ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 3 Issue 3 September 2023 / Page No: 23-28
Paper Id: JETA-V3I7P104 / Doi: 10.56472/25832646/JETA-V3I7P104

Original Article

LDPC Decoders for the Design and Implementation of High Performance and Low Cost Techniques

Syamsuddin Millang[1], Siti Nuraeni[2]

[1]*Research Scholar, Electronics Communication Engineering, University of Santo Tomas, Philippines.*
[2]*Lecturer, Electronics Communication Engineering, University of Santo Tomas, Philippines.*

Received Date: 25 July 2023 Revised Date: 15 August 2023 Accepted Date: 03 September 2023

Abstract*: Scaling of technology and increased integration density can cause parameter and noise changes, which can increase error rates at different stages of computing. Soft failures and single-event upsets are a persistent issue in memory applications. This work is primarily concerned with the design of an effective Multi Detector/Decoder (MLDD) for fault detection and fault correction in memory applications. One-step majority logic decoding is used to effectively discover and rectify errors in Euclidean Geometry Low Density Parity Check Codes (EG-LDPC). While majority decodable codes have the potential to fix a huge number of mistakes, their lengthy decoding times mean that errors are more likely to be missed, and the ML Decoding technique might waste time by identifying faults in both error- and error-free code words. The suggested fault-detection approach can identify the error in nearly three decoding cycles. Memory access time can be decreased when the data read is mistake free. For high code word sizes, the approach maintains small area overhead and low power consumption.*

Keywords: *LDPC Decoders, Multi Detector, ML Decoding, Memory.*

I. INTRODUCTION

A low-density parity-check (LDPC) code is a linear error correcting code, a means of conveying a message across a noisy transmission channel that is discussed in the field of information theory. A sparse bipartite graph is used to build an LDPC. Since the noise threshold may be adjusted practically close (or arbitrarily close on the BEC) to the theoretical maximum (the Shannon limit) for a symmetric memoryless channel, LDPC codes are considered capacity-approaching codes. The noise threshold establishes a maximum level of channel noise beyond which information loss is unlikely to occur. LDPC codes may be deciphered in time proportional to the length of their blocks using iterative belief propagation techniques.

The usage of LDPC codes is growing in fields that rely on secure, efficient data transmission over limited bandwidth or return channels with interfering noise. LDPC codes have lagged behind other codes, especially turbo codes, in terms of implementation. On August 29, 2013, the original patent for Turbo Codes was no longer valid. In honor of Robert G. Gallager, who introduced the LDPC idea in his 1960 PhD dissertation at the Massachusetts Institute of Technology, LDPC codes are often referred to as Gallager codes.

III. IMPLEMENTATIONOFLDPCINVHDL

A. Communication System Jargon:

LDPC codes have a sparse parity-check matrix definition. Typically, according to the sparsity limitations, this sparse matrix is created at random — LDPC code production is addressed in more detail below. Robert Gallager created the first of these codes in 1962.

Forney's factor graph notation is used to illustrate a section of an illustrative LDPC program below. The network in question has n nodes representing variables at the top, coupled to (nk) nodes representing constraints at the bottom. This is a popular way of graphically representing an (n,k) LDPC code. The bits of a valid message, when placed on the T's at the peak of the graph, where the conditions for the graph are met. The values of all lines that lead to a variable node (a box labeled "=") are the same, and the values of all lines that lead to a factor node (a box labeled "+") must add up to zero modulo two (that is, there

This is an open access article under the CCBY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/2.0/)

must be an even number of odd values).

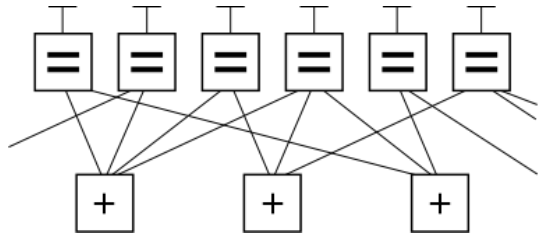

Figure 1: Communication System Jargon

Eight six-bit sequences (i.e., 000000, 011001, 110010, 101011, 111100, 100101, 001110, 010111) correspond to acceptable code words when all lines entering the image are disregarded. A three-bit message is shown here as a six-bit LDPC code fragment. The inclusion of redundancy improves the likelihood of successful channel recovery. With n equal to six and k equal to three, we get a (6, 3) linear code.

Onceagainignoringlinesgoingoutofthepicture,theparity-checkmatrixrepresentingthisgraphfragmentis

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}_1 \sim \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}_2 \sim \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}_3 \sim \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}_4.$$

Each bit in the received codeword is represented by a row in this matrix, and each of the three parity-check requirements is represented by a column. By performing some elementary row operations in GF(2) on the parity-check matrix H, we may extract the eight code words:

Step1:H.
Step2:Row1is addedtorow3.
Step3:Row2and3areswapped.
Step4:Row1isaddedtorow3.Fromthis,thegeneratormatrixGcanbeobtainedas[I_K|P](notingthatinthespecialcaseofthis being a binary code P=-P), or specifically.

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Finally, by multiplying all eight possible 3-bit strings by G, all eight valid code words are obtained. For example, the codeword for thebit-string'101'isobtainedby:

$$\begin{pmatrix} 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Asa check, the row space of G is orthogonal to H such that $GH^T = 0$

B. Encoding:

Bits of input data (D) are replicated and sent to several encoders during frame encoding. Typically, accumulators are employed as the constituent encoders, and each accumulator is responsible for generating a parity symbol. The code symbols are formed by transmitting the parity bits (P) along with a single copy of the original data (S0, K-1). The S bits from each individual encoder are thrown away. A different component code might make advantage of the parity bit.

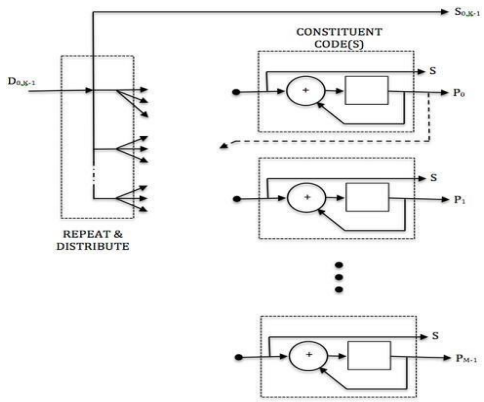

Figure 2: LDPC Encoder

Blocks of 64800 symbols (N=64800) are encoded using 43200 data bits (K=43200) and 21600 parity bits (M=21600) in the DVB-S2 rate 2/3 coding. Except for the initial parity bit, which encodes just 8 data bits, each constituent code (check node) encodes a full complement of 16 data bits. In this code, the first 4680 bits of data are utilized 13 times (in 13 parity codes), whereas the remaining data bits are used in 3 parity codes (an irregular LDPC code).

Traditional turbo codes, on the other hand, have two separate codes working in parallel to encode the same K-bit input block of data. These constituent encoders are 8- or 16-state recursive convolutional codes (RSC), and they are separated by a code inter leaver that shuffles one copy of the frame between them.

In contrast, the LDPC code employs a series of accumulators (low depth constituent codes) that each encode a relatively tiny subset of the input frame. Multiple 'convolutional codes' of low depth (2 states) are interconnected by repeat and disseminate to form the larger code. Interleaving is handled by the repeat and distribute procedures in the turbo code.

LDPC codes can sometimes have greater performance than turbo codes because they allow for more exact management of connections between the various constituent codes and the amount of redundancy for each input bit. Even at low code rates, turbo codes appear to outperform LDPCs, or at least the design of well-performing low rate Turbo Codes is simpler than it is for LDPCs.

For efficiency, the accumulators' hardware is recycled for further uses in the encoding procedure. To clarify, the same accumulator hardware is utilized to produce a second set of parity bits after the first set has been generated and saved.

C. Decoding:

Maximum probability decoding of an LDPC code over a binary symmetric channel is, like maximum likelihood decoding of any other code, an NP-complete issue. It is impractical to perform optimum decoding on an NP-complete code of any practical size. Although iterative belief propagation decoding is an inferior method, it yields outstanding results and is feasible in practice. The individual parity checks that make up an LDPC are treated as separate codes by the suboptimal decoding methods. Soft-in-soft-out (SISO) methods, including SOVA, BCJR, MAP, and their variants, are used to independently decode each SPC code. Each SISO decoder's soft decision data is updated by being compared to data from many redundant SPC decoders of the same data bit. Using the revised soft decision data, the SPC codes are decoded once more. In order to find a valid code word or run out of decoding possibilities, this process is repeated. The term "sum-product decoding" is commonly used to describe this method of decoding.

In computer science, the process of decoding SPC codes is known as "check node" processing, whereas the process of verifying the values of individual variables is known as "variable-node" processing. Sets of SPC codes are decoded in parallel in a realistic LDPC decoder implementation to boost throughput. However, on the binary erasure channel, belief propagation is really just an iterative constraint fulfillment procedure. Using the above example as a guide, imagine that the valid codeword 101011 is

broadcast via a binary erasure channel and received with the first and fourth bits erased, resulting in the question mark. 01? 11. The received message is represented by placing it on top of the factor graph, as the sent message must have complied with the coding requirements.

Since there is more than one unknown bit in each of the restrictions related to the initial bit, it is currently impossible to recover it. In order to decode the message, it is necessary to find constraints that relate to exactly one of the deleted bits. Only the second limitation is necessary in this case. Looking at the second condition, we see that the fourth bit must have been zero as that's the only value that would allow us to meet the requirement. This process is repeated as needed. With the updated fourth-bit value and the first constraint in hand, the original first-bit value may be reconstructed. To meet the leftmost requirement, the first bit must be set to 1.

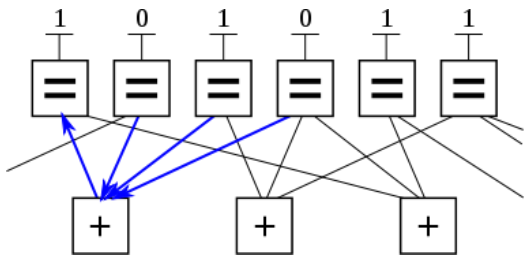

Figure 3: Decoding

This allows for iterative decoding of the message. In contrast, actual values representing probabilities and likelihoods of belief are transmitted between the variable nodes and the check nodes in other channel architectures. Multiplying the revised code word by the H parity-check matrix verifies the correctness of the solution:

$$\mathbf{z} = \mathbf{Hr} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The resultant codewords are verified to be correct due to the fact that the operation's output z (the syndrome) is the three-by-one-zero vector. In reality, almost all commercial LDPC decoders make advantage of soft-decision decoding and soft-decision message transmission, which is not demonstrated in this erasure example.

D. Updating Node Information:

There has been extensive research on the outcomes of various update schedules for variable nodes and constraint nodes in the recent years. Historically, "flooding" has been the method of choice for deciphering LDPC codes. In order to edit a variable node, it was necessary to update all of the constraint nodes first, and vice versa. Later work by Vila Casadoet al. investigated other update approaches, including the updating of variable nodes with the most recent information from check nodes.

The reasoning behind these algorithms is that nodes with the most rapidly changing values are the ones that require attention immediately. When the magnitude of a node's log-likelihood ratio (LLR) is big and stable over time, it doesn't need to be updated as often as a node that's LLRs vary more in sign and magnitude. In comparison to flooding-based scheduling algorithms, they provide faster convergence and lower error levels. The Informed Dynamic Scheduling (IDS)[15] technique is able to beat these lower error levels because it avoids trapping sets of adjacent codewords.[16] Iteration has a different meaning when non-flooding scheduling techniques are applied. An iteration is complete in a (n, k) LDPC code with rate k/n when all n variables and n k constraint nodes have been modified.

.

E. Lookup Table Decoding:

LDPC codes may be decoded using a lookup table on a low-power microcontroller. Applications exist that employ low-power processors and shorter block lengths (1024), despite the fact that high-power processors are often used to create codes like the LDPC.

As a result, the output bit may be predicted based on known input bits. Input statuses (both errored and non-errored) are cataloged in a table with n entries (for a block length of 1024 bits, this would be 1024 bits long).Each input bit is stored in a first-in, first-out (FIFO) register, and the value of that register is used to access the appropriate output from a table of precalculated values.

This technique allows for LDPC decoding to be performed on an MHz PIC chip, despite the usage of very high iterations with only the memory for the lookup table costing anything.

a) Code Construction

In order to develop LDPC codes for high block sizes, it is usual practice to first observe how decoders perform. LDPC decoders may be proven to have a noise threshold over which decoding is not performed, a phenomenon known informally as the cliff effect, when the block size goes to infinity. Finding the optimal split between arcs coming from check nodes and those coming from variable nodes allows us to fine-tune this cutoff. An EXIT chart is a helpful visual tool for approximating this cutoff.

b) There are two major approaches to building an LDPC code when this optimization is complete:

- Pseudorandom approaches
- Combinatorial approaches

Building on theoretical conclusions that a random construction provides acceptable decoding performance for big block sizes, a pseudo-random technique is used during building.[7] Pseudorandom codes often have complicated encoders, although the best decoders allow for simpler encoders. The desired qualities predicted at the theoretical limit of infinite block size are typically achieved at a limited block size by applying a variety of restrictions.
Small block-size LDPC codes can have their characteristics optimized using combinatorial methods, and codes with simple encoders can be generated using these same methods.

III. SUBSET OF LDPC CODES

A subset of LDPC codes, called RS-LDPC, is based on Reed-Solomon codes and is utilized in the 10 Gigabit Ethernet standards. Structured LDPC codes, such as the LDPC code used in the DVB-S2 standard, are simpler to implement and so need less expensive hardware. in particular, routines designed to ensure the H matrix is always in motion.

IV. EXPERIMENTS AND RESULTS

A. Simulation Result:

To ensure the code is error-free, simulation is used. The study makes use of the Verilog programming language. Errors in Verilog code, such as syntax mistakes, logical inconsistencies, etc., are easily detected. Modelsim is used for the simulation process. The following are simulation results for decoding performed in Verilog in Xilinx.

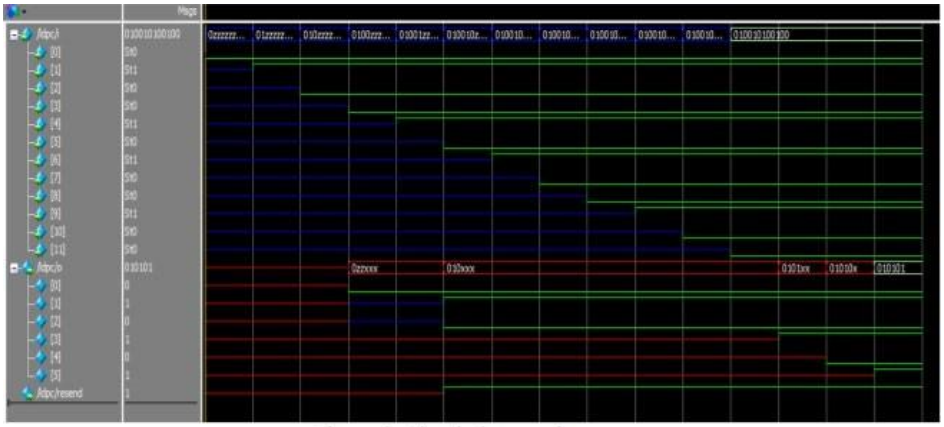

Figure 4: Simulation Result

B. Synthesis:

Mentor Graphics Leonardo Spectrum is used for the synthesis process. The Virtex IV technology is employed. Synthesis involves the examination of outputs such as chip size, latency, etc. The number of look-up tables (LUTs) is a good proxy for chip size. The lag causes a lag in signal propagation. Code synthesis results for the LDPC decoder implementation are displayed in the table below.

Table 1: Synthesis Result

Pass	Area (LUTs)	Delay (in ns)	DFF	PIs	POs
1	15	1	0	12	7

V. CONCLUSION AND FUTURE SCOPE

A. Conclusions:

Based on our research into various encoding and decoding methods and our comprehensive examination of LDPC code, we have concluded that it is a very effective block error correcting code. When we tested this LDPC-coded BPSK system in Matlab, we discovered that its error performance was very close to Shannon's limit; we then tested it in FPGA, and we found that the decoder, a variant of the Modified Sum-Product Algorithm, was effective as well. What follows is a brief synopsis of the project's final results.

- The error-correcting performance of the LDPC Code is almost as good as the error-correction limit set by Shanon.
- The effectiveness of the LDPC Code is contingent on a number of features of the parity matrix, including but not limited to the matrix's size, girth, regularity, etc.
- Because of their naturally parallel structure, cyclic codes are simple to build.
- The error performance of the Sum-product Algorithm (SPA) is better in the log domain than in the probability domain.
- MSPA's (Modified SPA) performance is comparable to that of SPA, and it's simple to construct in an FPGA.
- Several decoding algorithms, including Min-Sum and Message Passing, had their behavior analyzed, and it was determined that they were quite close to Shanons.
- Encoder and decoder FPGA implementations on Spartan 3E were analyzed, and it was determined that device use was allowable, or synthesizable.

B. Future Scope:

LDPC: The code is a strong contender for many modulation methods, including orthogonal frequency division multiplexing (OFDM), digital video broadcast standard 2 (DVB-S2), and wireless access point (Wi-Max) 802.3ad. Researching and implementing an LDPC-based OFDM system is a future objective. OFDM is a potential candidate for future communication techniques that require a lot of bandwidth, like 4G and 5G, and as there is a need of error correcting codes in this technique, we can use LDPC and then try to find its improvement before implementing it. When used with other types of Block error correcting codes, including RS codes, LDPC can increase performance. In future efforts, we shall take into account LDPC codes with RS-codes. As a last step, we'll experiment with different decoder architectures and algorithms in an effort to reduce the complexity of the implementation.

VI. REFERENCES

[1] Conde-Canencia, Laura, et al. "Modeling Adaptive Coded Modulation in real time partially reconfigurable mobile terminals." 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO). IEEE, 2012.

[2] Hailes, Peter, et al. "A flexible FPGA-based quasi-cyclic LDPC decoder." IEEE Access 5 (2017): 20965-20984.

[3] Wang, Zhongfeng, and Zhiqiang Cui. "Low-complexity high-speed decoder design for quasi-cyclic LDPC codes." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 15.1 (2007): 104-114.

[4] Preim, Bernhard, and Dirk Bartz. Visualization in medicine: theory, algorithms, and applications. Elsevier, 2007.

[5] McEliece, Robert J., David J. C. MacKay, and Jung-Fu Cheng. "Turbo decoding as an instance of Pearl's" belief propagation" algorithm." IEEE Journal on selected areas in communications 16.2 (1998): 140-152.

[6] Xiao, Xin, et al. "Designing finite alphabet iterative decoders of LDPC codes via recurrent quantized neural networks." IEEE Transactions on Communications 68.7 (2020): 3963-3974.