Parallelised Decoding of Turbo Product Codes Using a Many-Cored Processor

B. Sim, L. Cheng and J. Verfeld
School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg, South Africa
Tel: +27 11 717 7244/37, Fax: +27 11 403 1929
Email: benjamin.sim@students.wits.ac.za, {ling.cheng, jaco.versfeld}@wits.ac.za

***Abstract*—With the advent of the compute capable desktop GPU, or many-cored processor, parallel processing has become easily available and cost effective. Thus, the GPU is finding uses in many different fields. Parallel computing offers massive performance improvements, often providing real-time performance on previously cumbersome tasks. One such problem is the inherently parallel problem of turbo product code decoding. Due to the construction of a turbo product code, the decoding of these codes is ideally suited to a many-cored processor. Hence, a decoding algorithm which utilises a graph theoretic approach to efficiently parallelise the decoding of a turbo product code could offer significant performance improvements. The algorithm proposed attempts to optimise the decoding such that no redundant processing occurs. This limits the wasted processing time, which can be costly on platforms lacking processing power. Due to this efficient outlook, this algorithm would be ideally suited to compute-limited devices such as would be used in a mobile platform.**

***Index Terms*—CUDA, Graph Theory, LDPC Codes, Turbo Product Codes**

I. Introduction

In 1954, Elias described a coding technique which concatenated two simpler component codes together, to construct a more powerful coding technique [1]. This technique described by Elias is called a product code and was the first ever implementation of a collection of coding techniques known as turbo codes [2]. Turbo codes were some of the first codes to approach the Shannon channel capacity and have been used in some of the harshest channels available, both deep space and power line communications [3], [4]. The 'turbo' designation of these codes does not refer to the performance of the code itself, but rather to the decoding time. Turbo codes offer short decoding times relative to their often very long lengths [2]. In the case of a product code, the decoding performance can be adversely effected by both the size of the product code and the number of dimensions across which the product code extends.

Due to the poor decoding time performance of large turbo codes, this paper outlines an investigation into an efficient parallelised decoding algorithm for a Turbo Product Code (TPC). To facilitate the parallelised nature of the decoding the algorithm is designed to utilise a many-cored processor, such as a Graphics Processing Unit. The proposed algorithm attempts to parallelise the decoding of TPCs through analysis of the error distribution in the code block. To provide a consistent mathematical model for error distribution, the error shape is modelled through a graph theoretic approach.

Through the remainder of the paper a brief background into the topics of TPCs, many-cored processors and graph theory will be presented. The research motivation for the work is then discussed. This is followed by a very high-level discussion on the proposed solution. Finally a concluding statement is given to summarise the paper.

II. Background

To properly contextualise the reader, it is necessary to introduce some key aspects used in the proposed design. The following topics will be expanded: turbo product codes, many-cored processors and graph theory.

A. Turbo Product Codes

TPCs are a sub-class of modern coding techniques. Turbo codes utilise code concatenation of simpler component codes to construct a more powerful overall code, which offers performance greater than the sum of its component codes. A TPC is constructed by sequentially encoding a data block along all dimensions of the block. Theoretically this can extend until z dimensions, but practically is limited to fewer dimensions. Fig. 1 presents an example of the two dimensional TPC to illustrate the encoding along two dimensions.

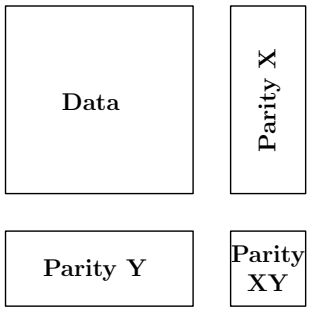

Fig. 1. A two dimensional TPC with horizontal parity 'X', vertical parity 'Y' and parity-on-parity 'XY'.

To encode the two dimensions of a two dimensional code block, as illustrated in Fig. 1, the data is encoded horizontally and then vertically to create two parity information sets, parity 'X' and 'Y'. During the vertical encoding, creating parity section 'Y', the parity section from the horizontal encoding is also encoded to create a parity-on-parity section 'XY' where the horizontal codes parity is encoded. As shown by Equation 1, the overall minimum distance of the

constructed product code is greater than the multiplication of all the minimum distances of all the z component codes.

$$d_{min} \geq d^1_{min} d^2_{min} \dots d^z_{min} \qquad \text{for } z \in \mathbb{N} \qquad (1)$$

Turbo product codes can be decoded by multiple different decoding algorithms such as the maximum-likelihood Chase or maximum a *posteriori* belief propagation algorithms [5], [6].

B. Many-Cored Processors

A many-cored processor is a processor with many cores capable of running many simultaneous threads. In the modern context, the most widely available many-cored processor is the GPU. The GPU has become an ideal platform for parallelised computation as the processors have been developed from the simple raster engines of the past, into powerful many-cored processors capable of complex computation. The Compute Unified Device Architecture (CUDA) provided a compute framework for the modern GPUs [7]. The CUDA provided the first approachable framework for parallelised computation, making the power of parallel processing available to the average developer. The processing power of multi-cored processors is being leveraged in many different aspects of telecommunications, from encryption to coding [8], [9]. Falcao *et al* achieved decoding performance for Low Density Parity Check (LDPC) on a multicored processor which approaches the performance of VLSI hardware implementations [9].

C. Graph Theory

Graph theory is a mathematical study of graphs. A graph is a mathematical construct which abstracts a problem into a system of objects with paired connections between objects [10]. These objects are denoted as *vertices* and all the connections are denoted as *edges*. It should be immediately evident, that certain problems will map to certain graph constructions, such as a tree or bipartite graph [10]. In telecommunications the bipartite graph is not a new construction, as Tanner described the construction of powerful LDPC codes using a bipartite graph. These graphs became known as Tanner graphs [11].

III. Research Motivation

An efficient parallel TPC decoding algorithm offers many benefits. The most notable, and immediately evident, benefit is the performance improvement offered by parallelised algorithms [8], [9]. Possibly less evident is the use of an efficient parallel TPC decoding algorithm on a mobile platform. As mobile platforms improve, so the GPUs present in these devices improve. Thus, it is not inconceivable that in the near future, these GPUs will become compute capable. Thus the mobile platform would be capable of software defined decoding on complex codes with performance comparable to hardware implementations [9].

IV. Proposed Solution

The main focus of the proposed solution is to correctly identify rows/columns on the code block which offer an optimal decoding. Optimal in this case, refers to the minimum number of rows/columns which can be decoded to correct the maximum number of errors in a code block, as illustrated in Figure 2. Thus a graph is used to describe the positions of errors relative to one another. The graph type used is a bipartite graph, where the two disjoint sets are the rows and columns. Using this, mapping multiple optimally decodable vertices can be selected to perform a decoding of applicable rows/columns. Ideally, these rows/columns would be processed in parallel to vastly improve the problem.

	1	2	3	4
A	1	0	1	E
B	E	0	E	0
C	E	1	1	0
D	0	1	E	1

Fig. 2. A toy example of an optimal decoding for a two dimensional product code, with a minimum distance of 3. An optimal decoding would be the decoding of columns 1, 3 and 4, requiring only three decoding, any other combination would require more decodings thus be sub optimal.

V. Conclusion

TPCs are a sub-class of turbo codes. TPCs often have high decoding complexity which results in poor decoding performance. This paper has presented an outline for a parallel TPC decoder. Using a graph theoretic approach the optimal vertices can be identified providing the most performant decoding. By focusing on efficiency, and limiting wasted processing time, the algorithm would be ideal for compute-limited hardware. Thus, such an algorithm could find use, not only, on powerful desktop or server GPUs but also mobile platforms.

References

[1] P. Elias, "Error-free coding," *IEEE Transactions on Information Theory*, vol. 4, no. 4, pp. 29–37, 1954.

[2] T. K. Moon, *Error Correction Coding*. John Wiley & Sons, 2005, ch. 11.

[3] R. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *Communications, IEEE Transactions on*, vol. 46, no. 8, pp. 1003 –1010, aug 1998.

[4] D. Divsalar and E. Pollara, "Low-rate turbo codes for deep-space communications," in *Information Theory, 1995. Proceedings., 1995 IEEE International Symposium on*, sep 1995, p. 35.

[5] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *Information Theory, IEEE Transactions on*, vol. 18, no. 1, pp. 170–182, 1972.

[6] C. Jego and W. Gross, "Turbo decoding of product codes based on the modified adaptive belief propagation algorithm," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*. IEEE, 2007, pp. 641–644.

[7] *NVIDIA CUDA Programming Guide*, 4th ed., NVIDIA Corporation, November 2011.

[8] S. Manavski, "Cuda compatible GPU as an efficient hardware accelerator for AES cryptography," in *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on*, nov. 2007, pp. 65 –68.

[9] G. Falcao, L. Sousa, and V. Silva, "Massively LDPC decoding on multicore architectures," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 2, pp. 309 –322, feb. 2011.

[10] W. Mayeda, *Graph Theory*. John Wiley and Sons, Inc, 1972.

[11] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

Benjamin Sim completed his BSc. Eng. in Information Engineering from University of the Witwatersrand. Currently, he is working toward a MSc. Eng in Electrical Engineering at the same institution. His research interests include coding theory and steganography.