International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on12th Jun 2021. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-06

DOI: 10.48047/IJIEMR/V10/I06/01

Title **Analysis of Energy-Efficient High-Throughput VLSI Architectures for Product-Like Codes**

Volume 10, Issue 06, Pages: 1-7

Paper Authors

PIDETI MADHAVI, Mrs. U.KRUPA, Dr T Raghavendra Vishnu

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Analysis of Energy-Efficient High-Throughput VLSI Architectures for Product-Like Codes

PIDETI MADHAVI
M-tech student Scholar Department of Electronics and Communication Engineering, Priyadarshini Institute Of Technology & Science For Women, chintalapudi Village, Duggirala Mandal, Near Tenali, Guntur, Andhra Pradesh, India.
Mrs. U.KRUPA M.Tech
Assistant professor Department of Electronics and Communication Engineering, Priyadarshini Institute Of Technology & Science For Women, chintalapudi Village, Duggirala Mandal, Near Tenali, Guntur, Andhra Pradesh, India.
Dr T Raghavendra Vishnu M.tech, Ph.D.,
Associate Professor & HOD Department of Electronics and Communication Engineering, Priyadarshini Institute Of Technology & Science For Women, chintalapudi Village, Duggirala Mandal, Near Tenali, Guntur, Andhra Pradesh, India.

Abstract: Implementing forward error correction (FEC) for modern long-haul fiber-optic communication systems is a challenge, since these high-throughput systems require FEC circuits that can combine high coding gains and energy-efficient operation. We present VLSI decoder architectures for product-like codes for systems with strict throughput and power dissipation requirements. To reduce energy dissipation, our architectures are designed to minimize data transfers in and out of memory blocks, and to use parallel non-iterative component decoders. The proposed results show that the architectures achieve better performance compared with the previous architectures.

Key words: Forward Error Correction (FEC), ACS block, Clock gating technique, Lookup table architectures, Throughput, Transmission energy.

I. INTRODUCTION

VLSI technology defined the edge of the ASIC business, which accelerated the push of powerful embedded systems in to adorable products. A Wireless Sensor Network is spatially distributed autonomous to monitor physical or environmental conditions in order to send their data through the network to a main location. But, designing the decoder on the receiver side encounters many challenges like algorithm complexity, large silicon area, high power consumption and low throughput. The proposed work is concentrated on designing the parallel turbo decoder which fits for the Wireless Sensor network applications. Turbo codes are on the verge of finding their way in numerous cutting edge applications like cellular communications and satellite communications. A turbo decoder is composed of modules that work in an iterative scheme. Some alternative algorithms exist for the algorithm incorporated in these decoder modules. Recent Application-Specific Integrated Circuit (ASIC) based decoder architectures have been designed [1] for achieving a high transmission throughput, rather than for low transmission energy. This addresses design and implementation aspects of parallel turbo decoders that reach the 326.4 Mb/s peak data-rate using multiple soft-input soft-output decoders that operate in parallel. The Error-Control Coding (ECC) is implemented in Wireless Sensor Networks (WSN), to determine the energy efficiency of the specific ECC implementations in WSN are been proposed [2]. ECC provides coding gain, resulting in transmission energy savings, at

the cost of added decoder power consumption but is not an error-control code, as there is no encoding. C.Schlegel (2011) defined the physical layer (PHY) and Medium Access Control (MAC) sub layer specifications for less data rate wireless connectivity with fixed, portable, and moving devices with no battery or very limited battery consumption requirements. The energy consumption of Amplitude Shift Key(ASK) is (i.e.14.013 mJ) less than Offset Quadrature Phase Shift Key(OQPSK) (i.e. 15.401 mJ) but due to its high sensitivity to noise this modulation scheme is rejected and OQPSK is chosen[3]. A modified version of the BCJR algorithm, that has redesigned vigorously is also used [4]. There are several simplified versions of the Maximum A posteriori Probability (MAP) algorithm, namely the log-MAP and the max-log-MAP algorithms. But in this case power consumption is high and it is sensitive to signal that produces more noise. J.M.Mathana (2014) presents VLSI architecture for an efficient turbo decoder using sliding window method. The speed of the implemented architecture is improved by modifying the value of the branch metrics [5]. But this method occupies a large amount of chip area.A low-complexity ACS (add compare and select) architecture is introduced in WSN decoder design. The entire decoder architecture is coded using Verilog HDL and it is synthesized using Xilinx EDA with Spartan 3E FPGA[6]. But this design is said to be more complicated and requires more cost because of large usage of components.

At the core of our product and staircase decoders we find non-iterative component decoders, which realize boundeddistance decoding of shortened binary Bose-ChaudhuriHocquenghem (BCH) codes [12] with error-correction capabilities in the range of 2–4. These component decoders are fully parallel and strictly feed-forward, which means that internal state registers can be avoided. As will be shown, this is key to high throughput and low latency. We will, e.g., showcase VLSI implementations of a number of staircase codes that are relevant for fiber-optic communication systems [7]. The implementations are capable of achieving in excess of 1- Tb/s information throughput, which is significantly higher than those of currently published state-of-the-art FEC implementations [13]–[17]. While high throughput requirements typically make power dissipation a serious design concern, this is not the case for our decoders, which dissipate only 1.3–2.4 W (or around 2 pJ/bit), depending on configuration and assumed input bit-error rate. Before we describe the VLSI architectures and implementations of product and staircase decoders, we will briefly introduce concepts pertaining to the FEC codes used: As component code, we use BCH(n, k, t), where n is the block length, k is the number of useful information bits, and t is the number of bit errors that the code can correct. Given that GF(2m) is the Galois field in which computations are performed, the BCH code parameters are related as $n = 2m-1$ and $n-k = mt$. In addition to the definitions above, the code rate, which is the proportion of a block that contains useful information, is defined as $R = k$ n , while the code overhead is defined as $OH = n-k$ $k = 1$ $R - 1$.

BCH codes can be shortened to allow for more flexibility in terms of code rate, i.e., this is a tradeoff between coding gain and information throughout. Shortening means a number of information-bit positions are fixed to zero and never transmitted, with the result that the code overhead is increased from the initial non-shortened BCH code. Hereon, we denote the block length and the number of information bits in the shortened codes as $ns = n - s$ and $ks = k - s$, respectively, where s is the number of removed bits.

II. COMPONENT DECODERS

Since the component decoders are central to efficient VLSI implementation of product-like codes, we will first introduce the BCH component decoders, with emphasis on the key equation solver. Variants of these decoders have been used in our previous work: We described 1- and 2-error correcting decoders in [18], while more advanced 3- and 4-error correcting implementations were used (but not described) in [11]. A typical BCH decoder employs syndrome calculation, the Berlekamp-Massey (BM) algorithm (to find the error-location polynomial), and Chien search (to find the errors). Different optimizations of the BM algorithm, such as the simplified inverse-free BM (SiBM) algorithm [19], can improve the implementation, however, a fundamental problem is that con ventional BM implementations are iterative and require at least t clock cycles to complete their operation.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

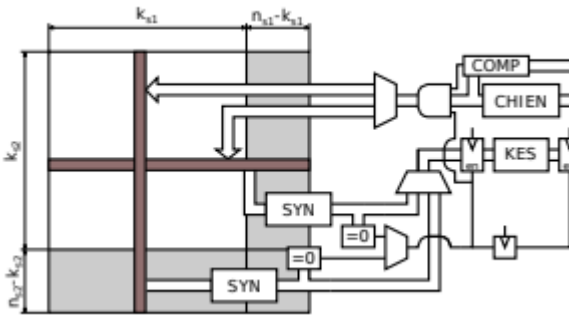

Figure .1: BCH component decoders used in a product-decoder architecture.

Aiming for high throughput and low latency, we have developed fully parallel BCH component decoders based on the direct-solution Peterson algorithm [20]. Fig. 1 shows a product-decoder architecture in which the BCH component decoders—comprising SYN, KES (see Section III-A) and CHIEN units—are integrated. (The memory block can just as well belong to a staircase decoder.) The component decoders are non-iterative and strictly feed-forward, thus, simplifying state-machine design and allowing for synchronous clock gating of a component decoder's pipeline. The decoders are implemented using bit-parallel polynomial-base GF(2m) multipliers [21]. Error-free component data are detected after the first stage in the decoder pipeline, i.e., the syndrome calculation stage. To save power, we use several techniques: If all syndromes are zero, the pipeline is gated sequentially and a flag is set to allow for memory-block gating. Each column and row in the product/staircase code uses separate syndrome calculation units to reduce logic signal switching (see Section IV), while the KES and Chien search units are shared between a row/column pair.

III. DECODER ARCHITECTURE OVERVIEW

After initially moving the received bits into the memory block closest to the channel, product and staircase codes are decoded using an iterative scheme. For product decoders, an iteration consists of two phases: With reference to Fig. 1, first all rows are decoded and errors are corrected, after which all columns are decoded and errors are corrected. This procedure is straightforwardly repeated for a given number of iterations. In staircase decoding, the iteration scheme is more complex in that each component code covers two spatially coupled blocks, as shown in the simplified decoder configuration in Fig. 2 which operates on a window of 3 blocks. Additional component decoders can be chained over the decoder window, as shown for the more useful 5-block decoder configuration in Fig. 3. Similar to the product decoding above, the staircase algorithm [6] entails decoding of rows followed by columns: After channel data have been moved into memory block 1, parallel decoding of all rows and columns of that block is followed by decoding of all rows and column in the next memory block. Successively all memory blocks get decoded and once this is completed, a new decoding phase commences in memory block 1. This process is repeated for a given number of iterations and once all iterations are completed, the data of memory block 1 are moved to memory block 2 and new channel data are moved into memory block 1, thus, shifting the decoding window. We now perform syndrome recomputation in order to avoid separate storage and multiplexers to move syndromes with the data block. (Overall, syndrome computation contributes to at most 10 % of total power dissipation in all considered configurations.)

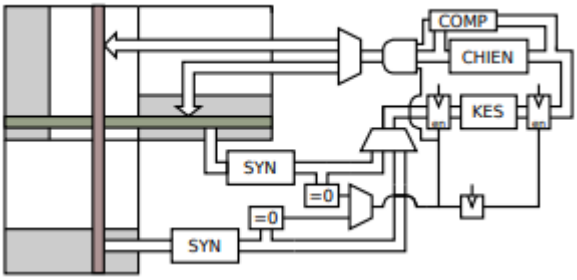

Figure 2: A 3-block staircase memory array supported by component decoders.

Since there are no inter column dependencies nor inter-row dependencies during each half iteration, our staircase decoder architecture first iterates over all rows in the entire window, then over all columns in the window. This reduces iteration time and increases throughput compared to decoders that operate on memory blocks sequentially in the decoder window. No significant difference in error-correction performance of staircase codes was found when comparing two MATLAB reference implementations. As switching power dissipation is proportional to the signal switching, we use replicated syndrome computation units for all decoders. By assigning one syndrome computation unit to rows and another unit to columns, fewer signals switch, since at most t bits are flipped in each row/column per iteration the majority of the XOR-gates in the syndrome computation are thus kept static reducing switching power dissipation. Consider the case of correcting one bit: One flipped bit causes at most dlog2 (n)e toggles in each of the XOR-trees for syndrome calculation. The KES and

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Chien search (the decoder back-end) units are shared between rows and columns.

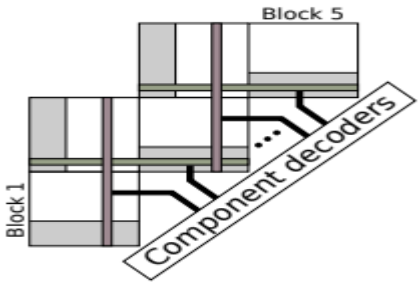

Figure 3: A 5-block staircase memory array with attached component decoders. The lower block (Block 1) is the memory block closest to the channel.

Row and column syndromes indicate presence of believed errors in corresponding rows and columns. In the product decoder, the codeword is believed to be correct once all syndromes are zero. If this is the case, the memory block is clock gated to save power. The state-based clock gating is somewhat more complex in the staircase memory array: Beside gating each memory block, if no errors are found within that block during component-code decoding, the whole staircase array is gated during component-code decoding and only clocked on write-back or window shifting. Since each full row or column is read out and decoded fully block parallel in these decoder architectures, we obtain decoders with very high throughput and low processing latency. Another advantage of using component decoders without any feedback loops is that the control state-machine implementation is straightforward.

IV. SIMULATION RESULTS

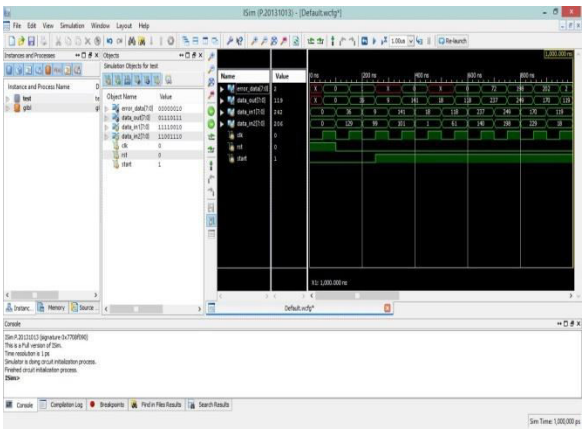

Figure 4: Simulation result of the product decoder architecture

Fig 4 shows the simulation result of the product decoder architecture in which input is given through the data_in signals and the output is observed in the data_out signal and the error data is observed in the error_data signal. With the help of error data signal we can easily find out whether the received data is correct or not in the decoder architecture.

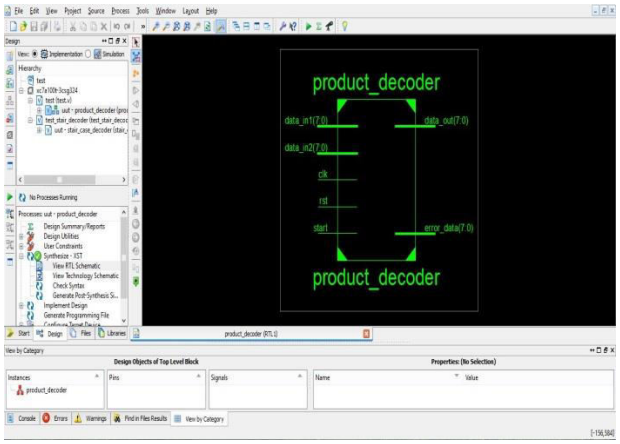

Figure 5: Block diagram of the product decoder architecture

Fig 5 shows the block diagram of the product decoder architecture. It is generated by synthesizing the design. It shows the number of input and output ports are used in the design.

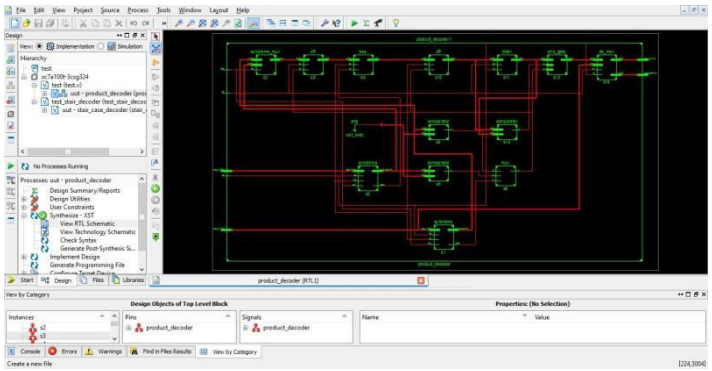

Figure 6: RTL schematic of the product decoder architecture

Fig 6 shows the RTL schematic of the product decoder architecture in which it shows the internal structure of the design.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Figure 7: Technology schematic of the product decoder architecture

Fig 7 shows the technology schematic of the product decoder architecture. In this the design is implemented with the help of LUT's only.

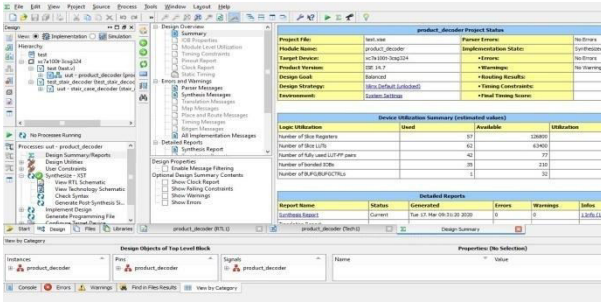

Figure 8: Summary report of the product decoder architecture

Fig 8 presents the summary report of the product decoder architecture in which it shows the count of the number of slice registers, LUT's and number of IOB's are utilized in the respective FPGA family

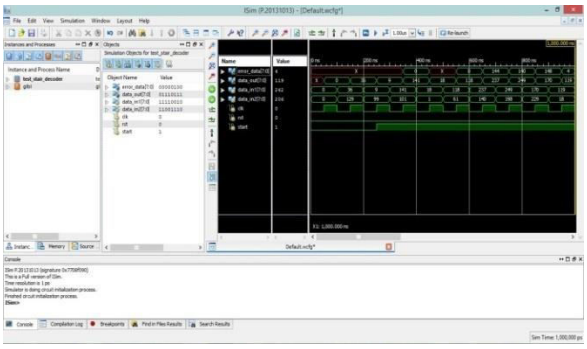

Figure 9: Simulation result of the stair case decoder architecture

Fig 9 shows the simulation result of the stair case decoder architecture in which input is given through the data_in signals and the output is observed in the data_out signal and the error data is observed in the error_data signal. With the help of error data signal we can easily find out whether the received data is correct or not in the decoder architecture.

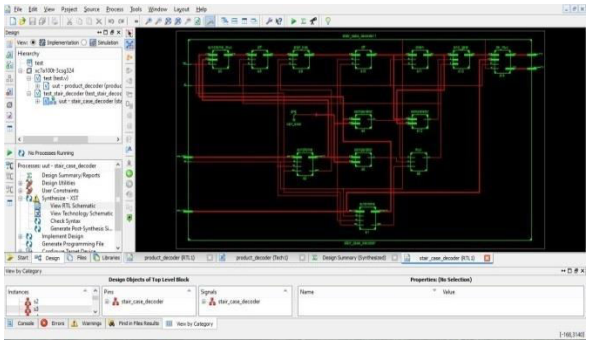

Figure 10: RTL schematic of the stair case decoder architecture

Fig 10 shows the RTL schematic of the staircase decoder architecture in which it shows the internal structure of the design.

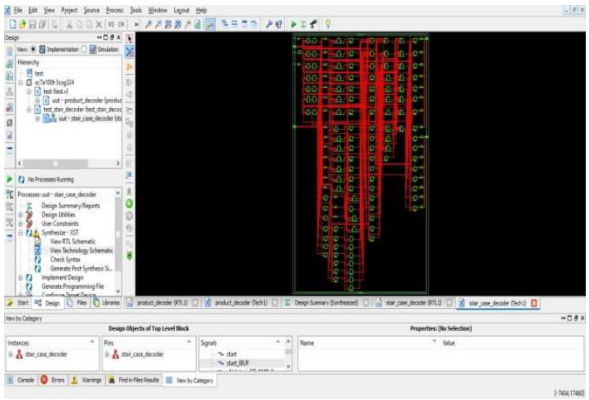

Figure 11: Technology schematic of the stair case decoder architecture

Fig 11 shows the technology schematic of the stair case decoder architecture. In this the design is implemented with the help of LUT's only.

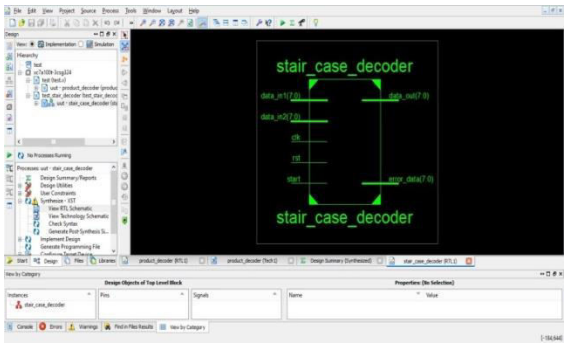

Figure 12: Block diagram of the stair case decoder architecture

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Fig 12 shows the block diagram of the stair case decoder architecture. It is generated by synthesizing the design. It shows the number of input and output ports are used in the design.

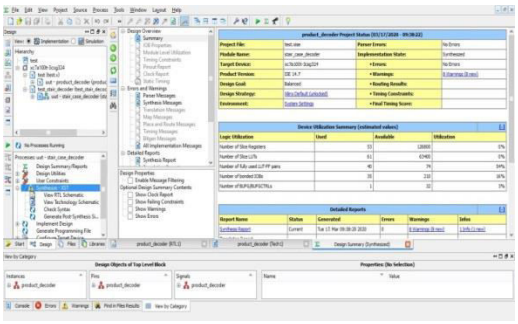

Figure 13: Summary report of the stair case decoder architecture

Fig 13 presents the summary report of the stair case decoder architecture in which it shows the count of the number of slice registers, LUT's and number of IOB's are utilized in the respective FPGA family.

Table: 1 Summary report

Name of the Component	LUT	Delay	Registers
product-decoder architecture	62	4.081 ns	57
Staircase component decoders	61	4.045 ns	53

V. CONCLUSION

We have implemented energy-efficient high-throughput VLSI decoders for product and staircase codes, power constrained fiber optic communication systems. The decoders have been implemented and evaluated using Xilinx ISE, allowing us to consider aspects of energy efficiency and related tradeoffs. The staircase decoders have a block-decoding latency of 483.6 ns, while the product decoder latencies are 64 ns. Effective use of clock gating to inhibit signals from switching is shown to significantly reduce energy dissipation of iterative decoders, both in memory blocks and in component decoders. All considered product and staircase decoders are estimated to dissipate less power, demonstrating the viability of high-throughput hard decision product and staircase decoders.

REFERENCES

[1] R.-J. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini, and B. Goebel, "Capacity limits of optical fiber networks," J. Lightwave Technol., vol. 28, no. 4, pp. 662–701, Sept./Oct. 2010.

[2] R.-J. Essiambre, G. Kramer, G. J. Foschini, and P. J. Winzer, "Optical fiber and information theory," in Conf. on Inform. Sciences and Systems, Princeton, NJ, 2010.

[3] K. Kikuchi, "Coherent optical communication systems," in Optical Fiber Telecommunications V-B, I. Kaminow and T. Li and A. Willner, Eds. New York: Academic, 2008, pp. 95–129.

[4] B. Spinnler, "Equalizer design and complexity for digital coherent receivers," IEEE J. Select. Topics Quantum Electron., vol. 16, no. 5, pp. 1180–1192, Sept./Oct. 2010.

[5] Y. Benlachtar et al., "Real-time digital signal processing for the generation of optical orthogonal frequency-division-multiplexed signals," IEEE J. Select. Topics Quantum Electron., vol. 16, no. 5, pp. 1235–1244, Sept./Oct. 2010.

[6] P. J. Winzer and R.-J. Essiambre, "Advanced optical modulation formats," in Optical Fiber Telecommunications V-B, I. Kaminow and T. Li and A. Willner, Eds. New York: Academic, 2008, pp. 23–93.

[7] T. Mizuochi, "Recent progress in forward error correction and its interplay with transmission impairments," IEEE J. Select. Topics Quantum Electron., vol. 12, no. 4, pp. 544–554, Jul. 2006.

[8] I. B. Djordjevic, M. Arabaci, and L. L. Minkov, "Next generation FEC for highcapacity communication in optical transport networks," J. Lightwave Technol., vol. 27, no. 16, pp. 3518–3530, Aug. 2009.

[9] G. P. Agrawal, Fiber-Optic Communication Systems. San Diego, CA: WileyInterscience, 1997.

[10] Nonlinear Fiber Optics. New York, NY: Elsevier Science & Technology, 2006.

[11], Lightwave Technology. Hoboken, NJ: Wiley-Interscience, 2005.

[12] A. Mecozzi, "Limits to long-haul coherent transmission set by the Kerr nonlinearity and noise of the in-line amplifiers," J. Lightwave Technol., vol. 12, no. 11, pp. 1993– 2000, Nov. 1994.

[13] P. Elias, "Error-free coding," IRE Trans. on Inform. Theory, vol. PGIT-4, pp. 29– 37, 1954.

[14] F. R. Kschischang, "Product codes," in Wiley Encyclopedia of Telecommunications, J. Proakis, Ed., 2003.

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

[16] R. M. Tanner, "A recursive approach to low complexity codes," IEEE Trans. Inform. Theory, vol. 27, no. 5, pp. 533–547, Sep. 1981.

[17] M. Lentmaier and K. S. Zigangirov, "On generalized low-density parity-check codes based on Hamming component codes," IEEE Commun. Lett., vol. 3, no. 8, pp. 248–250, Aug. 1999.

[18] S. Lin and D. J. Costello Jr., Error Control Coding. Upper Saddle River, NJ: Prentice Hall, 2004.

[19] E. R. Berlekamp, Algebraic Coding Theory. New York, NY: McGraw-Hill, 1968.

[20] R. T. Chien, "Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes," IEEE Trans. Inform. Theory, vol. IT-10, pp. 357–363, Oct. 1964.

[21] I. S. Reed and X. Chen, Error-Control Coding for Data Networks. Boston, MA: Kluwer Academic Publishers, 1999.

[22] E. R. Berlekamp, H. Rumsey, and G. Solomon, "On the solution of algebraic equations over finite fields," Inform. Contr., vol. 10, pp. 553–564, 1967.

[23] R. T. Chien, B. D. Cunningham, and I. B. Oldham, "Hybrid methods for finding roots of a polynomial with application to BCH decoding," IEEE Trans. Inform. Theory, vol. 15, pp. 329–335, Mar. 1969.

[24] J. Justesen, K. J. Larsen, and L. A. Pedersen, "Error correcting coding for OTN," IEEE Commun. Mag., vol. 48, no. 9, pp. 70–75, Sep. 2010.

[25] Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "An efficient 10GBASE-T ethernet LDPC decoder design with low error floors," IEEE J. SolidState Circuits, vol. 45, no. 4, pp. 843–855, Apr. 2010.

[26] A. Darabiha, A. Chan Carusone, and F. R. Kschischang, "Power reduction techniques for LDPC decoders," IEEE J. Solid-State Circuits, vol. 43, no. 8, pp. 1835– 1845, Aug. 2008.