Low-Complexity Tree Architecture For Finding The First Two Minima

V. Nagarjuna Reddy1, Syed Zahiruddin2

1M.Tech., PG Scholar, KSRM College of Engineering, Kadapa
2Associate Professor, Dept of ECE, KSRM College of Engineering, Kadapa

Abstract—This brief presents an area-efficient tree architecture for finding the first two minima as well as the index of the first minimum, which is essential in the design of a low-density parity check decoder based on the min–sum algorithm. The proposed architecture reduces the number of comparators by reusing the intermediate comparison results computed for the first minimum in order to collect the candidates of the second minimum. As a result, the proposed tree architecture improves the area–time complexity remarkably.

I. INTRODUCTION

This note constitutes an attempt to highlight some of the main aspects of the theory of low-density parity-check (LDPC) codes. It is intended for a mathematically mature audience with some background in coding theory, but without much knowledge about LDPC codes. The idea of writing a note like this came up during conversations that with Dr. Khorovshahi, head of the Mathematics Section of the Institute for Studies in Theoretical Physics and Mathematics in December 2002. The main motivation behind writing this note was to have a written document for Master’s and PhD students who want to gain an understanding of LDPC codes at a level where they feel comfortable deciding whether or not they want to pursue it as their research topic. The style is often informal not to compromise exactness. The note is by no means a survey. Left out a number of interesting aspects of the theory, such as connections to statistical mechanics, or artificial intelligence. The important topics of general Tanner graphs, and factor graphs have also been completely omitted. Connections to Turbo codes have also been left untouched. My emphasis in writing the notes has been on algorithmic and theoretical aspects of LDPC codes, and within these areas on statements that can be proved. In this paper didn’t discussed any of the existing and very clever methods for the construction of LDPC codes, or issues regarding their implementation. Nevertheless, that this document proves useful to at least some students or researchers interested in pursuing research in LDPC codes, or more generally codes obtained from graphs.

II. LITERATURE REVIEW

1. Jianguang Zhao, Farhad Zarkeshvari, On Implementation Of Min-Sum Algorithm And Its Modifications For Decoding Low-Density Parity-Check (LDPC) Codes, Vol53, No.4, April 2005

The effects of clipping and quantization on the performance of the min-sum algorithm for the decoding of low-density parity-check (LDPC) codes at short and intermediate block lengths are studied. It is shown that in many cases, only four quantization bits suffice to obtain close to ideal performance over a wide range of signal-to-noise ratios. Moreover, we propose modifications to the min-sum algorithm that improve the performance by a few tenths of a decibel with just a small increase in decoding complexity.

2. Mohammad Rakibul Islam, Optimized Min-Sum Decoding Algorithm For Low Density Parity Check Codes, Vol. 2, No. 12, 2011

Low Density Parity Check (LDPC) code approaches Shannon–limit performance for binary field and long code lengths. However, performance of binary LDPC code is degraded when the code word length is small. An optimized min-sum algorithm for LDPC code is proposed in this paper. In this algorithm unlike other decoding methods, an optimization factor has been introduced in both check node and bit node of the Min-sum algorithm. The optimization factor is obtained before decoding program, and the same factor is multiplied twice in one cycle.


Graphics Processor Units are used for many general purposes processing due to high compute power available on them. Regular, data-parallel algorithms map well to the SIMD architecture of current GPU. Irregular algorithms on discrete structures like graphs are harder to map to them. Efficient data-mapping primitives can play crucial role in mapping such algorithms onto the GPU. In this paper, we present a minimum spanning tree algorithm on Nvidia GPUs under CUDA, as a recursive formulation of Borůvka’s approach for undirected graphs.

4. Zhou Zhongl, Yunzhou Lil, Modified Min-Sum Decoding Algorithm For Ldpc Codes Based On Classified Correction, March 17, 2009

In this project, a modified min-sum decoding algorithm based on classified correction is proposed for low density parity check (LDPC) codes. Different from the single correction in the normalized Belief Propagation (BP)-based and offset BP-based algorithms, the proposed algorithm utilizes two corrections for both minimum and sub-minimum magnitudes of input messages in check nodes. Simulation results show that the proposed algorithm can achieve performance very close to that of the BP algorithm.
III. SYSTEM ANALYSIS

It is possible to reduce the number of comparators needed for the second minimum by reusing the comparison results performed for the first minimum. In the proposed architecture, a candidate set \( Y \) for MIN2 is first constructed by using the Prior comparison results, and then a comparison network is additionally constructed to select MIN2. This two-step approach is conceptually similar to SM sort but the second step is much faster in the proposed architecture.

![Figure: SM SORT](image1)

DESCRIPTION

Area-efficient tree architecture for finding the first two minima as well as the index of the first minimum, which is essential in the design of a low-density parity check decoder based on the min–sum algorithm. The proposed architecture reduces the number of comparators by reusing the intermediate comparison results computed for the first minimum in order to collect the candidates of the second minimum. As a result, the proposed tree architecture improves the area–time Complexity remarkably.

IV. BLOCK DIAGRAM

![Figure: Block Diagram](image2)

In line with our motivation and the previously explained normalized min-sum algorithm, we propose the optimized min-sum algorithm. The main feature of our proposed algorithm is the use of the optimization factor. Multiplication of both in check node and bit node update is the basic difference between optimized min-sum algorithm and Normalized Min-sum algorithm. In the Normalized Min-sum algorithm, normalizing factor was used for check node update only. Also in 2 Dimensional Normalized Min Sum algorithm, two different factors for check and bit node updates are used and multiplied in 3 different places, check node processing, A posteriori information and bit node processing. The advantage of the proposed algorithm is that only the optimization factor is used for both bit node and check node updates. Also, the Optimization factor is not multiplied in a posteriori information which reduces complexity of the algorithm.

The proposed algorithm is explained in where a flowchart is shown. First we initialize the bit to check message. Then we update the check message in the horizontal step. In this step, we multiply the Optimization factor with the check message. After that, we proceed to the vertical step. In this step, we update the posteriori information with the help of check message and then we update the bit node. Here, we multiply the Optimization factor with the check message. The last step is the decision making process. If the decoded codeword is correct, we stop there and take it as the output or otherwise repeat the whole decoding process until the iteration number reaches its maximum limit.

![Normalized minimum sum algorithm](image3)

DISADVANTAGES

- The minima technique includes both main digital signal processor and error correction (EC) block.
- To meet ultralow power demand, minima and maxima is used in technique. However, under the complexity, once the critical path delay \( T_{cp} \) of the system becomes greater than the sampling period \( T_{samp} \), the soft errors will occur.
- It leads to severe degradation in signal precision.
ADVANTAGES

✓ The detector has exponentially increased computing complexity with constellation size and number.
✓ It reduces the computational complexity.
✓ It reduces the number of comparators.

APPLICATIONS

✓ Mobile applications.
✓ Computer and cybernetics applications.
✓ Industrial applications.

V. INTRODUCTION TO ASICS AND PROGRAMMABLE LOGIC

The last 15 years have witnessed the demise in the number of cell-based ASIC designs as a means for developing customized SoCs. Rising NREs, development times and risk have mostly restricted the use of cell-based ASICs to the highest volume applications; applications that can withstand the multi-million dollar development costs associated with 1-2 design re-spins. Analysts estimate that the number of cell based ASIC design starts per year is now only between 2000-3000 compared to ~10,000 in the late 1990s.

The FPGA has emerged as a technology that fills some of the gap left by cell-based ASICs. Yet even after 20+ years of existence and 40X more design starts per year than cell-based ASICs, the size of the FPGA market in dollar terms remains only a fraction that of cell-based ASICs. This suggests that there are many FPGA designs that never make it into production and that for the most part, the FPGA is still seen by many as a vehicle for prototyping or college education and has perhaps even succeeded in actually stifling industry innovation.

This paper introduces a new technology, the second generation Structured ASIC, that is tipped to reenergize the path to innovation within the electronics industry. It brings together some of the key advantages of FPGA technology (i.e. fast turnaround, no mask charges, no minimum order quantity) and of cell-based ASIC (i.e. low unit cost and power) to deliver a new platform for SoC design.

This document defines requirements for development of Application Specific Integrated Circuits (ASICs). It is intended to be used as an appendix to a Statement of Work. The document complements the ESA ASIC Design and Assurance Requirements (AD1), which is a precursor to a future ESA PSS document on ASIC design.

Moore’s Law

In the 1960s Gordon Moore predicted that the number of transistors that could be manufactured on a chip would grow exponentially. His prediction, now known as Moore’s Law, was remarkably prescient. Moore’s ultimate prediction was that transistor count would double every two years, an estimate that has held up remarkably well. Today, an industry group maintains the International Technology Roadmap for Semiconductors (ITRS), that maps out strategies to maintain the pace of Moore’s Law.

MODEL-SIM

ModelSim PE, our entry-level simulator, offers VHDL, Verilog, or mixed-language simulation. Coupled with the most popular HDL debugging capabilities in the industry, ModelSim PE is known for delivering high performance, ease of use, and outstanding product support.

Figure: ModelSimSE

6.5 Welcome Menu

Figure: Tool Structure and Flow

The diagram below illustrates the structure of the ModelSim tool, and the flow of that tool as it is used to verify a design.

Model Technology’s award-winning Single Kernel Simulation (SKS) technology enables transparent mixing of VHDL and Verilog in one design. ModelSim’s architecture allows platform independent compile with the outstanding performance of native compiled code.
Figure: Flowchart for VHDL

An easy-to-use graphical user interface enables you to quickly identify and debug problems, aided by dynamically updated windows. For example, selecting a design region in the Structure window automatically updates the Source, Signals, Process, and Variables windows. These cross linked ModelSim windows create a powerful easy-to-use debug environment. Once a problem is found, you can edit, recompile, and re-simulate without leaving the simulator.

ModelSim PE fully supports the VHDL and Verilog language standards. You can simulate behavioral, RTL, and gate-level code separately or simultaneously. ModelSim PE also supports all ASIC and FPGA libraries, ensuring accurate timing simulations. ModelSim PE provides partial support for VHDL 2008.

VI. SIMULATION OUTPUTS

Low complexity realization

Figure: comparators, leading to a low complexity realization

The simulation of the proposed work is done using technology on model sim. The schematics of the 8 input mVG unit is obtained using basic logic gates and look-up-tables (LUTs). The transient analysis is done with the input bits given as voltage, 1.8 V is for logic-1 value and 0 V as logic-0, the waveform for dc power analysis.

AREA TIME COMPLEXITY

Figure: area time complexity

The delay obtained between input and maximum 1 output delay about 12 seconds. The total time elapsed for dc analysis with the peak memory consumption is reported. The total static power consumed by the design is about 221 micro-watts, i.e. minimum power consumption by the design.
VII. CONCLUSION

The low complexity design is a prominent requirement of iterative decoders. We have tried to propose a design that fulfills the necessity with the implementation of the sum min operator using improved architecture. The complexity is reduced as the number of registers and memory consumption decreases. The overall synthesis result provides satisfactory performance for propagation delay i.e. the speed of the output is improved. VLSI implementation can be easily achieved with minimum consumption of area and power as shown with back end analysis.

REFERENCE


