Memory Efficient Decoders using Spatially Coupled Quasi-Cyclic LDPC Codes

Vikram Arkalgud Chandrasetty, Sarah J. Johnson and Gottfried Lechner

Abstract—In this paper we propose the construction of Spatially Coupled Low-Density Parity-Check (SC-LDPC) codes using a Quasi-Cyclic (QC) algorithm. The QC based approach is optimized to obtain memory efficiency in storing the parity-check matrix in the decoders. A hardware model of the parity-check storage units has been designed for Xilinx FPGA to compare the logic and memory requirements for various approaches. It is shown that the proposed QC SC-LDPC code (with optimization) can be stored with reasonable logic resources and without the need of block memory in the FPGA, and a significant improvement in the processing speed is achieved.

Index Terms—Error correction codes, low-density parity-check codes, spatially coupled codes, digital circuits, field programmable gate arrays.

I. INTRODUCTION

Spatially Coupled LDPC (SC-LDPC) codes have recently drawn special interest in channel coding due to their excellent thresholds [1]. Their performances are achieved at large code lengths (over 100K) which results in increased decoding complexity [2]. Practical implementation of such large LDPC codes is a well known problem, particularly the storage of the parity-check matrix in hardware memory [3]. The structure of the matrix also significantly affects the complexity of the decoder. Quasi-Cyclic based LDPC matrices have proven advantages over unstructured (random) matrices in design complexity and encoding process [4]. They also enable collision-free parallel processing in the decoder. Therefore, it is of great interest to investigate both their applicability in the case of SC-LDPC codes and the scope for further improvements in decoding performance and reducing implementation complexity.

In this paper, we present the construction of SC-LDPC codes using QC techniques. The decoding performance of these codes are compared with that of standard LDPC codes. The inherent advantages of QC based codes and the diagonal structure of the SC-LDPC matrix is further explored to reduce the complexity of the decoder by reusing the circulants in the matrix. A detailed investigation is carried out to evaluate the performance in terms of bit error rate (BER) and hardware implementation complexity of these memory-optimized QC SC-LDPC codes. The FPGA resource requirements and speed of operation are compared with QC and PEG SC-LDPC by implementing a hardware model of these codes. Memory efficiency and speed improvements achievable by using the proposed optimized QC SC-LDPC codes are also analysed.

A sample Tanner graph structure of SC-LDPC codes is shown in Fig. 1. The SC-LDPC code starts with the protograph of a standard \((d_l,d_r)\)-regular LDPC code (see Fig. 1(a)) where \(d_l\) and \(d_r\) denotes the average bit degree and check degree respectively. An SC-LDPC protograph is constructed by coupling a chain of \(2L+1\) protographs of the regular LDPC code (see Fig. 1(b)). A particular code of length \(N\) is formed from the \((d_l,d_r,L)\) protograph chain by creating \(M\) copies of every node and edge in the coupled chain. If a particular bit node was coupled to a particular check node in the original chain each of its copies is then connected to one of the \(M\) copies of that check node in the final code. The choice of which copy each node is connected to will depend on the constructions of the code. Like any standard LDPC code, SC-LDPC codes can be constructed randomly or incorporating structure of some form. In the following section we present a QC algorithm for the construction of SC-LDPC codes. In a QC-SC-LDPC code the all of the \(M\) copies of a particular edge in the protograph are specified using a circulant. The length, \(N\), and rate, \(r\) of the SC-LDPC code in Fig. 1(c) is computed as follows:

\[
N = (2L+1) \times 2M, \quad r = \frac{2(L+1) - d_l}{(2L+1) \times 2}. \quad (1)
\]
II. MEMORY-EFFICIENT QUASI-CYCLIC SC-LDPC CODES

The quasi-cyclic (QC) [5] construction of LDPC code has various advantages including simple encoding [6], parallel decoding and memory efficiency in storing the matrix elements in the decoder [2]. Hence it is a good idea to have SC-LDPC codes in QC form. The construction of QC SC-LDPC matrices is presented in [8], [9]. [8] discusses the performance improvements achieved by spatial coupling particularly for quantum LDPC codes. Whereas [9] presents techniques to improve the upper bound on the minimum Hamming distance of members of the QC sub-ensembles. In this section we present an innovative technique to further optimize the QC based SC-LDPC codes to significantly reduce the hardware requirements of the decoder, especially the Block RAMs (BRAM) in an FPGA.

While designing an LDPC decoder, it is essential to store the structure of the LDPC matrix in the hardware to carry out the decoding process. The structure is normally stored in the form of BRAMs because of the enormous data required, particularly for large LDPC codes [10][11]. Compared to the codes that are constructed using random or progressive edge growth (PEG) algorithms, the QC based technique requires significantly less memory to store the matrix structure. This is because the latter requires memory for storing only the circulant, instead of the entire matrix. However, a systematic shifting circuit is needed to shift position of circulant to realize the actual and complete LDPC matrix in the decoder hardware [10]. Therefore, the decoder using QC based LDPC matrices require less memory but uses additional logic elements for special operations.

The QC SC-LDPC matrix consists of a chain of circulants making up the \((2L+1)\) protographs. For a \((d_l, 2d_l)\)-regular code, (i.e. a rate \(\frac{1}{2}\) code) each protograph in the chain corresponds to a set of \(2d_l\) individual circulants. We say that a circulant \(X\) is \(x\) cyclic shifts of the binary identity matrix. So the first column of the circulant \(X\) will have a ‘1’ entry in row \(x+1\), the second column of \(X\) will have a ‘1’ entry in row \(x+2\) etc.

The staircase structure of the SC-LDPC codes offers a unique opportunity to optimize the QC based technique to further reduce the hardware resources required for the decoder. We investigate whether a fixed set of circulants in the matrix (i.e. fewer than \(2L+1\) independent circulant sets) can be reused without affecting the decoding performance. To this end, we consider SC-LDPC codes constructed with one, two and three repeating sets of circulants. Fig. 2(a), Fig. 2(b) and Fig. 2(c) show one, two and three repeating sets respectively for a rate \(\frac{1}{2}\) \((3, 6, L)\) quasi-cyclic SC-LDPC code.

A. The effect of circulant reuse on code girth

For a quasi-cyclic SC-LDPC code with the reuse of \(u\) columns of circulants as in Fig. 2 we have the following constraint on its girth. Note that we consider column weights of 4 (or more) since, as will be shown later, these are the column weights we require for SC-LDPC codes.

**Lemma 1:** Given a quasi-cyclic SC-LDPC code with reuse \(u = 1, 2\) or 3, with \(d_l \geq 4\) and \(L \geq 2\). The girth of this code is at most \(4 + 2u\).

![Fig. 2: SC-LDPC parity-check sub-matrices from a (3,6)-regular protograph with quasi-cyclic column reuse.](image)

#### Proof Sketch:

We will consider the case \(r = 2\) and examine the first column of circulants for each protograph in the chain. We assume that the first four circulants in the first column set are given by \(A, B, C\) and \(D\) and the first four circulants in the second column set are given by \(E, F, G\) and \(H\). The columns are repeated as shown in Fig. 2(b) and so we have the following parity-check sub-matrix corresponding to the first column of the first four protographs:

\[
H = \begin{bmatrix}
A & \ldots & 0 & \ldots & 0 & \ldots & 0 & \ldots \\
B & \ldots & E & \ldots & 0 & \ldots & 0 & \ldots \\
C & \ldots & F & \ldots & A & \ldots & 0 & \ldots \\
D & \ldots & G & \ldots & B & \ldots & E & \ldots \\
0 & \ldots & H & \ldots & C & \ldots & F & \ldots \\
0 & \ldots & 0 & \ldots & D & \ldots & G & \ldots \\
\end{bmatrix}
\]

We will say that these four columns of circulants begin in the \(j_{0}\)-th, \(j_{1}\)-th, \(j_{2}\)-th and \(j_{3}\)-th columns of \(H\) respectively. Similarly we will say that these six rows of circulants begin in the \(i_{0}\)-th, to \(i_{6}\)-th rows of \(H\) respectively. Consider the \(i_{1} + b\)-th row of \(H\). \(H\) has a non zero entries in positions \(i_{1} + b, j_{0}\) and \((i_{1} + b, j_{1} + b - e)\). Consider the \(i_{3} + d\)-th row of \(H\). \(H\) has a non zero entry in positions \((i_{3} + d, j_{0})\) and \((i_{3} + d, j_{3} + d - e)\). Consider the \(i_{3} + b - e + g\)-th row of \(H\). \(H\) has a non zero entry in positions \((i_{3} + b - e + g, j_{1} + b - e)\) and \((i_{3} + b - e + g, j_{2} + b - e + g - b)\). Consider the \(i_{4} + d - e + g\)-th row of \(H\). \(H\) has a non zero entry in positions \((i_{4} + d - e + g, j_{2} + d - e + g - d)\) and \((i_{4} + d - e + g, j_{2} + d - e + g - g)\) and \((i_{4} + d - e + g, j_{2} + d - e + g - d)\).

There exist an 8-cycle connecting these eight non-zero entries as

\[
(i_{1} + b, j_{0}) \rightarrow (i_{1} + b, j_{1} + b - e) \rightarrow (i_{3} + b - e + g, j_{1} + b - e) \rightarrow (i_{3} + b - e + g, j_{2} + b - e + g) \rightarrow (i_{5} + d - e - e + g, j_{2} + b - e + g) \rightarrow (i_{5} + d - e - e + g, j_{3} + d - e) \rightarrow (i_{5} + d + d - d + d) \rightarrow (i_{3} + d, j_{3} + d - d) \rightarrow (i_{1} + b, j_{0}).
\]

Although the proof only requires that we find one 8-cycle which exists for any choices of \(a, b, \ldots, h\), in fact every column of \(H\) (and hence every codeword bit) is involved in 8-cycles in QC SC-LDPC codes with reuse-2 at column weight at least four regardless of which circulants are chosen. The proof for the cases \(u = 1\) and 2 follows a similar argument.
Fig. 3: Thresholds for LDPC codes from density evolution

Fig. 4: BER Performance of SC-LDPC codes for various code lengths

III. PERFORMANCE OF SC-LDPC CODES

The QC SC-LDPC codes were simulated to evaluate the BER performance on a binary input additive white Gaussian noise (BI-AWGN) channel. We used multi-edge density evolution to compute the threshold of SC-LDPC codes, shown in Fig. 3 and noted that \( L \geq 16 \) is necessary to achieve thresholds better than that of \((3, d_r)\) standard LDPC codes and that \( d_l = 4 \) has an improved threshold over \( d_l = 3 \) as \( L \) is increased. Also, as SC-LDPC codes are known to have good performance for very long codes, we compared decoding performance with code length as shown in Fig. 4. Given these results a fairly long SC-LDPC code of 100,360-bits (\( \approx 100K \)) with \( d_l = 4, L = 96 \) and \( M = 260 \) is considered for our simulation results. However, we note that even longer codes would perform noticeably better. Simulations were carried out using software models on a BI-AWGN channel. The sum-product algorithm was used for decoding with a maximum of 1000 iterations, and the simulation was run until at least 50 word errors were accumulated.

Fig. 5 shows a comparison of quasi-cyclic SC-LDPC codes with different levels of circulant reuse. In the construction process for all codes the circulants are chosen to avoid girth 4 in the resulting matrix. As would be expected due to girth limitations, the quasi-cyclic matrix with reuse of one and two circulant columns have poor performance. However, by reusing three circulant columns, the BER performance we obtained is as good as the standard QC SC-LDPC codes.

Lastly in Fig. 6 we compare the BER performance of the quasi-cyclic SC-LDPC codes with non-quasi-cyclic SC-LDPC codes and with standard LDPC codes. The non-quasi-cyclic SC-LDPC codes are constructed randomly and using a PEG algorithm modified for spatial coupling. The standard LDPC codes and constructed according to (PEG and QC). From Fig. 6 it is clear that the waterfall regions of SC-LDPC are better than standard LDPC (as predicted by density evolution). Also, the performance of random, PEG and QC based SC-LDPC codes follow each other very closely.

IV. ESTIMATION OF HARDWARE REQUIREMENTS FOR USING SC-LDPC CODES

The new QC SC-LDPC codes are compared with random SC-LDPC codes by designing a decoder hardware model. The model consists of units that are essential for storing the contents of the LDPC matrix and generating the appropriate address locations in a decoder. The block diagram of the hardware model is shown in Fig. 7. The hardware model consists of a clock and a synchronous reset as inputs. They
Table I: FPGA resources required for various SC-LDPC codes

| LDPC codes          | Rand or PEG | QC (Std.) | QC (Reuse-3) |
|---------------------|-------------|-----------|--------------|
| Code length         | 100K / 64K  | 100K / 64K| 100K / 64K   |
| Registers           | 286 / 268   | 320 / 302 | 322 /        |
| LUTs                | 185 / 170   | 890 / 819 | 609 /        |
| Slices              | 143 / 139   | 357 / 356 | 214 /        |
| RAM                 | 528 / 248   | 12 / 12   | -            |
| Clock (MHz)         | 188 / 215   | 180 / 182 | 264 /        |

Fig. 7: Block diagram of the hardware model for storing and processing quasi-cyclic SC-LDPC codes

The hardware models for both Random/PEG and QC SC-LDPC codes have been designed and synthesized using Verilog HDL. The designs are placed and routed for Xilinx Kintex-7 FPGA (XC7K355T) with LDPC codes of length 100K and 64K. The estimation of FPGA hardware requirements and maximum clock frequency achievable for the designs are shown in Table I. As expected, a large number of BRAMs are utilized by the PEG based codes compared to QC SC-LDPC codes, to store the matrix elements. With slightly increased logic units - registers and look-up tables (LUT), the standard QC codes offer a significant saving (up to 43 times in the case of 100K code lengths) of BRAMs compared to PEG based SC-LDPC codes. As stated earlier, the increased logic requirements are due to the Cyclic Shift unit in the QC based SC-LDPC hardware models. Further, it is also observed that QC SC-LDPC codes with reuse-3 do not require BRAMS, as the few number of circulant can be easily stored in LUTs. Elimination of BRAMs also results in a significant improvement in the speed of operation (up-to 40% increase in the maximum operating clock frequency) compared to PEG or standard QC SC-LDPC codes (for code length of 100K).

V. Conclusion

This paper has presented the construction of SC-LDPC codes using the QC technique. It demonstrates the threshold advantages achievable by SC-LDPC over standard LDPC codes. Memory optimized QC SC-LDPC codes are introduced to significantly reduce the complexity of the decoder for storing the matrix elements. It is shown that by reusing the circulant columns in the SC-LDPC matrix (with reuse-3), it is possible to obtain a memory efficient decoder without noticeably affecting the decoding performance. The advantages of using the optimized QC SC-LDPC codes have been demonstrated by designing a hardware model, which shows substantial reduction in memory requirements and a significant improvement in the operating clock frequency of the decoder.

References

[1] S. Kudekar, T. J. Richardson, and R. L. Urbanke, “Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC,” IEEE Transactions on Information Theory, vol. 57, no. 2, pp. 803–834, 2011.
[2] S. Kudekar and K. Kasai, “Threshold saturation on channels with memory via spatial coupling,” in IEEE International Symposium on Information Theory Proceedings, 2011, pp. 2562–2566.
[3] V. A. Chandrasetty and S. M. Aziz, “A highly flexible LDPC decoder using hierarchical quasi-cyclic matrix with layered permutation,” Journal of Networks, Academy Publisher, vol. 7, no. 3, pp. 441–449, 2012.
[4] Z. Xiaimiao and C. Fang, “Efficient partial-parallel decoder architecture for quasi-cyclic nonbinary LDPC codes,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 58, no. 2, pp. 402–414, 2011.
[5] M. Fossorier, “Quasi-cyclic low-density parity-check codes from circulant permutation matrices,” IEEE Transactions on Information Theory, vol. 50, no. 8, pp. 1788–1793, 2004.
[6] A. Mahdi, N. Kanistras, and V. Paliouras, “An encoding scheme and encoder architecture for rate-compatible QC-LDPC codes,” in IEEE Workshop on Signal Processing Systems, 2011, pp. 328–333.
[7] D. Yongmei, C. Ning, and Y. Zhiyu, “Memory efficient decoder architectures for quasi-cyclic LDPC codes,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 55, no. 9, pp. 2898–2911, 2008.
[8] M. Hagiwara, K. Kasai, H. Imai, and K. Sakaniwa, “Spatially coupled quasi-cyclic quantum LDPC codes,” in IEEE International Symposium on Information Theory, 2011, pp. 638–642.
[9] D. Mitchell, R. Smarandache, M. Lentmaier, and D. Costello Jr, “Quasi-cyclic asymptotically regular LDPC codes,” arXiv preprint arXiv:1007.0404, 2010.
[10] V. A. Chandrasetty and S. M. Aziz, “FPGA implementation of a LDPC decoder using a reduced complexity message passing algorithm,” Journal of Networks, Academy Publisher, vol. 6, no. 1, pp. 36–45, 2011.
[11] C. Xiaoheng, L. Shu, and V. Akella, “QSN: A simple circular-shift network for reconfigurable quasi-cyclic LDPC decoders,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 57, no. 10, pp. 782–786, 2010.
[12] H. Xiao-Yu, E. Eleftheriou, and D. M. Arnold, “Progressive edge-growth Tanner graphs,” in IEEE Global Telecommunications Conference, vol. 2, 2001, pp. 995–1001.