A Parallel Decoder Algorithm for Low Density Parity Check Convolutional Codes for the XInC Multi-threaded Microprocessor

Xin Sheng Zhou
Bruce F. Cockburn
Stephen Bates

Department of Electrical and Computer Engineering
University of Alberta

2nd Floor, ECERF Building
9107 - 116 Street
University of Alberta
Edmonton, Alberta, Canada T6G 2V4
E-mail: {xzhou, cockburn, sbates}@ece.ualberta.ca

Corresponding Author: Xin Sheng Zhou
Xin Sheng Zhou’s Program: M. Sc.
Recommended for Oral Presentation
A Parallel Decoder Algorithm for Low Density Parity Check Convolutional Codes for the XInC Multi-threaded Microprocessor

Xin Sheng Zhou, Bruce F. Cockburn, Stephen Bates
Department of Electrical and Computer Engineering
University of Alberta

Abstract: Convolutional low density parity check codes (LDPC-CCs) are a relatively new class of powerful error control codes. The advantage of these codes over other capacity-approaching codes (e.g. Turbo codes and block-oriented LDPC codes) is that they can encode data blocks of arbitrary length. The goal of our project is to investigate high-performance software implementations of the encoding and decoding algorithms for LDPC-CCs. In particular, we are interested in efficient implementations for the XInC multithreaded microprocessor developed by Eleven Engineering Inc. (Edmonton).

In communication systems, information must be transmitted reliably from end to end. However, due to channel distortions and inevitable noise sources, the quality of the received signals can be degraded to the point where bit errors occur in the received information. In order to detect and correct these errors, redundant information (i.e. check bits) can be added to the intended data (i.e. information bits) at the transmitter before the signal is sent into the channel. In his landmark 1948 paper, Claude Shannon showed that for any given channel bandwidth and signal power to noise power ratio (SNR), there exists a maximum bit rate at which information can be encoded over the channel and decoded without error at the receiver. Finding code constructions that reach the Shannon Limit has been a major challenge. Block-based Low Density Parity Check codes (LDPC-BCs) were first proposed by Gallager in the early 1960s. Along with more recent Turbo Codes, the LDPC-BCs are among the few error control codes whose performances approach the Shannon Limit. Unfortunately, LDPC-BCs codes were not pursued after their initial invention due to the demanding computational complexity of the required decoding algorithm. Recent improvements in hardware speed have made LDPC-BCs practical, and these codes have been adopted by several recent communication standards including the DVB-S2 and the IEEE 802.3 10G Base-T standards.

The LDPC-BCs have the disadvantage of requiring data that is partitioned into a fixed block size. In the late 1990’s, a new convolutional counterpart of the LDPC-BC codes, called the LDPC Convolutional Codes (LDPC-CCs), were proposed by Felstrom and Zigangirov. Compared with the LDPC-BCs, the LDPC-CCs have the following important advantages:

I. As with any convolutional code, and unlike for block codes, the LDPC-CC’s can encode data blocks of arbitrary size. This is a key advantage in the transmission of data packets of variable size (e.g. Internet file transfers) or streaming media (e.g. video).

II. LDPC-CC codes also have advantages in both the encoding and decoding steps. In an encoder for LDPC-BC’s, an entire block of arriving data must generally be stored to allow the parity bits to be calculated. In LDPC-CC’s, the required storage requirements for the encoder are much smaller, involving only recent data bits, recent code bits, and the encoder state. In an LDPC-BC decoder, by contrast, an entire codeword must be stored and processed iteratively as a whole, whereas in an LDPC-CC decoder, a pipeline of identical decoding steps processes the encoded data as it arrives in a continuous stream.

Recent work at the University of Alberta has led to the first hardware implementations of LDPC-CC encoders and decoders using both Field-Programmable Gate Arrays (FPGAs) and semicustom
Application-Specific Integrated Circuits (ASICs). The focus of our new work has been to attempt to efficiently exploit the parallel computing resources in commercially available microprocessors to implement efficient software-based decoders. The XInC chip from Eleven Engineering provides a multithreading environment with eight identical threads implemented directly in hardware. Computation occurs on a common RISC processing core, which is accessed in round-robin fashion by the eight threads. Each thread has its own set of registers and can be used as if it were running on a separate processor. Threads can share information by means of shared memory. This platform appeared to be a reasonably good match for the parallel structure of the decoding algorithm for LDPC-CCs.

An emulator for XInC chip was developed to support algorithm development and performance evaluation. The emulator can run 8 independent execution threads, all 18 XInC instructions, all of the registers and some of the peripherals. The emulator can be used to investigate the potential benefits of additional hardware units to facilitate error control coding (e.g. new instructions, new functional units, or a memory mapped coprocessor) as well as the performance of alternative parallel implementations. Finally, evaluation boards from XInC can be used to measure the performance of parallel algorithm and alternative LDPC codes in real wireless environments. Several dB of coding gain over existing scheme is expected in this work. The practical benefits would include greater reliable transmission range, greater reliable bit rates as the same range, and lower required power to achieve an acceptable bit error rate. For example, the following figure illustrates how the range could be increased by changing the SNR (rho value).

![Figure 1: Range versus Coding Gain](image)

References:
Robert G. Gallager, 1963, Low-Density Parity-Check Codes