A 5.35 mm^2 10GBASE-T Ethernet LDPC Decoder Chip in 90 nm CMOS

Alessandro Cevrero*†, Yusuf Leblebici*, Paolo Ienne†, and Andreas Burg‡

*Microelectronic Systems Laboratory, EPF Lausanne, 1015 Lausanne, Switzerland
Email: {alessandro.cevrero, yusuf.leblebici}@epfl.ch

†Processor Architecture Laboratory, EPF Lausanne, 1015 Lausanne, Switzerland
Email: paolo.ienne@epfl.ch

‡Integrated Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland
Email: apburg@iis.ee.ethz.ch

***Abstract*—A partially parallel low density parity check (LDPC) decoder compliant with the IEEE 802.3an standard for 10GBASE-T Ethernet is presented. The design is optimized for minimum silicon area and is based on the layered offset-min-sum algorithm which speeds up the convergence of the message passing decoding algorithm. To avoid routing congestion the decoder architecture employs a novel communication scheme that reduces the critical number of global wires by 50%. The prototype LDPC decoder ASIC, fabricated in 90 nm CMOS, occupies only 5.35 mm^2 and achieves a decoding throughput of 11.69 Gb/s at 1.2 V with an energy efficiency of 133 pJ/bit.**

I. Introduction

To support the high data rates in recent wireline communication standards, such as IEEE 802.3an [1] for 10 Gb/s communications, error correction coding is mandatory in order to maintain communication with a very low error rate. The coding schemes adopted for such applications are based on low density parity check (LDPC) codes, since they perform very close to the Shannon limit once decoded iteratively. Invented by Gallager in 1962 [2], these codes were initially considered too complex for economic implementation and only the availability of nanometer process technology has recently enabled their integration into many important wireless and wireline communication standards. In fact, with the considerable integration density of modern process technologies, the massively parallel message passing decoding algorithm makes these codes particularly attractive for applications that require very high throughput and are therefore ill-suited for sequential decoding algorithms. Unfortunately, we shall see that a fully parallel, straightforward VLSI implementation of an LDPC decoder is still difficult to achieve.

1) Decoding algorithm: The (2048, 1723) LDPC code used in 10GBASE-T is defined by a sparse parity check matrix $\mathbf{H}$ that has binary entries with its 2048 columns associated with the bits in a code block and its 384 rows corresponding to parity check equations. For decoding, the parity check matrix $\mathbf{H}$ is graphically represented by a factor graph comprised of 2048 *variable nodes* (VNs) and 384 *check nodes* (CNs). An edge exists between a check node j and a variable node i if $\mathbf{H}_{j,i} = 1$. Since the degree of each column of $\mathbf{H}$ in the code under consideration is 6 and the degree of each row is 32, 12,288 edges exist in the corresponding graph. VNs hold the intrinsic reliability of each bit L_i initialized by the channel output prior to the first decoding iteration. The message passing algorithm iteratively exchanges messages between the VNs and the CNs along the edges of the graph to update L_i. To facilitate the processing of the messages, the *offset min-sum (OMS) algorithm* is typically used which is better suited for VLSI implementation compared to the ideal sum-product algorithm. With the *flooding schedule* used in previous high-throughput LDPC decoder implementations, all CNs are processed before proceeding to process the VNs in each iteration. This structure can easily be mapped to hardware by directly instantiating and connecting all components of the factor graph. Unfortunately, the large number of instances requires considerable silicon area and the irregular interconnect between the VNs and CNs defined by the parity check matrix leads to severe routing congestion. A *partially parallel* approach reduces the number of CNs through resource sharing, leading to more compact implementation, but more clock cycles are required to complete a full iteration which makes it more difficult to achieve high throughput.

2) Prior art: Blansky et al. [3] fabricated the first fully parallel LDPC chip in 0.16 μm CMOS, achieving 1 Gb/s throughput with 50% silicon utilization, even with an optimized floor plan and manual buffer insertion for a code that is less complex than the one specified for 10GBASE-T. More recently, Mohsenin et al. [4] illustrated the limitations caused by routing congestion in a straightforward implementation with a reference design in 65 nm CMOS, which could only route at 25% area utilization with a maximum throughput of 2.3 Gb/s. To overcome this limitation, Mohsenin et al. [4] proposed a slightly sub-optimal algorithm that considerably reduces routing complexity with a small penalty in the error-rate performance. More on the architectural level, other authors [5] proposed to rely on bit-serial arithmetic to reduce the combinational circuit area and the number of interconnects by a factor equal to the word-length. The technique was applied for an ASIC with a block size smaller than the one specified for 10GBASE-T and synthesis results were reported for a bit-serial 10GBASE-T compliant decoder that occupies a cell area of 9.8 mm^2 in 90 nm CMOS. On the negative side, bit-serial arithmetic requires high clock frequencies and additional silicon area for more sequential elements.

The partially parallel approach was employed by Liu et al. [6] and Zhang et al. [7]. However, the design reported in [6] still has only 50% area utilization and occupies 14.5 mm^2 in a 90 nm process. The chip described in [7] is fully 10GBASE-T compliant and has a good silicon utilization above 80%. The design relies on a 7-stage pipeline to mitigate the delay of the combinational logic and of the interconnect. However, the necessary stall cycles have a negative impact on throughput and the additional registers inflate the active area.

Contribution: This paper presents a 10GBASE-T compliant LDPC decoder that is optimized for minimum silicon area. In contrast to all previous designs for 10GBASE-T that rely on a flooding schedule, we employ the *layered* offset min sum algorithm [8] which offers faster convergence and is inherently well suited for resource sharing. To solve the routing congestion problem, we rely on a message broadcasting architecture and we propose a novel *full-duplex* interconnect that allows to reduce the number of global wires by 50%. With these

978-1-4244-8298-6/10/$26.00 ®2010 IEEE

techniques an area-efficient decoder can be fabricated relying only on fully automated placement and routing without a detailed manual floor-plan. The fabricated ASIC exceeds the throughput required by the IEEE 802.3an standard, reaching 11.69 Gb/s and–to the best of our knowledge–is the smallest fabricated LDPC decoder for 10GBASE-T reported in literature when taking technology scaling into account.

II. Layered Decoder Architecture

The basic idea behind *layered decoding* is that the 384x2048 parity check matrix $\mathbf{H}$ can be partitioned into 6 partial parity check matrices (*layers*) each of size 64x2048 with column-weight one. The partial factor graph associated with each layer comprises only 64 instead of 384 CNs, but all 2048 VNs. A *layered schedule* processes the 6 layers of the code sequentially to complete one iteration. From an implementation perspective, this sequential processing enables resource sharing for the CN processors since only 64 of them are active at the same time. Compared to a partially parallel implementation of the flooding schedule, all VNs are always triggered after processing each individual subset of CNs and not only after all CNs have been processed. This expedites the convergence of the algorithm (c.f., Fig. 4) and therefore partially makes up for the higher number of cycles (six) required for a full iteration.

Fig. 1 depicts a sample 8×12 parity check matrix with two-layer partitioning together with the corresponding high-level block diagram. The depicted architecture instantiates only 4 CN processors which are time multiplexed between the first and second layer. For a 2 clock cycles per iteration schedule, the solid lines represent the connections activated during the first cycle, while the dashed lines represent the connections activated during the second cycle. Since the partitioning into layers is done in such a way that each column of a sub-matrix has a Hamming weight of one, each VN participates only in one parity check equation in each layer. For the ith VN, we define the Q-message Q_i as the outgoing message and the R-message $R_{l,i}$ as incoming message in the lth layer. The layered decoding allows to use the following simplified update rules for the computation of all Q-messages and for the update of all R-messages and intrinsic decoded values L_i in the lth layer of each iteration

$$Q_i = L_i - R_{l,i} \tag{1}$$

$$\bar{R}_{l,i} \leftarrow \max\left\{\min_{i' \in \mathcal{P}_{l,i}/i} \{|Q_{i'}|\} - \beta, 0\right\} \prod_{i' \in \mathcal{P}_{l,i}/i} \operatorname{sgn}(Q_{i'}) \tag{2}$$

$$\bar{L}_i \leftarrow Q_i + \bar{R}_{l,i} \tag{3}$$

In (1) to (3) the VN index i runs over all VNs and the sets $\mathcal{P}_{l,i}$ contain the indices of the other bits (VNs) participating in the same parity check as the ith bit in the same layer. The OMS parameter β is chosen at design-time by means of simulations and is set to $\beta = 1.0$ for our implementation. For the algorithm to work, R-messages are initialized with zero ($R_{l,i} = 0$) prior to the first iteration. Before proceeding to the next layer (or to the next iteration if the last layer was considered) L_i and $R_{l,i}$ are updated according to $L_i \leftarrow \bar{L}_i$ and $R_{l,i} \leftarrow \bar{R}_{l,i}$.

A. High-level architecture and operation

The high-level block diagram of the layered 10GBASE-T LDPC decoder is depicted in Fig. 2. The circuit is comprised of 2048 identical VN processors and of 64 identical CN processors.

The VN processors contain small memories to store the corresponding R-messages $R_{l,i}$ ($l = 1 \ldots 6$) and a 7-bit register that stores L_i. Prior to decoding start, the L_i registers are loaded with the channel outputs in the form of 5-bit log likelihood ratios. In each cycle, each VN processor computes a single Q-message according

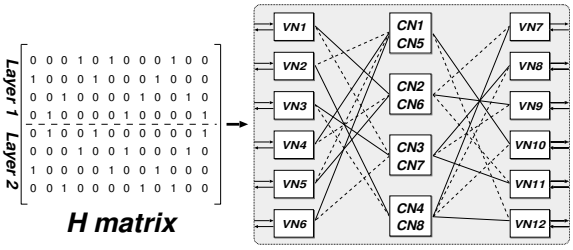

Fig. 1. Example of a structured parity check matrix and a corresponding layered decoder architecture.

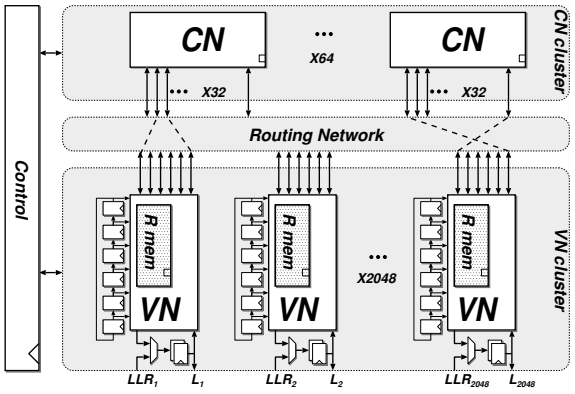

Fig. 2. LDPC decoder high-level block diagram.

to (1) and sends the result to a CN processor through the *routing network*. The CN processors are purely combinational. They compute 32 R-messages according to (2) and send them back to the VNs which update L_i by following (3).

We shall first describe the routing network and the distribution of Q- and R-messages without the full-duplex routing described in Section II-B. In this initial baseline configuration, routing of the Q-messages and the multiplexing of the inputs of the CN processors according to the current layer are realized using a tristate architecture instead of area-consuming distributed full 6-to-1 multiplexers in each of the 32 inputs of the 64 CNs. To this end, each VN has 6 output ports (associated with the six layers) and each CN processor has 32 input ports. Each output port of a VN connects to inputs of 6 (different) CN processors. Hence, each input of a CN processor is driven by more than one VN processor, but in each cycle only one (selected according to the current layer) is driving. For the distribution of the R-messages in reverse-direction, the CNs broadcast each of their 32 outputs to one of the inputs of six (different) VNs[1]. Depending on the current layer, then each VN can locally select the correct input.

B. On Chip Global Interconnects

The above described routing network and multiplexing structure for layered decoding already leads to a considerable reduction of the routing congestion compared to the interconnect of a fully parallel implementation of the flooding schedule. However, dedicated resources are still required to route messages from VNs to CNs and back with the same source/destination pair.

To avoid this overhead, we propose to send Q- and R-messages in forward and backward direction on the same physical wire between

[1]Compared to the fully parallel architecture, this message broadcasting inherent to the partially parallel approach reduces the number of global nets, which was shown to be advantageous in the context of global routing congestion [9] (though for a different LDPC decoder circuit topology).

a VN and a CN during the high- and the low-phase of the same clock cycle, respectively. This scheme further reduces the number of global nets in the routing network by 50%. To implement this idea, we shorten corresponding input and output ports in the VNs, we add tristate drivers to the outputs of the CNs and also shorten them to the corresponding inputs as illustrated in Fig. 3. To control the access to the shared wire during the two phases of the clock, the enable signal of the tristate drivers in the VNs and in the CNs are controlled by a copy of the clock signal which can be shifted to optimize the timing of the overall circuit. Additional latches are needed in the CN to hold the computed R-messages (and to avoid a combinational loop) during the low-phase of the clock in which the inputs of the CNs are no longer driven by the connecting VNs, but by their own outputs.

C. Variable Node Processor

A detailed block diagram of a VN processor is depicted in Fig. 3. In each cycle, the associated L_i is launched by flip-flops and the corresponding 4-bit R-message $R_{l,i}$ is subtracted with saturation to obtain a 4-bit Q-message Q_i. The local 6×4-bit R-memory is implemented using latches in a master-slave configuration, where the slave is shared across the 6 storage words since it is located after the output multiplexer. Write access is controlled by gating the clock signal with a one-hot encoded, glitch-free address input generated by a local 6-bit shift register that keeps track of the current layer. The same shift register also controls the access to the 6 merged input/output ports of the VN during the high-phase of the clock, always enabling only one-at-a-time to avoid driving conflicts between multiple VNs connected to the same port of the same CN. While the shift-register could in principle be shared across multiple VNs, we include a separate instance per VN to reduce the capacitive load and to simplify timing closure in the backend design.

D. Check Node Processor

The detailed CN block diagram is also depicted in Fig. 3. The corresponding circuit receives 32 Q-messages in each clock cycle and computes both the minimum and second minimum of the corresponding absolute values. The minimum finder has been implemented according to the tree structure proposed by Wey et al. [10]. Since this part of the datapath constitutes a considerable part of the critical path, we provided a gate level description to guide the synthesis engine. The outgoing 32 R-messages are obtained by comparing the minimum to each input and by selecting the second minimum only for the output for which the comparison is equal. The sign datapath computes the correct sign for both minimum and second minimum. The latches required for the full-duplex routing are placed within the CN processor datapath such that the area-penalty is minimized.

III. Chip Implementation

The physical implementation of the proposed decoder presents unique challenges due to the still significant number of global wires, the tristate interconnect, and the need to control the timing of the various clock gates and the full-duplex routing. Nevertheless, the design was fully implemented using a conventional standard cell library, without the need for custom components or in house EDA tools beyond a few scripts. The decoder was generated from a VHDL description. Since the 2048 CNs and the 64 VNs are identical, a bottom-up approach was used for synthesis with a timing budget that was carefully optimized to meet the 10GBASE-T throughput requirement with margin, but without sacrificing too much area to obtain the highest possible speed. Note that significantly faster implementations can be obtained at the expense of area with the same architecture by using more stringent timing constraints during synthesis.

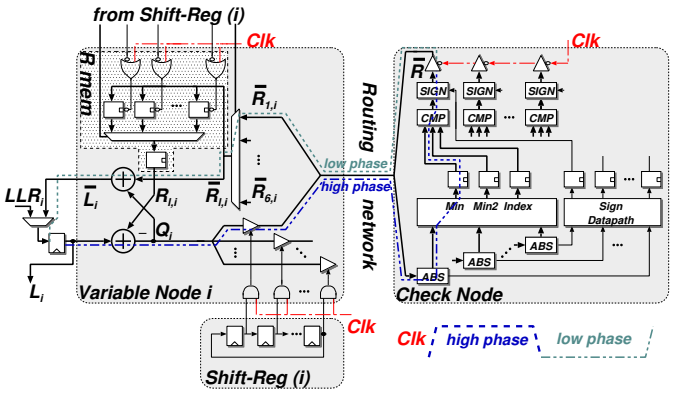

Fig. 3. CN and VN processors detailed schematic. The routing scheme among CN and VN is depicted.

A. Physical Design

The backend design was carried out using CAD tools from Cadence. Due to the large amount of identical blocks repeated on chip, a hierarchical placement and routing is expected to provide the best results. However, many design iterations are required before reaching a feasible floor-plan. Hence, a flat placement and routing approach has been adopted due to the good quality of results produced in the final layout.

With the chosen wordlengths, the final design contains a total of 49,156 full-duplex wires. The drawback of our tristate approach is that the wire delay grows rapidly with the length of the wire since buffer insertion is not possible. Furthermore, the automatic tools fail to properly size the tristate-drivers during timing optimization. As a remedy, we rely on the placement engine to minimize the wire-length by placing the VNs and the corresponding CNs in close proximity to each other, specifying routability as primary optimization objective. After placement, approximately 5% of the global wires still have a large capacitance, ranging from 200 fF up to 800 fF. As a consequence, gates directly connected to these nets exhibit a transition time of up to 800 ps and a maximum propagation delay of up to 2 ns. We handle this problem by resizing all the tristate buffers, using the available drive strengths in the cell library according to the net capacitance observed during the trial-routing phase. For short nets which are not on the critical path and do not violate transition-time constraints, minimum-size buffers are used to reduce the pin capacitance of the control network and to minimize active area. To avoid large cross current in the tristate buffers, the skew existing between tristate-enable pins has been reduced to 120 ps.

B. Implementation Results

1) Error-rate performance: The fixed-point error rate performance of the implemented circuit is shown in Fig. 4. Monte-Carlo simulations show that with only 4 iterations (thanks to the layered schedule) our fixed-point design operates within 0.2 dB of the ideal floating-point SPA algorithm with 20 iterations. With 6 iterations the gap is closed further to merely 0.1 dB.

2) ASIC Measurements: The ASIC has been fabricated in a 90 nm CMOS process. Out of 10 packaged samples, 8 were fully functional. The maximum operating frequency and power measurements were performed with HP83000 automatic test equipment, and correct operation was verified under different channel conditions with the Eb/N0 ranging from 2 dB up to 5 dB. The die photograph of the chip is shown in Fig. 6. Table I summarizes our measurement results. The design operates at a maximum frequency of 137 MHz at 1.2 V. The number of iterations is programmable and can be loaded before

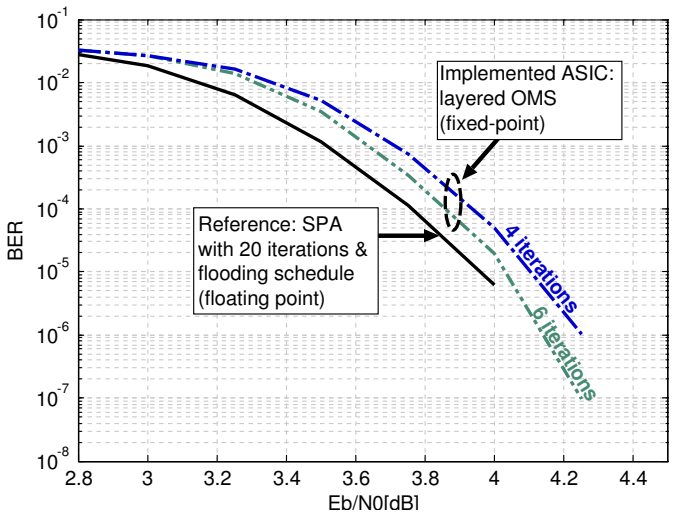

Fig. 4. Decoder bit error rate (BER) comparison for AWGN channel. 4-bit Q- and R-messages have been chosen for the fabricated prototype.

TABLE I
SUMMARY OF THE MEASURED RESULTS FOR THE (2048,1723) IEEE 8023.AN LDPC DECODER.

Technology	UMC 90 nm	
Core Area	1.56 x 3.43 mm	
Core Utilization	84.4%	
R-message Word Length	4-bit	
Q-message Word Length	4-bit	
L Word Length	7-bits	
R Storage	48-Kbit	
L Storage	14-Kbit	
Decoding Algorithm	Layered OMS	
Supply (V)	1.2	0.8
Throughput @ 4 iter. (Gb/s)	11.69	7.23
Clock Speed (MHz)	137	84.7
Power @ 3db SNR (mW)	1559	386.8
Leakage Power (mW)	40.80	18.52
Energy Efficiency (pJ/bit)	133.37	53.49

starting the decoding process. For our measurement the iteration count has been set to four, since it is sufficient to provide excellent correction capabilities. The overall throughput is 11.69 Gb/s and the power consumption is 1.56 W. This corresponds to an energy efficiency of 133 pJ/bit. Due to the fixed silicon area required for chip fabrication, we did not include an early termination unit. This technique could be applied to the proposed architecture with small area overhead resulting in an average decoding throughput above 40 Gb/s in the high Eb/N0 regime. Fig. 5 shows the effect of scaling the decoder's supply voltage on its maximum decoding throughput and the corresponding power consumption. Scaling down the voltage is an efficient technique to improve energy efficiency. As highlighted in the plot, at 0.8 V the chip can still reach 7.23 Gb/s, with an energy efficiency of 53.49 pJ/bit.

IV. CONCLUSION

This paper presents an LDPC decoder ASIC that is compatible with the 10GBASE-T standard. The manufactured circuit provides a throughput of 11.69 Gb/s with a silicon area of 5.35 mm^2 in 90 nm CMOS. Good area-utilization and high-throughput are concurrently achieved by using a layered decoding scheme that facilitates resource sharing and reduces the number of decoding iterations. The routing-congestion problem has been solved by reducing the number of global

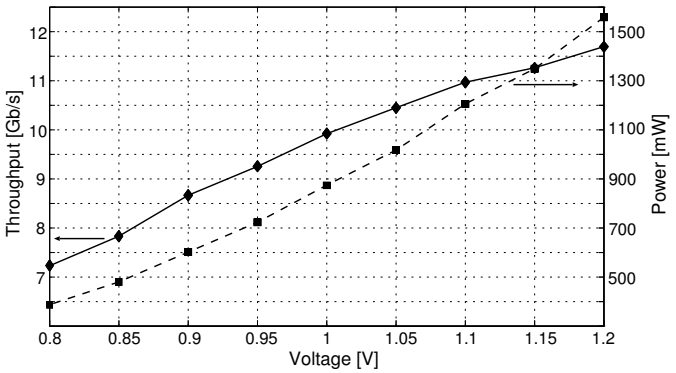

Fig. 5. Measured throughput and power of the fabricated ASIC.

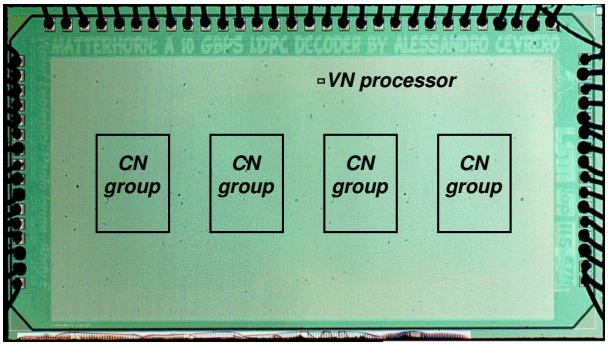

Fig. 6. Microphotograph of the fabricated ASIC.

nets using a tristate routing network and with a novel full-duplex communication scheme that further improves area utilization. Both techniques are not specific to the code under consideration and can also be applied to decoders for other regular LDPC codes.

REFERENCES

[1] *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, Sep 2006, IEEE Std. 802.3an.

[2] R. Gallager, "Low density parity check codes," *Trans. of the IRE Prof. Group on Inf. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[3] A. Blanksby and C. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, Mar. 2002.

[4] T. Mohsenin, D. Truong, and B. Baas, "Multi-split-row threshold decoding implementations for LDPC codes," in *IEEE Int. Symposium on Circuits and Systems*, May 2009, pp. 2449–2452.

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

[6] L. Liu and C.-J. Shi, "Sliced message passing: high throughput overlapped decoding of high-rate low-density parity-check codes," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 11, pp. 3697–3710, Dec. 2008.

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

[8] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1288–1299, Aug. 2005.

[9] A. Darabiha, A. Carusone, and F. Kschischang, "Block-interlaced LDPC decoders with reduced interconnect complexity," *IEEE Trans. Circuits Syst. II*, vol. 55, no. 1, pp. 74–78, Jan. 2008.

[10] C.-L. Wey, M.-D. Shieh, and S.-Y. Lin, "Algorithms of finding the first two minimum values and their hardware implementation," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 11, pp. 3430–3437, Dec. 2008.