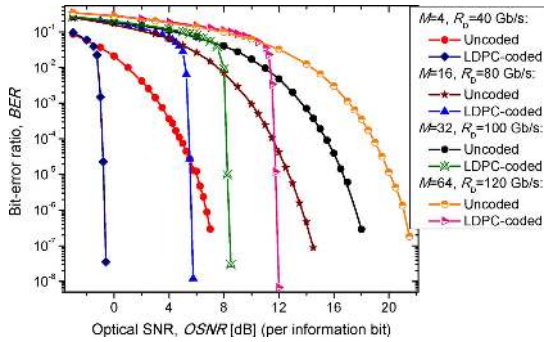

Fig. 12. BER performance of polarization multiplexed coded-OFDM, for DGD of 1200 ps. R_D denotes the aggregate data rate.

each polarization. The outputs of MZMs are combined using the polarization beam combiner (PBC). One DFB laser is used as CW source, with x- and y-polarization separated by polarization beam splitter (PBS).

The polarization-detector soft estimates of symbols carried by the kth subcarrier in the ith OFDM symbol, $s_{i,k,x(y)}$, are forwarded to the APP demapper, which determines the symbol LLRs $\lambda_{x(y)}(q)$ $(q=0,1,\ldots,2^b-1)$ of x- (y-) polarization by

$$\lambda_{x(y)}(q) = -\frac{\left(\mathrm{Re}\left[\tilde{s}_{i,k,x(y)}\right]-\mathrm{Re}\left[QAM\left(map\left(q\right)\right)\right]\right)^2}{\left(2\sigma^2\right)} - \frac{\left(\mathrm{Im}\left[\tilde{s}_{i,k,x(y)}\right]-\mathrm{Im}\left[QAM\left(map\left(q\right)\right)\right]\right)^2}{\left(2\sigma^2\right)} \quad (21)$$

where $\mathrm{Re}[]$ and $\mathrm{Im}[]$ denote the real and imaginary part of a complex number, QAM denotes the QAM-constellation diagram, σ^2 denotes the variance of an equivalent Gaussian noise process originating from ASE noise, and $\mathrm{map}(q)$ denotes a corresponding mapping rule. (b denotes the number of bits per constellation point.) Let us denote by $v_{j,x(y)}$ the jth bit in an observed symbol q binary representation $\boldsymbol{v}=(v_1,v_2,\ldots,v_b)$ for x- (y-) polarization. The bit LLRs needed for LDPC decoding are calculated from symbol LLRs in fashion similar to (18). The extrinsic LLRs are iterated backward and forward until convergence or predetermined number of iterations has been reached. The polarization-detector soft estimates can be obtained by employing: (i) polarization-time coding [32] similar to space-time coding proposed for use in MIMO wireless communication systems [33], (ii) using BLAST algorithm [34], (iii) by polarization interference cancellation scheme [34], or (iv) carefully performed channel matrix inversion [35].

In Fig. 12, we show both the uncoded and LDPC-coded BER performance of the polarization multiplexed LDPC-coded OFDM scheme from [35], against the polarization diversity OFDM scheme, for different constellations sizes. For DGD of 1200 ps, the polarization multiplexed scheme [35] performs comparable to the polarization-diversity OFDM scheme in terms of BER (the corresponding curves overlap each other), but it has two times higher spectral efficiency. The net effective coding gain increases as the constellation size grows. For $M=4$ QAM based polarization multiplexed coded-OFDM the net effective coding gain is 8.36 dB at BER of 10^{-7}, while for $M=32$ QAM based LPDC-coded OFDM (of aggregate data rate 100 Gb/s) the coding gain is 9.53 dB at the same BER.

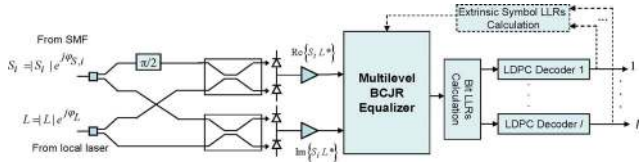

Fig. 13. LDPC-coded turbo equalization scheme configuration.

V. LDPC-Coded Turbo-Equalization (TE)

In this section we describe an LDPC-coded turbo equalization scheme [36], as a universal scheme that can be used simultaneously for: (i) suppression of fiber nonlinearities, (ii) PMD compensation, and (iii) chromatic dispersion compensation in multilevel coded-modulation schemes. The LDPC-coded turbo equalizer is composed of two ingredients: (i) the multilevel BCJR algorithm [36], [37] based equalizer, and (ii) the LDPC decoder. The transmitter configuration, for MLC, is already explained previous section [see Fig. 9(a)]. The receiver configuration of LDPC-coded trubo equalizer is shown in Fig. 13. The outputs of upper- and lower-balanced branches, proportional to $\mathrm{Re}\{S_iL^*\}$ and $\mathrm{Im}\{S_iL^*\}$ respectively, are used as inputs of multilevel BCJR equalizer, where the local laser electrical field is denoted by $L=|L|\exp(j\varphi_L)$ (φ_L denotes the laser phase noise process of the local laser) and incoming optical signal at time instance i with S_i.

The multilevel BCJR equalizer operates on a discrete dynamical trellis description of the optical channel. Notice that this equalizer is universal and applicable to any 2-D signal constellation such as M-ary PSK, M-ary QAM or M-ary polarization-shift keying (PolSK), and both coherent and direct detections. This dynamical trellis is uniquely defined by the following triplet: the previous state, the next state, and the channel output. The state in the trellis is defined as $\boldsymbol{s}_j=(x_{j-m},x_{j-m+1},\ldots,x_j,x_{j+1},\ldots,x_{j+m})=\boldsymbol{x}[j-m,j+m]$, where x_k denotes the index of the symbol from the following set of possible indices $\boldsymbol{X}=\{0,1,\ldots,M-1\}$, with M being the number of points in corresponding M-ary signal constellation. Every symbol carries $l=\log_2 M$ bits, using the appropriate mapping rule (natural, Gray, anti-Gray, etc.) The memory of the state is equal to $2m+1$, with $2m$ being the number of *symbols* that influence the observed symbol from both sides. An example trellis of memory $2m+1=3$ for 4-ary modulation formats (such as QPSK) is shown in Fig. 14. The trellis has $M^{2m+1}=64$ states $(\boldsymbol{s}_0,\boldsymbol{s}_1,\ldots,\boldsymbol{s}_{63})$, each of which corresponds to the different 3-symbol patterns (symbol-configurations).

The state index is determined by considering $(2m+1)$ symbols as digits in numerical system with the base M. For example, in Fig. 14, the quaternary numerical system (with the base 4) is used. (In this system 18 is represented by $(102)_4$.) The left column in dynamic trellis represents the current states and the right column denotes the terminal states. The branches are labeled by two symbols, the input symbol is the upper symbol of branch (the blue symbol), the output symbol is the central symbol of terminal state (the red symbol). Therefore, the current symbol is affected by both previous and incoming symbols. For the complete description of the dynamical trellis, the transition probability density functions (PDFs) $p(\boldsymbol{y}_j|x_j)=p(\boldsymbol{y}_j|\boldsymbol{s})$, $\boldsymbol{s}\in\boldsymbol{S}$ are needed; where $\boldsymbol{S}$ is the set of states in the trellis, and

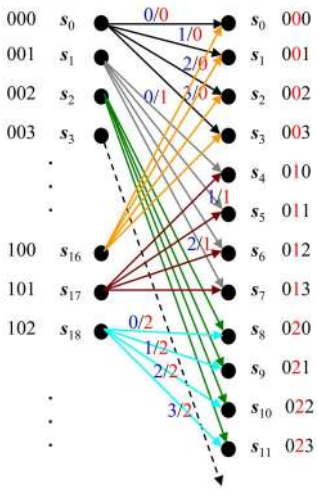

Fig. 14. Portion of trellis for 4-level BCJR equalizer with memory $2m+1=3$.

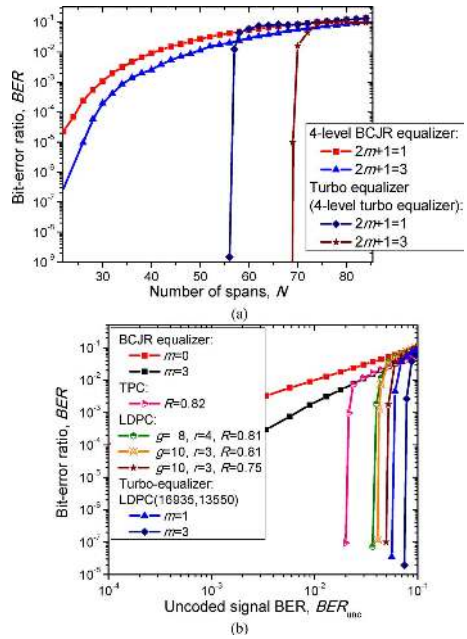

Fig. 15. BER performance of LDPC-coded turbo equalizer in the presence of fiber nonlinearities for: (a) QPSK modulation format with aggregate data rate of 100 Gb/s, and (b) RZ-OOK modulation format at 40 Gb/s. For both simulations, dispersion map shown in Fig. 16 is used.

$\boldsymbol{y}_j$ is the is the vector of samples (corresponding to the transmitted symbol index x_j). The conditional PDFs can be determined from *collected histograms* or by using *instanton-Edgeworth expansion* method [38]. The number of edges originating in any of the left-column states is M, and the number of merging edges in arbitrary terminal state is also M.

As an illustration of the potential of the proposed scheme, the BER performance of an LDPC-coded turbo equalizer is given in Fig. 15 for the dispersion map shown in Fig. 16 (launch power of 0 dBm and single channel transmission). EDFAs with a noise figure of 5 dB are deployed after every fiber section. The bandwidth of the optical filter is set to $3R_l$ and that of the electrical filter is set to $0.7R_l$, where $R_l = R_s/R$ with R_s being the symbol rate and R being the code rate (0.8). In Fig. 15(a), we

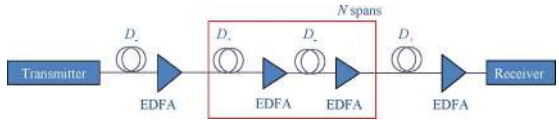

Fig. 16. Dispersion map under study is composed of N spans of length $L = 120$ km, consisting of $2L/3$ km of D_+ fiber followed by $L/3$ km of D_- fiber, with precompensation of -1600 ps/nm and corresponding postcompensation. The fiber parameters are given in Table II.

TABLE II
FIBER PARAMETERS

	D_+ fiber	D_- fiber
Dispersion [ps/(nm km)]	20	-40
Dispersion Slope [ps/(nm^2 km)]	0.06	-0.12
Effective Cross-sectional Area [μm^2]	110	50
Nonlinear refractive index [m^2/W]	$2.6\cdot10^{-20}$	$2.6\cdot10^{-20}$
Attenuation Coefficient [dB/km]	0.19	0.25

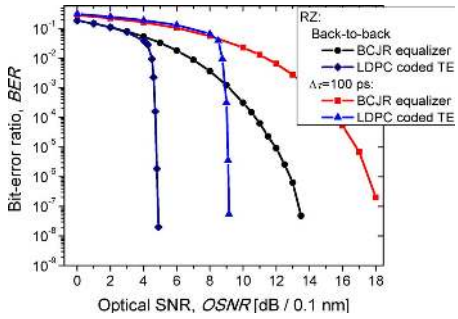

Fig. 17. BER performance of LDPC(16935,13550)-coded PMD TE with trellis memory $2m+1=7$.

present simulation results for QPSK transmission at the symbol rate of 50 Giga symbols/s. The symbol rate is appropriately chosen so that the effective aggregate information rate is 100 Gb/s. The figure depicts the uncoded BER and the BER after iterative decoding with respect to the number of spans, which was varied from 4 to 84. The propagation was modeled by solving the nonlinear Schrödinger equation using the split-step Fourier method. It can be seen from Fig. 15(a) that when a 4-level BCJR equalizer of state memory $2m+1=1$ and an LDPC(16935,13550) code of girth-10 and column weight 3 are used, we can achieve QPSK transmission at the symbol rate of 50 Giga symbols/s over 55 spans (6600 km) with a BER below 10^{-9}. On the other hand, for the turbo equalization scheme based on a 4-level BCJR equalizer of state memory $2m+1=3$ [see Fig. 15(a)] and the same LDPC code, we are able to achieve even 8160 km at the symbol rate of 50 Giga symbols/s with a BER below 10^{-9}. Notice that in both cases the BCJR equalizer trellis detection depth was equal to the codeword length. The BER performance comparison of LDPC-coded TE against large-girth LDPC codes and turbo-product codes for RZ-OOK system operating at 40 Gb/s (in effective information rate) is given in Fig. 15(b), for different trellis memories. LDPC-coded TE with state memory $2m+1=7$ provides almost 12 dB improvement over the BCJR equalizer with state memory of $m=0$ at BER of 10^{-8}.

In order to apply the proposed multilevel turbo equalizations scheme to real 100 Gb/s systems, the practical circuit implementation study would be mandatory. It is evident from Fig.

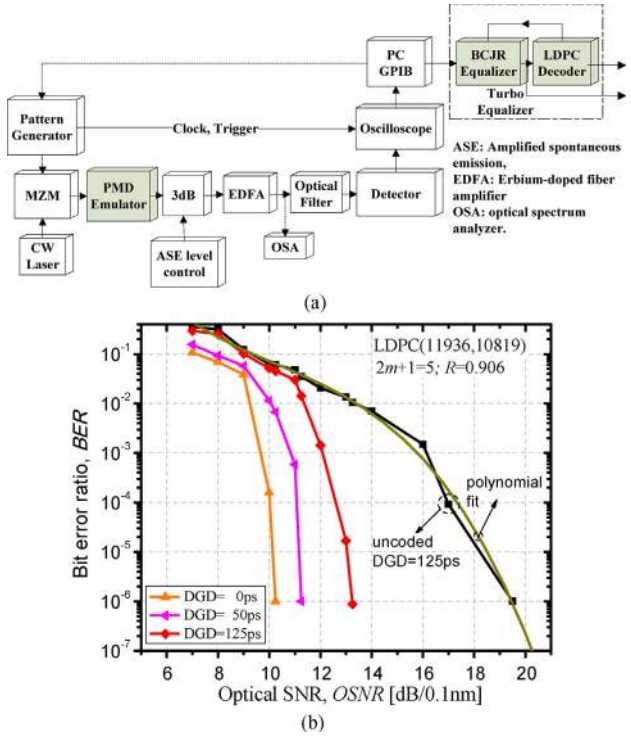

Fig. 18. (a) Experimental setup for PMD compensation study by LDPC-coded turbo equalization, and (b) BER performance of the PMD compensator.

13 that complexity of dynamic trellis grows exponentially, because the number of states is determined by M^{2m+1}, so that the increase in signal constellation leads to increase of the base, while the increase in channel memory assumption $(2m+1)$ leads to the increase of exponent. We have shown in the case of QPSK transmission [see Fig. 15(a)], that even small state memory assumption $(2m+1=3)$ leads to significant performance improvement with respect to the state memory $m=0$. For larger constellations and/or larger memories, a reduced complexity BCJR algorithm is to be used instead. For example, instead of detection of sequence of symbols corresponding to the length of codeword n, we can observe shorter sequences. Further, we do not need to memorize all branch metrics but several largest ones. In forward/backward metrics' update, we need to update only the metrics of those states connected to the edges with dominant branch metrics, and so on. Moreover, $\max^*(x,y)=\max(x,y)+\log[1+\exp(-|x-y|)]$ operation required in forward and backward recursion steps can be approximated by $\max(x,y)$ operation. Thus, forward and backward BCJR steps become the forward and backward Viterbi algorithms, respectively.

The nonlinear ISI turbo equalizer described above can also be used as a PMD compensator. The results of simulations, for 10 Gb/s transmission and ASE noise dominated scenario, are shown in Fig. 17 for a differential group delay (DGD) of $\Delta\tau = 100$ ps and a girth-10 LDPC code of rate 0.81. RZ-OOK of a duty cycle of 33% is observed. The bandwidth of super-Gaussian optical filter is set to $3R_l$, and the bandwidth of Gaussian electrical filter to $0.7R_l$, with R_l being the line rate. For DGD of 100 ps, the $R=0.81$ LDPC-coded turbo equalizer (for trellis memory $2m+1=7$) has a penalty of only 2 dB with respect to the back-to-back configuration.

In the rest of this section, we turn our attention to the experimental verification. The experimental setup for PMD compensation study by LDPC-coded turbo equalization is shown in Fig. 18(a). The LDPC-encoded sequence is uploaded into Anritsu pattern generator via GPIB card controlled by a PC. A zero-chirp Mach-Zehnder modulator is used to generate the NRZ data stream. The launch power is maintained at 0 dBm at the input of PMD emulator (with equal power distribution between states of polarization). The output of PMD emulator is combined with an ASE source immediately prior to the preamplifier. The ASE noise power is controlled by variable optical attenuator (VOA) in order to provide an independent optical signal-to-noise ratio (OSNR) adjustment at the receiver. A standard preamplified PIN receiver is used for direct detection and is preceded by another VOA to maintain a constant received power of -6 dBm. The sampling oscilloscope (Agilent), triggered by the data pattern, is used to acquire the received sequences, downloaded via GPIB card back to the PC which serves as an LDPC-coded turbo equalizer.

The experimental results for 10 Giga symbols/s (effective information rate) NRZ transmission are shown in Fig. 18(b), for different DGD values. The TE is based on a quasi-cyclic LDPC(11936,10819) code of code rate 0.906 and girth-10, with 5 outer and 25 sum-product decoding algorithm iterations. The OSNR penalty for DGD of 125 ps is about 3 dB at $\mathrm{BER}=10^{-6}$, while the coding gain improvement over BCJR equalizer (with memory $2m+1=5$) for $\mathrm{DGD}=125$ ps is 6.25 dB at $\mathrm{BER}=10^{-6}$. Larger coding gains are expected at lower BERs.

VI. Summary

In this invited paper, we described the large-girth binary LDPC code design, the min-sum-with-correction-term decoding algorithm and its FPGA implementation, and provided a class of nonbinary LDPC codes suitable for use in optical communications. We explained how to combine multilevel modulation and channel coding by using: (i) multilevel coding, and (ii) coded-OFDM. Furthermore, we described the LDPC-coded turbo-equalization scheme as a universal equalizer to deal simultaneously with fiber nonlinearities, PMD, and residual chromatic dispersion.

References

[1] E.-J. Essiambre, G. Raybon, and B. Mikkelsen, "Pseudo-linear transmission of high-speed TDM signals at 40 and 160 Gb/s," in *Optical Fiber Telecommunications IVB*, I. P. Kaminow and T. Li, Eds. San Diego, CA: Academic, 2002, pp. 233–304.

[2] G. P. Agrawal, *Nonlinear Fiber Optics*. San Diego, CA: Academic, 2001.

[3] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[4] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes," *IEEE Trans. Commun.*, no. 10, pp. 1261–1271, Oct. 1996.

[5] W. E. Ryan, "Concatenated convolutional codes and iterative decoding," in *Wiley Encyclopedia in Telecommunications*, J. G. Proakis, Ed. New York: Wiley, 2003.

[6] R. M. Pyndiah, "Near optimum decoding of product codes," *IEEE Trans. Commun.*, vol. 46, pp. 1003–1010, 1998.

[7] O. A. Sab and V. Lemarie, "Block turbo code performances for long-haul DWDM optical transmission systems," in *Proc. OFC*, 2001, vol. 3, pp. 280–282.

[8] T. Mizuochi *et al.*, "Forward error correction based on block turbo code with 3-bit soft decision for 10 Gb/s optical communication systems," *IEEE J. Sel. Topics Quantum Electron.*, vol. 10, no. 2, pp. 376–386, Mar. 2004.
[9] T. Mizuochi *et al.*, "Next generation FEC for optical transmission systems," in *Proc. Opt. Fib. Comm. Conf.*, 2003, vol. 2, pp. 527–528.
[10] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.
[11] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 9, pp. 533–547, Sep. 1981.
[12] D. J. C. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, pp. 399–431, 1999.
[13] S. Chung *et al.*, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon Limit," *IEEE Commun. Lett.*, vol. 5, pp. 58–60, Feb. 2001.
[14] I. B. Djordjevic, L. Xu, T. Wang, and M. Cvijetic, "Large girth low-density parity-check codes for long-haul high-speed optical communications," in *Proc. OFC/NFOEC*, San Diego, CA, 2008, JWA53.
[15] W. E. Ryan, "An introduction to LDPC codes," in *CRC Handbook for Coding and Signal Processing for Recording Systems*, B. Vasic, Ed. Boca Raton, FL: CRC, 2004.
[16] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, pp. 1788–1793, 2004.
[17] O. Milenkovic, I. B. Djordjevic, and B. Vasic, "Block-circulant low-density parity-check codes for optical communication systems," *IEEE/LEOS J. Sel. Topics Quantum Electron.*, vol. 10, pp. 294–299, Mar. 2004.
[18] H. Xiao-Yu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding of LDPC codes," in *Proc. IEEE Globecom*, Nov. 2001, vol. 2, pp. 1036–1036E.
[19] Y. Miyata, R. Sakai, W. Matsumoto, H. Yoshida, and T. Mizuochi, "Reduced-complexity decoding algorithm for LDPC codes for practical circuit implementation in optical communications," presented at the Optical Fiber Communication Conf., OWE5.
[20] M. Arabaci and I. B. Djordjevic, "An alternative FPGA implementation of decoders for quasi-cyclic LDPC codes," in *Proc. TELFOR*, Nov. 2008, pp. 351–354.
[21] Mitrion Users Guide,v1.5.0-001 ed. Mitrionics, Inc., 2008.
[22] M. Arabaci, I. B. Djordjevic, R. Saunders, and R. Marcoccia, "A class of nonbinary regular Girth-8 LDPC codes for optical communication channels," presented at the OFC/NFOEC, San Diego, CA, Mar. 22–26, 2009, JThA.
[23] M. C. Davey, "Error-Correction Using Low-Density Parity-Check Codes," Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 1999.
[24] C. Spagnol, W. Marnane, and E. Popovici, "FPGA implementations of LDPC over $\mathrm{GF}(2^m)$ decoders," in *Proc. IEEE Workshop on Signal Processing Systems*, Shanghai, China, 2007, pp. 273–278.
[25] A. Voicila, F. Verdier, D. Declercq, M. Fossorier, and P. Urard, "Architecture of a low-complexity non-binary LDPC decoder for high order fields," in *Proc. ISIT*, pp. 1201–1206.
[26] L. Lan, L. Zeng, Y. Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar, "Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: A finite field approach," *IEEE Trans. Inf. Theory*, vol. 53, pp. 2429–2458, 2007.
[27] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, pp. 1288–1299, 2005.
[28] I. B. Djordjevic and B. Vasic, "Multilevel coding in M-ary DPSK/differential QAM high-speed optical transmission with direct detection," *IEEE/OSA J. Lightw. Technol.*, vol. 24, no. 1, pp. 420–428, Jan. 2006.
[29] I. B. Djordjevic, M. Cvijetic, L. Xu, and T. Wang, "Using LDPC-coded modulation and coherent detection for ultra high-speed optical transmission," *IEEE/OSA J. Lightw. Technol.*, vol. 25, no. 11, pp. 3619–3625, Nov. 2007.
[30] I. B. Djordjevic and B. Vasic, "LDPC-coded OFDM in fiber-optics communication systems [Invited]," *OSA J. Opt. Netw.*, vol. 7, pp. 217–226, 2008.
[31] J. Hou, P. H. Siegel, L. B. Milstein, and H. D. Pfitser, "Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2141–2155, Sep. 2003.
[32] I. B. Djordjevic, L. Xu, and T. Wang, "PMD compensation in coded-modulation schemes with coherent detection using alamouti-type polarization-time coding," *Opt. Exp.*, vol. 16, no. 18, pp. 14163–14172.
[33] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2007.
[34] I. B. Djordjevic, L. Xu, and T. Wang, "PMD compensation in multilevel coded-modulation schemes with coherent detection using BLAST algorithm and iterative polarization cancellation," *Opt. Exp.*, vol. 16, no. 19, pp. 14845–14852, Sep. 2008.
[35] I. B. Djordjevic, L. Xu, and T. Wang, "Beyond 100 Gb/s optical transmission based on polarization multiplexed coded-OFDM with coherent detection," *IEEE J. Sel. Areas Commun., Optical Commun., Netw.*, vol. 27, no. 3, Apr. 2009.
[36] I. B. Djordjevic, L. L. Minkov, and H. G. Batshon, "Mitigation of linear and nonlinear impairments in high-speed optical networks by using LDPC-coded turbo equalization," *IEEE J. Sel. Areas Comm., Opt. Commun., Netw.*, vol. 26, no. 6, pp. 73–83, Aug. 2008.
[37] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 3, pp. 284–287, Mar. 1974.
[38] M. Ivkovic, I. Djordjevic, P. Rajkovic, and B. Vasic, "Pulse energy probability density Functions for long-haul optical fiber transmission systems by using instantons and edgeworth expansion," *IEEE Photon. Technol. Lett.*, vol. 19, no. 10, pp. 1604–1606, Oct. 2007.
[39] J. G. Proakis, *Digital Communications*. Boston, MA: McGraw-Hill, 2001.

Ivan B. Djordjevic (M'04) received B.S., M.S., and Ph.D. degrees in electrical engineering from University of Nis, Nis, Serbia, in 1994, 1997, and 1999, respectively.

He is an Assistant Professor of electrical and computer engineering at the University of Arizona, Tucson. Prior to this appointment in August 2006, he was with the University of Arizona (as a Research Assistant Professor); University of the West of England, Bristol, U.K.; University of Bristol, Bristol, U.K.; Tyco Telecommunications, Eatontown, NJ; National Technical University of Athens, Athens, Greece; and National Telecommunications Company "Serbia Telecom", Nis, Serbia. His current research interests include optical networks, error control coding, constrained coding, coded modulation, turbo equalization, OFDM applications, quantum error correction, and wireless communications. He directs the Optical Communications Systems Laboratory (OCSL) within the ECE Department, University of Arizona. He is an author of almost 100 journal publications and over 80 conference papers.

Dr. Djordjevic serves as an Associate Editor for *Research Letters in Optics* and as an Associate Editor for the *International Journal of Optics*.

Murat Arabaci (S'01) received the B.S. degree in electrical and electronics engineering from Osmangazi University, Eskisehir, Turkey, in 2003, and the M.S. degree in electrical engineering from the University of Arizona, Tucson, in 2006. He is currently pursuing the Ph.D. degree in electrical engineering at the University of Arizona.

His research interests include coding theory, communication theory, and information theory.

Lyubomir L. Minkov received the M.S. degree in electrical engineering from the Technical University of Sofia, Bulgaria. He is currently pursuing the Ph.D. degree in the Electrical and Computer Engineering Department, University of Arizona.

His current research interests include application of error correction coding in optical communication systems and modulation schemes in optical communications.