FPGA implementation of rate-adaptive spatially coupled LDPC codes suitable for optical communications

XIAOLE SUN* AND IVAN B. DJORDJEVIC

Department of Electrical and Computer Engineering, University of Arizona, 1230 E Speedway Blvd., Tucson, AZ 85719, USA
*sunxiaole@email.arizona.edu

Abstract: In this paper, we propose a unified field-programmable gate array (FPGA) structure for a rate-adaptive forward error correction (FEC) scheme based on spatially coupled (SC) LDPC codes derived from quasi-cyclic (QC) LDPC codes. We described the unified decoder structure in detail and showed that the rate adaptation can be achieved by a controller on-the-fly. By FPGA based emulation, the results show that, with comparable complexity, the SC codes provide larger coding gain. The implemented unified structure can be employed for any template QC-LDPC code to achieve a spatially-coupling based code-rate adaptation scheme.

© 2019 Optical Society of America under the terms of the OSA Open Access Publishing Agreement

1. Introduction

As the speed requirement of modern optical communication systems keeps increasing, forward error correction (FEC) becomes an essential technique to enable high-speed transmission ranging from long-haul to access networks. FEC with low-density parity-check (LDPC) codes has been studied for decades from LDPC block code to the most recent spatially-coupled (SC) LDPC code. SC-LDPC code as one of the asymptotically capacity-achieving codes was first introduced as LDPC convolutional code (LDPC-CC) in 1999 [1]. A simple approach to construct a LDPC-CC from a standard LDPC block code (LDPC-BC) was given. After the first introduction, many interesting research papers have been published on this promising type of LDPC codes. More recently, more thorough analysis has been presented to show the underlying theory of such new code. In [2], an iterative decoding threshold analysis for terminated regular LDPC convolutional codes was given for both binary erasure channel and AWGN channel. It was shown that the decoding thresholds for the LDPC-CC codes are not only better than the belief propagation decoding thresholds of corresponding regular LDPC block codes, but also even very close to the maximum likelihood (ML) decoding thresholds. A more comprehensive and general proof was given in [3] to show that the spatial coupling of individual codes increases the belief-propagation (BP) threshold of the new ensemble to come closer to its maximum a posteriori (MAP) threshold of the underlying ensemble. This phenomenon was termed as “threshold saturation via spatial coupling.”

On the other hand, to realize the superior performance of LDPC codes, researchers have been studied on the hardware architecture of such code through FPGA-based emulation and verification. In ref [4], an FPGA verification of quasi-cyclic (QC) LDPC codes has been done to show error floor below $10^{-15}$. Around the same time, a FPGA-based emulation has been presented in ref [5] for LDPC convolutional codes in high speed coherent optical transmission systems. More recently, related to the SC-LDPC codes, the authors in [6] presented a windowed decoding scheme for LDPC convolutional codes to provide the flexibility to set and change the decoding latency on the fly. The good performance of SC-LDPC codes at low BER range compared with polar codes, another capacity-achieving code, have been validated...
using an FPGA-based simulation [7]. More construction methods with hardware implementation architecture were proposed for this specific type of code in [8,9].

In this paper, we proposed a unified FPGA-based architecture for implementing SC-LDPC codes from a template QC-LDPC block code, which is suitable for high-speed implementation in optical communication systems. The unified structure leverages the hardware-friendly structure of a QC-LDPC code to realize the code rate-adaptation of the proposed coding scheme. The rate-adaptation capability provides the potential application in both free-space optical (FSO) and fiber-optics communication systems to deal with time-varying optical channels conditions. At the same time, the emulation results showed that the SC-LDPC codes provided extra SC coding gain, in addition to the coding gain improvement coming from code rate reduction.

The paper is organized as follows. In Section 2, we present the construction method of the proposed SC-LDPC codes that is derived from the template QC-LDPC code. In Section 3, we provide the detailed discussion on the implementation of the proposed LDPC coding scheme in which we show the overall structure of the unified structure. The emulation results, discussion and analysis are presented in Section 4. Finally, some relevant concluding remarks are provided in Section 5.

2. Construction of SC-LDPC codes derived from template QC-LDPC codes

In this section, we present a straightforward construction method of SC-LDPC codes, which extended from large-girth QC-LDPC block codes. We first describe the construction of large-girth QC-LDPC codes. The performance of the large-girth QC codes has been evaluated in our previous work [10]. Meanwhile, the QC-LDPC codes have low complexity structure which leads to an efficient encoder/decoder design. Then, based on the template QC-LDPC codes, we provide a simple method of constructing SC-LDPC codes from the template codes. The structure of such extended SC-LDPC codes will also share the low complexity structure as the template QC-LDPC code.

2.1 Construction of large-girth QC-LDPC codes

There are various algebraic or combinatorial methods to construct a structured QC-LDPC code. The design based on permutation matrices for QC-LDPC codes gets particular interests. The permutation matrices in the code parity-check matrix \( H \) leads to a memory efficient design for the decoding algorithm. The large-girth property gives a good waterfall region performance, since it provides a large minimum distance which is exponentially proportional to the girth. A QC-LDPC code, with column-weight \( J \) and row-weight \( K \), can be represented by a parity-check matrix \( H_{QC} \) as follows

\[
H_{QC} = \begin{pmatrix}
I(O_{0,0}) & I(O_{0,1}) & \cdots & I(O_{0,K-1}) \\
I(O_{1,0}) & I(O_{1,1}) & \cdots & I(O_{1,K-1}) \\
\vdots & \vdots & \ddots & \vdots \\
I(O_{J-1,0}) & I(O_{J-1,1}) & \cdots & I(O_{J-1,K-1})
\end{pmatrix},
\]

where \( I \) denotes a \( b \times b \) circulant permutation matrix, with \( O_{ij} \) representing the offset of each permutation matrix from an identity matrix. Given target values of \( J \) and \( K \), block size \( b \), and a desired girth \( g \), the offsets \( O_{ij} \) for each sub-matrix can be obtained which satisfy the Theorem 2.1 given in [11]. Column weight \( J = 3 \) LDPC codes are widely adopted in optical communication applications due to complexity reason. Therefore, for the template QC-LDPC block code, we construct a \( (J = 3, K = 15, b = 1129) \) code which has girth \( g = 8 \). The constructed code has a rate of \( R \geq 1 - J/K = 0.8 \). The codeword length is hence given by \( K \times b = 16935 \) bits, with \( K \times b - r = 13550 \) information bits, where \( r \) represents the rank of the parity check matrix \( H \).
2.2 Construction of SC-LDPC codes derived from template QC-LDPC code

The construction of the SC-LDPC code we introduce in this paper is straightforward from the template QC-LDPC code, which serves as a basic building block. By simply copy-and-shift operation, the block code is extended to a longer SC-LDPC code. To save the memory consumption for hardware implementation, we copy the base QC-LDPC block code for \( c \) times. Then the duplicated QC-LDPC code is shifted by multiple sub-blocks. Although the shift length can be arbitrary in the number of bits, we limit the shift operation to be a sub-block based for a memory efficient implementation. If we chose a coupling length of \( m = 5 \) with \( c = 3 \) of the constructed (3,15,1129) QC-LDPC codes, the result SC-LDPC code will be

\[
H_{sc} = \begin{bmatrix}
I(O_{0,0}) & \cdots & I(O_{0,16}) & \cdots & I(O_{0,14}) \\
I(O_{1,0}) & \cdots & I(O_{1,16}) & \cdots & I(O_{1,14}) \\
I(O_{2,0}) & \cdots & I(O_{2,16}) & \cdots & I(O_{2,14}) \\
I(O_{0,0}) & \cdots & I(O_{0,4}) & \cdots & I(O_{0,14}) \\
I(O_{1,0}) & \cdots & I(O_{1,4}) & \cdots & I(O_{1,14}) \\
I(O_{2,0}) & \cdots & I(O_{2,4}) & \cdots & I(O_{2,14}) \\
I(O_{0,0}) & \cdots & I(O_{0,4}) & \cdots & I(O_{0,14}) \\
I(O_{1,0}) & \cdots & I(O_{1,4}) & \cdots & I(O_{1,14}) \\
I(O_{2,0}) & \cdots & I(O_{2,4}) & \cdots & I(O_{2,14}) \\
\end{bmatrix}
\]

(2)

The codeword length of the SC-LDPC code will be \( b \times (c \times K - m \times (c-1)) \), and again \( c \) is the number of coupled template QC-LDPC codes and \( m \) is the coupling length in blocks.

3. FPGA implementation of SC-LDPC codes

As we described in section 2, the constructed SC-LDPC code will have the same structure as the template QC-LDPC code for each sub-code layer, and as such it is suitable for high-speed implementation. This property allows us to implement the SC-LDPC codes with comparable complexity as the QC-LDPC code with some straightforward modifications. Hence, we can build a re-configurable decoder with unified structure, in which we can choose between QC and SC-LDPC codes. The structure of the unified decoder is described in this section. In Fig. 1, the high-level schematic diagram of our FPGA-based emulation platform is given. It consists of an addictive white Gaussian noise (AWGN) generator module, the QC/SC-LDPC unified decoder engine, an error counter module, and a virtual input/output (VIO) module for BER monitoring purpose.

The simple copy-and-shift method allow us to select using either QC or SC LDPC codes for the decoder and set specific parameters for SC-LDPC codes without changing the main structure of decoder. The structure of the unified decoder is shown in Fig. 2. The unified decoder consists of a memory block storing \( H \) matrix (HBRAM) of the template QC-LDPC
code, a memory block storing received LLR (lvBRAM) for each codeword bit, an controller generating addresses for each processor, a set of variable node processors (VNPs), and a set of check node processors (CNPs) calculating the extrinsic messages, and a memory block storing the updated messages (lcvBRAM). For the template QC-LDPC code we selected in this paper, namely the (3,15,1129) QC-LDPC code, the HBRAM will be $15 \times 3 \times \log_2(1129) = 495$ bits since only the offset of each sub-matrix needs to be stored. For the SC-LDPC codes, because of the multiple instances of the $H_{QC}$ matrices, only one copy of the offsets is needed to represent $H_{SC}$. Therefore, the same HBRAM size also applies to SC-LDPC codes. In a more general case where the coupled code layers are not the same, which is known as time-variant SC-LDPC code, the HBRAM size will be $c$ times larger than the template code. In the unified decoder, the key component is the controller. The controller generates the proper valid signals based on the code configuration information and organizes the read and write addresses for each memory module. We adopt the layered scaled min-sum algorithm for decoding algorithm with scale factor $\alpha = 0.75$. In this scheme, the VNP will perform the sum operation while the CNP will perform a first two minimum values search operation. In each layer, the CNP will update the lcvBRAM with the most up-to-date information, so that, in the next layered iteration, the VNP will use the updated information from the previous layer. For the QC code, all the VNPs will have the same weights, e.g. each VNP sums three values for a CNP. However, for the SC-LDPC code, depending on the coupling length $m$, portion of the VNPs will have higher weights. For example, as shown in Fig. 3, a SC-LDPC code is composed of three (3,15,1129) QC-LDPC codes with coupling length $m = 5$. In this case, 10 out of 35 VNPs have weight 6 while the rest are of weight 3. Therefore, two types of VNPs are needed to perform additions. The controller will generate addresses for each VNP of connected check nodes and received LLR position from the channel. The process starts from a new codeword being stored in lvBRAM. The size of lvBRAM is given by (codeword length \times LLR resolution) bits. In our emulation, we choose 8 bits resolution to ensure that the degradation of code performance is negligible. For the lcvBRAM, we need to store the passing messages for the layered decoding algorithm. The size of lcvBRAM is then given by $J \times K \times b$ bytes for the QC-LDPC code. The lcvBRAM size for the SC-LDPC code is $c$ times larger than the corresponding template QC-LDPC code. Because there are $J \times K \times c$ non-empty sub-matrices in the $H_{SC}$ matrix, in this scheme, the controller needs to let VNP and CNP know the layer index or label of the current incoming values. Then the CNP needs to update the corresponding position in lcvBRAM. As illustrated in Fig. 4, by giving layer index to CNP and VNP, we do not need to reserve memory space for the empty sub-matrices. Therefore, the memory is independent of the coupling length $m$, which enables a unified architecture to implement multiple SC-LDPC codes. For a FPGA-based implementation, there are two types of energy dissipation contributors to the overall energy consumption: switching power from signal toggling/moving and leakage power from transistors. The leakage power is proportional to the area of the implemented circuit. Apart from the optimized memory organization described above, the processor units in the decoder were also optimized in terms of the resource utilization. The VNP is realized by a simple adder tree. The number of pipeline stages in the adder tree is dependent on the clock frequency. In our emulation, only one pipeline stage is used for a maximum nine 8-bits number addition to get highest throughput. Therefore, the clock frequency is set to 200 MHz to take the data from the front side to the end side while meeting the timing requirements of the design. The CNP is also implemented by a tree structure. In the CNP, the first minimum value and its index were found by a bottom to top search. Then the second minimum value was found by comparing values in the tree with the first minimum value. For the detail implementation structure, we refer an interested reader to ref [12]. To reduce the signal toggling, when the QC-LDPC code was chosen, all SC processors and corresponding area in lcvBRAM for the SC layers were disabled to prevent unnecessary data moving and storage.
4. Emulation results and discussion

We use the Kintex UltraScale kcu105 evaluation kit from Xilinx for our emulation platform. Modules shown in Fig. 1 are implemented in the programmable logic. In the emulation, we employ a uniform quantization scheme for all values in decoding algorithm. The emulation is conducted over binary AWGN channel assuming the BPSK modulation, which is reasonable
assumption for amplified spontaneous emission (ASE) dominated channels. To compare the performance, we set the maximum layered iteration number to be $I_{\text{max, QC}} = 36$ for the template QC-LDPC code and $I_{\text{max, SC}} = 108$ for the extended SC-LDPC codes. Therefore, each block layer in the code will be updated 12 times before a final decision was made for all codes. The bit-error rate (BER) vs. signal-to-noise ratio (SNR), expressed in dB scale, performances of the proposed LDPC codes are shown in Fig. 5. Once more, all the SC-LDPC codes are extended from $c = 3$ template $(3,15,1129)$ QC-LDPC codes.

As we can see from the Fig. 5, spatially coupled codes extended from the template QC-LDPC code provide higher coding gain. The gain is higher as the coupling length increases. It is worth to note that spatially coupling would induce code rate loss, which means the SC-LDPC codes will have lower rates. The effective code rates for $m = 3$, $m = 5$, and $m = 7$ SC-LDPC codes are 0.76, 0.733, and 0.7, respectively. To illustrate the coupling coding gain, the curves showed in Fig. 5 have already accounted for the code rate loss. In emulation results, we did not observe an error floor down to $10^{-14}$. There is a coupling gain ranging from 0.3 dB to 0.6 dB compared to the template QC-LDPC code. This unified decoder provides a feasible solution for a wide range of spatially coupled LDPC codes. Any terminated spatially coupled LDPC codes can be represented by a $H_{\text{sc}}$ matrix as Eq. (2) or Fig. 3 with different coupled code layers. The unified structure gives an efficient infrastructure to utilize the spatial-coupling structure of such code for a rate adaptive scheme.

5. Concluding remarks

We proposed a simple copy-and-shift method for construction of SC-LDPC codes derived from the large-girth template QC-LDPC codes. The complexity of the decoder of such code remains relatively low. The BER performance has been verified through FPGA-based emulation. The results have shown that the proposed SC-LDPC codes exhibit a superior performance over the template QC-LDPC code. From the implementation perspective, the structure of proposed SC-LDPC code is suitable for high-speed implementation and optical communication applications thanks to its low complexity and rate-adaptation capability.

![Fig. 5. FPGA emulation BER performance of proposed SC-LDPC codes.](image)

**Funding**

Office of Naval Research (ONR) MURI program (N00014-13-1-0627).
References


