1 Executive Summary

Modern flash memory is reliant on error-correcting code to ensure error-free operation. Current flash memory architectures often use linear block codes for this purpose (e.g. Reed-Solomon, Hamming or BCH codes). However, the recent rediscovery of LDPC (Low Density Parity Check) codes, which can achieve superior performance close to the Shannon Limit, has generated much interest in the NAND memory industry. These codes could be used to further improve error correction capability in flash memory, thus allowing for more densely packed memory cells and thus larger capacity drives.

The general aim of this project is to produce a MATLAB simulation of how an error correction system using LDPC codes would work for flash memory. By combining both an error generation and an error correction model, it will be possible to benchmark these rediscovered codes, and subsequently compare them to current generation technologies.

Contents

2 Introduction

3 Overview of Linear Block Codes

Linear block codes are one of the two main classes of forward error correction (FEC), with the other main type being convolutional coding. A linear block code essentially takes a block of binary data, and adds additional redundant data onto it. This block can then be transmitted over a noisy channel, and subsequently decoded at the receiver. The redundant bits in the block are used as parity check equations, which allows a linear block code to both detect and correct errors.

3.1 Definitions for Linear Block Codes

All linear block codes can be described using a set of standard terms and symbols. For this project, all codes use a binary alphabet of $\{0, 1\}$ and hence all operations are over this binary field. n is the block length, the total size of the output codeword. k is the message length, the size of the information vector prior to encoding. The rate of any linear block code, R, is defined as:

$$R = \frac{k}{n} \tag{1}$$

The rate is a measure of the number of information bits compared to the total number of transmitted codeword bits. A high rate code will be more efficient in terms of useful information transmitted, but will have a poorer error correction capability. In Flash Memory, very high rate ($R > 0.9$) codes are used in order to maximise the amount of usable storage space. Conversely, an example use of low rate codes would be in deep-space probe transmissions, where receiving error-free data is more important than speed of transmission.

A linear block code can be represented in two ways: through the $k \times n$ generator matrix $\mathbf{G}$, or the $(n - k) \times n$ parity check matrix $\mathbf{H}$. Each is the null-space of the other, such that:

$$\mathbf{G}\mathbf{H}^{\mathsf{T}} = 0 \tag{2}$$

Unsurprisingly, the generator matrix $\mathbf{G}$ is used in the transmission side when encoding data, and the parity check matrix $\mathbf{H}$ is used at the receiver to detect and correct any errors. A *systematic* generator

matrix takes the form:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \tag{3}$$

At the transmitter, if we take a $1 \times k$ input vector of binary data $\mathbf{x}$, the method of encoding this data into a codeword $\mathbf{c}$, is simply a multiplication operation:

$$\mathbf{c} = \mathbf{x}\mathbf{G} \tag{4}$$

At the receiver, a similar operation is performed:

$$\mathbf{s} = \mathbf{H}\mathbf{c}^{\mathsf{T}} \tag{5}$$

where $\mathbf{s}$ is known as the *syndrome*. If the syndrome is the all zero vector, then error free transmission has occured. Conversely, if any bit of the syndrome is 1, then this represents a particular error pattern.

4 Overview of Flash Memory Technology

5 Decoding of LDPC Codes

6 The AWGN channel: Simulation & Results

7 Modelling a memory-specific noise channel

8 Decoding in the non-Gaussian case

9 The memory channel: Simulation & Results

10 Conclusions