AN AREA- EFFICIENT HIGH- THROUGHPUT LDPC DECODER ARCHITECTURE FOR 50G- PON SYSTEMS

ABSTRACT

LDPC codes are an appealing error correction mechanism for ensuring data integrity in new memory generations. This study describes a minimal hardware design for a high- speed Low- Density Parity Check (LDPC) decoder that is suitable for 50G Passive Optical Network (50G- PON).The suggested design aims to provide extremely fast decoding throughput with excellent error correction performance while reducing hardware complexity and resource usage. A phased decoding technique is used to optimise the trade- off between processing speed and implementation efficiency. The quasi- cyclic features of the LDPC parity- check matrix are employed to create a fixed and regular connection topology between check and variable nodes, greatly simplifying hardware routing. In addition, a block- level column cyclic shift mechanism is used on the parity- check matrix to decrease control overhead and boost parallelism. Simplified message normalization and quantization approaches are used to reduce computing complexity while maintaining decoding accuracy. A pipelined processing architecture is used to create the decoder, which allows for continuous high- speed operation. According to testing data, the proposed architecture is a viable and efficient option for next- generation 50G- PON systems, with a decoding throughput of 49.7664 Gbps on a single FPGA.

KEYWORDS: Low- density parity check (LDPC), 50G passive optical network (PON), parity- check matrix, bank allocation, and pipeline conflicts.

INTRODUCTION

The continuing rise of broadband applications has created a demand for optical access networks capable of supporting extremely high data rates while maintaining consistent transmission quality. In this context, 50G Passive Optical Network (50G-PON) technology has garnered significant attention since it allows for multi-gigabit connection in current access networks. Strong error-control techniques are required to provide consistent data transmission at such high rates, and The Low-Density Parity-Check (LDPC) codes are widely used because to their superior error-correction capabilities. Despite its benefits, developing LDPC decoders for 50G-PON systems poses significant design problems, notably in satisfying rigorous throughput requirements while reducing hardware complexity and resource consumption. Traditional decoder solutions sometimes entail substantial memory utilization, irregular connectivity, and high computational cost, limiting their usability in real-time, high-speed situations. As a result, adopting efficient architecture strategies—such as phased decoding methods, regularized interconnection topologies, and pipeline processing—is critical. An LDPC decoder architecture that is both simple and fast.designed for 50G-PON applications is presented in this study. The suggested solution greatly lowers implementation costs by utilizing the quasi-cyclic structure of LDPC codes and using streamlined message normalization and quantization techniques. Furthermore, a pipelined processing framework is employed to enhance decoding throughput without increasing hardware cost. The resulting architecture shows its potential for next-generation optical access networks by achieving nearly 50 Gbps decoding performance on a single FPGA platform.

Turbo codes and Low-Density Parity-Check (LDPC) codes are two types of near Shannon limit codes that can achieve extremely low bit error rates in low signal-to-noise ratio (SNR) situations.The LDPC decoding algorithm provides greater parallelization, reduced achievement complexity, short decoding delay, and In comparison to the Turbo code, there are no error floors at high SNRs decoding algorithm Among the most well-known Turbo codes and Low-Density Parity Check (LDPC) codes are similar to Shannon limit codes. The LDPC decoding method outperforms the Turbo code decoding technique in terms of parallelization. The complicated interconnects and high memory needs resulting from the sparse parity generator matrix are the main problems with the VLSI implementation of LDPC decoders. Based on recent research on structured LDPC codes, this paper suggests a low complexity design with lower memory requirements for LDPC decoding. Almost all of the upcoming communication standards take into account LDPC codes. Due to their advantages in developing a high throughput, low latency decoder, Low Density Parity Check Codes (LDPC), one of the Shannon limit codes, have gotten a lot of attention. For near Shannon limit performance, a suboptimal decoding technique called Sum of Product (SP) and its approximate counterpart called Offset Min-Sum (MS) have been developed. By eliminating the non-linear processes required in SP, offset MS lowers the decoding complexity.

Typically, iterative message-passing techniques are used to decode the LDPC codes. For LDPC decoding, there are two types of messages. scheduling schemes: sequential (layered) scheduling and flooded scheduling. According to studies, layered scheduling beats Traditional flooding scheduling for a high number of iterations and improves convergence time in terms of iteration

count. The majority of LDPC decoder implementations now in use rely on layered decoding. In contrast to turbo codes, LDPC codes can be written in a quasicyclic (QC) fashion to enable high throughput implementation.Fully parallelized architectures have been proposed and designs with multiple-Giga bps throughputs have been reported for certain standards. However, to maintain a reasonable die-size, for fully parallelized architectures, the underlying LDPC codes usually have to be short, such as the codes defined. There are always trade-offs between complexity and throughput/latency for moderate and lengthy codes.

The partially parallel architecture offers a decent compromise between hardware expense and throughput. The number of PUs decreases significantly compared to the fully parallel architecture since The PU is shared over numerous rows or columns. Determining which rows or columns are addressed in a PU is critical since decoding actions are concurrent in nature. In order to reduce the total number of cycles, the grouping should take into account the dependencies between rows and columns by overlapping the decoding operations. For quasi-cyclic LDPC codes, a heuristic scheduling approach has been described; however, it is not applicable to generic LDPC codes. An effective scheduling strategy for general LDPC codes is proposed in these articles. The idea of matrix permutation serves as the foundation for the suggested algorithm.

RELATED WORKS

The growing bandwidth and low latency demands of contemporary broadband and wireless services have significantly influenced the development of optical access networks. Early studies have highlighted the role of passive optical networks (PONs) in supporting high-speed communication, particularly for emerging 5G and beyond wireless infrastructures. In this regard, Kim et al. [1] highlighted the significance of effective physical-layer designs to satisfy demanding performance requirements by demonstrating a High-speed, low-latency PON architecture suitable for 5G systems.

With the progression toward higher data rates, 50G-PON has been standardized as a key next-generation access technology. Industry white papers and standardization documents, such as the ZTE 50G-PON white paper [2] and the ITU-T G.9804.3 recommendation [3], provide detailed specifications for physical layer requirements, including modulation formats, data rates, and forward error correction (FEC) schemes. These publications emphasize the importance of strong FEC mechanisms for ensuring reliable transmission under high-speed and long-distance operation circumstances.

Forward error correction has long been an important part of optical communication systems. Schmalen, et al. The transition from traditional block codes to more advanced approaches like as Low-Density Parity-Check (LDPC) codes was highlighted in [4], which offered a comprehensive overview of FEC algorithms used in Optical core and access networks. LDPC codes are particularly desirable because to their near-Shannon-limit performance, making them suitable for high-capacity optical communications.

Earlier optical transmission systems relied heavily on Reed-Solomon (RS) codes due to their easy

decoding structures and resilience. Buerner et al. [5] established the feasibility of hardware-based FEC at large data rates by proposing A high-speed RS-coded design for optical systems with a bandwidth of 43 Gbps.However, RS codes' limited error-correction capability and scalability difficulties rendered them less acceptable when data rates grew above tens of gigabits per second.Simpler FEC methods were adequate to satisfy system requirements in legacy PON standards like G-PON, which is defined in ITU-T G.984.3 [6].However, these approaches are no longer adequate for 50G-PON systems, where Higher spectral efficiency and more stringent error performance are required. In the end, LDPC-based FEC has been included in subsequent standards, prompting substantial research into decoder designs capable of supporting exceptionally high throughput while maintaining an acceptable hardware cost.

Although LDPC codes have great error-correction performance, practical implementation at 50G-PON data rates remains difficult. Conventional LDPC decoders sometimes need complicated interconnection networks, enormous memory requirements, and high computational complexity, making them unsuitable for real-time, high-speed optical access systems. In order to satisfy the strict criteria of next-generation PON standards, current research has increasingly concentrated on low-complexity, high-throughput LDPC decoder designs that take advantage of organized parity-check matrices, efficient decoding schedules, and pipelined processing.

PROBLE DEFINITION

Stricter constraints on data speed, latency, and transmission reliability have been brought about by the quick development of optical access networks, especially with the introduction of 50G Passive Optical Network (50G-PON) technology. Strong forward error correction algorithms, such as Low-Density Parity-Check (LDPC) codes, are utilized to meet these criteria. Although LDPC codes give excellent error correcting performance,there are substantial obstacles to its hardware implementation at extremely high data speeds.

The primary problem lies in designing an LDPC decoder that can sustain near-50 Gbps throughput while operating within the limited hardware resources of practical platforms such as Field-Programmable Gate Arrays (FPGAs). Conventional LDPC decoder architectures typically rely on complex interconnection networks, large memory buffers, and computationally intensive decoding schedules. These factors result in high area consumption, increased power usage, and reduced scalability, making them unsuitable for real-time 50G-PON applications.

Additionally, irregular message flow between check nodes and variable nodes restricts the possible clock frequency and adds routing complexity.Achieving high throughput often requires extensive parallelism, which further increases resource utilization and design complexity. In addition, high-precision message representation improves decoding accuracy but significantly increases hardware cost and processing latency.

Thus, the development of a low-complexity LDPC decoder architecture that minimizes hardware resource consumption and implementation complexity while meeting the incredibly high throughput needs of 50G-PON systems is the main issue addressed in this work.This includes

reducing interconnection irregularity, simplifying message computation and storage, and enabling efficient pipelined processing without degrading error-correction performance.

PROPOSED SYSTEM

In order to satisfy the demanding specifications of 50G Passive Optical Network (50G-PON) systems, this paper suggests a high-throughput and hardware-efficient LDPC decoder architecture. The suggested system aims to achieve ultra-high decoding speed while minimizing implementation complexity and resource use.

On the algorithmic level, a phased decoding method is used to optimize the decoding schedule. The design decreases simultaneous node updates by splitting the decoding process into well-defined phases, allowing for improved data flow control and memory access conflicts to be avoided. This method enables for effective utilization of computing resources while maintaining decoding performance.

To simplify hardware implementation, the proposed decoder uses the LDPC Parity-Check Matrix's Quasi-cyclic structure. The use of a fixed and regular connection configuration between changeable nodes and check nodes significantly reduces wire overhead and eliminates the need for elaborate routing networks. In addition, a block-wise column cyclic shift mechanism is added to the parity-check matrix, allowing for flexible decoding while maintaining a basic and hierarchical design.

The message-passing calculations are further improved with reduced normalization and quantization procedures. These strategies minimize arithmetic complexity and memory needs while retaining Reliable error correcting capabilities. Fixed-point representations are carefully chosen to balance decoding accuracy with hardware efficiency.

The overall architecture incorporates a deeply pipelined processing framework to support continuous data flow and high clock frequency operation. Pipeline stages are designed to overlap computation and memory access, thereby maximizing throughput without increasing hardware area. The suggested LDPC decoder is therefore a viable option for next-generation high-speed optical access networks since it can achieve near-50 Gbps decoding throughput on a single FPGA platform.

(2) a modified version for layered decoding updates, residue-based layered decoding, and (3) a comprehensive strategy based on a series of offline algorithms to improve the resource efficiency of layered scheduling LDPC decoders. Offline strategies for improving decoder hardware utilization efficiency in both message mapping and read access scheduling problems. FPGAs are used to develop a unique residue-based layered decoding approach that alleviates access scheduling limitations caused by pipeline issues. Our simulation findings show that the imprecision produced by errors in the offline prediction of the maximum energy improves the decoding capabilities.

LDPC Codes

LDPC codes are block codes that employ parity-check matrices with a limited number of non-zero elements apiece. H's sparseness guarantees both a linearly increasing decoding difficulty and a linearly increasing minimum distance. The only difference between an LDPC code and any other block code is that H must be sparse. In reality, if existing block codes can be represented by a spare parity-check matrix, they can be used with LDPC iterative decoding algorithms. However, it is typically impossible to evaluate a simple parity-check matrix for an existing code. Instead, LDPC codes are formed by first building a sparse parity-check matrix and then influencing a code-generation matrix. The main difference between conventional block codes and LDPC codes is how they are decoded. Classic block codes are often short and algebraically designed to make this task simpler because they are decrypted using ML-like decoding techniques. LDPC codes, on the other hand, are generated with H's characteristics in mind since they are decoded iteratively using a graphical representation of the parity check matrix.

BLOCK DIAGRAM

Fig 1 Block diagram

BLOCK DIAGRAM DESCRIPTION

CONSTRUCTION

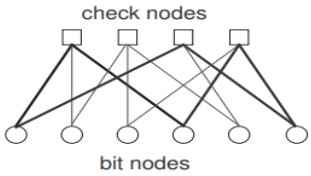


Fig 2 LDPC Construction

Gallager was able to get typical LDPC codes, which have a banded structure in H. A single column of H is added. Using this strategy, go from left to right one at a time. The non-zero entries in each column are chosen at random from rows that are not yet full, and the weight of each column is calculated to provide the right bit amount distribution. The row degree distributions for H will be less precise if there are rows at any stage with more. empty places than columns that need to be added. Until the precise row degrees are attained, the development might be restarted or reversed by a few columns.

LDPC Decoding Process

Iterative decoding of LDPC codes is achieved via a message passing technique. This approach checks nodes that are connected by edges in the bipartite network and exchanges belief information among the variable nodes. Let Ln be the trustworthy information for variable node n;

Ln,m is the information conveyed from variable node n to check node m; En,m is the extrinsic information formed in check node m and transmitted to variable node n; and In is the intrinsic information from the received signal. The belief data is deployed in two steps and regularly updated. To update Ln, the check nodes send the updated belief information (new En,m) to the variable nodes that are linked to them, while the variable nodes provide their belief information (Ln,m) to the check nodes in the first phase. In this case, N(m) represents the set of variable nodes in the bipartite graph that are linked to check node m. In a similar vein, M(n) is the set of check nodes connected to variable node n.

Variable Depth Pipeline

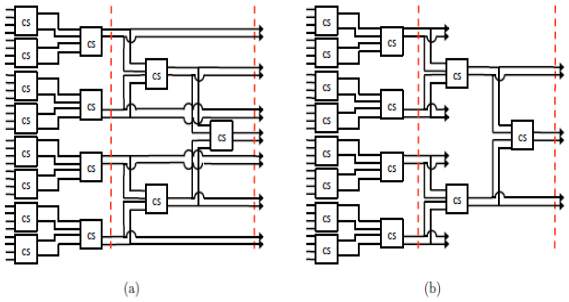


Fig 3 Variable depth pipeline

Since the check node requires many CS stages to compute its result, pipeline stages may be required if the row degree of any of the layers in at least one of the matrices is significant. The output must also be obtained early in the tree if any of the matrices have a large l. For the big l matrices, a pipeline stage can be eliminated by placing it at the same location where an output needs to be taken. The pipeline depth decreases the worst-case number of cycles required to decode a frame by varying according to the current code. This reduces the frequency of the decoder and conserves electricity.

VNU processing block

The system consists of m VNUs that process dv β messages in a serial way and output dv α messages in a serial fashion (one α message per clock cycle). The VNU processing blocks read and write m β messages in a single clock cycle. Each of the m VNUs handles the variable node and a-posteriori updates for a column in the B matrix.

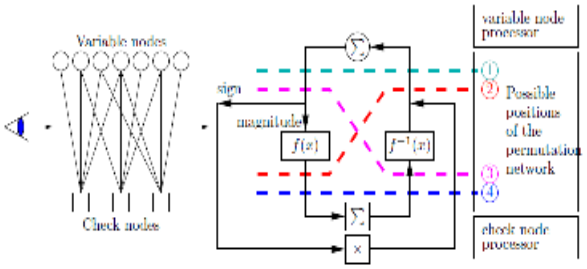


Fig 4 VNU processing block

XOR Encryption and Decryption

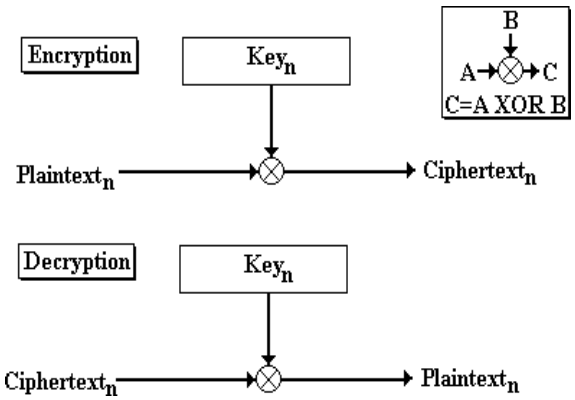


Fig 5 XOR encryption process

XOR encryption, also known as Exclusive-OR encryption is a popular method for encrypting books in a way that is difficult for the ordinary person to understand. The XOR encryption algorithm quite useful for storing data, such as laughter save details and additional statistics that are kind of stored locally on a user's computer. Although it wouldn't be a major concern if they were tampered with, you would want to deter citizens from doing so.

As a split of additional composite encryption techniques, XOR encryption is just as good. The concept is that it is difficult to identify either the novel disposition or the XOR encryption key if you are unable to do so. However, data nearly always contains patterns (JSON utilizes '{' and '}' characters, XML has plenty of '<' and '>' characters, etc.), so if a big shot decides on the prototype and undoes yet another nature, they will have the key to unlocking everything.This is why it is not completely safe. However, in reality, XOR encryption offers a wealth of useful applications. As long as security isn't the top priority, any deterrent that is added to data that users will have easy access to but You do not want them to meddle with it with is a perfect contender. The idea is simple: you name a key spirit and apply the key to each nature in the cord that you wish to encrypt. You just need to go through the string and use the key once again to decrypt the contents.

Residue Based Layered LDPC Decoder

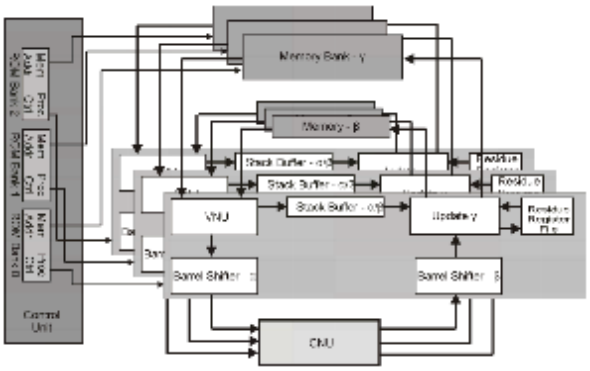


Fig 6 Residue Based Layered LDPC Decoder

In contrast to most layered decoding systems, which use AP-LLRs to route α and β signals, it uses distinct Processing units (VNU, CNU, and AP-LLR updates). Because α and β communications are less quantized than the AP-LLR messages, cost reductions are achieved in this manner. We employ basic architecture with reverse writeback.method because it provides easier control while developing decoders for irregular LDPC codes. A straightforward bidirectional shift register, the stack buffer works with using dc regular and quasi-regular codes (i.e., ±1 variation). The data-processing block uses pipelining.

OFF-LINE ALGORITHMS

The base graph G and the architecture-specific parameters (i) the nbanks parameter value, (ii) the write-back update strategy, (iii) the nlatency parameter value, and (iv) the maximum permitted.The memory message mapping and message read access scheduling procedures use the number of consecutive residual updates (n_) as input. Following the application of the off-line algorithms and optimization steps, the outputs include layer processing information (i.e., the number of memory accesses for each check-node in the layer order, as well as the number of stalls required after the check-node processing) and nbanks ROMs (one for each bank), which store the bank address, BS shift amounts, and valid bit information of each AP-LLR access according to the order specified by ψ. The three basic stages are outlined. The proposed technique does not rely on a single write-back methodology. To incorporate any updating technique, just overwrite the pd function computation.

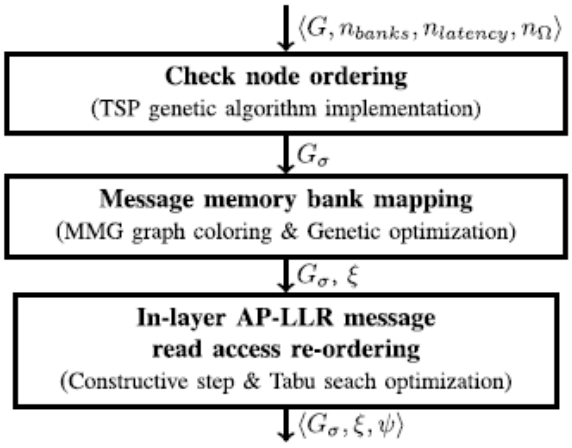
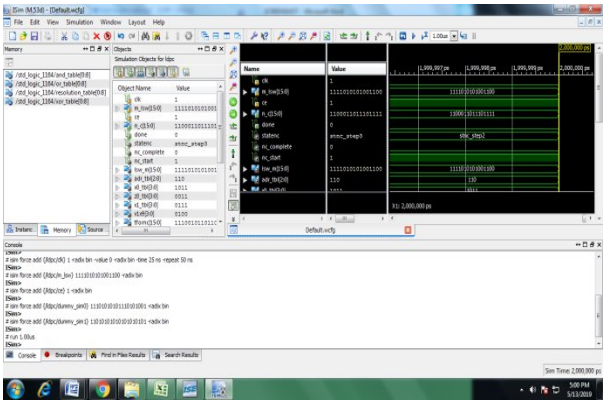


Fig 7 Offline process

ERROR CORRECTION AND PARITY CHECKS

Because we will only receive binary messages in this case, broadcast messages will consist of 0s and 1s. To construct a codeword for the message, forward error control coding requires supplementing these communication bits with intentional redundancy in the form of extra check bits. These check bits are required to ensure that codewords are sufficiently distinct from one another so that the receiver may correctly infer the message even if part of the codeword's bits are broken during transmission across the channel.

RESULT AND DISCUSSION



COMPARISON

PARAMETER	EXISTING SYSTEM	PROPOSED SYSTEM
EFFICIENCY	93.0%	95.6%
DELAY	4.103 ns	3.283ns
MEMORY	242256 kilobytes	187740 kilobytes
COMBILATION TIME	3.52 sec	3.52 sec
FREQUENCY	508.647 MHz	71.045 MHz
CLOCK PERIOD	1.966 ns	1.950 ns

CHART

CONCLUSION

This study demonstrated an efficient LDPC decoder architecture designed to meet the rigorous needs of 50G Passive Optical Network systems. The major goal was to obtain an extraordinarily high decoding throughput while reducing hardware complexity and resource use. The proposed design reduces implementation costs and simplifies interconnections by utilizing a phased decoding approach and The Quasi-cyclic nature of the LDPC Parity-Check Matrix.

Block-level cyclic shifting, together with enhanced message normalization and quantization

algorithms, reduces computing complexity while maintaining error-correction efficiency. In addition, a pipelined processing design allows for continuous high-speed operation with decoding throughput of almost 50 Gbps on a single FPGA device. The suggested system strikes an optimal compromise between performance, efficiency, and practicality, making it appropriate for next-generation optical access networks.

Overall, the findings show that hierarchical architectural optimization and simpler decoding algorithms may address the problems of ultra-high-speed LDPC decoding. The suggested decoder offers a scalable and resource-efficient solution that may be used to future high-capacity PON standards and other high-speed communication networks.

REFERENCE

[1] K. Kim, K.-H. Doo, H. H. Lee, S. Kim, H. Park, J.-Y. Oh, and H. S. Chung, ‘‘High speed and low latency passive optical network for 5G wireless systems,’’ J. Lightw. Technol., vol. 37, no. 12, pp. 2873–2882, Jun. 2019.

[2] ‘‘White paper on 50G-PON technology, V.2.0,’’ ZTE, White Paper, 2020. Accessed: Feb. 11, 2025. [Online]. Available: https://www.zte.com.cn/

[3] G.9804.3: 50-Gigabit-Capable Passive Optical Networks (50G-PON): Physical Media Dependent (PMD) Layer Specification. Accessed: Sep. 2021. [Online]. Available: https://www.itu.int/rec/T-REC-G.9804.3- 202109-I

[4] L. Schmalen, A. J. D. L. van Wijngaarden, and S. Ten Brink, ‘‘Forward error correction in optical core and optical access networks,’’ Bell Labs Tech. J., vol. 18, no. 3, pp. 39–66, Dec. 2013.

[5] T. Buerner, R. Dohmen, A. Zottmann, M. Saeger, and A. J. van Wijngaarden, ‘‘On a high-speed Reed–Solomon coded architecture for 43 Gb/s optical transmission systems,’’ in Proc. Int. Conf. Microelectron., Nis, Serbia, May 2004, pp. 743–746.

[6] G.984.3: Gigabit-capable Passive Optical Networks (G-PON): Transmission Convergence Layer Specification. Accessed: Jan. 2014. [Online]. Available: https://www.itu.int/rec/

[7] IEEE Standard for Ethernet—Amendment 9: Physical Layer Specifications and Management Parameters for 25 Gb/s and 50 Gb/s Passive Optical
Networks, IEEE IEEE Standard 802.3ca, 2020.

[8] R. G. Gallager, Low-Density Parity-Check Codes. Cambridge, MA, USA:MIT Press, 1963.

[9] T. J. Richardson and R. L. Urbanke, ‘‘The capacity of low-density paritycheck codes under message-passing decoding,’’ IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[10] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, ‘‘Design of capacity-approaching irregular low-density parity-check codes,’’ IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[11] 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.

[12] G. D. Forney, ‘‘Codes on graphs: Normal realizations,’’ IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 520–548, Feb. 2001.

[13] E. Sharon, S. Litsyn, and J. Goldberger, ‘‘An efficient message-passing schedule for LDPC decoding,’’ in Proc. 23rd IEEE Conv. Electr. Electron.
Engineers, Tel Aviv-Yafo, Israel, Nov. 2004, pp. 223–226.

[14] D. Hocevar, ‘‘A reduced complexity decoder architecture via layered decoding of LDPC codes,’’ in Proc. IEEE Workshop Signal Process. Syst.,
Austin, TX, USA, Oct. 2004, pp. 107–112.

[15] P. Radosavljevic, A. de Baynast, and J. R. Cavallaro, ‘‘Optimized message passing schedules for LDPC decoding,’’ in Proc. 39th Asilomar Conf. Signals, Syst. Comput., Pacific Grove, CA, USA, 2005, pp. 591–595.