UNIVERSITY OF CALIFORNIA
Los Angeles

Absorbing Set Analysis of LDPC Codes and Read-Channel Quantization in Flash Memory

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Jiadong Wang

2012

ABSTRACT OF THE DISSERTATION

Absorbing Set Analysis of LDPC Codes and Read-Channel Quantization in Flash Memory

by

Jiadong Wang

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2012

Professor Richard D. Wesel, Co-chair

Professor Lara Dolecek, Co-chair

High-capacity NAND flash memories achieve high-density by storing more than one bit per cell. Storage systems require extremely low block-error-rates, making powerful error-correcting codes with low-error floors necessary. Low-density parity-check (LDPC) codes are well known to approach the capacity of the additive white Gaussian noise (AWGN) channel, but they often suffer from error floors and require soft information to achieve better performance. This dissertation tackles these two problems.

The first part of this dissertation introduces the cycle consistency matrix (CCM) as a powerful analytical tool for characterizing and avoiding absorbing sets in separable circulant-based (SCB) LDPC codes. Each potential absorbing set in an SCB LDPC code has a CCM, and an absorbing set can be present in an SCB LDPC code only if the associated CCM is not full column-rank. Using this novel observation, a new code construction approach selects rows and columns from the SCB mother matrix to systematically and provably eliminate dominant absorbing sets by forcing the associated CCMs to be full column-rank. Simulation results both in software and in hardware demonstrate new codes that have steeper error-floor slopes and provide at least one

order of magnitude of improvement in the low FER region.

This dissertation also shows how identifying absorbing-set-spectrum equivalence classes within the family of SCB codes with a specified circulant matrix significantly reduces the search space of code matrices with distinct absorbing set spectra. For a specified circulant matrix, SCB codes all share a common mother matrix and thereby retain standard properties of quasi-cyclic LDPC codes such as girth, code structure, and compatibility with existing high-throughput hardware implementations. SCB codes include a wide variety of LDPC codes such as array-based LDPC codes as well as many common quasi-cyclic codes. Hence the CCM approach should find wide application.

The second part of this dissertation focuses on coding for flash memory. Traditional flash memories employ simple algebraic codes, such as BCH codes, that can correct a fixed, specified number of errors. This dissertation investigates the application to flash memory of low-density parity-check (LDPC) codes which are well known for their ability to approach capacity in the AWGN channel. We obtain soft information for the LDPC decoder by performing multiple cell reads with distinct word-line voltages. The values of the word-line voltages (also called reference voltages) are optimized by maximizing the mutual information between the input and output of the multiple-read channel. Our results show that using this soft information in the LDPC decoder provides a significant benefit and enables the LDPC code to outperform a BCH code with comparable rate and block length over a range of block error rates. Using the maximum mutual-information (MMI) quantization in the LDPC decoder provides an effective and efficient estimate of the word-line voltages compared to other existing quantization techniques.

The dissertation of Jiadong Wang is approved.

Yingnian Wu

Lieven Vandenberghe

Kung Yao

Lara Dolecek, Committee Co-chair

Richard D. Wesel, Committee Co-chair

University of California, Los Angeles

2012

To my parents

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Professor Richard D. Wesel. His passion for research inspired and motivated me during these five years. He taught me not only the methodology to do fascinating research, but also the conscientious attitude to deal with problems. He constantly encouraged me to pursue my own research interest and gave me thoughtful direction when I met obstacles in my research. His warm smile will always light me up even when I faced very difficult problems. As a person, he is extremely kind, respectful and honest. I think I learnt from him to be a better person for these five years and I will appreciate it for my whole life.

I would also like to thank all my other committee members: Professors Lara Dolecek, Kung Yao, Lieven Vandenberghe and Yingnian Wu, for their instructions, insight and help in various aspects. One of my research topics, absorbing set analysis for SCB codes, is inspired and supported by Professor Dolecek. It is a great pleasure to work with her. I have benefited a lot from her solid mathematical skills and resourceful ideas of coding theory. It is a great experience to productively publish five papers in a row with her in just one year. I also want to thank Dr. Dariush Divsalar from JPL. I am very grateful for his useful guidance and help.

I am also very fortunate to have my excellent colleagues in CSL group: Bike Xie, Yuan-Mao Chang, Thomas Courtade, Tsung-Yi Chen, Adam Williamson, Yang Guo and Kasra Vakilinia. Bike helped me a great deal when I joined CSL as junior member. His enthusiasm inspired me as I began to do research. Tom joined CSL the same year as me and we share a lot of fun experience together. He is brilliant and I enjoyed working with him. Tsung-Yi is also a great researcher. We solved many problems together and also enjoyed many conference trips together. These are all valuable memories to me.

I also thank all of my friends who I met in UCLA. I have made many good friends here: Zicong Zhou, Ruihao Ke, Jiajun Zhang, Lichao Chen, Jinchao Li, Mingtian Zhao, Zhongyang Zhang and a lot more. I really enjoyed the time when they are around and appreciate their help during these five years. My life will be less colorful without them.

Finally and most importantly , I owe everything to my parents who love and support me no matter what happens. They are the source of power behind me, and this journey would never be possible if they were not there for me. To them, I dedicate this dissertation.

VITA

2007	B.Eng. (Department of Automation), Tsinghua University, Beijing, China
2008	M.Sc. (Electrical Engineering), UCLA
2007–2012	Research Assistant, Electrical Engineering Department, UCLA
2009–2010	Teaching Assistant, Electrical Engineering Department, UCLA
2010	Intern Engineer, Inphi Inc.
2011	Intern Engineer, Qualcomm Inc.

SELECTED PUBLICATIONS

J. Wang, L. Dolecek, Z. Zhang and R. Wesel, "Combinatorial Analysis of Absorbing Sets for Separable Circulant-Based LDPC Codes". *Submitted to IEEE Trans. Inform. Theory*.

M. Griot, A. I. Vila Casado, W.-Y. Weng, H. Chan, J. Wang, and R. Wesel, "Nonlinear Trellis Codes for Binary-Input Binary-Output Multiple-Access Channels With Single-User Decoding". *IEEE Trans. Comm.*, Feb. 2012, vol. 60, no.2, pp 364–374.

J. Wang, G. Dong, T. Courtade, H. Shankar, T. Zhang and R. Wesel, "Soft Information for LDPC Decoding in Flash Memories". *In preparation*.

J. Wang, T. Courtade, and R. Wesel, "Nonlinear Turbo Code Design for Z Channels and Superposition Coding". *In preparation*.

J. Wang, G. Dong, T. Zhang and R. Wesel, "Use Mutual-Information Optimized Quantization in LDPC decoding for Flash Memory". *in Non-Volatile Memory Workshop 2012*.

J. Wang, T. Courtade, H. Shankar and R. Wesel, "Soft Information for LDPC Decoding in Flash: Mutual-Information Optimized Quantization". *Globecom*, Houston, TX, December 2011.

T.-Y Chen, D. Divsalar, J. Wang and R. Wesel, "Protograph-Based Raptor-Like LDPC Codes for Rate-Compatibility with Short Blocklengths". *Globecom*, Houston, TX, December 2011.

T. Courtade, J. Wang and R. Wesel, "Superposition Coding To Support Multiple Streams, Priorities, and Channel Capacities In The Context Of GMSK". *MILCOM*, Baltimore, MD, November 2011.

J. Zhang, J. Wang, S.G. Srinivasa and L. Dolecek, "Achieving Flexibility in LDPC Code Design by Absorbing Set Elimination". *Asilomar*, Pacific Grove, CA, November 2011.

J. Wang, L. Dolecek, Z. Zhang and R. Wesel, "Absorbing Set Spectrum Approach for Practical Code Design". *ISIT*, Saint-Petersburg, Russia, August 2011.

J. Wang, L. Dolecek and R. Wesel, "Controlling LDPC Absorbing Sets via the Null Space of the Cycle Consistency Matrix," *ICC*, Kyoto, Japan, June 2011.

J. Wang, L. Dolecek and R. Wesel, "LDPC Absorbing Sets, the Null Space of the Cycle Consistency Matrix, and Tanner's Constructions," *ITA*, San Diego, Feb. 2011.

L. Dolecek, J. Wang and Z. Zhang, "Towards Improved LDPC Code Designs Using Absorbing Set Spectrum Properties," *ISTC*, Brest, France, Sept. 2010.

CHAPTER 1

Introduction

1.1 Low Density Parity Check codes

Low-Density Parity-Check (LDPC) codes were first introduced by Gallager in his thesis in the early 1960's [Gal63]. An (n, d_v, d_c) LDPC code was defined as a code of block length n in which each column of the parity check matrix contains d_v ones and each row contains d_c ones. With respect to the regular structure of Gallager's codes, which refer to uniform column and row weight, these codes are now called regular LDPC codes. Gallager also invented soft-decision and hard-decision iterative decoders that were based on message passing. He showed simulation results for codes with block length around 500 bits by using hard-decision decoding, and these results showed the great potential of LDPC codes for error correction. Unfortunately however, the codes that Gallager used to simulate were not long enough to approach Shannon capacity over Hamming bound, and the limited computational capability at that time was another barrier for the simulations of long length codes. During the next decades before the invention of turbo codes, LDPC codes were largely neglected.

The message passing algorithm, which is defined on graphs, has become further understood during the last thrity years. Tanner [Tan81] introduced bipartite graphs to describe low-density codes and the sum-product algorithm based on these graphs. Wiberg *et al.* [Wib96] extended Tanner graphs by including state variables which are invisible to decoders. Pearl [Pea88] systematically described the "belief propagation" algorithm operating on Bayesian networks. It has been recently shown that the forward/backward algorithm for turbo codes, the belief propagation algorithm for LDPC codes, and many other decoding algorithms for other graph-based codes,

are variations of the generalized sum-product (S-P) algorithm operating on the so-called factor graphs (see [KFL01] [For01]). To avoid confusion in notation, we will call all variations of the generalized S-P algorithms "message passing".

In the mid-1990's, Berrou et al. [BGT93] demonstrated the impressive capacity-approaching capability of turbo codes, which activized recent interest in turbo codes and other long random linear codes. It is well-known that Turbo codes share many properties with LDPC codes, especially in the way that message passing is performed in the iterative decoders. These similarities led to an explosion of interest in early work on LDPC codes. In 1999, MacKay et al. [Mac99] showed that LDPC codes have near capacity performance and proposed several empirical rules for how to construct good LDPC codes with random selection. Luby et al. [LMS01] formally showed that properly constructed irregular LDPC codes can approach capacity more closely than regular ones. Richardson, Shokrollahi and Urbanke [RSU01] created a systematic method called "density evolution" to design and analyze the degree distribution asymptotically for extremely long sequence in various channel models.

Some other interesting and important research topics related to LDPC codes emerged in addition to the above fundamental topics. For our purpose, the decoding complexity per bit for message passing depends on the topology of the graph rather than block length, which makes the decoding of extremely long blocks possible. However, the encoding complexity is quadratic in block length if dense generator matrices are used. An almost linear time systematic encoder[RU01] was proposed by converting the parity-check matrix to "approximate" lower triangular form by permutation. The permutation transform doesn't affect code performance because it preserves the sparsity of the original parity-check matrix. It is worthwhile that the transformed generator matrix with a special shape allows itself to encode most of the non-systematic bits recursively in linear time. Chung *et al.* [CRU01] proposed a Gaussian approximation approach to reduce the density evolution algorithm to a one-dimensional problem, with only a little loss in accuracy of performance. Mao [MB01] showed that different scheduling in the message passing decoder gives different performance when the SNR is high enough. Davey [DM98] showed by their simulations that LDPC codes over GF(q) outperform binary LDPC codes. Fos-

sorier [Fos01] designed a reliability-based decoder to tighten the performance gap between message passing decoder and maximum likelihood (ML) decoder. Kou [KLF01] designed LDPC codes based on finite geometries with quasi-cyclic structures and very good minimum distance properties.

We are able to find the performance threshold for infinitely long codes whose associated bipartite graphs are assumed to follow a tree-like structure by density evolution. With this method, Richardson *et al.* [RSU01] designed rate one-half LDPC codes that achieve bit error rate (BER) 10^{-6} within less than 0.1 dB from the capacity limit. But they used the block length of 1 million bits to achieve this performance, which is too long for many applications. Bipartite graphs representing moderate-length codes without nodes of degree one inevitably have many short cycles, while in Richardson's analysis these cycles are neglected for the density evolution. Cycles in bipartite graphs would degrade the optimality of the practical message passing decoder. In other words, the existence of cycles would ruin the conditional independence on the neighbours of a node, therefore graph separation is inaccurate and so is Pearl's polytree algorithm [Pea88], of which belief propagation is a special case.

With respect to the limitation of finite length and the sub-optimality of message passing in a graph with cycles, in the low FER region a flattening of the frame error rate (FER) curve called the error floor usually occurs for LDPC codes with moderate block lengths and high code rates. Prior work indicates that certain sub-graphs called trapping sets [Ric03], and, in particular, a subset of trapping sets called absorbing sets [DZW10] are a primary cause of the error floor in practical implementations. Part of this dissertation focuses on the error floor region and the absorbing set analysis for a class of structured regular LDPC codes.

The other part of this dissertation focuses on the application of LDPC codes in flash memory. Flash memory can store large quantities of data in a small device that has low power consumption and no moving parts. Error control coding for flash memory is becoming more important in a variety of ways as the storage density increases. The increasing number of levels (and smaller distance between levels) means that variations in cell behavior from cell to cell (and over time

due to wear-out) lower the signal-to-noise ratio of the read channel making a stronger error-correction code necessary. Reductions in feature size make inter-cell interference more likely, adding an equalization or interference suppression component to the read channel [LHC02]. Also, the wear-out effect is time varying, introducing a need for adaptive coding to maximize the potential of the system.

LDPC codes have typically been decoded with soft reliability information while flash systems have typically only provided hard reliability information to their decoders. This dissertation demonstrates that at least some soft information is crucial to successfully reaping the benefits of LDPC coding in flash memory. We also explore how much soft information is necessary to provide most of the benefits and how flash systems could be engineered to provide the needed soft information without an unnecessary penalty in complexity or processing time.

1.2 Dissertation outline

In this dissertation, we study the error floor of LDPC codes and the LDPC application for flash memory, in Chapter 2 and Chapter 3 respectively. Chapter 2 introduces the cycle consistency matrix (CCM) as a powerful analytical tool for characterizing and avoiding absorbing sets in separable circulant-based (SCB) LDPC codes. Each absorbing set in an SCB LDPC code has a CCM, and an absorbing set can be present in an SCB LDPC code only if the associated CCM is not full column-rank. Using this novel observation, a new code construction approach selects rows and columns from the SCB mother matrix to systematically and provably eliminate dominant absorbing sets by forcing the associated CCMs to be full column-rank. Simulation results both in software and in hardware demonstrate that the new codes have steeper error-floor slopes and provide at least one order of magnitude of improvement in the low FER region.

The CCM-based analysis shows that quasi-cyclic code families described in [TSS04], [Fos04] and [Fan00] include codes with good absorbing set spectra with a proper choice of parameters. This chapter also shows how identifying absorbing-set-spectrum equivalence classes within the family of SCB codes with a specified circulant matrix significantly reduces the search space of

code matrices with distinct absorbing set spectra.

Chapter 3 explores how to obtain the soft information from flash memory read channel for the LDPC decoder. This chapter first uses pulse-amplitude modulation (PAM) with Gaussian noise to model Flash cell threshold voltage levels, and investigates how to optimize the word-line voltages by maximizing the mutual information (MMI) between the input and the output of the equivalent read channel. After choosing the word-line voltage for each of the reads, the multiple-read channel can be represented by a probability transition matrix and the data can be decoded with a standard belief-propagation algorithm. Then we extend the MMI approach to other channel models such as retention noise model.

Chapter 3 also explores how the quantized setting should be considered in the selection of the LDPC degree distribution. LDPC codes are usually designed with the degree distribution optimized for the AWGN channel [RSU01]. However, our simulations show that, in the quantized setting, adjusting this "optimal" degree distribution can significantly improve performance.

CHAPTER 2

Combinatorial Analysis of Absorbing Sets for Separable Circulant-Based LDPC Codes

For LDPC codes operating over additive white Gaussian noise channels and decoded using message-passing decoders with limited precision, absorbing sets have been shown to be a key factor in error floor behavior. Focusing on this scenario, this chapter introduces the cycle consistency matrix (CCM) as a powerful analytical tool for characterizing and avoiding absorbing sets in separable circulant-based (SCB) LDPC codes. Each potential absorbing set in an SCB LDPC code has a CCM, and an absorbing set can be present in an SCB LDPC code only if the associated CCM is not full column-rank. Using this novel observation, a new code construction approach selects rows and columns from the SCB mother matrix to *systematically and provably* eliminate dominant absorbing sets by forcing the associated CCMs to be full column-rank. Simulation results both in software and in hardware demonstrate that the new codes have steeper error-floor slopes and provide at least one order of magnitude of improvement in the low FER region.

This chapter also shows how identifying absorbing-set-spectrum equivalence classes within the family of SCB codes with a specified circulant matrix significantly reduces the search space of code matrices with distinct absorbing set spectra. For a specified circulant matrix, SCB codes all share a common mother matrix and thereby retain standard properties of quasi-cyclic LDPC codes such as girth, code structure, and compatibility with existing high-throughput hardware implementations. SCB codes include a wide variety of LDPC codes such as array-based LDPC codes as well as many common quasi-cyclic codes. Hence the CCM approach should find wide

application.

2.1 Introduction

Low-density parity-check (LDPC) codes were introduced by Gallager [Gal63] and are well-known for approaching capacity with iterative decoding [RSU01]. However, in the low FER region a flattening of the frame error rate (FER) curve called the error floor usually occurs for LDPC codes with moderate block lengths and high code rates. This performance degradation is due at least in part to the sub-optimality of message passing in a graph with cycles.

For LDPC codes operating over additive white Gaussian noise channels and decoded using message-passing decoders with limited precision, absorbing sets have been shown in [ZDN06, ZDN08, ZDN09] to be a key factor in error floor behavior. This chapter focuses this scenario and on one suitable class of regular LDPC codes: separable, circulant-based (SCB) codes. For a specified circulant matrix, SCB codes share a common mother matrix and include array-based LDPC codes as well as many common quasi-cyclic codes. SCB codes retain standard properties of quasi-cyclic LDPC codes such as girth, code structure, and compatibility with existing high-throughput hardware implementations. These codes might be applicable for high-throughput data storage applications when message passing decoding is used with limited precision.

Prior work indicates that certain sub-graphs called trapping sets [Ric03], and, in particular, a subset of trapping sets called absorbing sets [DZW10] are a primary cause of the error floor in practical implementations. Absorbing sets are trapping sets that are stable under bit-flipping operations.

Since trapping sets are partially due to the sub-optimality of the iterative decoding algorithm, one possible direction to avoid the trapping sets is to improve the decoding algorithm. More effective message-passing algorithms [CGW10, KKH09], better iteration averaging schemes [VF06, LM05], more efficient quantizations [WC07, ZDN09] and post-processing for absorbing sets [ZDN08] all can improve the error floor.

A complementary direction to improving the error floor is to design a parity check matrix with certain properties. The Approximate Cycle EMD[1] (ACE) algorithm [TVW04] or the Progressive Edge Growth (PEG) algorithm [XB04] can be used for a column-by-column construction. Algebraic methods for constructing LDPC codes also demonstrate good error floors as compared to randomly constructed codes. Results in [LZT07], [CXD04], [ZHL10], [HDL11], and [NCM10] present some notable work in this area.

Recent papers have proposed methods to improve the absorbing set spectrum. Introducing additional check nodes [LHM06] or increasing the girth [MKL06] eliminates small trapping sets for some codes. The algorithm in [NCM10] and [NVM10] constructs quasi-cyclic codes from Latin squares so that the Tanner graph does not contain certain trapping sets.

A very clever idea of deterministically constructing structured LDPC codes free of trapping sets was developed in [ABA11b] [ABA11a]: carefully chosen edges in the lifted graph of the original code are swapped to provably eliminate the presence of detrimental trapping sets. Our approach has a similar spirit as that of [ABA11b] [ABA11a]: we also provide a deterministic method for ensuring that the resultant parity check matrix does not have detrimental absorbing sets while maintaining the structure of the matrix. While [ABA11b] [ABA11a] focuses on successively performing graph lifting of a base graph for the elimination of detrimental structures, we focus on a fixed bit node degree for a whole code family. We do not enlarge the graph nor do we alter bit/check node degrees. As a result, the two methods can be further combined for an even more powerful code search and construction.

This chapter introduces the cycle consistency matrix (CCM) as a powerful analytical tool for characterizing and avoiding absorbing sets in separable circulant-based (SCB) LDPC codes. Each absorbing set in an SCB LDPC code has a CCM, and an absorbing set can be present in an SCB LDPC code only if the associated CCM is not full column-rank. Using this novel observation, a new code construction approach selects rows and columns from the SCB mother matrix to systematically and provably eliminate dominant absorbing sets by forcing the associated CCMs

[1]EMD is the acronym for Extrinsic Message Degree.

to be full column-rank. Simulation results both in software and in hardware demonstrate that the new codes have steeper error-floor slopes and provide at least one order of magnitude of improvement in the low FER region.

The CCM-based analysis shows that quasi-cyclic code families described in [TSS04], [Fos04] and [Fan00] include codes with good absorbing set spectra with a proper choice of parameters. This chapter also shows how identifying absorbing-set-spectrum equivalence classes within the family of SCB codes with a specified circulant matrix significantly reduces the search space of code matrices with distinct absorbing set spectra.

Section 2.2 introduces separable circulant-based (SCB) codes and the cycle consistency matrix (CCM). Sections 2.3 and 2.4 identify the CCMs for the dominant [2] absorbing sets of example families of SCB codes with column weights 5 and 4, respectively. These two sections then select specific rows from the SCB mother matrix to eliminate certain dominant absorbing sets by forcing the associated CCMs to be full column-rank. These two sections also show how selecting specific columns from the SCB mother matrix can further eliminate the remaining dominant absorbing sets, again by forcing the associated CCMs to be full column-rank [3] While we provide a detailed analysis of these two important choices for the column weight, it is important to mention tat the analysis does not depend on the Section 2.6 provides simulation results demonstrating a substantial performance improvement obtained by the new codes both in hardware and in software. Section 2.7 delivers the conclusions.

2.2 SCB Codes and the Cycle Consistency Matrix

We now introduce separable, circulant-based (SCB) codes and summarize their key properties in Section 2.2.1. Section 2.2.2 first summarizes absorbing sets as the combinatorial objects of interest and then introduces the main concept of this chapter: the cycle consistency matrix (CCM)

[2]For the purposes of the argument we qualitatively refer to the small absorbing sets as dominant.

[3]It is important to recognize that the CCM approach is not dependent on the variable node degree; rather the chapter focuses on two important choices of variable node degrees for high-rate codes not previously studied in the open literature.

of an absorbing set. The section concludes with a theorem providing necessary and sufficient conditions for the existence of a given absorbing set in SCB codes in terms of its CCM.

2.2.1 Separable, Circulant-based LDPC codes

Circulant-based LDPC codes are composed of circulant matrices and form a subset of (r, c) regular LDPC codes, where r is the variable-node degree and c is the check-node degree. Each circulant matrix is $p \times p$ where p represents a prime number. r and c are both less than or equal to p. The structure of these codes is particularly compatible with high-throughput hardware implementations [ZDN09].

The parity-check matrix of circulant-based LDPC codes can be described as follows:

$$H_{p,f}^{r,c} = \begin{bmatrix} \sigma^{f(0,0)} & \sigma^{f(0,1)} & \sigma^{f(0,2)} & \dots & \sigma^{f(0,c-1)} \\ \sigma^{f(1,0)} & \sigma^{f(1,1)} & \sigma^{f(1,2)} & \dots & \sigma^{f(1,c-1)} \\ \sigma^{f(2,0)} & \sigma^{f(2,1)} & \sigma^{f(2,2)} & \dots & \sigma^{f(2,c-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \sigma^{f(r-1,0)} & \sigma^{f(r-1,1)} & \sigma^{f(r-1,2)} & \dots & \sigma^{f(r-1,c-1)} \end{bmatrix},$$

where σ is the following $p \times p$ circulant matrix:

$$\sigma = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix},$$

and $f(i, j)$ is any function mapping the (row-index, column-index) pairs to the integers $\{0, \dots, p-1\}$.

A column (row) group is a column (row) of circulant matrices. Each variable node has a label (j, k) with $j \in \{0, ..., c-1\}$ being the index of the corresponding column group and $k \in \{0, ..., p-1\}$ identifying the specific column within the group. Similarly, each check node has a label (i, l) where $i \in \{0, ..., r-1\}$ and $l \in \{0, ..., p-1\}$.

This chapter focuses on separable, circulant-based (SCB) codes, which are defined as follows:

Definition 1 (Separable, Circulant-Based (SCB) Code) *An SCB code is a circulant-based LDPC code with a parity-check matrix $H_{p,f}^{r,c}$ in which $f(i,j)$ is separable, i.e., $f(i,j) = g_r(i) \cdot g_c(j) \mod p$.* ■

Parity check matrices of SCB codes with a specified circulant matrix can be viewed as originating from a common SCB mother matrix $H_{p,f_m}^{p,p}$ with $f_m(i,j) = i \cdot j \mod p$. The functions $g_r(i)$ and $g_c(j)$ effectively specify which rows and columns of the mother matrix are selected for the resultant SCB matrix. The ranges of $g_r(i)$ and $g_c(j)$ are both $\{0, \ldots, p-1\}$.

SCB codes include, for example, the constructions in [TSS04], [Fos04], and [Fan00]. The girth of all SCB codes is guaranteed to be at least 6 by the SCB constraint on the submatrix exponent value $f(i,j) = g_r(i) \cdot g_c(j)$ (since all entries in each of g_r and g_c are distinct, [DZW10]).

The SCB structure imposes certain conditions [DZW10] on the variable and check nodes:

Bit Consistency: The neighboring check nodes of a variable node must have distinct row-group (i) labels.

Check Consistency: The neighboring variable nodes of a check node must have distinct column-group (j) labels.

Pattern Consistency: (As shown in [DZW10]) Since every entry in the SCB mother matrix with the value 1 satisfies $k + ij = l \mod p$, if two variable nodes corresponding to columns (j_1, k_1) and (j_2, k_2) share a check node in row group i, they must satisfy:

$$k_1 + ij_1 = k_2 + ij_2 \mod p. \tag{2.1}$$

Note that the converse also holds: If (2.1) is satisfied, then the two variable nodes (j_1, k_1) and (j_2, k_2) share a check node in row group i of the SCB mother matrix.

Cycle Consistency: As shown in [DZW10], the equations of the form (2.1) for any length-$2t$ cycle in an SCB mother matrix, which involve t variable nodes with column-group labels

j_1 through j_t and t check nodes with row-group labels i_1 through i_t, show that the cycle must satisfy:

$$\sum_{m=1}^{t} i_m(j_{(m+1) \bmod t} - j_m) = 0 \bmod p. \tag{2.2}$$

After reviewing absorbing sets below, (2.2) is used to construct a necessary matrix equation for an absorbing set to exist based on the cycles contained in that absorbing set.

2.2.2 Absorbing sets and the Cycle Consistency Matrix

An LDPC code with parity-check matrix H is often viewed as a bipartite (Tanner) graph $G_H = (V, F, E)$, where the set V represents the variable nodes, the set F represents the check nodes, and the set E corresponds to the edges between variable and check nodes.

For a variable node subset $V_{\text{as}} \subset V$, analogous to G_H, let $G_{\text{as}} = (V_{\text{as}}, F_{\text{as}}, E_{\text{as}})$ be the bipartite graph of the edges E_{as} between the variable nodes V_{as} and their neighboring check nodes F_{as}. Let $o(V_{\text{as}}) \subset F_{\text{as}}$ be the neighbors of V_{as} with odd degree in G_{as} and $e(V_{\text{as}}) \subset F_{\text{as}}$ be the neighbors of V_{as} with even degree in G_{as}. We refer to the nodes in $e(V_{\text{as}})$ as "satisfied check nodes" because they will satisfy their parity-check equations when all the nodes in V_{as} are in error.

Definition 2 (Absorbing Set (cf. [DZW10])) *An (n_v, n_o) absorbing set is a set $V_{as} \subset V$ with $|V_{as}| = n_v$ and $|o(V_{as})| = n_o$, where each node in V_{as} has strictly fewer neighbors in $o(V_{as})$ than in $e(V_{as})$.* ■

Moreover, if each variable node in $V \backslash V_{as}$ has strictly fewer neighbors in $o(V_{as})$ than in $F \backslash o(V_{as})$, an (n_v, n_o) absorbing set is called an (n_v, n_o) *fully* absorbing set [DZW10]. An important property of fully absorbing sets is that they are stable under bit-flipping decoding in which the bit values of variable nodes are flipped if a majority of their neighboring check nodes are not satisfied.

Fig. 2.1 shows an example of a $(4, 8)$ absorbing set, which has 4 variable nodes and 8 unsatisfied check nodes.

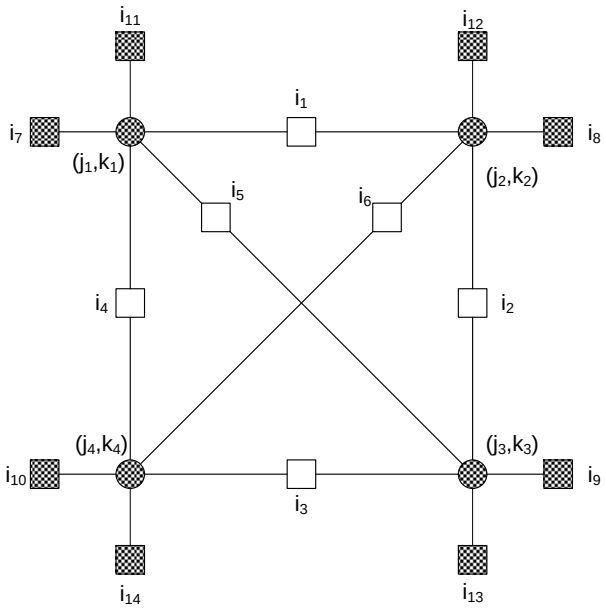


Figure 2.1: Depiction of a $(4, 8)$ absorbing set. Black circles are the four bit nodes (variable nodes) of the absorbing set. The white squares are the satisfied neighboring check nodes, and the black squares are the eight unsatisfied neighboring check nodes.

Suppose there are n_v variable nodes in an absorbing set. Let $j_1, \ldots, j_{n_v}$ be the column-group labels of these n_v nodes in the SCB mother matrix. Define $u_m = j_m - j_1$ for $m \in \{2, ..., n_v\}$ and define $\mathbf{u} = [u_2, ..., u_{n_v}]$. For each cycle in the absorbing set, by replacing the difference of j's with the difference of u's, (2.2) may be written as

$$\sum_{m=2}^{t} (i_{m-1} - i_m) u_m = 0 \mod p, \tag{2.3}$$

where $2t$ is the length of that cycle. Note that the sequence of check-node row groups $\{i_1, i_2, \ldots, i_t\}$ will be different for different cycles reflecting the particular cycle trajectories.

Every cycle in the absorbing set satisfies an equation of the form (2.3). Taken together, these equations produce a matrix equation: $\mathbf{Mu} = 0 \mod p$, where $\mathbf{M}_{\ell m}$ is the coefficient of u_m in (2.3) for the ℓth cycle.

A key property of $\mathbf{M}$ is that $\mathbf{Mu} = 0 \mod p$ completely characterizes the requirement that every cycle in G_{as} satisfies (2.3). Even so, it is not necessary for $\mathbf{M}$ to include a row for every

cycle in the absorbing set.

A cycle need not be included in $\mathbf{M}$ if it is a linear combination of cycles already included in $\mathbf{M}$. Thus the number of rows needed in $\mathbf{M}$ is the number of linearly independent cycles in G_{as}. Some definitions from graph theory [R 06] are necessary to establish the number of linearly independent cycles in G_{as} and hence how many rows are needed for $\mathbf{M}$.

Definition 3 (Incidence Matrix) *For a graph with n vertices and q edges, the (unoriented) incidence matrix is an $n \times q$ matrix B with $B_{ij} = 1$ if vertex v_i and edge x_j are incident and 0 otherwise.* ■

Note that since each edge is incident to exactly two vertices, each column of B has exactly two ones.

The incidence matrix of a graph is useful for identifying the cycles in the graph because every cycle has the property that the indicator vector $\mathbf{x}_c$ of the edges in the cycle satisfies $B\mathbf{x}_c = 0 \mod 2$. In fact, the edges identified by the vector $\mathbf{x}_c$ form a cycle (or a union of cycles) if and only if $B\mathbf{x}_c = 0 \mod 2$. This is formalized in the definition below.

Definition 4 (Binary Cycle Space) *The binary cycle space (bcs) of a graph, which is the set of binary vectors that are indicator vectors of the edges in a cycle or a union of cycles, is the null space of its incidence matrix over $GF(2)$.* ■

For any absorbing-set bipartite graph G_{as} a "variable-node" (VN) graph can be constructed whose only vertices are V_{as} and for which two vertices in the VN graph are connected iff there is a check node that connects them in G_{as}. For a code with girth greater than 4, multiple edges are not allowed between vertices since they will introduce a length-4 cycle. Fig. 2.2 shows the VN graph for the absorbing set graph in Fig. 2.1. It is for the VN graph that we construct the incidence matrix B_{as}. If each satisfied check node in G_{as} has degree 2, then the incidence matrix B_{as} is simply the transpose of the submatrix of $H_{p,f}^{r,c}$ whose rows correspond to the satisfied check nodes of G_{as} (a subset of F_{as}) and whose columns correspond to the variable nodes in V_{as}. If a shared check node has degree-N, its neighbors form a size-N clique in the VN graph.

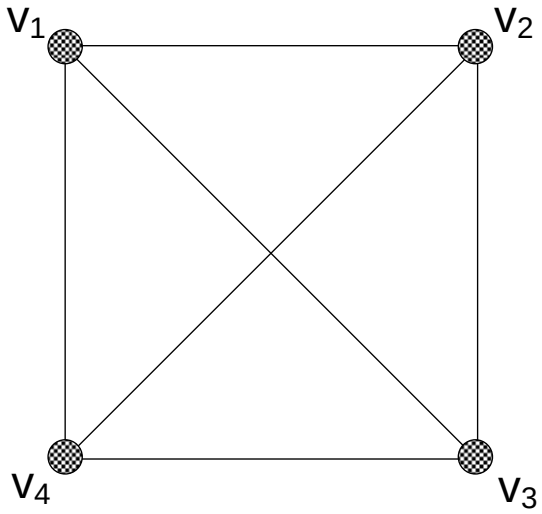


Figure 2.2: Variable-node (VN) graph of the $(4, 8)$ absorbing set of Fig. 2.1.

The number of linearly independent cycles in an absorbing set is the dimension D_{bcs} of its binary cycle space. D_{bcs} is the size of the null space of the incidence matrix B_{as}:

$$D_{\text{bcs}} = q - \text{rank}(B_{\text{as}})\,, \tag{2.4}$$

where q is the number edges in the VN graph of the absorbing set. Knowing the number of linearly independent cycles allows us the specify an efficient $\mathbf{M}$, which we define to be a Cycle Consistency Matrix as follows:

Definition 5 (Cycle Consistency Matrix) *A Cycle Consistency Matrix* $\mathbf{M}$ *of an absorbing-set graph* G_{as} *has* $|V_{as}| - 1$ *columns and* D_{bcs} *rows. The rows of* $\mathbf{M}$ *correspond to* D_{bcs} *linearly independent cycles in* G_{as}*. Each row has the coefficients of* $\mathbf{u}$ *in* (2.3) *for the corresponding cycle.* ■

Note that $\mathbf{M} \cdot \mathbf{u} = 0 \mod p$ completely characterizes the requirement that every cycle in G_{as} satisfies (2.3).

Recall that the vector $\mathbf{u}$ contains difference information about the column groups involved in the absorbing set: the value of the first column group is subtracted from each of the others.

The vector $\mathbf{u}$ cannot be an all-zero vector because the Check Consistency condition requires that variable nodes sharing a check node have distinct column groups, and a zero entry indicates that the variable node is in the first ($j = j_1$) column group. Thus a necessary condition for the existence of a given absorbing set is that its $\mathbf{M}$ does not have full column-rank in $GF(p)$.

Definition 6 (Extensible VN Graph) *If the VN graph of an absorbing set G_{as}^{A} is a sub-graph of the VN graph of at least one other absorbing set G_{as}^{B} with the same number of variable nodes, then we say the VN graph of the absorbing set G_{as}^{A} is extensible.* ■

Equipped with these definitions, the next theorem gives necessary and sufficient conditions for the existence of absorbing sets in SCB codes.

Theorem 1 *Given a proposed absorbing set graph $G_{as} = (V_{as}, F_{as}, E_{as})$, where every variable node is involved in at least one cycle,[4] specified column group labels of the variable nodes in V_{as} in the SCB mother matrix, and specified row-group labels of the check nodes in F_{as} in the SCB mother matrix, the following are necessary conditions for the proposed absorbing set to exist in each daughter SCB LDPC code (with a parity check matrix H that includes the specified row and column groups of that SCB mother matrix): (1) The CCM for G_{as} does not have full column-rank; (2) Variable nodes in V_{as} satisfy the Bit Consistency condition and can form a difference vector $\mathbf{u}$ in the null space of the CCM; and (3) Each check node in F_{as} satisfies the Check Consistency condition. Taken together, these conditions are also sufficient if the VN graph of this absorbing set is not extensible.*

Proof: Each of the three conditions has already been shown to be a necessary condition for the existence of G_{as} in an SCB. If all of these three conditions are satisfied, all the cycles corresponding to the rows of the CCM exist in G_H because they can be constructed as follows: Conditions (1) and (2) ensure a sequence of variable node column groups $[j_1, j_2, ..., j_{|V_{as}|}]$ that form a vector $[u_2, ..., u_{|V_{as}|}]$ in the null space of the CCM. For any fixed k_1, we can compute

[4] *If the variable node degree is at least 2, then each variable node in a given absorbing set must be a part of at least one cycle.*

$k_2, ..., k_{|V_{as}|}$ using the Pattern Consistency requirement of equation (2.1). Any linear combination of these cycles exists in G_H as well.

These cycles cover every edge except edges that connect a variable node to a degree-1 check node in G_{as}. If (1) a variable node's unsatisfied check node is the same as another variable node's unsatisfied check node, or (2) a variable node's unsatisfied check node is the same as one satisfied check node in the graph, or (3) two of the satisfied check nodes are the same, then there exist other independent cycles in the VN graph which extend the VN graph.

In these cases, the original structure is extensible and the conditions in the theorem are not sufficient to establish the existence of the absorbing set G_{as} originally considered. (The conditions in the theorem might instead be caused by the presence of an absorbing set whose VN graph is an extension of the VN graph of the originally considered absorbing set.) However, if the original VN graph is not extensible, the above constructed solution establishes the existence of the proposed absorbing set G_{as} in G_H. ■

Corollary 1 *Given a non-extensible* (n_{v1}, n_{o1}) *absorbing-set graph* $G^1_{as} = (V^1_{as}, F^1_{as}, E^1_{as})$ *whose VN grahp is a sub-graph of the VN graph of another* (n_{v2}, n_{o2}) *absorbing-set graph* $G^2_{as} = (V^2_{as}, F^2_{as}, E^2_{as})$. *Then the existence of* G^1_{as} *is a necessary condition for the existence of* G^2_{as}.

Proof: Suppose the CCMs of G^1_{as} and G^2_{as} are $\mathbf{M}_1$ and $\mathbf{M}_2$ respectively. If the VN graph of G^1_{as} is a sub-graph of the VN graph of G^2_{as}, the independent cycles of G^1_{as} will also be independent in G^2_{as} and thus at least one valid $\mathbf{M}_2$ has $\mathbf{M}_1$ as a sub-matrix:

$$\mathbf{M}_2 = \begin{bmatrix} \mathbf{M}_1 & 0 \\ \mathbf{A} & \mathbf{B} \end{bmatrix}, \tag{2.5}$$

where the sub-matrix $[\mathbf{A} \quad \mathbf{B}]$ represents the other linearly independent cycles in G^2_{as}, which are not included in G^1_{as}. Therefore if there exists a valid $\mathbf{u}_2$ such that $\mathbf{M}_2\mathbf{u}_2 = 0 \mod p$, the first $n_{v1} - 1$ elements would also be a valid $\mathbf{u}_1$ such that $\mathbf{M}_1\mathbf{u}_1 = 0 \mod p$. This shows that the existence of G^1_{as} is a necessary condition of the existence of G^2_{as}. ■

2.3 Analytical results for $r = 5$ SCB codes

We now consider the $H^{r,p}_{p,f(i,j)}$ LDPC codes as an instance of high-performance practical LDPC codes with a hardware-friendly structure. We consider both $r = 4$ and $r = 5$ SCB codes, but we begin with the $r = 5$ case because it is easier to analyze in the sense of eliminating the dominant absorbing sets. This section provides an example with $r = 5$ (five row groups) that shows how to design an SCB code with a specified circulant matrix that eliminates the dominant absorbing sets by selecting rows from the SCB mother matrix to force the CCMs associated with the dominant absorbing sets to be full column-rank.

This example of SCB code design involves two classes of SCB codes:

- Array-based codes [Fan00] are the most elementary SCB codes in which the first r rows of the SCB mother matrix $H^{p,p}_{p,f}, f(i,j) = i \cdot j$ comprise the parity-check matrix. We will refer to this class as the elementary array-based (EAB) codes.

- As shown in [DWZ10], a careful selection of the r row-groups from the overall SCB mother matrix can improve performance over the EAB codes. Thus, selected-row (SR) SCB codes are our second class of SCB codes. The parity-check matrix for these codes is $H^{r,p}_{p,f}, f(i,j) = g_r(i) \cdot j$ where $g_r(i)$ is called the row-selection function (RSF). We will often represent an RSF as the vector $\begin{bmatrix} g_r(0) & g_r(1) & g_r(2) & g_r(3) & g_r(4) \end{bmatrix}$.

Theorem 1 shows that an absorbing set may be avoided either by forcing the associated CCM to be full column rank or by precluding $\mathbf{u}$ from being in the null space of $\mathbf{M}$. Corollary 1 shows that if a non-extensible absorbing set does not exist, then all absorbing sets whose VN graphs contain the VN graph of this absorbing set also do not exist. The CCM approach carefully selects the RSF (and possibly also the analogous column selection function (CSF)) to systematically eliminate small absorbing sets, in the order of the size of the VN graph of the absorbing sets.

Section 2.3.1 establishes that $(4, 8)$ absorbing sets are the smallest possible for a general $r = 5$ code family. Then Corollary 2 shows that $(4, 8)$ absorbing sets indeed exist for this $r = 5$ array-based code family. This theoretical result is also consistent with previous experimental

results of a sum-product decoding algorithm implemented in software and on a hardware emulator [ZDN09] for which it was shown that decoding errors due to $(4, 8)$ absorbing sets dominate the low BER region of an $r = 5$ array-based code.

Sections 2.3.1, 2.3.2 and 2.3.3, respectively, show that $(4, 8)$, $(5, 9)$ and $(6, 8)$ absorbing sets exist in the EAB code with $r = 5$ and that carefully selecting row groups from the SCB mother matrix can systematically eliminate these absorbing sets. Section 2.3.4 provides several good RSFs that can eliminate small absorbing sets for $r = 5$ SCB codes. Section 2.3.5 explores the absorbing set spectrum of the existing quasi-cyclic LDPC codes with the Tanner construction [TSS04]. Section 2.3.6 identifies equivalence classes among SCB codes.

2.3.1 $(4, 8)$ absorbing sets

In this section we analyze $(4, 8)$ absorbing sets. The main result states that $(4, 8)$ absorbing sets exist in SCB codes and specifically in EAB codes (Corollary 3), but that $(4, 8)$ absorbing sets can be provably eliminated from SR-SCB codes using a suitable RSF, as shown in Corollary 4. These theoretical results will be substantiated by experimental results in Section 2.6.

First note that $(4, 8)$ absorbing sets are smallest possible for any regular code with girth at least 6 and variable node degree $r = 5$ [Dol10]. To see why this is true, note that the cases with $n_v < 4$ can be directly eliminated by the girth condition, since each of n_v variable nodes needs to have at least 3 satisfied checks. When $n_v = 4$, the girth constraint prevents any variable node from sharing more than 3 checks with other variable nodes in the absorbing set, so that n_o must necessarily be 8 and each pair of variable nodes in the absorbing set shares a distinct satisfied check. Thus $(4, 8)$ absorbing sets are only possible when the girth equals 6. Therefore, the $(4, 8)$ absorbing set in Fig. 2.1 is the smallest possible absorbing set in the $H^{5,p}_{p,f(i,j)}$ code family. More general results regarding the minimality of absorbing sets are provided in [Dol10].

This section shows that the $(4, 8)$ absorbing set always exists in $r = 5$ EAB codes. However, by carefully selecting row groups from the SCB mother matrix, $r = 5$ SR-SCB codes can systematically eliminate all $(4, 8)$ absorbing sets.

The $(4, 8)$ absorbing set provides an example of how to construct the CCM for a given absorbing set. Using the technique of Section 2.2, the absorbing set graph in Fig. 2.1 induces the variable-node (VN) graph shown in Fig. 2.2. There are five cycles in the variable-node graph, but not all of these cycles need to be explicitly represented in the CCM. The incidence matrix B_{as} of the VN graph shown in Fig. 2.2 with $q = 6$ and $V_{\text{as}} = 4$ is

$$B_{\text{as}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \tag{2.6}$$

Remark 1 *In this absorbing set, since every satisfied check node has degree-2, the incidence matrix is the transpose of the submatrix $\hat{H}_{as}$ of the parity-check matrix that only includes the absorbing-set variable nodes and satisfied check nodes:*

$$\hat{H}_{as} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{2.7}$$

The rank of B_{as} in (2.6) is 3 under $GF(2)$. Thus the dimension of the binary cycle space is $D_{\text{bcs}} = q - \text{rank}(B_{\text{as}}) = 6 - 3 = 3$, which means that three linearly independent cycles span the binary cycle space.

Selecting the following three linearly independent cycles in Fig. 2.1: $v_1 - v_2 - v_3$, $v_1 - v_2 - v_4$, $v_1 - v_3 - v_4$ (here and in the remainder v_l corresponds to the bit node labeled (j_l, k_l)) produces the CCM in (2.8) and its determinant in (2.9) as follows:

$$\mathbf{M} = \begin{bmatrix} i_1 - i_2 & i_2 - i_5 & 0 \\ i_1 - i_6 & 0 & i_6 - i_4 \\ 0 & i_5 - i_3 & i_3 - i_4 \end{bmatrix}, \tag{2.8}$$

$$\det \mathbf{M} = -(i_1 - i_2)(i_6 - i_4)(i_5 - i_3) - (i_2 - i_5)(i_1 - i_6)(i_3 - i_4)\,. \tag{2.9}$$

Using Fig. 2.1 and (2.9), Theorem 1 leads to the following corollary regarding $(4, 8)$ absorbing sets:

Corollary 2 *Given an SCB mother matrix with a specified p, the existence of a selection of integers for row group labels $i_1, \ldots, i_6$ satisfying the Bit Consistency conditions associated with the absorbing sets shown in Fig. 2.1 and satisfying* $\det \mathbf{M}{=}0$ *(with* $\det \mathbf{M}$ *given in* (2.9)*) is necessary and sufficient for the existence of the* $(4, 8)$ *absorbing sets shown in Fig. 2.1.*

Proof: Since the VN graph in Fig. 2.2 of the $(4, 8)$ absorbing sets is a fully connected graph, it is not extensible without introducing a parallel edge which implies a length-4 cycle in the corresponding bipartite graph, an operation that would violate the property that the girth is at least 6. Because the VN graph is non-extensible, Theorem 1 applies. Hence $\det \mathbf{M}{=}0 \mod p$ and Bit Consistency are both necessary conditions. If the Check Consistency is also satisfied, the three conditions together are sufficient by Theorem 1. The rest of the proof shows that identifying a selection of integers for row group labels $i_1, \ldots, i_6$ satisfying the Bit Consistency conditions and satisfying $\det \mathbf{M}{=}0$ implies the existence of j_1, j_2, j_3, j_4 satisfying the Check Consistency conditions.

Consider the Bit Consistency conditions associated with each of the four variable nodes in Fig. 2.1. Our concern is only regarding Bit Consistency conditions that involve satisfied check nodes. Hence the six row-group labels of interest are $\{i_1, \ldots, i_6\}$. For example, the Bit Consistency conditions applied to (j_1, k_1) require that $i_1{\neq}i_5$, $i_1{\neq}i_4$, and $i_4{\neq}i_5$.

There are six different row-group labels of interest, but only five possible row groups since $r = 5$. At least one pair of row-group labels must share the same row group. The inequalities implied by the Bit Consistency conditions allow only $i_1 = i_3$, $i_2 = i_4$, and $i_5 = i_6$.

Just as shown in [DZW10] for $(4, 4)$ absorbing sets, if all three of these equalities are satisfied, the Cycle Consistency conditions require all variable nodes to have the same column group.

which violates the Check Consistency conditions.

Visualizing the $(4, 8)$ absorbing set as a triangular-base pyramid with the four variable nodes as vertices, every point is symmetric to every other point and the three possible equalities allowed by the Bit Consistency conditions are all isomorphisms. Thus, we only need to consider one case of one equality being satisfied and one case of two equalities being satisfied.

In summary, there are only two possible non-isomorphic row-group labelings for the $(4, 8)$ absorbing set as follows [DWZ10]: for any bijective assignment of $\{0, 1, 2, 3, 4\}$ to $\{t, x, y, z, w\}$, $(i_1, i_2, i_3, i_4, i_5, i_6)$ can be either (x, y, x, y, z, w) (Assignment 1) or (x, t, w, y, z, z) (Assignment 2). Applying (2.9) to these two assignments yields the following:

$$\det \mathbf{M} = (y - x)\left((z - x)(w - y) + (z - y)(w - x)\right) \tag{2.10}$$

for Assignment 1, and

$$\det \mathbf{M} = (z - w)(x - t)(y - z) - (y - w)(x - z)(z - t) \tag{2.11}$$

for Assignment 2.

If $\det \mathbf{M}{=}0 \mod p$, there exists a non-zero solution to $\mathbf{M}{\cdot}\mathbf{u}{=}\ 0 \mod p$, where $\mathbf{u}{=}[u_2, u_3, u_4]^T$. With either Assignment 1 or Assignment 2, the resulting $\mathbf{M}$ has six nonzero entries such that if one of $\{u_2, u_3, u_4\}$ is nonzero, all three must be nonzero. Furthermore, the structure of $\mathbf{M}$ ensures that any nontrivial solution to $\det \mathbf{M}{=}0 \mod p$ will have u_2, u_3, and u_4 all distinct. Thus, for a fixed j_1, we can find j_2, j_3, and j_4 without contradiction to Check Consistency (since all are distinct). Fixing one specific k value will determine all others according to Cycle Consistency, yielding the variable nodes of $(4, 8)$ absorbing sets in the code. Therefore, a row-group labeling that satisfies Bit Consistency and $\det \mathbf{M}{=}0 \mod p$ is a necessary and sufficient condition for the existence of $(4, 8)$ absorbing sets. ■

Remark 2 *Now we examine how Corollary 2 applies to EAB and SR-SCB codes. For EAB codes,* $\{x, y, z, w, t\} = \{0, 1, 2, 3, 4\}$. *For Assignment 1 there are no solutions that achieve*

$\det \mathbf{M}{=}0 \mod p$ *for prime* p *large enough (*$p{>}17$*). However, for Assignment 2 there are* 8 *solution sets for* (x, y, z, w, t)*:* $\{(4,3,2,0,1)$*,* $(4,1,2,0,3)$*,* $(3,4,2,1,0)$*,* $(3,0,2,1,4)$*,* $(1,4,2,3,0)$*,* $(1,0,2,3,4)$*,* $(0,3,2,4,1)$*,* $(0,1,2,4,3)\}$*. Note that in all solutions* z *is always* 2*.* ■

As an example, let's find one set of js and ks so that $\det \mathbf{M}{=}0 \mod p$. Among the EAB Assignment-2 solutions above, consider the last one: $(x = 0, y = 1, z = 2, w = 4, t = 3)$ so that $(i_1, i_2, i_3, i_4, i_5, i_6) = (0, 3, 4, 1, 2, 2)$. For this solution set,

$$\mathbf{M} = \begin{bmatrix} i_1 - i_2 & i_2 - i_5 & 0 \\ i_1 - i_6 & 0 & i_6 - i_4 \\ 0 & i_5 - i_3 & i_3 - i_4 \end{bmatrix} = \begin{bmatrix} -3 & 1 & 0 \\ -2 & 0 & 1 \\ 0 & -2 & 3 \end{bmatrix}, \tag{2.12}$$

$$\det \mathbf{M} = -(-3)(1)(-2) - (1)(-2)(3) = 0\,. \tag{2.13}$$

As shown in (2.13), $\det \mathbf{M}{=}0$ for any p. Suppose $p = 47$ and $u_2 = 1$. Solving $\mathbf{Mu} = 0$ gives us $u_3 = 3, u_4 = 2$. Selecting $j_1 = 0$ gives $j_2 = 1, j_3 = 3, j_4 = 2$. From Pattern Consistency, taking $k_1 = 0$ gives $k_2 = 0, k_3 = 41, k_4 = 45$. Thus, once a nonzero solution to $\det \mathbf{M}{=}0 \mod p$ is specified (for example by selecting u_2), specifying any j_1 and k_1 leads directly to a solution that identifies a specific absorbing set.

Continuing with this specific choice of $(i_1, i_2, i_3, i_4, i_5, i_6)$, there are $p - 1$ ways to select u_2, each of which yields a distinct nonzero solution to $\det \mathbf{M}{=}0 \mod p$. (Selecting $u_2{=}0$ gives the all-zeros solution.) For each of these solutions there are p ways to select each of j_1, and k_1, yielding a total of $p^2(p - 1)$ solutions.

Thus there are there $\mathbf{\Theta}(p^3)$ such solutions. As with this specific choice, for each of the eight solutions above there are $\mathbf{\Theta}(p^3)$ absorbing sets. The overall number of absorbing sets is thus $\mathbf{\Theta}(p^3)$. Here and in the remainder of the chapter we use the standard asymptotic notation for $\mathbf{\Theta}$: a positive function $f(p)$ is $\mathbf{\Theta}(p^k)$ if there exist constants c_1 and c_2 such that $0 < c_1 \leq c_2 < \infty$ for which $c_1 p^k \leq f(p) \leq c_2 p^k$ for all $p \geq 0$.

For sufficiently large p (p>17), the $(4, 8)$ absorbing set identified above is always a $(4, 8)$ fully absorbing set since otherwise there would exist a variable node (j_5, k_5) outside the absorbing set incident to at least three of the checks labeled i_7 through i_{14}. Such a configuration would either violate the girth constraint (have girth less than 6) or it would imply the existence of a new absorbing-set configuration spanning four variable nodes, these being the node (j_5, k_5) and three variable nodes from the starting $(4, 8)$ absorbing set. These four variable nodes would necessarily be connected such that their common constraint is given in (2.10), previously shown to not exist for sufficiently large p.

To avoid the $(4, 8)$ absorbing set we need to consider alternatives to the EAB selection of row-group labels. SR-SCB codes can select the row-group labels to force the determinant of the CCM to be nonzero for the $(4, 8)$ absorbing sets. The goal is to identify a set of five (since $r = 5$) row-group labels such that any valid assignment of those labels to i_1 through i_6 satisfies $\det \mathbf{M} \neq 0 \mod p$ in (2.11). One such example is the RSF $[0, 1, 2, 4, 6]$. For a valid assignment of $\{0, 1, 2, 4, 6\}$ to i_1 through i_6, recall that the Bit Consistency conditions allow only $i_1 = i_3$, $i_2 = i_4$, and $i_5 = i_6$. Any such valid assignment of these labels has $\det \mathbf{M} \neq 0 \mod p$ in (2.11) for any prime p greater than 23. Therefore in SR-SCB codes with a well chosen RSF there are no $(4, 8)$ absorbing sets for sufficiently large p. Another example is the RSF $[0, 1, 3, 8, 19]$ which avoids all $(4, 8)$ absorbing sets for $p = 47$ and certain other p's as described in [DWZ10].

The above analysis proves the following corollaries.

Corollary 3 *The $(4, 8)$ (fully) absorbing sets exist in SCB mother matrices and in particular in all EAB codes described by the parity check matrix $H^{5,p}_{p,i\cdot j}$, and their number scales as $\mathbf{\Theta}(p^3)$ in the EAB code.*

Corollary 4 *There are no $(4, 8)$ absorbing sets in the SR-SCB codes described by the parity check matrix $H^{5,p}_{p,g_r(i)\cdot j}$, for prime p large enough and with a proper choice of RSF.*

2.3.2 $(5, 9)$ absorbing sets

Assuming an RSF that avoids the $(4, 8)$ absorbing sets, this section proves that the $(5, 9)$ absorbing sets are the smallest remaining. The CCM approach shows that the $(5, 9)$ absorbing sets always exist in the EAB codes (Corollary 8), but are avoided for SR-SCB codes by some of the RSFs that precluded the $(4, 8)$ absorbing sets (Corollary 9). We start by establishing a series of intermediate results, proven in Corollaries 5, 6 and 7.

Corollary 5 *Assuming* $(4, 8)$ *absorbing sets do not exist,* $(5, b)$ *absorbing sets also do not exist for* $b < 9$*.*

Proof: For $r = 5$, the total number of edges in a $(5, b)$ absorbing set is 25. The number of edges that connect to satisfied check nodes is even. The number of remaining edges b can only be odd. Thus if $b < 9$, b can only take on values from $\{1, 3, 5, 7\}$. For these b values, the absorbing set graph has at least 18 edges connecting to satisfied check nodes and thus the corresponding VN graph has nine or more edges. This forces the VN graph to either have parallel edges (corresponding to length-4 cycles) or contain as a subgraph the VN graph of a $(4, 8)$ absorbing set. Since length-4 cycles are not possible in SCB codes, Corollary 1 implies that the $(5, b)$ absorbing sets with $b < 9$ do not exist if $(4, 8)$ absorbing sets are absent. ■

Corollary 6 *Assuming* $(4, 8)$ *absorbing sets do not exist, the* $(5, 9)$ *absorbing set is the smallest possible absorbing set, and there are no other* $(5, b)$ *absorbing sets.*

Proof: From Corollary 5, $b \geq 9$ in the absence of $(4, 8)$ absorbing sets. Since the number of edges that go to unsatisfied check nodes is at most 10 and b is odd, the only possible $(5, b)$ absorbing set is the $(5, 9)$ absorbing set. We first prove that the VN graph of $(5, 9)$ absorbing sets does not contain the $(4, 8)$ absorbing-set graph as a sub-graph.

Suppose a $(5, 9)$ absorbing set has a $(4, 8)$ absorbing set as a subgraph. We label the five variable nodes as v_1, v_2, v_3, v_4 and v_5. Without loss of generality, let v_1, v_2, v_3 and v_4 form a $(4, 8)$ absorbing set. If v_5 connects to a satisfied check node in the $(4, 8)$ absorbing set, for

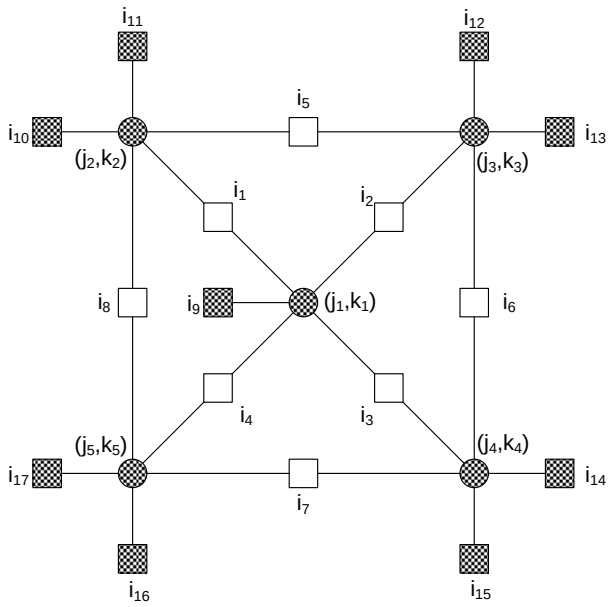


Figure 2.3: Depiction of a $(5, 9)$ absorbing set. Black circles are bit nodes in the absorbing set, white squares are their satisfied checks, and black squares are their unsatisfied checks.

example check node c_1 that connects v_1 and v_2, v_1 and v_2 will have more unsatisfied check nodes than satisfied check nodes. Then v_5 has also to connect to one of v_1's unsatisfied check node c_2 to make v_1 satisfy the absorbing set property. However, v_1, c_1, v_5 and c_2 form a length-4 cycle, which leads to a contradiction.

Thus v_5 cannot connect to the satisfied check nodes of v_1, v_2, v_3 and v_4. Since v_5 has at least three satisfied check nodes, it has to connect to at least three of the formerly unsatisfied check nodes of v_1, v_2, v_3 and v_4. This makes the total number of unsatisfied check nodes seven or less and certainly not nine. This again leads to a contradiction. Therefore $(5, 9)$ absorbing sets cannot have $(4, 8)$ absorbing sets as a subset. ■

The only remaining possible configuration of a $(5, 9)$ absorbing set is where one variable node has four satisfied check nodes and four variable nodes have three satisfied check nodes. Fig. 2.3 depicts this configuration, which has a VN graph that does not contain the VN graph of a $(4, 8)$ absorbing as a sub-graph. Corollary 7 below builds the necessary and sufficient conditions of this absorbing set.

Corollary 7 *For matrix* $\mathbf{M}$ *in* (2.14)*, a row-group label assignment that satisfies the Bit Consistency conditions and* $\det \mathbf{M} = 0 \mod p$ *is necessary and sufficient for the existence of* $(5, 9)$ *absorbing sets in Fig. 2.3.*

Proof: With an analysis similar to that of $(4, 8)$ absorbing sets, the binary cycle space for Fig. 2.3 has dimension 4. We construct the following CCM by selecting the following linearly independent cycles: v_1−v_2−v_3, v_1−v_2−v_5, v_1−v_3−v_4, v_1−v_4−v_5:

$$\mathbf{M} = \begin{bmatrix} i_1 - i_5 & i_5 - i_2 & 0 & 0 \\ i_1 - i_8 & 0 & 0 & i_8 - i_4 \\ 0 & i_2 - i_6 & i_6 - i_3 & \\ 0 & 0 & i_3 - i_7 & i_7 - i_4 \end{bmatrix}. \tag{2.14}$$

As in the proof of Corollary 2, we can show that a row-group label assignment that satisfies the Bit Consistency conditions and $\det \mathbf{M} = 0 \mod p$ is necessary and sufficient for the existence of a $(5, 9)$ absorbing set, where

$$\begin{aligned} \det \mathbf{M} =& (i_1 - i_5)(i_8 - i_4)(i_2 - i_6)(i_3 - i_7) \\ & - (i_1 - i_8)(i_5 - i_2)(i_6 - i_3)(i_7 - i_4) \mod p\,. \end{aligned} \tag{2.15}$$

■

Under the Bit Consistency conditions, there are 5 possible non-isomorphic check labeling patterns of $(i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8)$ as follows: (x, y, z, w, z, w, x, y), (x, y, z, w, t, w, x, y), (x, y, z, w, t, w, x, z), (x, y, z, w, t, x, t, y), and (x, y, z, w, t, w, t, y), where different letters correspond to distinct row-group labels.

The EAB codes have the set of row-group labels $\{x, y, z, w, t\}$ drawn from the set $\{0, 1, 2, 3, 4\}$. For the 4th pattern (x, y, z, w, t, x, t, y), there are 8 solution sets $(x, y, z, w, t) \in \{(4, 0, 1, 3, 2)$, $(4, 0, 3, 1, 2)$, $(3, 1, 4, 0, 2)$,$(3, 1, 0, 4, 2)$, $(1, 3, 4, 0, 2)$, $(1, 3, 0, 4, 2)$, $(0, 4, 1, 3, 2)$, $(0, 4, 3, 1, 2)\}$ that always have $\det \mathbf{M}{=}0$ in (2.15). The other four patterns have a nonzero determinant for p large enough. Once the labels of the check nodes are selected (cf. Fig. 2.3), the variable node labels (pairs (j_1, k_1) through (j_5, k_5)) can be selected in $\Theta(p^3)$ ways as in the $(4, 8)$ case.

For SR-SCB codes with a proper choice of RSF, $\det \mathbf{M} \neq 0 \mod p$ for p large enough. One such example is using an RSF of $[0, 1, 2, 4, 7]$ where $\det \mathbf{M} \neq 0 \mod p$ for the prime p greater than 89 except for the set $p \in \{101, 103, 131, 179\}$. Therefore we can conclude with the following two corollaries.

Corollary 8 $(5, 9)$ *absorbing sets exist in EAB codes described by the parity check matrix* $H^{5,p}_{p,i\cdot j}$*, and their number scales as* $\mathbf{\Theta}(p^3)$.

Corollary 9 *There are no* $(5, 9)$ *absorbing sets in the SR-SCB codes described by the parity check matrix* $H^{5,p}_{p,g_r(i)\cdot j}$*, for prime* p *large enough and with a proper choice of RSF* $g_r(i)$.

2.3.3 $(6, 8)$ absorbing sets

This (fairly long) section considers the $(6, 8)$ absorbing sets, which are the smallest remaining after the $(4, 8)$ and $(5, 9)$ absorbing sets. We will investigate the $(6, 8)$ absorbing sets both for EAB codes and for SR-SCB codes that preclude the $(4, 8)$ and $(5, 9)$ absorbing sets. The six subsections that follow examine respectively the six candidate configurations of $(6, 8)$ absorbing sets to be studied. The configurations are shown in Fig. 2.4 through Fig. 2.9 and constitute all possible choices.

Section summary. We first prove in subsection 2.3.3.1 that the check node degree is not larger than 2 in the SCB codes that preclude $(4, 8)$ absorbing sets. Combinatorial and consistency arguments show (subsections 2.3.3.3, 2.3.3.4 and 2.3.3.6) that three of the remaining five configurations are not present for p sufficiently large in either the EAB code or in SR-SCB codes that preclude the $(4, 8)$ and $(5, 9)$ absorbing sets.

The other two configurations have the cardinality $\mathbf{\Theta}(p^3)$ in the EAB code. However, both of these configurations contain a $(4, 8)$ absorbing set as a subset and thus cannot be present in SR-SCB codes that preclude the $(4, 8)$ absorbing sets. These two configurations are analyzed in subsections 2.3.3.2 and 2.3.3.5.

With this comprehensive analysis of $(6, 8)$ absorbing sets we can conclude that our optimized

SR-SCB codes do not have any $(6, 8)$ absorbing sets.

2.3.3.1 Configuration 1 - check nodes with degree>2

This case is precluded by the following lemma:

Lemma 1 *If a* $(6, 8)$ *absorbing set has a check node that connects to more than two variable nodes in the absorbing set graph, it will contain* $(4, 8)$ *absorbing sets as a sub-graph.*

Proof: An absorbing set contains a $(4, 8)$ absorbing set as a sub-graph if the VN graph of the original absorbing set contains a four-node clique, which is the VN graph of a $(4, 8)$ absorbing set. We consider separately the satisfied and unsatisfied check nodes and show that if a single check node of either type has degree larger than 2, a four-node clique is guaranteed. For the satisfied check nodes, if the degree is larger than two, it must be four or six. Recalling that a degree-N check node in an absorbing set implies an N-node clique in its VN graph, we see that satisfied check nodes can only have degree 2 if four-node cliques are to be avoided.

Now we consider the unsatisfied check nodes. Because the girth constraint precludes parallel edges, the VN graph of a $(6, 8)$ absorbing set has at most $\binom{6}{2} = 15$ edges. If it has all 15 possible edges, the VN graph is a six-node clique and contains $\binom{6}{4} = 15$ four-node cliques. Each edge in a six-node clique is an edge in exactly 6 of the 15 four-node cliques. Thus, removing one edge from a six node clique reduces the number of four-node cliques contained by the graph by 6, from 15 to 9. Removing a second edge from the six-node clique can remove at most six more four-node cliques, so that at least three remain. Thus, if the VN graph of a $(6, 8)$ absorbing set has 13 or more edges, it must contain a $(4, 8)$ absorbing set as a sub-graph.

With six nodes and $r = 5$, the $(6, 8)$ absorbing-set graph has 30 edges between variable nodes and check nodes. If all eight unsatisfied check nodes are singly connected, this accounts for 8 of the 30 edges. Since the satisfied check nodes all have degree 2, the remaining 22 edges of the absorbing set become the eleven edges of its VN graph, which is not enough to guarantee a four-node clique. Consider what happens when some of the unsatisfied check nodes have degree 3

(any larger degree would trivially introduce a four-node clique). Suppose the number of degree-1/2/3 check nodes is n_1/n_2/n_3 respectively. Then we have $n_1 + n_3 = 8$ unsatisfied check nodes, and $3n_3 + 2n_2 + n_1 = 30$ edges between variable and check nodes. These two equations lead to $n_3 + n_2 = 11$. Since the number of the edges in the VN graph is $n_{VN} = 3n_3 + n_2 = 11 + 2n_3$, n_{VN} will be 13 or more if $n_3 > 0$. Therefore if unsatisfied check nodes have degree 3, the $(4, 8)$ absorbing set will exist. Fig. 2.4 shows an example of such a $(6, 8)$ absorbing set, with its two degree-3 check nodes and nine degree-2 check nodes. Having more than two degree-3 check nodes forces more than 15 edges in the VN graph which in turn forces parallel edges in the VN graph causing a violation of the girth constraint. Hence, all cases have been considered, and any $(6, 8)$ absorbing set with a check node having degree larger than two must contain a $(4, 8)$ absorbing set as a sub-graph. ■

Based on Lemma 1, attention is now restricted to cases where all check nodes in the absorbing set graph have degree at most 2.

In a candidate $(6, 8)$ absorbing set, variable nodes can have 3, 4 or 5 satisfied checks. To maintain 8 unsatisfied checks, there can be at most 2 variable nodes with 5 satisfied checks. Suppose first that there are two such variable nodes. Since there are a total of 8 unsatisfied checks, the other 4 variable nodes must each have 3 satisfied and 2 unsatisfied checks. This necessarily implies the configuration shown in Fig. 2.5 which we discuss next.

2.3.3.2 Configuration 2 - Fig. 2.5

Since the configuration in Fig. 2.5 has two overlapping $(4, 8)$ absorbing sets, the existence of this $(6, 8)$ absorbing set relies on both CCM determinants of these two $(4, 8)$ absorbing sets that share an edge. Therefore with a proof similar to that of Corollary 2, we can show the following result:

Lemma 2 *A necessary and sufficient condition for the existence of* $(6, 8)$ *absorbing sets in Fig. 2.5 is* $\det \mathbf{M}_1 = 0 \mod p$ *and* $\det \mathbf{M}_2 = 0 \mod p$*, where* $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *are CCMs of the two internal* $(4, 8)$ *absorbing sets.*

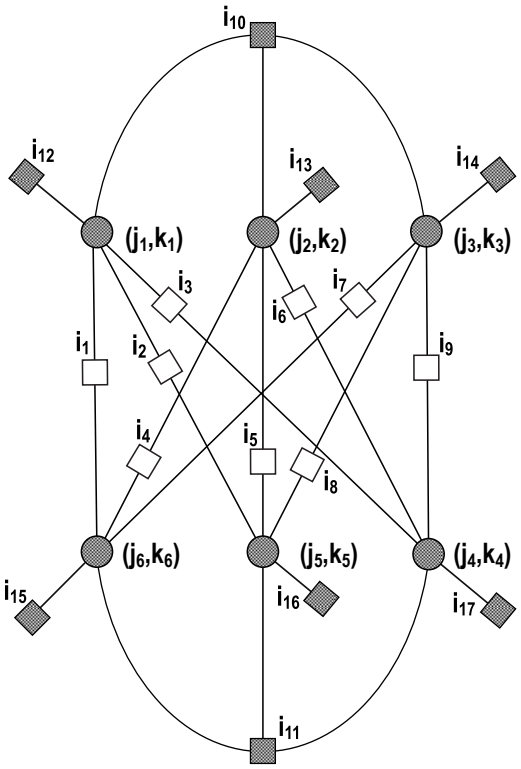


Figure 2.4: Candidate 1 configuration of a $(6, 8)$ absorbing set. In this configuration two check nodes (i_{10} and i_{11}) are connected to three variable nodes. Its VN graph is a six-node clique and that it contains 15 $(4, 8)$ absorbing sets as sub -graphs. In all other candidate $(6, 8)$ configurations that we consider, check nodes have degree ≤ 2.

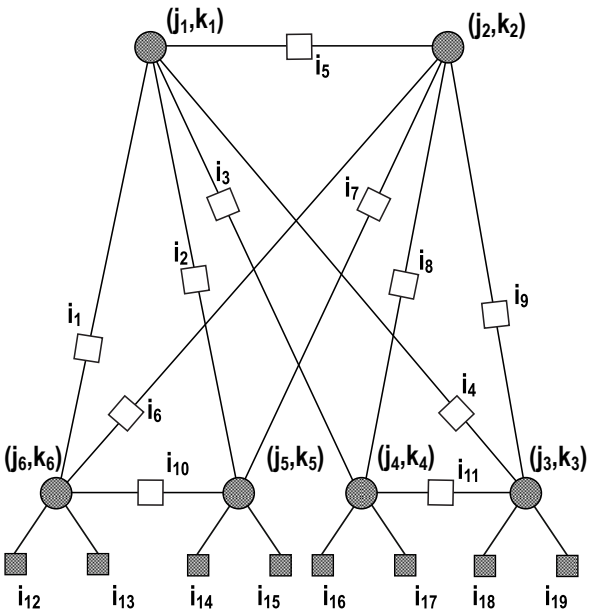


Figure 2.5: Candidate 2 configuration of a $(6, 8)$ absorbing set. In this configuration two variable nodes have 5 satisfied check nodes which forces the remaining four variable nodes each to have exactly two satisfied check nodes. Note the $(4, 8)$ absorbing sets (v_1, v_2, v_3, v_4) and (v_1, v_2, v_5, v_6).

For the EAB codes, we can further show the following result:

Corollary 10 *In the EAB codes, there are* $p^2(p-1)$ $(6,8)$ *absorbing sets with the topology shown in Fig. 2.5 for* p *larger than* 17.

Proof: We may start with the substructure spanning 4 bit nodes (j_1, k_1), (j_2, k_2), (j_5, k_5) and (j_6, k_6). Recall from Remark 2 in Section 2.3.1 that discussed the $(4,8)$ absorbing sets for EAB codes that out of six check nodes shared by these bit nodes, exactly two have the repeated label (it being label $z = 2$). Recall also from Section 2.3.1 that there are $p^2(p-1)$ ways of assigning values of the bit nodes and the check nodes in this $(4,8)$ substructure.

By symmetry of the configuration it suffices to consider the case when this repeated label is the one corresponding to i_5 and i_{10} and when this repeated label corresponds to some other pair of parallel edges. The latter case is not possible since the bit nodes (j_1, k_1), (j_2, k_2), (j_3, k_3) and (j_4, k_4) themselves constitute a $(4,8)$ absorbing set and the propagation of the check labels through the proposed configuration necessarily violates the check label constraints. In the former case, by the Bit Consistency constraints, the labeling of the checks incident to (j_3, k_3) and (j_4, k_4) is unique for each of the nodes (without assuming these two nodes themselves share an edge). Moreover, for the independently selected values of (j_3, k_3) and (j_4, k_4), we show by Pattern Consistency constraint that they indeed have a common check i_{11}, itself labeled "2". As a result, once the values (j_1, k_1), (j_2, k_2), (j_5, k_5) and (j_6, k_6) and their shared checks are pinned down – which can be done in $p^2(p-1)$ ways – the rest of the proposed $(6,8)$ configuration follows uniquely. As a result, the cardinality of $(6,8)$ absorbing sets of the type in Fig. 2.5 is $p^2(p-1)$. Moreover, it can be verified by the configurations of the check node labels, that these absorbing sets are in fact fully absorbing sets. ■

Similarly, the following corollaries are consequences of Corollary 3 and 4.

Corollary 11 *The* $(6,8)$ *absorbing sets in Fig. 2.5 exist in EAB codes described by the parity check matrix* $H^{5,p}_{p,i\cdot j}$*, and their number scales as* $\mathbf{\Theta}(p^3)$.

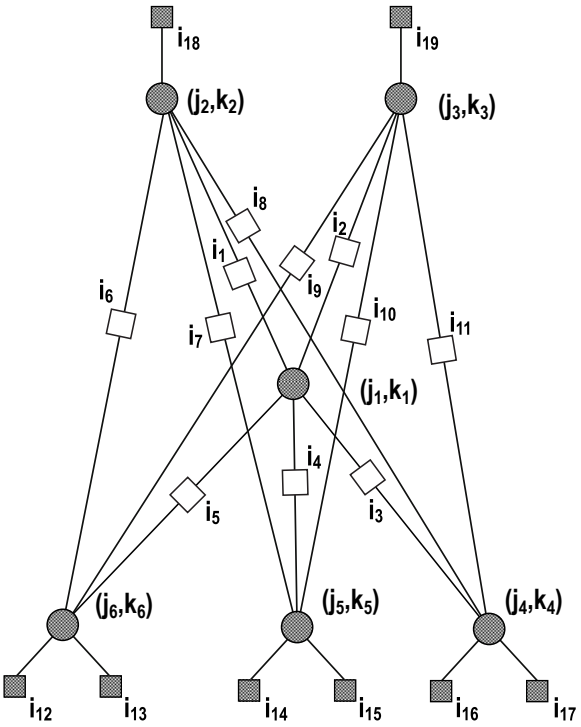


Figure 2.6: Candidate 3 configuration of a $(6, 8)$ absorbing set. This is the first of two configurations with exactly one variable node that has 5 satisfied check nodes.

Corollary 12 *There are no* $(6, 8)$ *absorbing sets with the structure of Fig. 2.5 in the SR-SCB codes described by the parity check matrix* $H^{5,p}_{p,g_r(i)\cdot j}$*, for prime* p *large enough with a proper choice of RSF (that precludes (4,8) absorbing sets).*

Suppose now that there is exactly one variable node in the absorbing set having all five checks satisfied. The variable nodes in the absorbing set must necessarily be arranged either as in Fig. 2.6 or Fig. 2.7.

2.3.3.3 $(6, 8)$ configuration candidate 3 - Fig. 2.6

Lemma 3 *In the Tanner graph corresponding to $H^{5,p}_{p,f(i,j)}$ for the EAB codes and SR-SCB codes, there are no $(6, 8)$ absorbing sets for p large enough of the type shown in Fig. 2.6.*

Proof: Without loss of generality we may assign check node labels for the checks emanating from the variable node (j_1, k_1) as follows: $i_1 = x$,$i_2 = y$, $i_3 = z$, $i_4 = w$, and $i_5 = t$, where x, y, z, w, t are the five distinct check labels.

The binary cycle space for Fig. 2.6 has dimension 6. We construct the following CCM by selecting the following linearly independent cycles: $v_1 - v_2 - v_6, v_1 - v_3 - v_6, v_1 - v_2 - v_5, v_1 - v_3 - v_4, v_1 - v_3 - v_5, v_1 - v_2 - v_4$:

$$\mathbf{M} = \begin{bmatrix} x - i_6 & 0 & 0 & 0 & i_6 - t \\ 0 & y - i_9 & 0 & 0 & i_9 - t \\ x - i_7 & 0 & 0 & i_7 - w & 0 \\ 0 & y - i_{11} & i_{11} - z & 0 & 0 \\ 0 & y - i_{10} & 0 & i_{10} - w & 0 \\ x - i_8 & 0 & i_8 - z & 0 & 0 \end{bmatrix}. \tag{2.16}$$

The rank of the matrix is at most 5 so in fact we may consider the 5×5 submatrix formed by the first five rows (call it B). If the matrix B is full rank, then $\mathbf{M}$ is full rank. Hence $\det(B) = 0$ is necessary for the existence of the absorbing sets of this type. This condition can be expressed as

$$\begin{aligned} -(i_{11} - z)[-(x - i_6)(i_9 - t)(i_7 - w)(y - i_{10}) \\ +(x - i_7)(i_6 - t)(y - i_9)(i_{10} - w)] = 0 \mod p. \end{aligned} \tag{2.17}$$

Also consider the bottom-left 4×4 submatrix (call it A). Note that the Bit Consistency conditions ensure that every element of $\mathbf{M}$ that is not explicitly zero in (2.16) must be nonzero. Thus, if the matrix A is full rank, then $\mathbf{M}$ is full rank (rank-5) because either of the top two rows provide a row linearly independent from the bottom four rows. Hence $\det(A) = 0$ is necessary

for the existence of absorbing sets of the type shown in Fig. 2.6. This condition can be expressed as

$$\begin{aligned} &-(x-i_7)(i_{10}-w)(y-i_{11})(i_8-z) \\ &+(x-i_8)(i_7-w)(i_{11}-z)(y-i_{10}) = 0 \mod p. \end{aligned} \tag{2.18}$$

For the values of i_6, i_7, i_9, i_{10} and i_{11} in the set described by $f(i,j)$ both for the EAB and for the SR-SCB codes, and such that the i labels of check nodes sharing a variable node are distinct (see Fig. 2.6), (2.17) and (2.18) evaluate to zero for only finite number of values of the parameter p.

For $g_r(i) = i$ (the EAB code) $\det(B) \neq 0$ for $p > 23$. We can also find many SR-SCB codes where (2.17) and (2.18) evaluate to zero only for a finite number of values of the parameter p. For example, with the SR-SCB code defined by the RSF $[0, 1, 2, 4, 7]$, $\det(A) \neq 0$ for $p > 89$. Thus for p sufficiently large, neither the EAB code nor well-designed SR-SCB codes contain $(6, 8)$ absorbing sets of the type shown in Fig. 2.6. ■

Incidentally, there do exist values of p for certain RSFs for which both $\det(A) = 0$ and $\det(B) = 0$. For example, for $p = 11, 17, 19, 23$ with the RSF $[0, 1, 2, 3, 4]$ both conditions hold. As we show in Corollary 13 below, $\det(A) = 0$ and $\det(B) = 0$ together are a sufficient condition for the existence of absorbing sets of the type shown in Fig. 2.6 if $(4, 8)$ absorbing sets have been excluded by the RSF. We also identify RSFs such as $[0, 1, 2, 4, 8]$ where both determinants are zero for an infinite number of p values.

Corollary 13 *In the Tanner graph corresponding to $H^{5,p}_{p,f(i,j)}$, if $(4, 8)$ absorbing sets have been excluded by the RSF, $(6, 8)$ absorbing sets of the type shown in Fig. 2.6 exist if and only if $\det(A) = 0$ and $\det(B) = 0$, where A and B are as defined in the proof of Lemma 3.*

Proof: We already established above that both $\det(A) = 0$ and $\det(B) = 0$ are necessary for the existence of $(6, 8)$ absorbing sets of the type shown in Fig. 2.6. At this point we need only establish that these two conditions are also sufficient.

The corollary to be proven assumes that the row labels $i_1, \ldots, i_{11}$ have been specified so that the Bit and Check Consistency conditions are satisfied and such that $\det(A) = 0$ and $\det(B) = 0$.

The first three rows of A are always linearly independent. (Recall that every element of $\mathbf{M}$ that is not explicitly zero in (2.16) must be nonzero.) Thus if $\det(A) = 0$ the bottom row of A is a linear combination of the other 3 rows. Thus, the bottom row of $\mathbf{M}$ is a linear combination of the three rows of $\mathbf{M}$ just above it, and $rank(B) = rank(\mathbf{M})$.

Thus, if $\det(A) = 0$ and $\det(B) = 0$, $\mathbf{M}$ is not full rank, and there exists a non-zero $\mathbf{u}$ such that equation (2.16) is satisfied.

If $(4, 8)$ absorbing sets have been excluded by the selected RSF, then the $(6, 8)$ absorbing set in Fig. 2.6 is not extensible. Thus by Theorem 1 $\det(A) = 0$ and $\det(B) = 0$ is a sufficient condition for the existence of the $(6, 8)$ absorbing set in Fig. 2.6. That is, with the row labels $i_1, \ldots, i_{11}$ specified so that the Bit and Check Consistency conditions are satisfied and such that $\det(A) = 0$ and $\det(B) = 0$, selecting any j_1, j_2, and k_1 will lead to the the entire specification of the absorbing set as with Corollary 2. ■

2.3.3.4 $(6, 8)$ configuration candidate 4 - Fig. 2.7

Lemma 4 *In the EAB codes corresponding to $H^{5,p}_{p,i\cdot j}$, there are no $(6, 8)$ absorbing sets with the topology shown in Fig. 2.7 for p large enough.*

Proof: The binary cycle space for Fig. 2.7 has dimension 6. We construct the following CCM by selecting the following linearly independent cycles: $v_1{-}v_2{-}v_3$, $v_1{-}v_2{-}v_4$, $v_1{-}v_3{-}v_4$, $v_1{-}v_3{-}v_6$, $v_1{-}v_4{-}v_5$, $v_1{-}v_5{-}v_6$:

$$\mathbf{M} = \begin{bmatrix} i_1 - i_2 & i_2 - i_5 & 0 & 0 & 0 \\ i_1 - i_6 & 0 & i_6 - i_4 & 0 & 0 \\ 0 & i_5 - i_3 & i_3 - i_4 & 0 & 0 \\ 0 & i_{11} - i_5 & 0 & 0 & i_7 - i_{11} \\ 0 & 0 & i_{10} - i_4 & i_8 - i_{10} & 0 \\ 0 & 0 & 0 & i_9 - i_8 & i_7 - i_9 \end{bmatrix}. \tag{2.19}$$

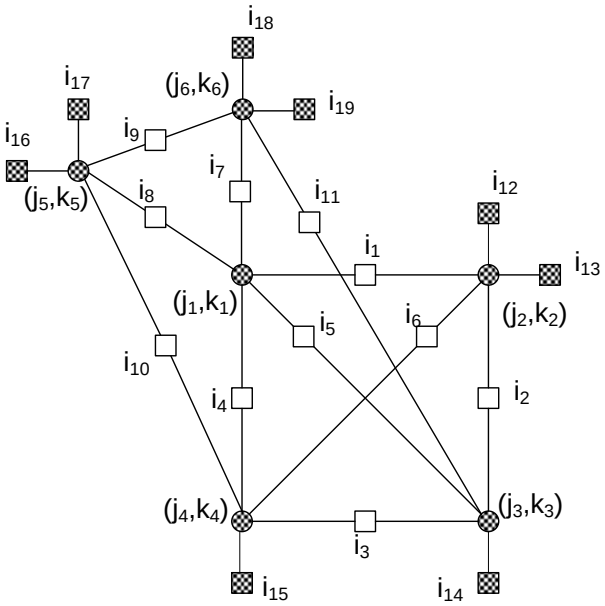


Figure 2.7: Candidate 4 configuration of a $(6, 8)$ absorbing set. This is the second of the two configurations with exactly one variable node that has 5 satisfied check nodes.

It suffices to consider the case when the labels $i_1, i_2, i_3, i_4, i_5, i_6$ adopt one of the following two assignments: $(i_1, i_2, i_3, i_4, i_5, i_6) = (x, t, w, y, z, z)$ or (z, t, z, y, x, w).

As with the proof of Lemma 3, $\mathbf{M}$ is not full column-rank if $\det A = 0 \mod p$ and $\det B = 0 \mod p$, where A is the top-left 3×3 submatrix of $\mathbf{M}$ and B is the bottom-right 4×4 submatrix of $\mathbf{M}$. The matrix A corresponds to the $(4, 8)$ absorbing set involving variable nodes v_1, v_2, v_3, and v_4. As shown in Corollary 3, EAB codes always contain $(4, 8)$ absorbing sets. Thus there is always an assignment of EAB row group labels to $i_1, i_2, i_3, i_4, i_5, i_6$ such that $\det A = 0 \mod p$.

When $\det A = 0 \mod p$, the second row of A is a linear combination of the the first and third rows of A. Thus the second row of $\mathbf{M}$ is a linear combination of the first and third rows. Therefore the rank of $\mathbf{M}$ only depends on the submatrix $\hat{\mathbf{M}}$ that contains all the rows of $\mathbf{M}$ except the second row. Thus for the first assignment,

$$\begin{aligned}\det \hat{\mathbf{M}} =&(x-t)[(z-w)(i_{10}-y)(i_7-i_{11})(i_9-i_8)\\ &-(i_{11}-z)(w-y)(i_8-i_{10})(i_7-i_9)] \equiv 0 \mod p\end{aligned} \tag{2.20}$$

is a necessary condition for the existence of the $(6,8)$ absorbing set with the configuration shown in Fig. 2.7. Similar analysis can be done for the second assignment. This constraint cannot be satisfied for $p > 41$ for i_1 to i_{11} taking values in the set $\{0,1,2,3,4\}$ and such that the Bit Consistency constraints are satisfied for both labellings. ■

Note that the VN of the configuration shown in Fig. 2.7 contains a subgraph that is the VN graph of a $(4,8)$ absorbing set. The following again is an easy consequence of Corollary 4.

Corollary 14 *The configuration shown in Fig. 2.7 is not possible in SR-SCB codes for which* $(4,8)$ *absorbing sets are absent.*

In the remainder we consider the case when no bit node in the absorbing set has all five satisfied checks. This constraint implies one of the configurations shown in Fig. 2.8 and Fig. 2.9.

2.3.3.5 $(6,8)$ configuration candidate 5 - Fig. 2.8

Lemma 5 *In the Tanner graph corresponding to* $H^{5,p}_{p,i\cdot j}$ *there are* $\Theta(p^3)$ $(6,8)$ *absorbing sets of the type shown in Fig. 2.8 for* p *large enough.*

Proof: Note that by the property of $(4,8)$ absorbing sets satisfied check nodes in the sub-structure spanning bit nodes (j_1,k_1), (j_2,k_2), (j_3,k_3), and (j_4,k_4) can be labelled by either $(i_1,i_2,i_3,i_4,i_5,i_6)=(x,t,w,y,z,z)$ or $(i_1,i_2,i_3,i_4,i_5,i_6)=(z,t,z,y,x,w)$.

Using the technique of Section 2.2.2 we construct the CCM for this configuration under labeling $(i_1,i_2,i_3,i_4,i_5,i_6)=(x,t,w,y,z,z)$. The binary cycle space for Fig. 2.8 has dimension 6. We construct the following CCM by selecting the following linearly independent cycles: $v_1-v_2-v_3, v_1-v_2-v_4, v_1-v_3-v_4, v_1-v_2-v_5, v_1-v_3-v_6-v_4, v_1-v_5-v_6-v_4$:

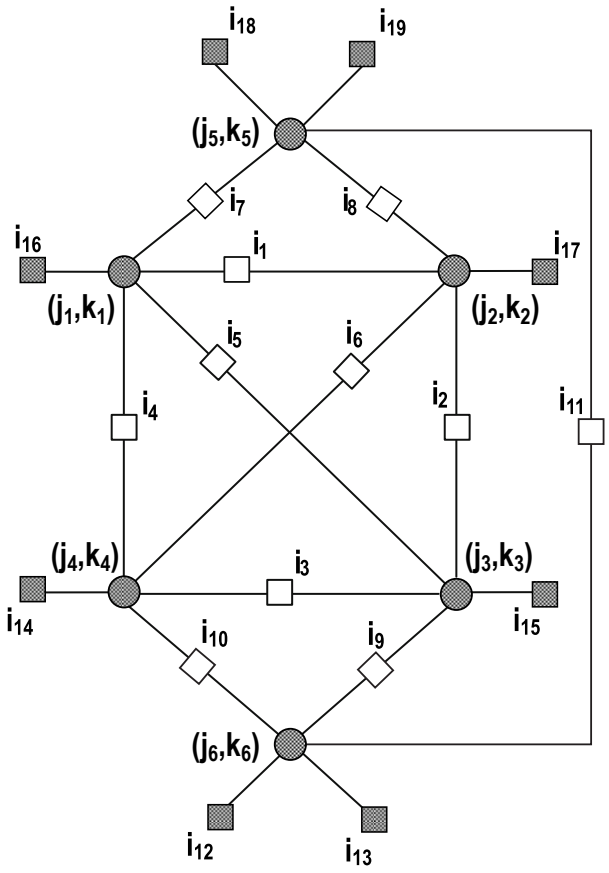


Figure 2.8: Candidate 5 configuration of a $(6, 8)$ absorbing set. This is the first of two configurations with no variable nodes that have 5 satisfied check nodes.

$$\mathbf{M} = \begin{bmatrix} x-t & t-z & 0 & 0 & 0 \\ x-z & 0 & z-y & 0 & 0 \\ 0 & z-w & w-y & 0 & 0 \\ x-i_5 & 0 & 0 & i_5-i_4 & 0 \\ 0 & i_9-w & w-i_{10} & 0 & i_{10}-i_9 \\ 0 & 0 & y-i_{10} & i_{11}-i_4 & i_{10}-i_{11} \end{bmatrix}. \tag{2.21}$$

Analogous to the proof of Lemma 4, the top-left 3-by-3 matrix has a zero determinant and without loss of generality, suppose the second row of $\mathbf{M}$ is linearly dependent of the other five rows. Denote the other five rows as $\hat{\mathbf{M}}$. Its determinant is

$$\begin{aligned} \det \hat{\mathbf{M}} &= -(x-i_5)(t-z)(w-y)(i_{10}-i_9)(i_{11}-i_4) \\ &+ [-(z-w)(i_5-i_4)((w-i_{10})(i_{10}-i_{11}) - (i_9-i_{10})(i_{10}-i_9)) \\ &+ (i_9-w)(w-y)(i_5-i_4)(i_{10}-i_{11})](x-t) \mod p. \end{aligned} \tag{2.22}$$

In fact this determinant always evaluates to zero for every value of p for 6 non-isomorphic edge labellings. For each such labelling, once the values of j_1, k_1 and j_2 are selected (which can be done in $p^2(p-1)$ ways), the rest of values in the configuration follows uniquely. Therefore there are $6p^2(p-1)$ such absorbing sets.

For certain small values of p, $p \in \{2, 3, 7, 11, 13, 31, 47\}$, there are additional solutions for which (2.22) evaluates to zero, raising the total number of solutions to $8p^2(p-1)$. Nonetheless, the scaling $\mathbf{\Theta}(p^3)$ of the cardinality of the absorbing sets still holds.

In the latter case, for $(i_1, i_2, i_3, i_4, i_5, i_6) = (z, t, z, y, x, w)$ we likewise establish a matrix relating the labels of the check nodes $(i_7, i_8, i_9, i_{10}, i_{11})$ in Fig. 2.8 incident to bit nodes (j_5, k_5) and (j_6, k_6) to adjacent checks. Again, by imposing the Bit Consistency conditions, we conclude that the above constraint cannot hold for $p > 23$ and therefore such a labelled configuration is in fact not possible for large enough p. ■

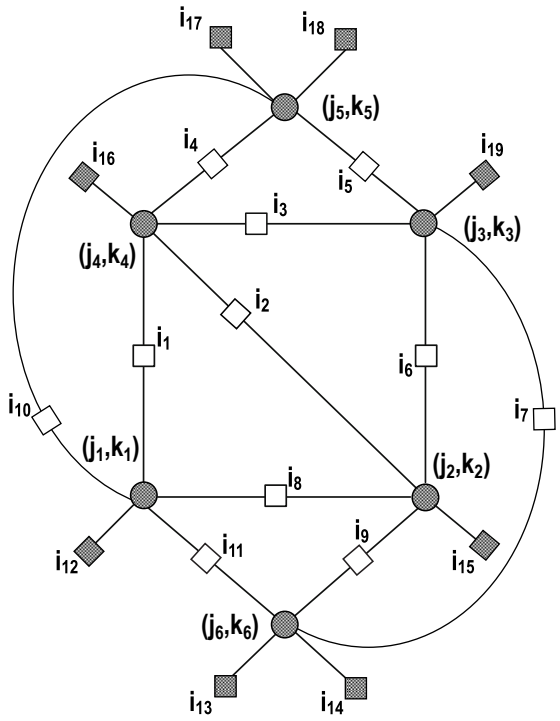


Figure 2.9: Candidate 6 configuration of a $(6, 8)$ absorbing set. This is the second of two configurations with no variable nodes that have 5 satisfied check nodes.

As in the previous case, the VN graph of the configuration shown in Fig. 2.8 has a subgraph that is the VN graph of a $(4, 8)$ absorbing set. The following again is an easy consequence of Corollary 4.

Corollary 15 *The configuration shown in Fig. 2.8 is not possible in SR-SCB codes for which* $(4, 8)$ *absorbing sets are absent.*

2.3.3.6 $(6, 8)$ configuration candidate 6 - Fig. 2.9

The last configuration we consider is the one in Fig. 2.9.

Lemma 6 *In the Tanner graph corresponding to* $H^{5,p}_{p,f(i,j)}$*, for* p *large enough there are no* $(6, 8)$

absorbing sets of the type shown in Fig. 2.9 in either the EAB codes or in the SR-SCB codes.

Proof: Using the technique of Section 2.2.2 we construct the CCM for this configuration. The binary cycle space for Fig. 2.9 has dimension 6. We construct the following CCM by selecting the following linearly independent cycles: $v_1 - v_2 - v_4, v_1 - v_4 - v_5, v_2 - v_3 - v_4, v_3 - v_4 - v_5, v_2 - v_3 - v_6, v_1 - v_2 - v_6$:

$$\mathbf{M} = \begin{bmatrix} i_8 - i_2 & 0 & i_2 - i_1 & 0 & 0 \\ 0 & 0 & i_1 - i_4 & i_4 - i_{10} & 0 \\ i_6 - i_2 & i_3 - i_6 & i_2 - i_3 & 0 & 0 \\ 0 & i_5 - i_3 & i_3 - i_4 & i_4 - i_5 & 0 \\ i_9 - i_6 & i_6 - i_7 & 0 & 0 & i_7 - i_9 \\ i_8 - i_9 & 0 & 0 & 0 & i_9 - i_{11} \end{bmatrix}. \tag{2.23}$$

The inspection of $\mathbf{M}$ reveals that $rank(\mathbf{M}) = 4$ if and only if $\det(A) = 0$ and $\det(B) = 0$, where

$$A = \begin{bmatrix} i_8 - i_2 & 0 & i_2 - i_1 & 0 \\ 0 & 0 & i_1 - i_4 & i_4 - i_{10} \\ i_6 - i_2 & i_3 - i_6 & i_2 - i_3 & 0 \\ 0 & i_5 - i_3 & i_3 - i_4 & i_4 - i_5 \end{bmatrix}, \tag{2.24}$$

$$B = \begin{bmatrix} i_8 - i_2 & 0 & i_2 - i_1 & 0 & 0 \\ 0 & 0 & i_1 - i_4 & i_4 - i_{10} & 0 \\ i_6 - i_2 & i_3 - i_6 & i_2 - i_3 & 0 & 0 \\ i_9 - i_6 & i_6 - i_7 & 0 & 0 & i_7 - i_9 \\ i_8 - i_9 & 0 & 0 & 0 & i_9 - i_{11} \end{bmatrix}. \tag{2.25}$$

Their determinants are

$$\begin{aligned} \det(A) &= (i_8 - i_2)[-(i_1 - i_4)(i_3 - i_6)(i_4 - i_5) \\ &+ (i_4 - i_{10})((i_3 - i_6)(i_3 - i_4) - (i_2 - i_3)(i_5 - i_3))] \end{aligned} \tag{2.26}$$

$$\begin{aligned}\det(B) = (i_4 - i_{10})[&-(i_8 - i_2)(i_9 - i_{11})(i_2 - i_3)(i_6 - i_7) \\ &+ (i_2 - i_1)((i_6 - i_2)(i_6 - i_7)(i_9 - i_{11}) \\ &- (i_3 - i_6)((i_9 - i_6)(i_9 - i_{11}) - (i_7 - i_9)(i_8 - i_9)))]\end{aligned} \tag{2.27}$$

Then in a manner similar to the proof of Lemma 3, we can show that the determinant of the corresponding matrix evaluates to zero only for finitely many choices for p for either selection of $g_r(i)$. If the determinant is equal to zero, we can prove the existence by constructing a valid solution, as stated in the proof of Theorem 1. In particular for the EAB codes, it suffices for p to be > 29 and $p \neq 41$ for the configuration not to exist.

For $p < 29$, and with the RSF equal to $[0, 1, 2, 3, 4]$, both $\det(A)$ and $\det(B)$ equal $0 \mod p$, which means this configuration exists for any prime number $p < 29$ with the RSF equal to $[0, 1, 2, 3, 4]$. We also identify RSFs such as $[0, 1, 2, 3, 6]$ where both determinants are zero for an infinite number of p values, and thus this configuration always exists for any prime number p with the RSF $[0, 1, 2, 3, 6]$. ■

2.3.4 Summary for $(4, 8)$, $(5, 9)$ or $(6, 8)$ absorbing sets

We have now considered all possible $(6, 8)$ configurations. The following theorem is a consequence of previously established results in Corollary 2, Lemma 1 through Lemma 6, and Corollary 3 through Corollary 15.

Theorem 2 *In $r = 5$ EAB codes the number of $(4, 8)$, $(5, 9)$ and $(6, 8)$ absorbing sets scales as $\Theta(p^3)$ whereas in SR-SCB codes with a properly selected RSF such as the examples given below there are no $(4, 8)$, $(5, 9)$ or $(6, 8)$ absorbing sets for large enough p.*

Table 2.1 presents some examples of RSFs that produce $r = 5$ SR-SCB codes for which $(4, 8)$, $(5, 9)$ and $(6, 8)$ absorbing sets do not exist. Additionally, $[0, 1, 2, 4, 7]$ for $p > 179$ and $[0, 1, 3, 4, 5]$ for $p > 271$ are generally good to eliminate $(4, 8)$, $(5, 9)$ and $(6, 8)$ absorbing sets.

Remark 3 *For sufficiently small p's, $(4, 8)$, $(5, 9)$ and $(6, 8)$ absorbing sets cannot be eliminated*

Table 2.1: Row selection functions for SR-SCB codes with $r = 5$ and a specified p ($p > 61$) that avoid all $(4, 8)$, $(5, 9)$, and $(6, 8)$ absorbing sets.

p	Rate $\frac{k}{n} = 1 - \frac{rp-r+1}{p^2}$	RSF
67	0.9263	$[0, 1, 2, 4, 17]$
73	0.9323	$[0, 1, 2, 3, 11]$
79	0.9373	$[0, 1, 2, 6, 7]$
83	0.9403	$[0, 1, 2, 3, 7]$
89	0.9443	$[0, 1, 2, 4, 11]$

simultaneously for the SR-SCB codes. Here are a few examples that remove most of these absorbing sets for $p < 50$: (1) for $p = 47$ with RSF= $[0, 1, 2, 3, 5]$, only the absorbing sets in Fig. 2.3 and Fig. 2.6 exist; (2) for $p = 43$ with RSF=$[0, 1, 2, 4, 6]$, only the absorbing set in Fig. 2.3 exists. ■

2.3.5 Absorbing set spectrum in Tanner construction

We now apply our analysis to the Tanner construction in [TSS04], whose codes are in the family of SCB codes. The Tanner-construction codes have the parity-check matrix $H^{r,c}_{p,a^i \cdot b^j}$ with $a, b \in GF(p)$ and $o(a) = r$ and $o(b) = c$, where $o(a)$ indicates the multiplicative order of a in $GF(p)$. We refer to this class of codes as Tanner-construction codes throughout the chapter.

Lemma 7 *In the Tanner graph corresponding to quasi-cyclic LDPC $H^{5,p}_{p,a^i \cdot b^j}$ in [TSS04], no $(4, 8)$ or $(6, 8)$ absorbing sets exist with parameters selected in Table I of [TSS04].*

Proof: For the codes in Table I of [TSS04] with girth greater than 6, neither $(4, 8)$ nor $(6, 8)$ absorbing sets exist since they contain length-6 cycles. For the codes in Table I of [TSS04] with girth $= 6$, we have analyzed each code using the CCM approach and confirmed that in each case neither $(4, 8)$ nor $(6, 8)$ absorbing sets exist. This analysis is too lengthy to include in this

chapter, but we provide the following example.

Consider the girth $= 6$ code with $p = 31, c = 6, r = 5$ as an example, with $a = 2, b = 6$. The matrix $H^{5,p}_{p,f(i,j)}$ here is a sub-matrix of the SR-SCB code with RSF $[1, 2, 4, 8, 16]$. Thus $f(i, j) = a^i b^j$ is achieved because the RSF enforces a^i and the submatrix of the SR-SCB code selects the columns to enforce b^j. We set up a system of equations as before, and conclude that for $(4, 8)$ absorbing sets, the only possible labeling for $p = 31$ is $(i_1, i_2, i_3, i_4, i_5, i_6) = (x, t, w, y, z, z)$. There are five non-isomorphic solutions to (2.11): $(x, y, z, w, t) = (2, 4, 1, 16, 8)$, $(1, 8, 2, 4, 16)$, $(1, 2, 4, 16, 8)$, $(1, 4, 8, 2, 16)$, $(1, 2, 16, 8, 4)$.

Each of the solutions corresponds to the CCM equation:

$$\mathbf{Mu} = \begin{bmatrix} x-t & t-z & 0 \\ x-z & 0 & z-y \\ 0 & z-w & w-y \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = 0. \tag{2.28}$$

Since $\det \mathbf{M} = 0$ and $\text{rank}(\mathbf{M}) = 2$, equation (2.28) is equivalent to

$$\begin{bmatrix} x-t & t-z & 0 \\ x-z & 0 & z-y \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = 0. \tag{2.29}$$

If we expand the u's with $u_i = j_i - j_1$, equation (2.29) is equivalent to

$$\begin{bmatrix} z-x & x-t & t-z & 0 \\ y-x & x-z & 0 & z-y \end{bmatrix} \begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \end{bmatrix} = 0. \tag{2.30}$$

We denote the matrix in equation (2.30) as $\mathbf{R}$.

Suppose the null space of each realization of the matrix $\mathbf{R}$ (as a function of the (x, y, z, w, t) values) is $N_i, 1 \leq i \leq 5$. For any $(4, 8)$ absorbing set, (j_1, j_2, j_3, j_4) should be in $\bigcup_{1 \leq i \leq 5} N_i$. Because of the Tanner construction, the column groups for this code are the powers of 6 mod 31.

Denote this set by $Y = \{1, 6, 5, 30, 25, 26\}$ the index of the selected column groups in the quasi-cyclic code. The 4-tuple (j_1, j_2, j_3, j_4) should also be in $\underset{i,j,k,l}{\cup}(Y(i), Y(j), Y(k), Y(l))$ where i, j, k, l are distinct integers between 1 and 6. However, in this case,

$$\{\underset{i,j,k,l}{\cup}(Y(i), Y(j), Y(k), Y(l))\} \cap \{\underset{1\leq i\leq 5}{\cup} N_i\} = \emptyset.$$

Thus $(4, 8)$ absorbing sets do not exist and consequently $(6, 8)$ absorbing sets in Figs. 2.5, 2.7 and 2.8 also do not exist.

Similarly, the null space of matrices considered in equations (2.16) and (2.23) in Lemma 3 and 6 does not have a non-trivial intersection with $\underset{i,j,k,l}{\cup}(Y(i), Y(j), Y(k), Y(l))$, which eliminates the possibility of existence of $(6, 8)$ absorbing sets in Fig. 2.6 and 2.9. ■

Codes listed in Table I of [TSS04] are mostly moderate-rate codes (the rate being around 0.5). The construction of codes with higher rates from this code family requires more columns to be selected from the SCB mother matrix. However, when more columns are selected, the Tanner construction cannot guarantee the absence of certain absorbing sets.

Lemma 8 *In the Tanner graph corresponding to quasi-cyclic LDPC with $H^{5,p}_{p,a^i\cdot b^j}$ in [TSS04], for higher rates in this class, $(4, 8)$, $(5, 9)$ and $(6, 8)$ absorbing sets can exist.*

Proof: For higher rates in this class, for instance with parameters $p = 31$, $c = 10$, $r = 5$, $a = 2$, $b = 15$ (with Y=1,15,8,27,2,30,16,23,4,29), $(4, 8)$ and $(6, 8)$ absorbing sets exist due to $\underset{i,j,k,l}{\cup}(Y(i), Y(j), Y(k), Y(l)) \cap \underset{1\leq i\leq 5}{\cup} N_i$ being non-empty, and thus equation (2.16) and (2.23) have nonzero solutions with Y. ■

With the same row selection, the parity check matrix of the SR-SCB code has the parity check matrix of the Tanner construction as a submatrix. According to the conclusions of Theorem 2, we can modify Tanner construction by using an optimized row selection to avoid all the $(4, 8)$, $(5, 9)$ and $(6, 8)$ absorbing sets for large p.

Lemma 9 *In the Tanner graph corresponding to quasi-cyclic LDPC codes (in [TSS04]) described by $H^{5,p}_{p,a^i\cdot b^j}$ with higher rates and large enough p, all of $(4, 8)$, $(5, 9)$ and $(6, 8)$ absorbing sets can be eliminated by only modifying the mapping sequence of the rows.*

Proof: By replacing the row mapping sequence $g_r(i) = \alpha^{i-1}$ by $g_r(i) = [0, 1, 2, 4, 7]$, the quasi-cyclic LDPC $H^{5,p}_{p,\hat{f}(i,j)}$ becomes a new class of SCB LDPC codes with $\hat{f}(i,j) = g_r(i) \cdot b^j$. Since $H^{5,p}_{p,\hat{f}(i,j)}$ is a submatrix of the mother matrix $H^{p,p}_{p,i\cdot j}$, it follows that $(4,8)$,$(5,9)$ and $(6,8)$ absorbing sets can be structurally eliminated for p large enough using the exactly the same analysis as presented in Sections 2.3.1 through 2.3.3. ■

Remark 4 *We already showed that one can easily check the existence of certain absorbing sets in Tanner construction [TSS04] (Lemmas 8 and 9). Other popular quasi-cyclic LDPC codes, such as codes in [Fos04] and [Fan00] can also be viewed as being constructed based on a selection of certain rows and columns of the SCB matrix. Codes in the Section III-B2/B3 of [Fos04] use the SCB structure with parity-check matrix* $H^{r,c}_{p,f(i,j)}$, *where* $f(i,j) = i \cdot j$, $0 \leq i \leq r-1$, $0 \leq j \leq c-1$ *or* $f(i,j) = (a^i - 1)(b^j - 1)$, $0 \leq i \leq r-1$, $0 \leq j \leq c-1$. *Array codes [Fan00] use the first* r *row groups of* $H^{p,p}_{p,i\cdot j}$. *The approach developed for the SCB construction can therefore be easily applied to codes in [Fos04] and [Fan00].*

2.3.6 Equivalence Classes for SR codes

The above analysis leads to some helpful code equivalence conditions.

Since the order of the elements in the RSF only permutes the matrix $H^{r,c}_{p,g_r(i)\cdot j}$ and does not change the code properties, we can assume that the RSF vector $[a_1, a_2, ..., a_r]$ is ordered in ascending order.

Consider a *difference matrix* D of the RSF, where $D_{ij} \equiv a_j - a_i \mod p, 1 \leq i, j \leq r$. If $\tilde{D} = D$ or $\tilde{D}$ is the reflection of D on its antidiagonal, we say that $\tilde{D}$ and D are equivalent difference matrices. The following lemma establishes some absorbing-set equivalence classes.

Lemma 10 *The following pairs of RSFs are equivalent in the sense that they have the same absorbing sets:*

1. $[a_1, a_2, ..., a_r] \equiv [a_1, a_2, ..., a_r] + \textit{constant} \mod p$

2. $[a_1, a_2, ..., a_r] \equiv [a_1, a_2, ..., a_r] \times$ *constant*$\neq 0 \mod p$

3. $[a_1, a_2, ..., a_r] \equiv [\tilde{a}_1, \tilde{a}_2, ..., \tilde{a}_r]$ *if they have equivalent difference matrices.*

Proof: With results proven in Lemma 1 through Lemma 6, and in Corollary 2 through Corollary 15, the existence of any particular absorbing set hinges on on whether the determinant of the associated CCM is zero. Since the determinant is only a function of the differences between the elements of the RSF, if two RSFs $[a_1, a_2, ..., a_r]$ and $[\tilde{a}_1, \tilde{a}_2, ..., \tilde{a}_r]$ share any of the three equivalent conditions, $[a_1, a_2, ..., a_r]$ has a zero determinant if and only if $[\tilde{a}_1, \tilde{a}_2, ..., \tilde{a}_r]$ has a zero determinant. ■

Remark 5 *Since the null space of CCM also only depends on the difference of the column-group indices, analogous equivalence conditions can be established column-wise.* ■

The following is a consequence of Lemma 10.

Corollary 16 *Any RSF is equivalent to an RSF* $[0, 1, x, y, z]$*, where* x, y, z *are positive integers.*

Proof: With condition (1) in Lemma 10, a mapping vector is equivalent to any mapping vector that differs by a constant in $GF(p)$. Thus we can subtract the smallest value in the mapping vector and obtain a 0 in the mapping vector. Then, with condition (2) in Lemma 10 and the multiplicative property of $GF(p)$, we can multiply the mapping vector by some value in $GF(p)$ to make one of the non-zero elements equal to 1. Therefore any mapping vector is equivalent to a mapping vector that contains 0 and 1. ■

This result enables a reduced search of structured matrices with good error-floor properties. For example, a row-mapping vector $[0, 1, 2, 4, 6]$ is equivalent to $[1, 2, 3, 5, 7]$, and to $[0, 2, 4, 8, 12]$.

2.4 Theoretical results for $r = 4$

This section provides an example with $r = 4$ (four row groups) that shows how to design an SCB code with a specified circulant matrix that eliminates the dominant absorbing sets by selecting

rows and columns from the SCB mother matrix to force the CCMs associated with the dominant absorbing sets to be full column-rank.

In addition to the EAB codes and SR-SCB codes introduced in Section 2.3, removing a few column groups from an SR-SCB code provides further improvement. Hence, shortened SR (SSR) SCB codes form our third class of SCB codes. The parity-check matrix for these codes is $H_{p,f}^{r,c}$, with $f(i,j) = g_r(i) \cdot g_c(j)$ where $g_c(j)$ is called the column-selection function (CSF). Note that for a $p \times p$ circulant matrix, EAB and SR-SCB codes have p column groups (p^2 binary columns), but SSR-SCB codes have fewer column groups since $g_c(j)$ selects a subset of the possible column groups.

Section 2.4.1 identifies the $(6,4)$ absorbing sets as dominant for EAB codes with $r = 4$. Section 2.4.2 analyzes the three possible $(6,4)$ absorbing set configurations and shows how carefully selecting four row groups from the SCB mother matrix can eliminate two of the three possible configurations. Section 2.4.3 provides an efficient provable algorithm to eliminate all $(6,4)$ absorbing sets by combining the row selection of Section 2.4.2 with column selection in which some column groups of the SCM mother matrix are removed.

2.4.1 Identifying the dominant absorbing sets

From the previous results in [DZW10], $(6,4)$ absorbing sets are the smallest possible structure for EAB codes with $r = 4$ for $p > 19$. Hardware simulations [ZDN09] also demonstrate that $(6,4)$ absorbing sets are the dominant cause of the error floor for example $r = 4$ EAB codes.

Based on these results, a key goal will be to design an $r = 4$ SCB code that avoids all $(6,4)$ absorbing sets. The lemma below establishes that the new code design approach does not introduce other smaller absorbing sets that were avoided by the codes discussed above.

Lemma 11 *In the Tanner graph corresponding to $H_{p,g_r(i)\cdot j}^{4,p}$, there is no absorbing set smaller than $(6,4)$ for p large enough with a careful choice of the row-selection function $g_r(i)$.*

Proof: The smallest possible absorbing sets for an $H_{p,g_r(i)\cdot j}^{4,p}$ SCB code are $(4,4)$, $(5,2)$, $(5,4)$

and $(6,2)$ absorbing sets (cf. [DZW10]). The $(4,4)$, $(5,4)$ and $(6,2)$ absorbing sets for $r=4$ are sub-graphs of the $(4,8)$, $(5,9)$ and $(6,8)$ absorbing sets, respectively, for $r=5$. From the analysis of absorbing sets for $r=5$ in the previous section (see also [DWZ10]), a careful choice of the $r=5$ row-selection function (RSF) $\tilde{a}(i)$ eliminates the $(4,8)$, $(5,9)$ and $(6,8)$ absorbing sets for p large enough. Taking any 4-element subset $g_r(i)$ of such an $\tilde{a}(i)$, for example the RSF $g_r(i)$ where $[g_r(0), g_r(1), g_r(2), g_r(3)] = [0,1,3,4]$, as a subset of $[0,1,3,4,5]$ that can eliminate the $(4,8)$, $(5,9)$ and $(6,8)$ absorbing sets for $p > 271$, avoids the $(4,4)$, $(5,4)$ and $(6,2)$ absorbing sets for $r=4$. Any RSF that avoids the $(4,4)$ absorbing set also avoids the $(5,2)$ absorbing set [DZW10]. Thus the resulting $H^{4,p}_{p,g_r(i)\cdot j}$ avoids $(4,4)$, $(5,2)$, $(5,4)$ and $(6,2)$ absorbing sets. ■

Remark 6 *Since the SSR code is a shortened version of the SR code, obtained by removing certain variable nodes, no smaller absorbing set will be introduced in the SSR code.* ■

In Section 2.4.2, we show that SR codes always have $(6,4)$ absorbing sets, irrespective of the choice of $g_r(i)$. Avoidance of all such configurations using shortening is the subject of Section 2.4.3.

2.4.2 $(6,4)$ absorbing sets in SR-SCB codes

Three distinct configurations of $(6,4)$ absorbing sets are possible for $r=4$ SCB codes. This section shows which of these configurations are possible in EAB and SR-SCB codes. The first configuration exists in the EAB code and in every possible SR-SCB code. The second configuration exists in the EAB code but can be avoided by a proper choice of the RSF for the SR-SCB code. The third configuration does not exist in either the EAB code or the SR codes.

2.4.2.1 The first (6,4) configuration

Fig. 2.10 shows the first configuration of a $(6,4)$ absorbing set in an $r=4$ SCB code. The following lemma establishes that the EAB code and all SR codes have this configuration.

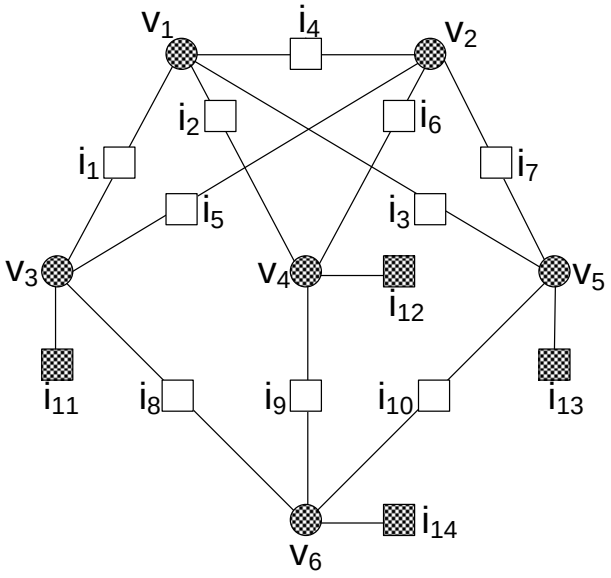


Figure 2.10: Depiction of the first (6, 4) absorbing set configuration.

Lemma 12 *In the Tanner graph corresponding to the EAB code and all SR codes with* $H^{4,p}_{p,f(i,j)}$ *there are* $(6, 4)$ *absorbing sets for any* p *with the configuration shown in Fig. 2.10.*

Proof: Using the technique of Section 2.2.2 we construct the CCM for this configuration. The binary cycle space for Fig. 2.10 has dimension 5. We construct the following CCM by selecting the following linearly independent cycles: $v_1 - v_2 - v_3, v_1 - v_2 - v_4, v_1 - v_2 - v_5, v_1 - v_3 - v_6 - v_4, v_1 - v_5 - v_6 - v_4$,

$$\mathbf{M} = \begin{bmatrix} i_4 - i_5 & i_5 - i_1 & 0 & 0 & 0 \\ i_4 - i_6 & 0 & i_6 - i_2 & 0 & 0 \\ i_4 - i_7 & 0 & 0 & i_7 - i_3 & 0 \\ 0 & i_1 - i_8 & i_9 - i_2 & 0 & i_8 - i_9 \\ 0 & 0 & i_2 - i_9 & i_{10} - i_3 & i_9 - i_{10} \end{bmatrix}. \tag{2.31}$$

To avoid this absorbing set we need to force the determinant of the CCM to be nonzero. Note that $\det(\mathbf{M})$ is computed as

$$\begin{aligned} &\mathbf{M}_{11}\mathbf{M}_{23}\mathbf{M}_{34}\mathbf{M}_{42}\mathbf{M}_{55} \\ &- \mathbf{M}_{12}\big(\mathbf{M}_{23}\mathbf{M}_{31}\mathbf{M}_{45}\mathbf{M}_{54} - \mathbf{M}_{21}\mathbf{M}_{34}(\mathbf{M}_{43}\mathbf{M}_{55} - \mathbf{M}_{45}\mathbf{M}_{53})\big), \end{aligned}$$

where $\mathbf{M}_{ij}$ denotes the (i, j) entry in $\mathbf{M}$.

From the proof of Lemma 8 in [DZW10], there are only two non-isomorphic row-group labelings for the check nodes of Fig. 2.10. It is sufficient to consider only $(i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10})$ $= (x, y, z, w, y, z, x, z, x, y)$ or $(x, y, z, w, y, z, x, z, w, y)$ with x, y, z, and w taking distinct values in the range of RSF. The first labeling yields

$$\mathbf{M} = \begin{bmatrix} w-y & y-x & 0 & 0 & 0 \\ w-z & 0 & z-y & 0 & 0 \\ w-x & 0 & 0 & x-z & 0 \\ 0 & x-z & x-y & 0 & z-x \\ 0 & 0 & y-x & y-z & x-y \end{bmatrix}, \tag{2.32}$$

and $\det(\mathbf{M}) = 0$ regardless of the specific values of w, x, y, z taken as mutually distinct integers $\bmod p$.

Thus there exists a non-zero solution to $\mathbf{M} \cdot \mathbf{u} \equiv 0 \mod p$. One solution to this equation is

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} \equiv \begin{bmatrix} (x-y)(z-y)(x-z) \\ (w-y)(z-y)(x-z) \\ (w-z)(y-x)(x-z) \\ (w-x)(y-x)(z-y) \\ (w-y)(x-z)(z-y) + (y-x)(w-z)(x-y) \end{bmatrix}. \tag{2.33}$$

For this absorbing set, the Check Consistency condition requires $u_2 \neq 0$, $u_3 \neq 0$, $u_4 \neq 0$, $u_5 \neq 0$, $u_2 \neq u_3$, $u_2 \neq u_4$, $u_2 \neq u_5$, $u_3 \neq u_6$, $u_4 \neq u_6$, and $u_5 \neq u_6$. These requirements as well as the Bit Consistency inequalities are met since x, y, z, w are mutually distinct.

The solution in (2.33) satisfies the Bit, Check and Cycle Consistency constraints. This is a sufficient condition for the existence of a $(6, 4)$ absorbing sets with the configuration of Fig. 2.10. Any four distinct values between 0 and $p - 1$ for $\{x, y, z, w\}$ identify a labeling of this first type that identifies an absorbing set in the EAB code and every SR SCB code.

Consider $(i_1, i_2, \ldots, i_{10}) = (x, y, z, w, y, z, x, z, w, y)$, the second labeling. In this case $\det(\mathbf{M}) \neq 0$ for $\{x, y, z, w\} = \{0, 1, 2, 3\}$. Thus $\det(\mathbf{M}) \not\equiv 0 \mod p$ for p large enough,

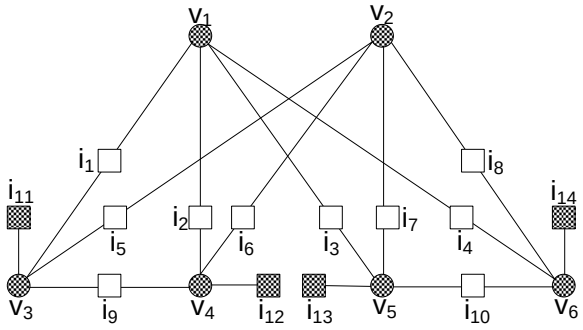


Figure 2.11: Depiction of the second candidate (6, 4) absorbing set.

and there is no such $(6, 4)$ configuration with this labeling in the EAB code for p large enough. The EAB code is one possible SR code. Other careful choices of the RSF $g_r(i)$, produce other SR codes that likewise do not have this $(6, 4)$ configuration. ■

2.4.2.2 The second (6,4) configuration

Fig. 2.11 shows the second possible configuration of a $(6, 4)$ absorbing set in an $r = 4$ SCB code. The following lemma establishes that the EAB code has this configuration but well-designed SR codes avoid it.

Lemma 13 *In the Tanner graph corresponding to $H^{4,p}_{p,f(i,j)}$, $(6, 4)$ absorbing sets of the type shown in Fig. 2.11 exist in the EAB code, but do not exist in certain SR codes for p large enough.*

Proof: Again using the technique of Section 2.2.2 we construct the CCM for this configuration. The binary cycle space for Fig. 2.10 has dimension 5. We construct the following CCM by selecting the following linearly independent cycles: $v_1 - v_3 - v_4, v_1 - v_5 - v_6, v_2 - v_3 - v_4, v_2 - v_5 - v_6, v_1 - v_4 - v_2 - v_5$,

$$\mathbf{M} = \begin{bmatrix} 0 & i_1 - i_9 & i_9 - i_2 & 0 & 0 \\ 0 & 0 & 0 & i_3 - i_{10} & i_{10} - i_4 \\ i_5 - i_6 & i_9 - i_5 & i_6 - i_9 & 0 & 0 \\ i_7 - i_8 & 0 & 0 & i_{10} - i_7 & i_8 - i_{10} \\ i_6 - i_7 & 0 & i_2 - i_6 & i_7 - i_3 & 0 \end{bmatrix}. \tag{2.34}$$

Note that the determinant is computed as follows:

$$\begin{aligned}\det(\mathbf{M}) = &- \mathbf{M}_{12}\mathbf{M}_{25}\mathbf{M}_{33}\mathbf{M}_{41}\mathbf{M}_{54} + \mathbf{M}_{13}\mathbf{M}_{25}\mathbf{M}_{32}\mathbf{M}_{41}\mathbf{M}_{54} \\ &+ \mathbf{M}_{12}\mathbf{M}_{24}\mathbf{M}_{31}\mathbf{M}_{45}\mathbf{M}_{53} - \mathbf{M}_{12}\mathbf{M}_{25}\mathbf{M}_{31}\mathbf{M}_{44}\mathbf{M}_{53} \\ &- \mathbf{M}_{12}\mathbf{M}_{24}\mathbf{M}_{33}\mathbf{M}_{45}\mathbf{M}_{51} + \mathbf{M}_{13}\mathbf{M}_{24}\mathbf{M}_{32}\mathbf{M}_{45}\mathbf{M}_{51} \\ &+ \mathbf{M}_{12}\mathbf{M}_{25}\mathbf{M}_{33}\mathbf{M}_{44}\mathbf{M}_{51} - \mathbf{M}_{13}\mathbf{M}_{25}\mathbf{M}_{32}\mathbf{M}_{44}\mathbf{M}_{51}.\end{aligned}$$

As with the previous lemma, the proof of Lemma 8 in [DZW10] identifies exactly two non-isomorphic labelings for $(i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10})$ in Fig. 2.11. These are $(x, y, z, w, y, x, w, z, z, y)$ and $(x, y, z, w, y, z, w, x, w, y)$ for $\{x, y, z, w\} = \{0, 1, 2, 3\}$.

For the first labeling $\det(\mathbf{M})$ is zero for distinct x, y, z, and w. Thus the configuration in Fig. 2.11 with this labeling exists in the EAB code. For the second labeling, $\det(\mathbf{M})$ is not zero for distinct x, y, z, and w. Therefore the configuration with this labeling does not exist in the EAB codes for p large enough.

For both labelings, SR codes can avoid the configuration in Fig. 2.11 with a careful choice of RSF. One such example is the RSF $[0, 1, 3, 4]$. Since the largest prime factor of $\det(\mathbf{M})$ is 31, the mapping vector avoids this configuration for $p > 31$. ■

2.4.2.3 The third (6,4) configuration

Fig. 2.12 shows the third configuration of a $(6, 4)$ absorbing set in an $r = 4$ SCB code. The following lemma establishes that neither the EAB nor well-designed SR codes have this configuration.

Lemma 14 *In the Tanner graph corresponding to $H^{4,p}_{p,f(i,j)}$ there are no $(6, 4)$ absorbing sets for p large enough with the configuration shown in Fig. 2.12 in either the EAB or in the SR code.*

Proof: Similar to the proof of Lemmas 12 and 13, the minimum number of independent cycles needed in Fig. 2.12 is 5 and the CCM can be constructed using the cycles $v_1 - v_3 - v_4$,

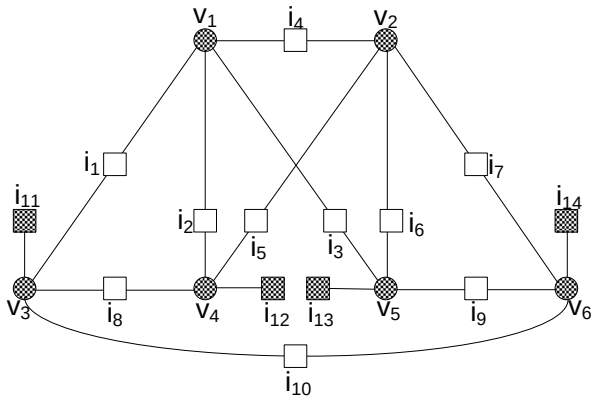


Figure 2.12: Depiction of the third candidate (6, 4) absorbing set.

$v_1 - v_2 - v_4$, $v_1 - v_2 - v_5$, $v_2 - v_5 - v_6$, $v_1 - v_3 - v_6 - v_5$:

$$\mathbf{M} = \begin{bmatrix} 0 & i_1 - i_8 & i_8 - i_2 & 0 & 0 \\ i_4 - i_5 & 0 & i_5 - i_2 & 0 & 0 \\ i_4 - i_6 & 0 & 0 & i_6 - i_3 & 0 \\ i_6 - i_7 & 0 & 0 & i_9 - i_6 & i_7 - i_9 \\ 0 & i_1 - i_{10} & 0 & i_9 - i_3 & i_{10} - i_9 \end{bmatrix}. \tag{2.35}$$

where

$$\begin{aligned} \det(\mathbf{M}) =& \mathbf{M}_{12}\mathbf{M}_{23}(\mathbf{M}_{31}(\mathbf{M}_{44}\mathbf{M}_{55} - \mathbf{M}_{45}\mathbf{M}_{54}) - \mathbf{M}_{34}\mathbf{M}_{41}\mathbf{M}_{55}) \\ &+ \mathbf{M}_{13}\mathbf{M}_{21}\mathbf{M}_{34}\mathbf{M}_{45}\mathbf{M}_{52}. \end{aligned} \tag{2.36}$$

With the proof of Lemma 8 in [DZW10], there are seven non-isomorphic labelings for $(i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10})$. Fixing $(i_1, i_2, i_3, i_4) = (x, y, z, w)$, we need only consider $(i_5, i_6, \ldots, i_{10}) = (x, y, z, z, x, y)$, (x, y, z, z, x, w), (x, y, z, z, w, y), (x, y, z, w, w, y), (z, x, y, w, w, z), (z, y, x, w, w, y), and (z, y, x, w, w, z) .

For these seven labelings $\det(\mathbf{M})$ is not zero for $\{x, y, z, w\} \in \{0, 1, 2, 3\}$ for p large enough. Therefore the configuration in Fig. 2.12 with this labeling does not exist in the EAB codes for p large enough. With a numerical evaluation of the determinant, $p > 13$ is sufficient for the absence of this configuration in the EAB codes. Moreover, for all of the seven labelings, SR-SCB codes can avoid the configuration in Fig. 2.12 with careful choice of the row mapping. One

such example is the EAB RSF. Another is to choose the RSF $[0, 1, 3, 4]$, which can avoid this configuration for $p > 19$. ■

2.4.3 Eliminating $(6, 4)$ absorbing sets with shortening

For a sufficiently large p, well-designed SR codes avoid the $(6, 4)$ absorbing set configurations in Fig. 2.11 and 2.12. However, as shown in Lemma 12, SR codes cannot eliminate the $(6, 4)$ absorbing set configuration in Fig. 2.10. We now consider shortened SR (SSR) codes that retain only certain column groups from the SCB mother matrix (reducing the rate). A well-chosen column selection $g_c(j)$ allows the Tanner graph corresponding to $H^{4,c}_{p,g_r(i)\cdot g_c(j)}$ to avoid all $(6, 4)$ absorbing sets.

We begin with an SR code using well-selected RSF $g_r(i)$, for instance $[0, 1, 3, 4]$, that already avoids the $(6, 4)$ absorbing set configurations in Figs. 2.11 and 2.12 for p large enough. We then choose a CSF $g_c(j)$ to also avoid the $(6, 4)$ absorbing set configurations in Fig. 2.10. Choosing a column selection $g_c(j)$ reduces to choosing a submatrix of $H^{4,p}_{p,g_r(i)\cdot j}$ by eliminating certain variable nodes. This operation cannot introduce smaller absorbing sets.

One solution to $\mathbf{M} \cdot u \equiv 0 \mod p$ is equation (2.33). The rank of $\mathbf{M}$ in (2.32) is 4, and therefore this single solution forms a basis of the null space. Multiplying u by a constant c, for $1 \leq c \leq p-1$, also results in a solution. These $p-1$ solutions cover all of the feasible solutions described by the null space. For fixed $u_1, ..., u_5$, we can choose j_1 from $0, 1, ..., p-1$ and obtain $j_2, ..., j_6$. Thus there are $p(p-1)$ ways to find j_1 to j_6 for a fixed $\{x, y, z, w\}$. Since there are $4!$ ways to assign check node labels based on the set $\{x, y, z, w\}$ for a fixed row mapping, there are at most $24p(p-1)$ possible vectors $[j_1, j_2..., j_6]$ that can form the configuration in Fig. 2.10. These vectors form the set $\tilde{V}$ of vectors, which completely characterizes the instances of this absorbing set configuration.

If a CSF is applied, each variable node group label j is in a set J where $J \subset \{0, 1, ..., p-1\}$ and we can only choose $[j_1, j_2..., j_6]$ such that $j_m \in J, m = 1, ..., 6$. There are $\binom{|J|}{6}$ possible $[j_1, j_2..., j_6]$ vectors and they form a set of vectors V. If $V \cap \tilde{V} = \emptyset$, the new code does not have

the $(6, 4)$ configuration in Fig. 2.10. We can find such a CSF with the greedy column-cutting procedure described in Algorithm 1 or the column-adding procedure described in Algorithm 2.

Algorithm 1 Greedy column-cutting algorithm.

1: % Initialization: C begins with all columns.
2: $C = \{0, 1, ..., p-1\}$.
3: Collect all the vectors $[j_1, j_2..., j_6]$, that form the configuration in Fig. 2.10 with $j_n \in C, n = 1, ..., 6$ and they form a set $W = \tilde{V}$.
4: % Proceed one column at a time removing columns that % preclude vectors in $\tilde{V}$ until no vector in $\tilde{V}$ is possible.
5: **while** $|W| > 0$ **do**
6: Find the most frequent j in W, say j_m.
7: Replace C by $C \setminus j_m$.
8: Remove every $[j_1, j_2..., j_6]$ that involves j_m from W.
9: **end while**
10: % C contains the column groups of the designed code.

Remark 7 *A similar technique could be applied to increase the girth [MKL06] instead of eliminating the smallest absorbing set. However, simply increasing the girth would not guarantee a better performance, see e.g. [NVM10].* ■

2.4.4 SSR codes with Tanner construction

This section shows that for $r = 4$, the row-selection function $g_r(i)$ of the Tanner construction [TSS04] will always introduce $(4, 4)$ absorbing sets for the case set forth in the following lemma. For a an element of $GF(p)$ let $o(a)$ denote its multiplicative order in $GF(p)$.

Lemma 15 *In the Tanner graph corresponding to the quasi-cyclic LDPC code with the parity-check matrix $H^{4,p-1}_{p,g_r(i)\cdot g_c(j)}$, where $g_r(i) = a^i, o(a) = 4, g_c(j) = b^j, o(b) = p-1$, $(4, 4)$ absorbing sets as shown in Fig. 2.13 always exist.*

Algorithm 2 Column-adding algorithm.

% Initialization: Select an initial six columns for C that % do not form a vector in $\tilde{V}$.

$J = \{0, 1, ..., p-1\}$.

Collect all the ordered vectors $[j_1, j_2..., j_6]$, that form the configuration in Fig. 2.10 with $j_n \in J, n = 1, ..., 6$, into the set $\tilde{V}$.

Choose a distinct 6-element set C, $C \subseteq J$.

while for some ordering $[\hat{j}_1, \hat{j}_2..., \hat{j}_6]$ of the elements of C, $[\hat{j}_1, \hat{j}_2..., \hat{j}_6] \in \tilde{V}$ **do**

    Choose another distinct 6-element set C randomly such that $C \subseteq J$.

end while

% Proceed one column group at a time, adding columns % to C that do not introduce the absorbing set.

$J = \{0, 1, ..., p-1\} \setminus C$

while $|J| > 0$ **do**

    Select a j_m randomly from J.

    if every $[\hat{j}_1, \hat{j}_2..., \hat{j}_6] \notin \tilde{V}$ for every $\hat{j}_1, \hat{j}_2..., \hat{j}_6 \in \{C \cup j_m\}$ **then**

        $C = C \cup j_m$

    end if

    $J = J \setminus j_m$

end while

% C contains the column groups of the designed code.

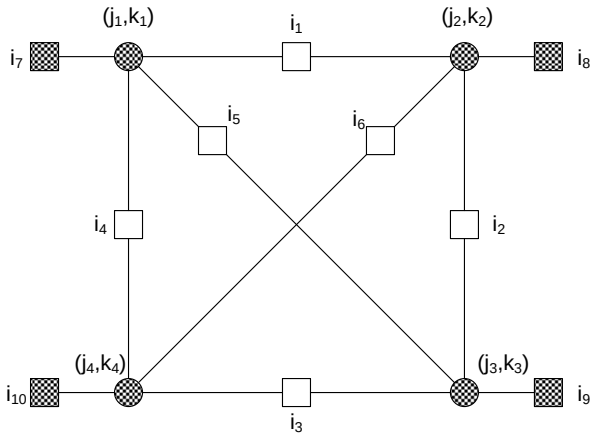


Figure 2.13: Depiction of the $(4, 4)$ absorbing set configuration.

Proof: With the analysis similar to that of the proof Corollary 2, it follows that there is only one possible non-isomorphic check labeling for this configuration that satisfies Bit Consistency and Check Consistency: $(i_1, i_2, i_3, i_4, i_5, i_6)$ is (x, y, x, y, z, w), where $\{x, y, z, w\} \subset (g_r(0), g_r(1), g_r(2), g_r(3))$. Therefore the necessary and sufficient condition for the existence of $(4, 4)$ absorbing sets can be also formulated as

$$\det \mathbf{M} = (z - x)(w - y) + (z - y)(w - x) = 0 \mod p. \tag{2.37}$$

Since $g_r(i) = \alpha^i, o(\alpha) = 4$, we can assign $x = 1, w = \alpha, y = \alpha^2, z = \alpha^3$. Then the determinant can be represented as

$$\begin{aligned} \det \mathbf{M} &= (\alpha^3 - 1)(\alpha - \alpha^2) + (\alpha^2 - \alpha^3)(1 - \alpha) \\ &= 2(1 - \alpha + \alpha^2 - \alpha^3) \\ &= 2(1 - \alpha)(1 + \alpha^2) \, . \end{aligned} \tag{2.38}$$

With the fact that $o(\alpha) = 4, \alpha^4 = 1$. Also, $o(\alpha^2) = 2$ so that the only possible value of α^2 in $GF(p)$ is $p - 1$. Thus $\det \mathbf{M} = 0 \mod p$.

According to the analysis in the proof of Corollary 2, $\det \mathbf{M} = 0 \mod p$ implies that there exists a non-zero $\mathbf{u} = [u_2, u_3, u_4]^T$, where $u_2 \neq 0, u_3 \neq 0, u_4 \neq 0$ are required to satisfy the bit-consistency conditions. Since $o(\beta) = p - 1$, the parity check matrix $H^{4,p-1}_{p,g_r(i)\cdot g_c(j)}$ includes

almost all the column groups except the one with index 0. Thus by carefully selecting a non-zero j_1 such that $j_1 \neq p - u_2, j_1 \neq p - u_3, j_1 \neq p - u_4$, we can always find $j_2 \neq 0, j_3 \neq 0, j_4 \neq 0$, and the resulting $[j_1, j_2, j_3, j_4]$ will produce the $(4, 4)$ absorbing sets for any specific k_1 value. ■

Remark 8 *Tanner constructions with $o(\beta) < p - 1$ and $r = 4$ sometimes can avoid all $(4, 4)$ absorbing sets by either increasing the girth or by making the intersection between null space of* $\mathbf{M}$ *and variable node space to be empty. However, $(4, 4)$ absorbing sets may still exist in some constructions with $o(\beta) < p - 1$ and $r = 4$. In these cases they dominate the error floor. For example, with $p = 67, g_r(i) = 11^i, g_c(j) = 5^j, o(11) = 4, o(5) = 30$, the resulting code has dominant $(4, 4)$ absorbing sets.*

2.5 Discussion for $r = 3$

For the EAB codes with $r = 3$, the $(3, 3)$ and $(4, 2)$ absorbing sets in Fig. 2.14 and 2.15 are the smallest ones [DZW10]. Since the $(3, 3)$ absorbing set is composed of a length-6 cycle, increasing the girth of the code is necessary to eliminate such absorbing sets. Thus the SR-SCB code cannot avoid the $(3, 3)$ absorbing sets. We can still eliminate these absorbing sets in SSR-SCB codes with the CCM approach to shorten the code such that the variable nodes cannot form a difference vector that falls into the null space of the CCM, which is equivalent to increasing the girth in this case. The resulting SSR-SCB codes have much lower rates and are not the interest of this chapter. Similarly, we can apply the CCM approach to the $(4, 2)$ absorbing set and easily prove the existence in any EAB and SR-SCB codes. The SSR-SCB codes can avoid the $(4, 2)$ absorbing set by shortening to low rates and are the interest of this chapter.

2.6 Results

In this section we experimentally demonstrate the performance improvement achieved by the proposed modifications. We show that it is *consistently valid across various choices of decoding*

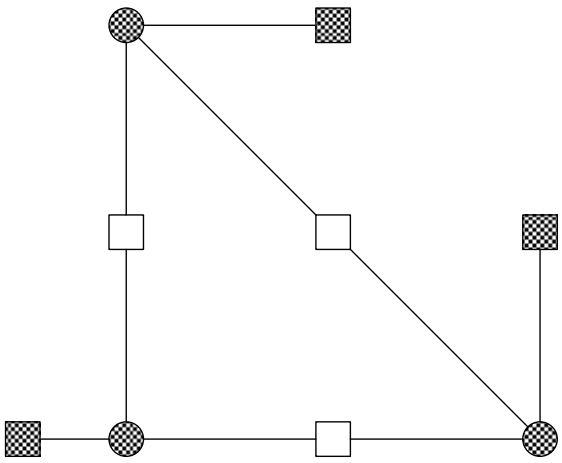


Figure 2.14: Depiction of the $(3, 3)$ absorbing set configuration.

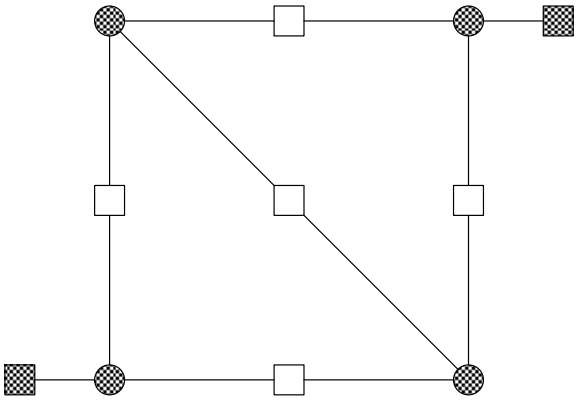


Figure 2.15: Depiction of the $(4, 2)$ absorbing set configuration.

Table 2.2: Hardware error profiles for three $(2209, 1978)$ codes, one EAB code and two SR-SCB codes. More input frames were emulated for the higher SNR level. The total number of collected errors is denoted n.e.

$p = 47$ $(2209, 1978)$ EAB code									
SNR	n.e.	(4,8)	(5,9)	(6,8)	(6,10)	(7,9)	(8,6)	(8,8)	(8,10)
5.6dB	272	97	27	64	10	16	17	11	5
5.8dB	211	121	20	34	9	3	3	4	3
$p = 47$ $(2209, 1978)$ SR-SCB-1 RSF = [0,1,3,8,19]									
SNR	n.e.	(4,8)	(5,9)	(6,8)	(6,10)	(7,9)	(8,6)	(8,8)	(8,10)
5.6dB	197	0	30	36	27	25	3	17	13
5.8dB	90	0	25	16	22	16	3	10	1
$p = 47$ $(2209, 1978)$ SR-SCB-2 RSF = [0,1,2,4,6]									
SNR	n.e.	(4,8)	(5,9)	(6,8)	(6,10)	(7,9)	(8,6)	(8,8)	(8,10)
5.6dB	179	0	30	28	29	22	5	9	10
5.8dB	116	0	25	16	22	16	3	10	1

implementations and platforms both in software and in hardware. In simulations, we use 200 iterations and a $Q4.2$ fixed-point quantization, 4 bits to the left of the radix point to represent integer values and 2 bits to the right of the radix point to represent fractional values. We simulate sum-product algorithm [ZDN09] and soft-xor algorithm [MS03] for different codes. We also identify the absorbing sets responsible for each error and report an error profile for each simulation. As expected, when an SR-SCB code or SSR-SCB code is designed to preclude certain absorbing sets, they do not appear in the error profile obtained by simulation.

2.6.1 SCB codes for $r = 5$

Consistent with the results presented at the end of Section 2.3.1, Table 2.2 and Fig. 2.16 show the performance of three $(2209, 1978)$ codes: the EAB code and two SR-SCB codes. All three

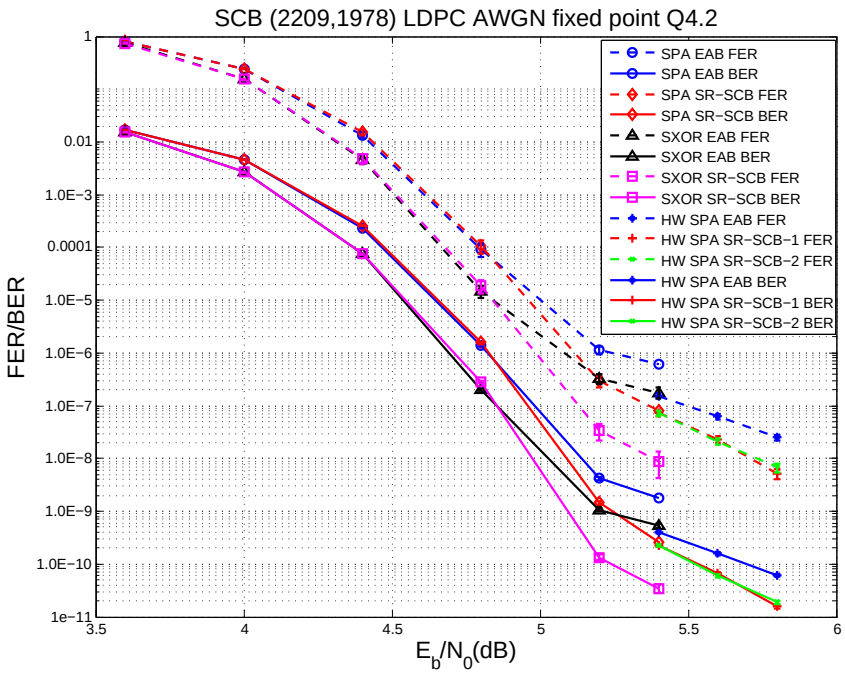


Figure 2.16: Performance comparison of the $(2209, 1978)$ EAB and SR-SCB LDPC codes.

codes are SCB codes with the following parameters: check node degree = 47, bit node degree = 5, and $f(i, j) = g_r(i) \cdot j$. The EAB code uses $g_r(i) = i$, SR-SCB-1 uses the RSF $[0, 1, 3, 8, 19]$, i.e. $(i, g_r(i)) \in \{(0, 0), (1, 1), (2, 3), (3, 8), (4, 19)\}$, and SR-SCB-2 uses the RSF $[0, 1, 2, 4, 6]$.

Regardless of the decoding algorithm (sum-product algorithm (SPA) or soft-XOR (SXOR)), $(4, 8)$ absorbing sets cause a significant fraction of decoding errors for the EAB code. With SR-SCB codes, there are no such errors (as must be the case according to the analysis of Section 2.3.1). Moreover, removing the $(4, 8)$ absorbing sets does not lead to an increase in higher-order absorbing sets. Rather, removing the $(4, 8)$ absorbing sets also decreases higher order absorbing sets that contain the $(4, 8)$ absorbing set or some of its cycles. Thus, the SR-SCB codes have fewer total errors than the EAB code.

We also implemented a sum-product decoder on an FPGA platform that works for all three codes since the basic array-code structure is preserved in the SR-SCB versions. A substantial simulation speedup via the FPGA platform allows us to extend the BER curve down to 10^{-11}, as shown in Fig. 2.16. Note that the SR-SCB codes improve the slope in the error floor region

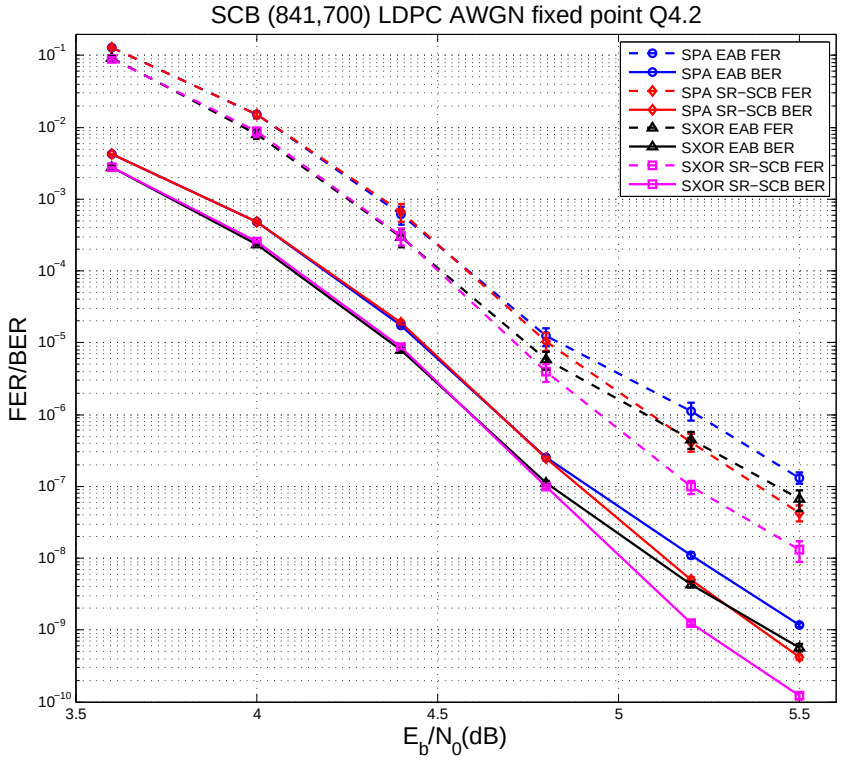


Figure 2.17: Performance comparison of the $(841, 700)$ EAB and SR-SCB LDPC codes.

significantly for both FER and BER. The BER improvement is more than one order of magnitude at 5.4 dB.

Table 2.2 gives the error profiles for these three codes for the two highest SNRs simulated on the FPGA. Consistent with the software-based simulation and the theoretical analysis, $(4, 8)$ absorbing sets dominate the error floor of the EAB array code. In both of the SR-SCB codes, having virtually the same performance (see Fig. 2.16), the $(4, 8)$ absorbing sets are completely eliminated, the BER curve is lowered, and its slope is improved. Another noticeable effect is the reduction of the number of $(6, 8)$ absorbing set errors after the code modification, which is attributed to the elimination of the $(4, 8)$ substructures, which are contained by some $(6, 8)$ structures.

Fig. 2.17 shows that elimination of the $(4, 8)$ absorbing set can provide a noticeable improvement even at short block lengths. This figure compares two $(841, 700)$ codes with $p = 29$, the

EAB code and the SR-SCB code with RSF $[0, 1, 2, 4, 6]$ that precludes all $(4, 8)$ absorbing sets.

Table 2.3 and Fig. 2.18 show the performance improvement over the EAB code of an SR-SCB code that eliminates of all the $(4, 8)$ and $(6, 8)$ absorbing sets with the RSF $[0, 1, 2, 4, 6]$. This comparison is for $(1849, 1638)$ codes with $p = 43$.

Table 2.4 and Fig. 2.19 show simulations comparing longer block-length EAB and SR-SCB codes. The performance improvement of the SR-SCB code is due to the structural elimination of the $(4, 8)$, $(5, 9)$ and $(6, 8)$ absorbing sets using the RSF $[0, 1, 2, 4, 17]$ which was identified in Table 2.1 of Section 2.3.3.6 as completely avoiding these three absorbing sets. As expected, the error profile as shown in Table 2.4 shows that the SR-SCB code completely eliminates the $(4, 8)$, $(5, 9)$ and $(6, 8)$ absorbing sets.

2.6.2 SCB codes for $r = 4$

Fig. 2.20 and Table 2.5 show the performance of $(2209, 2024)$ EAB code and SR-SCB codes both with check node degree = 47 and bit node degree = 4. The EAB code has $g_r(i) = i$ and the SR-SCB code uses RSF $[0, 1, 2, 4]$. Consistent with the theoretical analysis, the $(6, 4)$ absorbing sets dominate the error floor of the EAB code. The reduction of the $(6, 4)$ absorbing sets provides the SR performance improvement.

We also compare the performance of a high-rate quasi-cyclic (QC) code under Tanner construction [TSS04] with a similar-rate SSR code. The QC code has the following parameters: $p = 61$, $f(i, j) = a^i \cdot b^j$, $a = 11$, $b = 5$, $o(a) = 4$ and $o(b) = 30$, where $o(a)$ indicates the multiplicative order of a in $GF(p)$. Compared to the SSR code, this is a code with the same variable-node degree, a similar block length ($N = 1830$), and a similar rate (0.8683). Using the CCM based analysis, one can show that this code does not have $(6, 4)$ absorbing sets, although it does have $(4, 4)$ absorbing sets (due to an inappropriate row mapping). The SSR code has the following parameters: $p = 79$, RSF = $[0, 1, 3, 4]$, and $b(i) \in \{2, 6, 7, 14, 17, 18, 22, 26, 27, 30, 36, 37, 38, 46, 47, 49, 55, 56, 57, 58, 61, 62, 65, 66, 67, 76, 77, 78\}$. We thus obtain a code with the same variable-node degree, a similar block length ($N = 2212$) and a similar rate (0.8585).

Table 2.3: Hardware error profiles for the EAB $(1849, 1638)$,code (top), and the SR-SCB code (bottom).

$p = 43$ $(1849, 1638)$ EAB code										
SNR	n.e.	(4,8)	(5,9)	(6,8)	(6,10)	(7,9)	(7,11)	(8,6)	(8,8)	(8,10)
5.6dB	236	98	24	37	11	15	5	8	12	5
5.8dB	182	103	19	29	7	7	4	0	1	5
6.0dB	160	98	22	15	3	4	1	4	3	2

$p = 43$ $(1849, 1638)$ SR-SCB code RSF $= [0, 1, 2, 4, 6]$										
SNR	n.e.	(4,8)	(5,9)	(6,8)	(6,10)	(7,9)	(7,11)	(8,6)	(8,8)	(8,10)
5.6dB	126	0	23	0	23	17	8	5	8	10
5.8dB	92	0	27	0	15	13	6	6	6	6
6.0dB	48	0	19	0	14	4	3	3	1	1

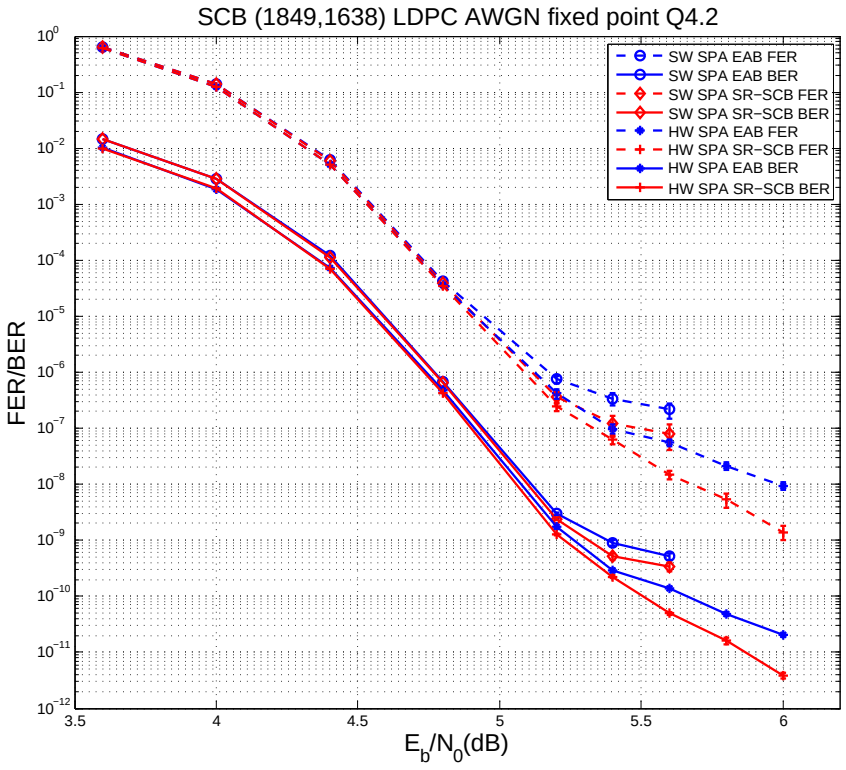


Figure 2.18: Performance comparison of the $(1849, 1638)$ EAB and SR-SCB LDPC codes.

Table 2.4: Error profiles for the EAB SCB $(4489, 4158)$, code (top), and the SR-SCB code (bottom), both with $p = 67$. n.e. is the number of collected errors.

$(4489, 4158)$ EAB code										
SNR	n.e.	(4,8)	(5,9)	(6,8)	(6,10)	(7,9)	(7,11)	(8,6)	(8,8)	(8,10)
5.6dB	150	67	17	22	7	6	5	6	6	3
5.8dB	139	83	18	16	6	5	1	3	3	1
6.0dB	131	77	18	22	5	1	1	2	1	1
6.2dB	107	85	10	5	4	2	0	0	0	0

$(4489, 4158)$ SR-SCB code RSF $= [0, 1, 2, 4, 17]$										
SNR	n.e.	(4,8)	(5,9)	(6,8)	(6,10)	(7,9)	(7,11)	(8,6)	(8,8)	(8,10)
5.6dB	106	0	0	0	25	15	6	15	13	6
5.8dB	140	0	0	0	35	29	14	16	6	8
6.0dB	60	0	0	0	25	7	5	9	5	3

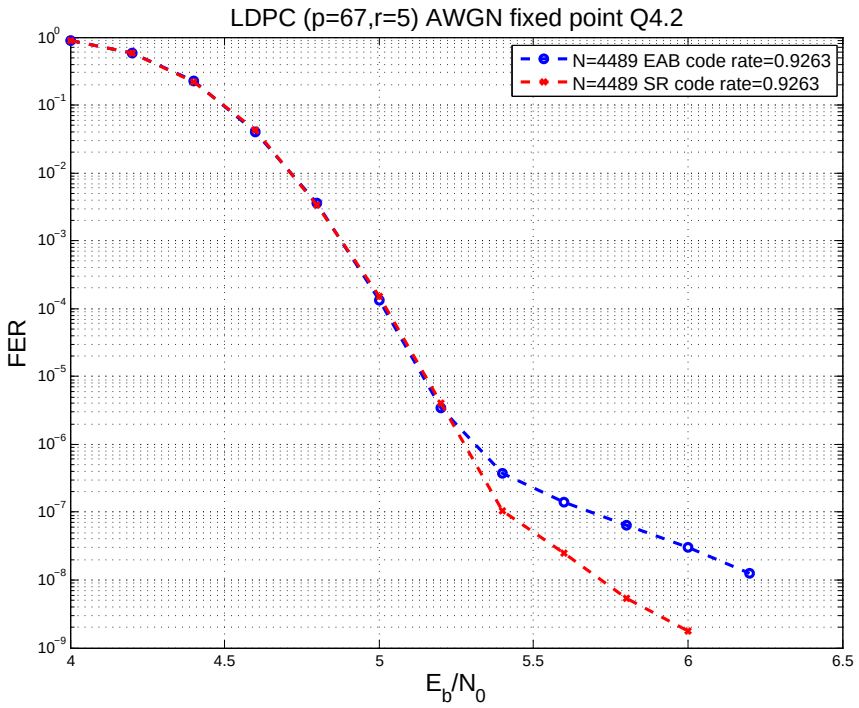


Figure 2.19: Performance comparison of $(4489, 4158)$ EAB and SR-SCB codes.

This code provably eliminates $(6, 4)$ absorbing sets without introducing smaller absorbing sets. The profiles in Table 2.5 also support this claim. Similar analysis can be applied to the codes in [Fos04] as well.

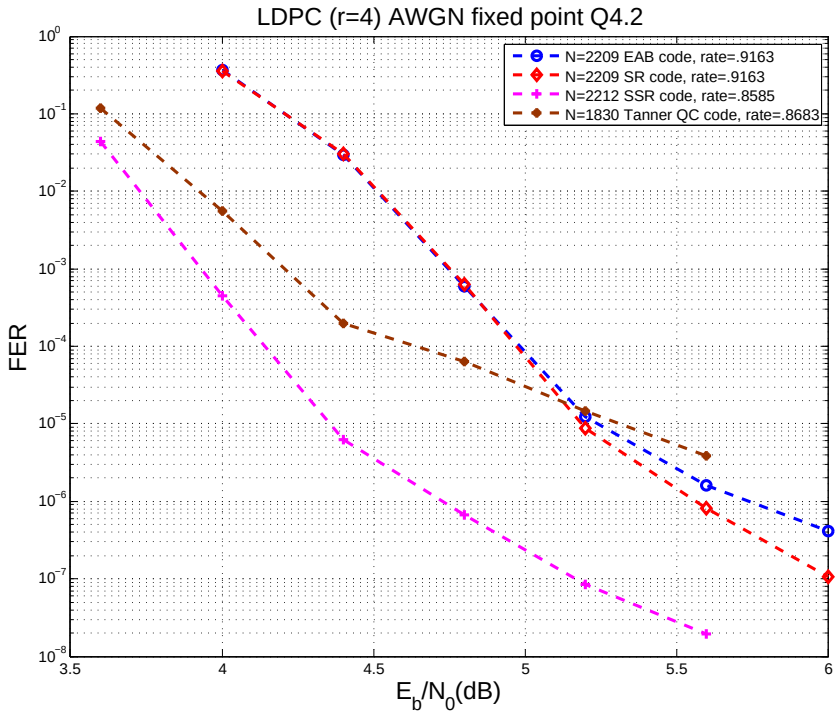


Figure 2.20: Performance comparison of EAB, SR, SSR, and a code from [TSS04].

Table 2.5: Error profiles for $(2209, 2024)$ EAB ($p = 47$), SR-SCB ($p = 47$), and SSR-SCB ($p = 79$) codes.

SNR	n.r.	n.e.	(6,4)	(6,6)	(7,4)	(8,2)	(8,4)	(9,4)	(10,4)	(12,4)
$(2209, 2024)$ EAB code										
5.6dB	2.0E8	322	236	2	2	27	3	1	37	1
6.0dB	8.0E8	329	329	0	0	0	0	0	0	0
$(2209, 2024)$ SR-SCB code RSF $= [0, 1, 2, 4]$										
5.6dB	2.0E8	167	38	3	0	40	45	3	2	0
6.0dB	8.0E8	88	4	0	0	2	48	3	0	0
$(2212, 1899)$ SSR-SCB code RSF $= [0, 1, 3, 4]$										
5.2dB	8.0E8	98	0	6	5	21	23	1	16	5
5.6dB	1.6E9	32	0	0	0	0	11	0	0	0

2.7 Conclusion

This chapter proposes the SR-SCB and SSR-SCB code families based on circulant matrices that are suitable for applications operating at low FER levels. We introduced a novel cycle consistency matrix description of dominant absorbing sets to guide code design and analysis. Our approach is a deterministic method that can provably eliminate certain absorbing sets in a large family of circulant-based codes, and can do so without changing the code's properties such as girth, rate and implementation complexity. This approach thus offers a class of codes with provably better performance than some known constructions. Theoretical findings were substantiated by experimental results showing consistent and significant improvement in the low BER region over a range of decoding algorithms, both in software and in hardware. A useful feature of the proposed approach from the implementation standpoint is that it can be easily combined with better decoding algorithms for the maximum improvement of the overall performance. Future work involves extending the analysis to other channel types and irregular codes.

CHAPTER 3

Soft Information for LDPC Decoding in Flash: Mutual-Information Optimized Quantization

High-capacity NAND flash memory achieves high density storage by using multi-level cells (MLCs) to store more than one bit per cell. Although this larger storage capacity is certainly beneficial, the increased density increases the raw bit error rate (BER), making powerful error correction coding necessary. Traditional flash memories employ algebraic codes, such as BCH codes, that can correct a fixed, specified number of errors. This chapter investigates the application of low-density parity-check (LDPC) codes which are well known for their ability to approach capacity in the AWGN channel. We obtain soft information for the LDPC decoder by performing multiple cell reads with distinct word-line voltages. The values of the word-line voltages (also called reference voltages) are optimized by maximizing the mutual information (MI) between the input and output of the multiple-read channel. Our results show that using this soft information in the LDPC decoder provides a significant benefit and enables the LDPC code to outperform a BCH code with comparable rate and block length over a range of block error rates. Using maximum mutual-information (MMI) quantization in the LDPC decoder provides an effective and efficient estimate of the word-line voltages. Constraining MMI quantization to satisfy the constant-ratio condition simplifies the optimization with negligible performance loss for four-level cells. This chapter also shows that quantization can affect how the LDPC degree distribution should be optimized and provides an example where adjusting the degree distribution away from the"optimal" AWGN distribution can improve performance in a quantized setting.

3.1 Introduction

Flash memory can store large quantities of data in a small device that has low power consumption and no moving parts. The original NAND flash memories used only two levels. This was called single-level-cell (SLC) flash because there is only one nonzero charge level. Devices currently available use multiple levels and are referred to as multiple-level cell (MLC) flash. Four and eight levels are currently in use, and the number of levels will increase further to provide more storage capability [LLa08][TSa09].

Error control coding for flash memory is becoming more important in a variety of ways as the storage density increases. The increasing number of levels (and smaller distance between levels) means that variations in cell behavior from cell to cell (and over time due to wear-out) lower the signal-to-noise ratio of the read channel making a stronger error-correction code necessary. Reductions in feature size make inter-cell interference more likely, adding an equalization or interference suppression component to the read channel [LHC02]. Also, the wear-out effect is time-varying, introducing a need for adaptive coding to maximize the potential of the system.

Low-density parity-check (LDPC) codes are well-known for their capacity-approaching ability in the AWGN channel [RSU01]. LDPC codes have typically been decoded with soft reliability information while flash systems have typically provided only hard reliability information to their decoders. This chapter demonstrates that at least some soft information is crucial to successfully reaping the benefits of LDPC coding in flash memory. The chapter explores how much soft information is necessary to provide most of the LDPC performance benefit and how flash systems could be engineered to provide the needed soft information without an unnecessary penalty in complexity or processing time. This chapter also explores how the lack of full soft information changes the LDPC code design process.

This chapter first uses pulse-amplitude modulation (PAM) with a Gaussian noise to model flash cell threshold voltage levels, and investigates how to optimize the word-line voltages to provide the maximum mutual information (MMI) between the input and the output of the equivalent read channel. After choosing the word-line voltage for each of the reads, the multiple-read

channel can be represented by a probability transition matrix and the data can be decoded with a standard belief-propagation algorithm. The chapter then extends the MMI approach to the retention noise model [WDZ11b] which more accurately models flash memories. The MMI approach is also explored in [LT05] [KY] for the design of the message-passing decoders of LDPC codes to optimize the quantization of the output of a binary-input channel.

In [DXZ11], a heuristic quantization algorithm sets the word-line voltages to the value where the two adjacent pdfs have a constant ratio R. The constant-ratio method leaves the parameter R unspecified. The MMI approach provides a natural way to select R in the constant-ratio method. Simulation results show that the constant-ratio approach with MMI selection simplifies the MMI optimization with negligible performance loss for four-level cells as long as the underlying LDPC code is well designed for the quantized channel.

Indeed, the quantized setting has a significant impact on the LDPC code design because small absorbing sets, which were discussed earlier in this dissertation, can dominate performance. This chapter explores how the quantized setting should be considered in the selection of the LDPC degree distribution. LDPC codes are usually designed with the degree distribution optimized for the AWGN channel [RSU01]. However, our simulations show that, in the quantized setting, adjusting this "optimal" degree distribution to avoid small absorbing sets can significantly improve performance.

Section 3.2 introduces the basics of the NAND flash memory model and LDPC codes. Section 3.3 shows how to obtain word-line voltages by maximizing the MI of the equivalent read channel. This section presents the MMI optimization approach and the constant-ratio method for SLC and MLC flash for the PAM-Gaussian model and the retention noise model. Section 3.4 provides simulation results demonstrating the benefits of using soft information with word-line voltages selected as described in Section 3.3 and compares the full MMI optimization with optimization constrained using the constant-ratio method. Section 3.5 delivers the conclusions.

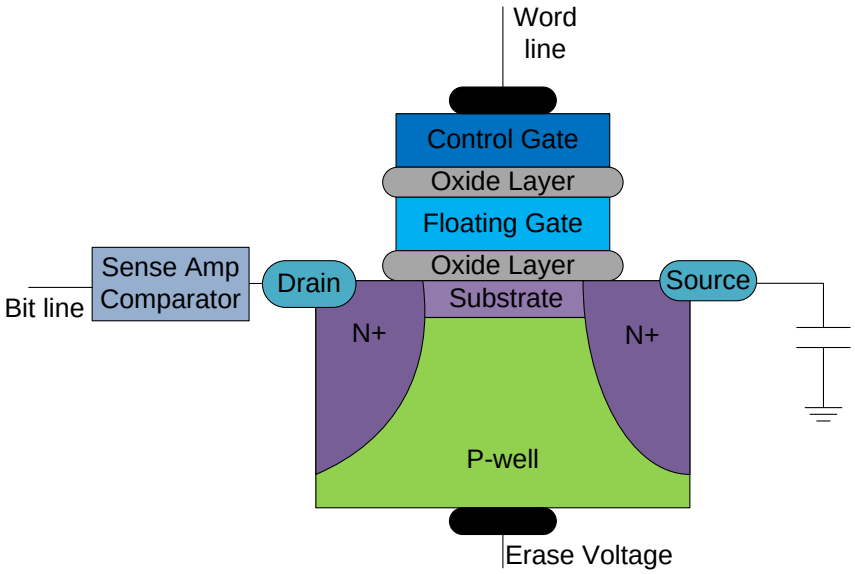


Figure 3.1: A NAND flash memory cell.

3.2 Background

This section introduces the basics of NAND flash memory and LDPC codes.

3.2.1 Basics of NAND Flash Memory

This chapter focuses on the NAND architecture for flash memory. Fig. 3.1 shows the configuration of a NAND flash memory cell. Each memory cell in the NAND architecture features a transistor with a control gate and a floating gate. To store information, a charge level is written to the cell by adding a specified amount of charge to the floating gate through Fowler-Nordheim tunneling by applying a relatively large voltage to the control gate [BCM03].

To read a memory cell, the charge level written to the floating gate is detected by applying a specified word-line voltage to the control gate and measuring the transistor drain current. The drain current is compared to a threshold by a sense amp comparator. If the drain current is above the comparator threshold, then the word-line voltage was sufficient to turn on the transistor, indicating that the charge written to the floating gate was insufficient to prevent the transistor

from turning on. If the drain current is below the threshold, the charge added to the floating gate was sufficient to prevent the applied word-line voltage from turning on the transistor. The sense amp comparator only provides one bit of information about the charge level present in the floating gate. A bit error occurring at this threshold-comparison stage is called a *raw bit error*.

The word-line voltage required to turn on a particular transistor (called the threshold voltage) can vary from cell to cell for a variety of reasons. For example, the floating gate can be overcharged during the write operation, the floating gate can lose charge due to leakage in the retention period, or the floating gate can receive extra charge when nearby cells are written [MH09].

The probability density function of the variation of threshold voltage from its intended value is often modeled by a Gaussian distribution. In this chapter, we initially assume an i.i.d. Gaussian threshold voltage for each level of an MLC flash memory cell such that an m-level flash cell is equivalent to an m-PAM communication system with AWGN noise, except that the threshold voltage cannot be directly observed. Rather, at most one bit of information about the threshold voltage may be obtained by each cell read.

More precise models such as the model in [MH09] in which the lowest and highest threshold voltage distributions have a higher variance and the model in [LZ10] in which the lowest threshold voltage (the one associated with zero charge level) is Gaussian and the other threshold voltages have Gaussian tails but a uniform central region are sometimes used. The model in [WDZ11b] is similar to [LZ10], but is derived by explicitly accounting for real dominating noise sources, such as inter-cell interference, program injection statistics, random telegraph noise and retention noise. This chapter uses the model of [WDZ11b] to study the MMI approach in a more realistic setting than the initial i.i.d. Gaussian assumption.

3.2.2 Basics of LDPC codes

LDPC codes [Gal63] are linear block codes defined by sparse parity-check matrices. By optimizing the degree distribution, it is well-known that LDPC codes can approach the capacity of an

AWGN channel [RSU01]. Several algorithms have been proposed to generate LDPC codes for a given degree distribution, such as the ACE algorithm [TJV04], and the PEG algorithm [HEA01].

Designing LDPC codes with low error-floors is crucial for the flash memory application since storage systems usually require block-error-rates lower than 10^{-15}. This topic has generated a significant amount of recent research including [Ric03] [WDW11b] [WDW11a] [ICV08] [NVM10] [HDL11].

In addition to their powerful error-correction capabilities, another appealing aspect of LDPC codes is the existence of low-complexity iterative algorithms used for decoding. These iterative decoding algorithms are called belief-propagation algorithms. Belief-propagation decoders commonly use soft reliability information about the received bits, which can greatly improve performance. Conversely, a quantization of the received information which is too coarse can degrade the performance of an LDPC code.

Traditional algebraic codes, such as BCH codes, use bounded distance decoding and can correct up to a specified, fixed number of errors. Unlike these traditional codes, it can be difficult for LDPC codes to guarantee a specified number of correctable errors. However the average bit-error-rate performance can often outperform that of BCH codes in Gaussian noise.

The remainder of this chapter studies how quantization during the read process affects the performance of LDPC decoding for flash memory. In the next section, we present a general quantization approach for selecting word line voltages for reading the flash memory cells in both the SLC and the MLC cases.

3.3 Obtaining Soft Information

Since the sense amp comparator provides at most one bit of information about the threshold voltage (or equivalently about the amount of charge present in the floating gate), decoders for error control codes in flash have historically used hard decisions on each bit from the sense-amp comparator outputs.

Soft information can be obtained in two ways: either by reading from the same sense amp comparator multiple times with different word line voltages (as is already done to read multi-level flash cells) or by equipping a flash cell with multiple sense amp comparators on the bit line, which is essentially equivalent to replacing the sense amp comparator (a one-bit A/D converter) with a higher-precision A/D converter.

These two approaches are not completely interchangeable. The real goal is to detect soft information about the threshold voltage. Each additional read of a single sense amp comparator can provide additional useful soft information about the threshold voltage if the word line voltages are well-chosen.

In contrast, multiple comparators may not give much additional information if the drain current vs. word-line-voltage curve (the classic I-V transistor curve) is too nonlinear. If the drain current has saturated too low or too high, the outputs from more sense-amp comparators are not useful in establishing precisely how much charge is in the floating gate. If the word line voltage and floating gate charge level place the transistor in the linear gain region, then some valuable soft information is provided by multiple sense amp comparators.

Our work focuses on the first technique described above in which soft information is obtained from multiple reads using the same sense-amp comparator with different word line voltages.

This section investigates the potential improvement of increasing the resolution beyond one bit and studies how best to obtain this increased resolution. In [DXZ11], the use of soft information was explored and the poor performance of uniformly spaced word-line voltages was clearly established.

This chapter takes an information-theoretic perspective on optimizing the word-line voltages. Similar to other work (not in the context of flash memory) such as [LT05] [KY], this chapter seeks to quantize so as to create an effective read channel that has the maximum mutual information (MMI). We study quantization models with different numbers of reads for both SLC and MLC flash memory.

In the course of our analysis, we choose the word-line voltages for each quantization to

achieve MMI between the input and output of each equivalent read channel. Theoretically, this choice of word-line voltages maximizes the amount of information provided by the quantization. The next subsection studies the simple PAM-Gaussian model. We first explore an example of SLC with just one additional read to provide extra soft information. Then, the section looks at the benefit of additional reads for SLC and MLC. After that, we extend the analysis to the retention noise model and study constraining the MMI approach with the constant-ratio method. Numerical results are given in Section 3.4.

3.3.1 PAM-Gaussian Model

This subsection describes how to select word-line voltages to achieve MMI in the context of a simple model of the flash cell read channel as PAM transmission with Gaussian noise. MMI word-line voltage selection is presented using three examples: SLC with two reads, SLC with three reads, and 4-level MLC with six reads.

For SLC flash memory, each cell can store one bit of information. Fig. 3.2 shows the model of the threshold voltage distribution as a mixture of two identically distributed Gaussian random variables. When either a "0" or "1" is written to the cell, the threshold voltage is modeled as a Gaussian random variable with variance $\sqrt{N_0/2}$ and either mean $-\sqrt{E_s}$ (for "0") or mean $+\sqrt{E_s}$ (for "1").

For SLC with two reads using two different word line voltages, which should be symmetric (shown as q and $-q$ in Fig. 3.2), the threshold voltage is quantized according to three regions shown in Fig. 3.2: the green region, the red region, and the union of the blue and purple regions. This quantization produces the effective discrete memoryless channel (DMC) model is shown in Fig. 3.3 (a) with input $X \in \{0, 1\}$ and output $Y \in \{0, e, 1\}$.

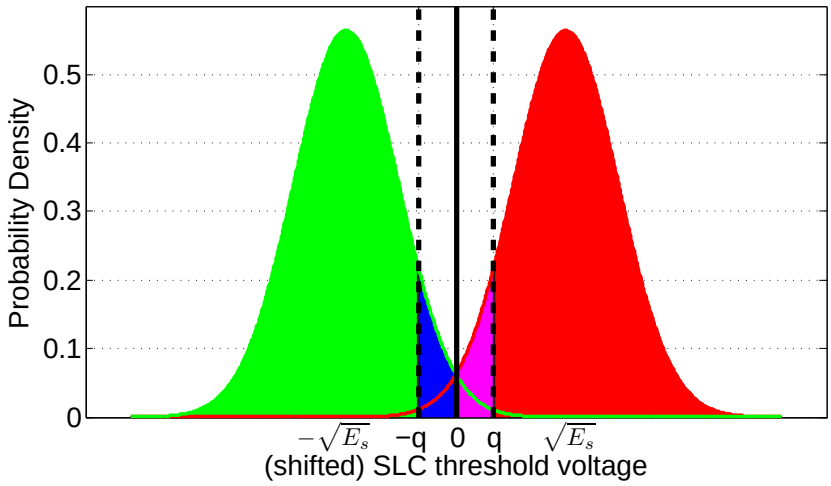
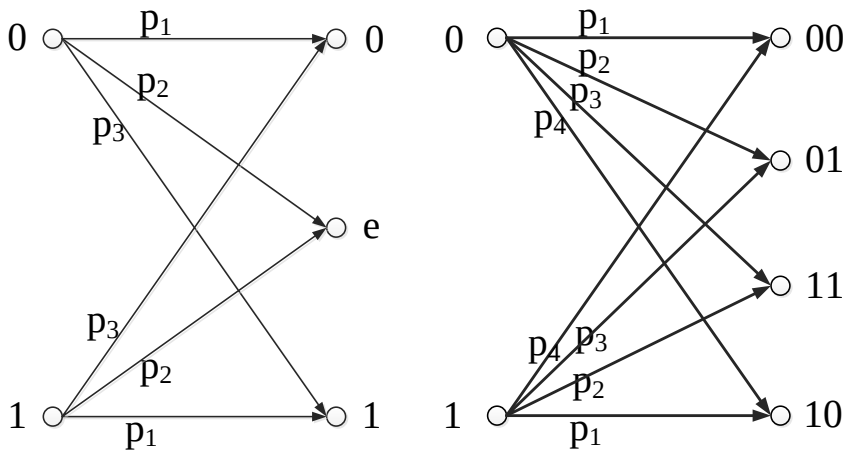


Figure 3.2: Identically distributed Gaussian model for SLC threshold voltages. Also shown are word-line voltages for two reads (the dashed lines) and three reads (all three lines). The quantization regions are indicated by color, with the middle region for two reads being the union of the blue and purple regions.



(a) Two reads

(b) Three reads

Figure 3.3: Equivalent discrete memoryless channel model for SLC with (a) two reads and (b) three reads with distinct word-line voltages.

Assuming X is equally likely to be 0 or 1, the MI between the input and output of the resulting DMC can be calculated as

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H\left(\frac{p_1+p_3}{2}, p_2, \frac{p_1+p_3}{2}\right) - H\left(p_1, p_2, p_3\right), \end{aligned} \tag{3.1}$$

where the crossover probabilities shown in Fig. 3.3 (a) are $p_1 = 1 - Q^-$, $p_2 = Q^- - Q^+$, and $p_3 = Q^+$ with

$$Q^- = Q\left(\frac{\sqrt{E_s} - q}{\sqrt{N_0/2}}\right) \text{ and } Q^+ = Q\left(\frac{\sqrt{E_s} + q}{\sqrt{N_0/2}}\right). \tag{3.2}$$

For fixed E_s/N_0, the MI in (3.1) is a quasi-convex function of q and can be maximized numerically to find the parameter q that yields the MMI. Fig. 3.4 and 3.5 show how MI varies as a function of q for three values of E_s/N_0 for two and three reads. The assumption here is that choosing q to maximize the MI should provide approximately optimal LDPC decoding performance for a given level of quantization. The optimum q^* is a function of E_s/N_0 as shown in Fig. 3.6. For example, if $E_s/N_0 = 3.241$ dB, $q^* = 0.2188\sqrt{E_s}$, and if $E_s/N_0 = 6.789$ dB, $q^* = 0.1253\sqrt{E_s}$. However, we note that the dependence of q on E_s/N_0 is not dramatic for practical SNR ranges because when E_s is known, the MI is only slightly reduced when, for example, the optimal q for $E_s/N_0 = 3.241$ dB is used when the actual SNR is $E_s/N_0 = 6.789$ dB. The selection of q becomes more critical as the channel SNR degrades, so the value of q should be set according to the worst expected noise conditions.

For SLC with three reads for each cell, the word-line voltages should again be symmetric (shown as q, 0, and $-q$ in Fig. 3.2). The threshold voltage is quantized according to the four differently colored regions shown in Fig. 3.2. This quantization produces the effective DMC model is shown in Fig. 3.3 (b) with input $X \in \{0, 1\}$ and output $Y \in \{00, 01, 10, 11\}$.

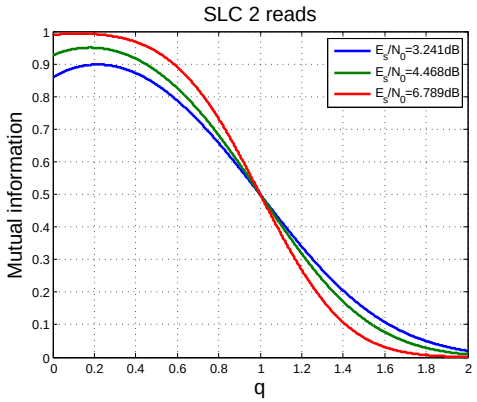


Figure 3.4: MI vs. q (for $E_s = 1$) for $E_s/N_0 = 3.241$ dB, 4.468 dB and 6.789 dB for SLC flash with two reads.

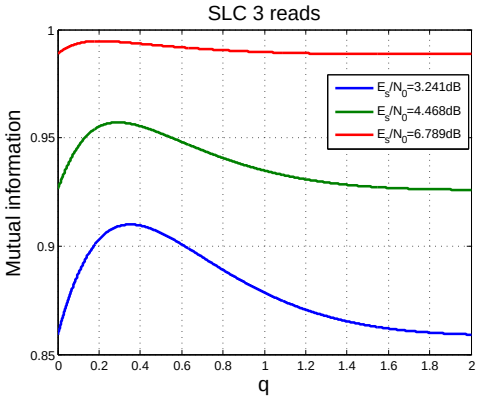


Figure 3.5: MI vs. q (for $E_s = 1$) for $E_s/N_0 = 3.241$ dB, 4.468 dB and 6.789 dB for SLC flash with three reads.

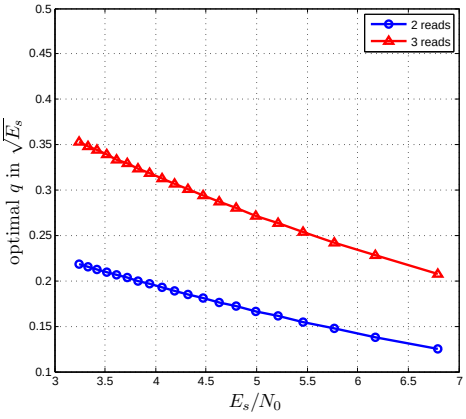


Figure 3.6: Optimal q (for $E_s = 1$) as a function of E_s/N_0 for SLC Flash with two and three reads.

Assuming X is equally likely to be 0 or 1, the MI between the input and output of this DMC can be calculated as

$$\begin{aligned} I(X;Y) =& H(Y) - H(Y|X) \\ =& H\left(\frac{p_1+p_4}{2}, \frac{p_2+p_3}{2}, \frac{p_3+p_2}{2}, \frac{p_4+p_1}{2}\right) \\ & - H(p_1, p_2, p_3, p_4), \end{aligned} \tag{3.3}$$

where the crossover probabilities are $p_1 = 1 - Q^-$, $p_2 = Q^- - Q^0$, $p_3 = Q^0 - Q^+$,and $p_4 = Q^+$ with Q^- and Q^+ as above and

$$Q^0 = Q\left(\frac{\sqrt{E_s}}{\sqrt{N_0/2}}\right). \tag{3.4}$$

Fig. 3.7 shows how MI increases with the number of reads for the SLC case. The top curve shows the MI possible with full soft information (where the receiver would know the threshold voltage exactly). The bottom curve shows the MI available with a single read, which is what is available with a typical SLC implementation. With two reads, the MI is improved enough to close about half of the gap between the single-read MI and the MI of full soft information. The MI with 3 reads is larger than the MI with 2 reads. This approach can be extended to however many reads are desired, but with diminishing returns as shown in Fig. 3.7. Fig. 3.8 shows how the raw bit error probability requirement to achieve an MI of 0.9021 increases (relaxes) as the number of reads increases.

For 4-level MLC flash memory, each cell can store 2 bits of information. Extending the previously introduced SLC model in the natural way, we model the MLC read channel as a 4-PAM signal with AWGN noise. To minimize the raw bit error rate, we also use the Gray labeling $(00, 01, 11, 10)$ for these four levels. Typically in 4-level MLC flash, each cell is compared to 3 word-line voltages and thus the output of the comparator has 4 values (i.e., four distinct quantization regions). If we consider three additional word-line voltages (for a total of six), the threshold voltage can be quantized to seven distinct regions. This leads to an effective DMC model with four inputs and seven outputs. The qualitative behavior of MMI and required BER as the number of reads is increased is essentially the same as in Figs. 3.7 and 3.8 for SLC.

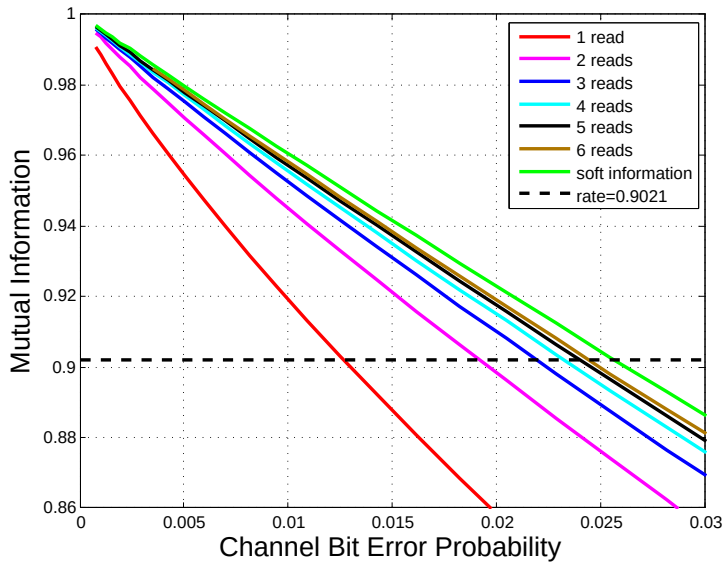


Figure 3.7: MI provided by different quantizations for SLC. The dashed horizontal line indicates the operating rate of our simulations. When a MI curve is below the dashed line, the read channel with that quantization cannot possibly support the attempted rate.

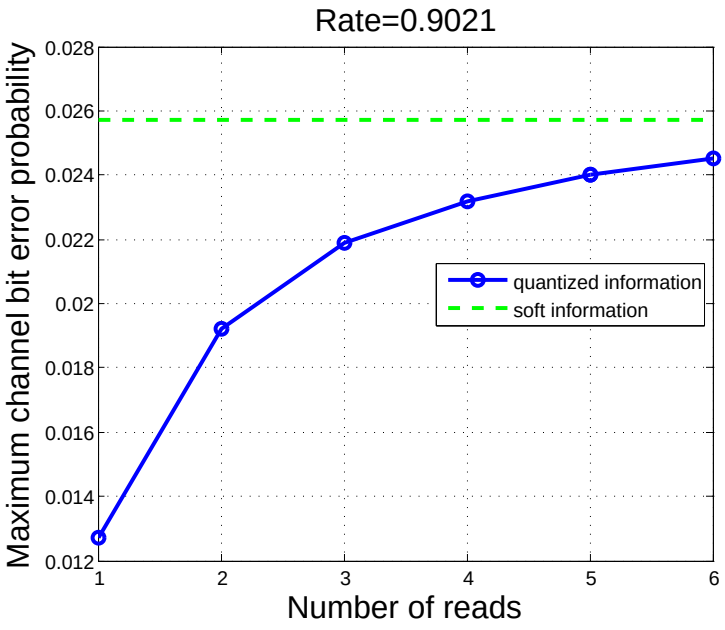


Figure 3.8: Maximum tolerable channel bit error probability (raw BER) supported by different quantizations for SLC operating at rate 0.9021.

If we consider three additional word-line voltages (for a total of six), the threshold voltage can be quantized to seven distinct values as shown in Figure 3.9. The resulting DMC is given in Figure 3.10. Since the channel is symmetric, the crossover probabilities for the channel model are symmetric in the upper and lower half of the figure, i.e., $p_{11} = p_{44}$, $e_{1a} = e_{4c}$, $p_{12} = p_{43}$, etc.

Similar to the SLC analysis, the MI between the input and output can be calculated as

$$\begin{aligned}
&I(X;Y)\\
=&H(Y) - H(Y|X)\\
=&H\left(\frac{p_{11}+p_{21}+p_{24}+p_{14}}{4}, \frac{p_{12}+p_{22}+p_{23}+p_{13}}{4},\right.\\
&\frac{p_{13}+p_{23}+p_{22}+p_{12}}{4}, \frac{p_{14}+p_{24}+p_{21}+p_{11}}{4},\\
&\frac{e_{1a}+e_{2a}+e_{2c}+e_{1c}}{4}, \frac{e_{1b}+e_{2b}+e_{2b}+e_{1b}}{4},\\
&\left.\frac{e_{1c}+e_{2c}+e_{2a}+e_{1a}}{4}\right)\\
&-\frac{1}{2}H(p_{11}, p_{12}, p_{13}, p_{14}, e_{1a}, e_{1b}, e_{1c})\\
&-\frac{1}{2}H(p_{21}, p_{22}, p_{23}, p_{24}, e_{2a}, e_{2b}, e_{2c}),
\end{aligned} \tag{3.5}$$

where all of the crossover probabilities can be calculated in the same manner as with SLC. Thus, in order to choose the optimal quantization levels q_1, q_2, and q_3 for a fixed E_s/N_0, we maximize the MI given in equation (3.5).

3.3.2 Retention Model

We can extend the MMI analysis of Section 3.3.1 to any model for the flash memory read channel. As an example, we model a 4-level 6-read MLC as a 4-input 7-output discrete channel as shown in Fig. 3.10. Instead of assuming Gaussian noise distributions as in Section 3.3.1, this section numerically computes the probability transition matrix using the retention noise model from [WDZ11b]. Fig. 3.11 shows the four conditional threshold-voltage probability density functions generated according to [WDZ11b] and the resulting six MMI word-line voltages after six months retention time. While the conditional noise for each transmitted (or written) threshold voltage is

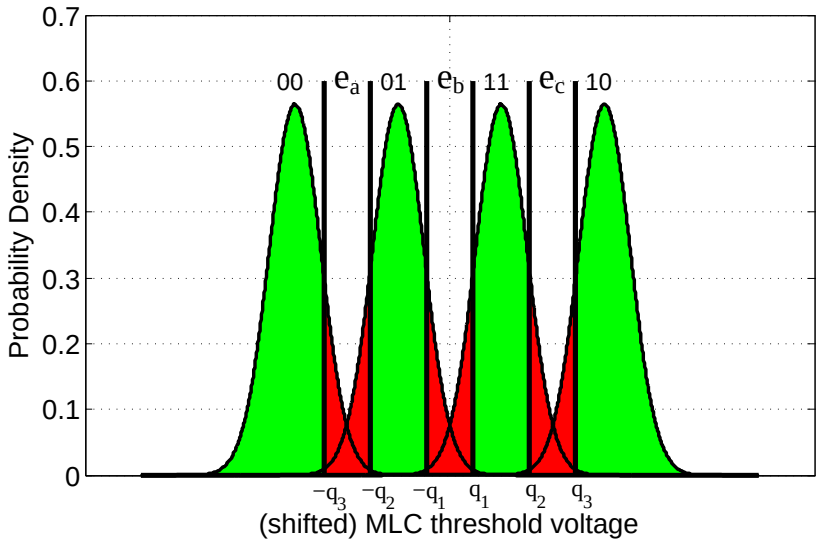


Figure 3.9: Channel model for 4-MLC with 6 reads.

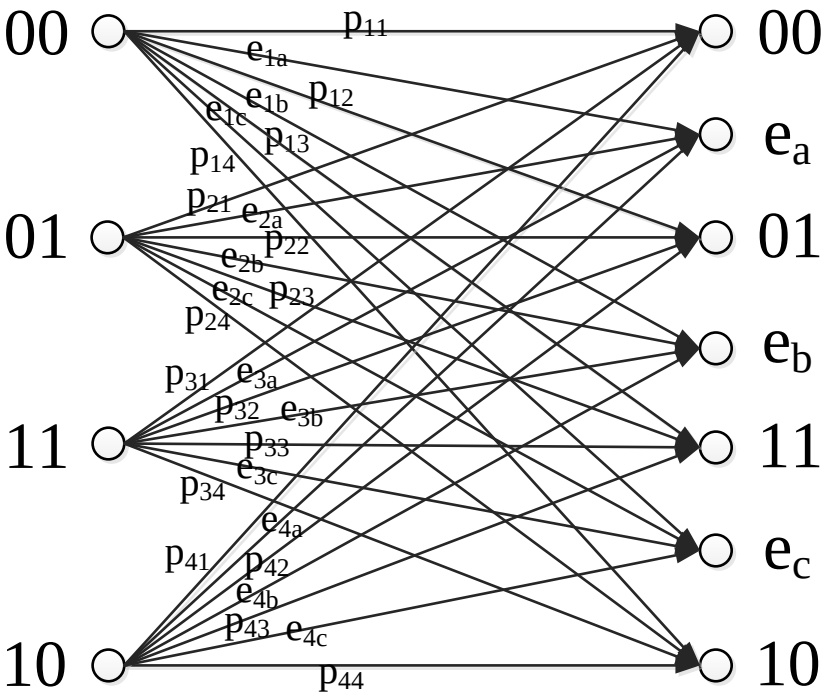


Figure 3.10: Quantization model for 4-MLC with 6 reads.

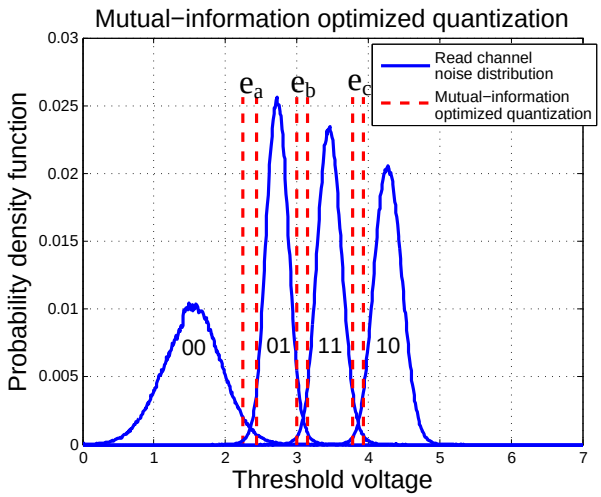


Figure 3.11: Mutual-information optimized quantization for the 6-month data.

similar to that of a Gaussian, the variance of the conditional distributions varies greatly across the four possible threshold voltages. Note that the lowest threshold voltage has by far the largest variance.

Since the retention noise model is not symmetric, we need to numerically compute all the probabilities in Fig. 3.10 and calculate the MI between the input and output:

$$\begin{aligned}
I(X;Y) =& H(Y) - H(Y|X) \\
=& H\left(\frac{p_{11}+p_{21}+p_{31}+p_{41}}{4}, \frac{p_{12}+p_{22}+p_{32}+p_{42}}{4},\right. \\
& \frac{p_{13}+p_{23}+p_{33}+p_{43}}{4}, \frac{p_{14}+p_{24}+p_{34}+p_{44}}{4}, \\
& \frac{e_{1a}+e_{2a}+e_{3a}+e_{4a}}{4}, \frac{e_{1b}+e_{2b}+e_{3b}+e_{4b}}{4}, \\
& \left.\frac{e_{1c}+e_{2c}+e_{3c}+e_{4c}}{4}\right) \\
& - \frac{1}{4}H(p_{11}, p_{12}, p_{13}, p_{14}, e_{1a}, e_{1b}, e_{1c}) \\
& - \frac{1}{4}H(p_{21}, p_{22}, p_{23}, p_{24}, e_{2a}, e_{2b}, e_{2c}) \\
& - \frac{1}{4}H(p_{31}, p_{32}, p_{33}, p_{34}, e_{3a}, e_{3b}, e_{3c}) \\
& - \frac{1}{4}H(p_{41}, p_{42}, p_{43}, p_{44}, e_{4a}, e_{4b}, e_{4c}). \qquad (3.6)
\end{aligned}$$

The MI in (3.6) is in general not a quasi-concave function in terms of the word-line voltages $q_1, q_2, ..., q_6$, although it is quasi-concave for the simple model of two symmetric Gaussians with symmetric word-line voltages studied in [WCS11]. Since (3.6) is a continuous and smooth function and locally quasi-concave in the range of our interest, we can numerically compute the maximum MI with a careful use of bisection search.

In [DXZ11], a heuristic quantization method constrains the word-line voltages to values where the two adjacent pdfs have a constant ratio R. This method leaves the selection of R to empirical simulation. The constant-ratio method can be viewed as a constraint that can be applied to MMI optimization in order to reduce the search space. As we will see in the next section, constraining the quantization using the constant-ratio method can also simplify optimization by yielding a convex or quasi-convex problem.

3.4 Simulation Results

In this section we explore how to design an LDPC code in a quantized setting and demonstrate the benefits of LDPC decoding using soft information provided through multiple reads. A rate-0.9021 BCH code with block length $n = 9152$ and dimension $k = 8256$ provides a baseline for comparison. For our simulations, we used three rate-0.9021 irregular LDPC codes, both with block length $n = 9118$ and dimension $k = 8225$. Each of the three codes employs a distinct degree distribution. However, each code was designed using the ACE algorithm [TJV04], and the stopping-set check algorithm [RW04] to optimize the LDPC matrix while maintaining the prescribed degree distribution. All of the simulations were performed using a sequential belief propagation decoder.

3.4.1 Degree distribution in a quantized setting

Two of the LDPC codes studied feature distinct degree distributions with maximum variable degree 19. For Code 1, the degree distribution is the usual optimal degree distribution for AWGN [RSU01]. For Code 2, the initial AWGN-optimal degree distribution is adjusted to improve performance in a quantized setting as follows:

Hard decoding makes the belief propagation process especially vulnerable to small absorbing sets such as $(4, 2)$, $(5, 1)$, $(5, 2)$ absorbing sets. Fig. 3.12 shows a (4,2) absorbing set found in Code 1. Note that all of the variable nodes in this (4,2) absorbing have degree 3. To preclude this absorbing set, we increase all the degree-3 variable nodes to have degree 4 to produce Code 2. Code 2 significantly outperforms Code 1 under hard decoding.

We also simulated another code with the maximum variable degree 24 with degree distribution optimized for AWGN (Code 3). Fig. 3.13 shows frame error rate versus retention time under hard decoding for these three codes.

Code 3 has an even better density evolution threshold in AWGN than Code 1, but the newly designed Code 2 with the lower AWGN threshold still outperforms it under hard quantization.

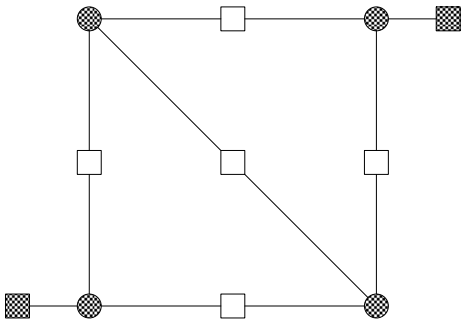


Figure 3.12: A (4,2) absorbing set which is avoided by precluding degree-3 nodes

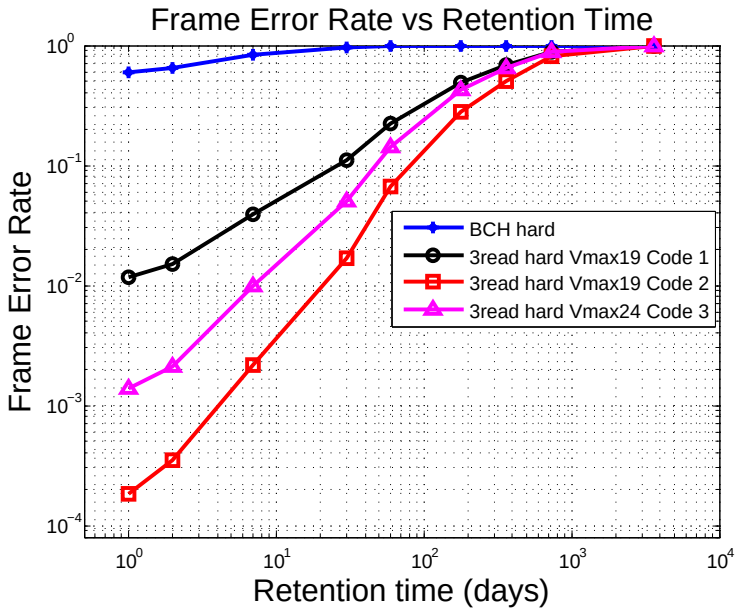


Figure 3.13: Simulation results for 4-level MLC using hard quantization.

This demonstrates that a superior AWGN threshold does not necessarily imply superior performance under hard decoding. When simulated in AWGN with full resolution soft decoding, Code 3 performs better than Code 1, and Code 1 performs better than Code 2 . However, when simulated under hard decoding, Fig. 3.13 shows that Code 2, which has the worst density evolution threshold under AWGN, gives the best performance because precluding degree-3 nodes has removed the small absorbing sets that were dominating performance.

3.4.2 Maximizing mutual information to minimize frame error rate

Fig. 3.14 shows MI as a function of R for four-level MLC for both the simple 4-PAM Gaussian model and the more realistic retention model of [WDZ11b] and the corresponding frame error rates as a function of R for Code 2, which, as described above, is an LDPC code who's degree distribution has been adjusted to preclude small absorbing sets that are troublesome for quantized channels. The Gaussian and retention channels in this figure were selected so that they had an identical MMI for six-read (seven-level) unconstrained MMI optimization. For both models the constant-ratio method provides the same MMI as obtained by the unconstrained MMI optimization. Furthermore, it is striking how similar the MMI vs. R behavior is for the two different channel models. For the Gaussian model, MI is a concave function of R. The curve of MI vs. R for the retention model closely follows the Gaussian model curve, but is not a strictly concave function.

The MMI approach is a way to select quantization levels in the hope of optimizing frame-error-rate (FER) performance. Fig. 3.14 shows the frame error rate (FER) performance as a function of R for both the Gaussian model and the retention model. The value of R that provides the maximum MI also delivers the lowest FER of 3×10^{-5}, thus confirming the essential hypothesis of the MMI approach. Note that while the MI range is small (approximately 0.01 bits), this variation in MI corresponds to more than an order of magnitude of FER variation.

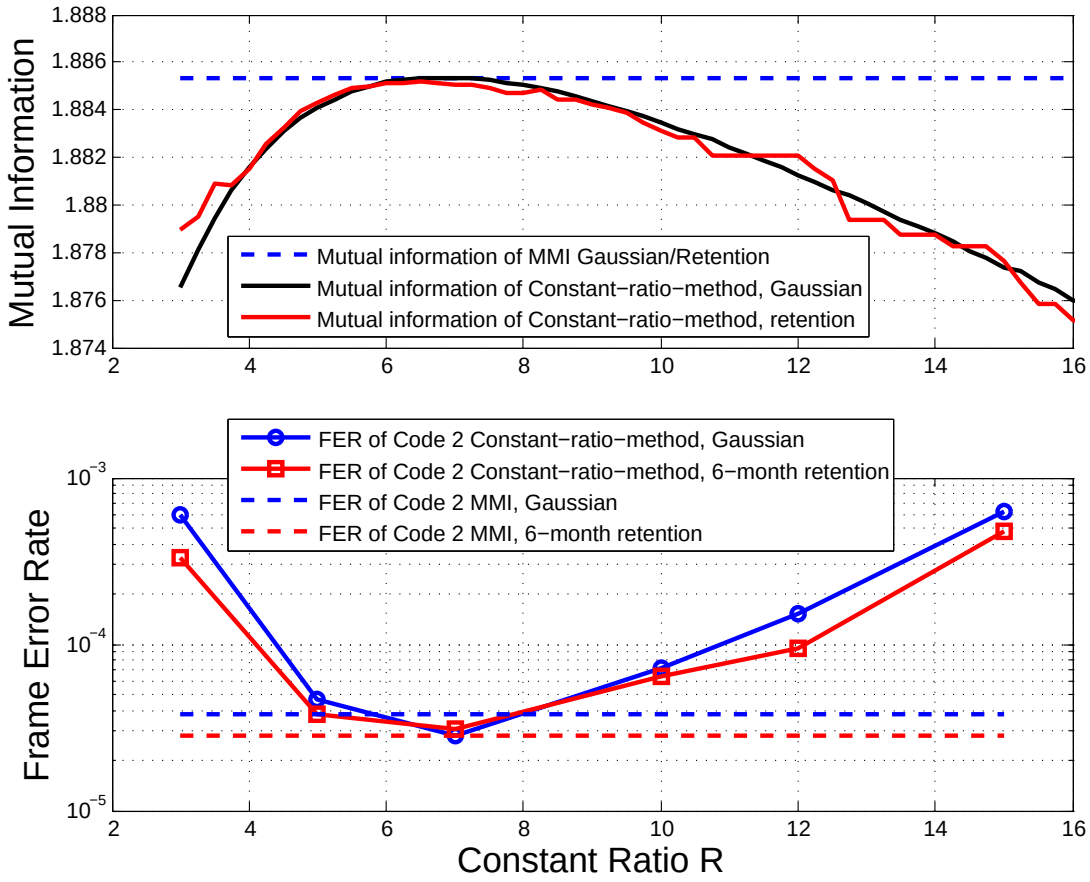


Figure 3.14: MI as a function of the constant-ratio value R for the 4-PAM Gaussian model with $E_s/N_0 = 10.76dB$ and the retention model for 6 months. Code 2 is simulated. These two models both have an MMI of 1.885 bits. Note that the minimum FER occurs for the value of R that maximizes mutual information, and that the FER increases quickly as R is varied away from the optimum point.

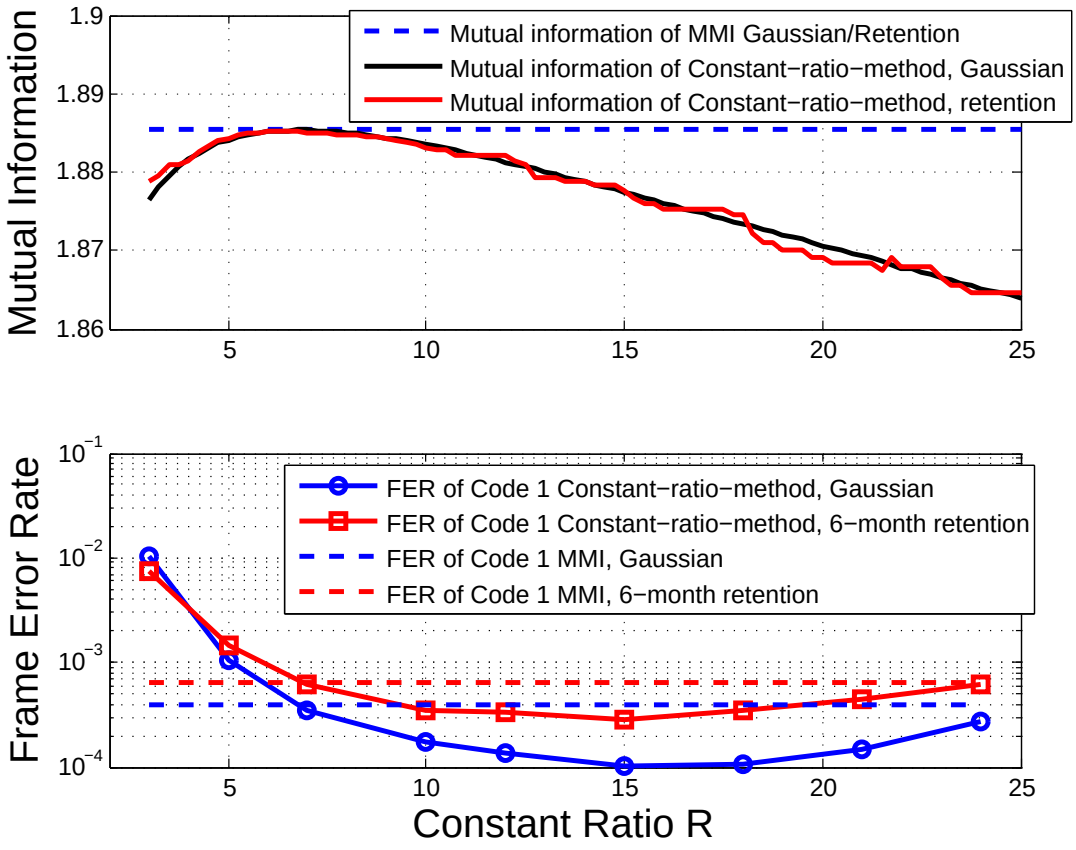


Figure 3.15: MI as a function of the constant-ratio value R for the 4-PAM Gaussian model with $E_s/N_0 = 10.76dB$ and the retention model for 6 months. Code 1 is simulated. These two models both have an MMI of 1.885 bits.

If a code is not well-designed for the specific quantized channel, the MMI approach may not provide the best performance as shown in Fig. 3.15. Code 1 is designed for the AWGN channel, but, as discussed above, does not perform well in the quantized setting. Fig. 3.15 shows that the minimum FER achieved is 10^{-4} on the quantized Gaussian channel and about 3×10^{-4} on the quantized retention channel. The value of R that produces the minimum FER is 15 while the maximum MI is achieved by $R = 7$. While the actual difference in FER performance between $R = 7$ and $R = 15$ is small, this figure shows that the MMI approach does not identify the minimum FER quantization for every code.

In particular, in this example where a code has small absorbing sets that limit its performance

in a quantized setting the maximum MMI quantization does not minimize FER. It is important to note that for the best codes, and hence the ones that would be used in practice, the MMI approach is quite effective, as shown in FIg. 3.14 . It is an area for further investigation to understand more fully the mechanism by which Code 1 performs better with a quantization that provides less mutual information than the maximum possible, even if the variation in FER is small between that provided by the MMI approach and the best FER found across possible R values.

3.4.3 Soft information with extra reads

Since the BCH decoder is limited to using hard decisions from the comparator, we first simulate the LDPC decoder using only hard decisions in order to make a fair baseline comparison. The BCH and LDPC 1-bit curves in Figs. 3.16 and 3.17 show that the LDPC code outperforms the BCH code in this range of frame error rates, but not significantly so. The red dashed vertical line gives the Shannon limit for operating at rate 0.9021 on this channel with a single bit of reliability information. Frame error rate (FER) is plotted vs. channel bit error probability (raw bit error probability). The frame sizes are the block lengths, $k = 8256$ for BCH and $k = 8225$ for LDPC.

Providing an additional bit of reliability information to the LDPC decoder through increased quantization resolution improves performance significantly, recovering almost all of the performance available with full soft information. This can be observed by comparing the Shannon limits corresponding to varying levels of soft information with their respective simulations in Figs. 3.16 and 3.17.

We also plot the frame error rate versus the traditional signal-to-noise ratio E_s/N_0 in Fig. 3.18 for SLC, where each E_s/N_0 corresponds to an equivalent raw bit error rate in Fig. 3.16.

Of course the BCH code will also benefit from the use of soft information. However, we were unable to perform simulations of a BCH decoder utilizing soft information (such as erasures) for inclusion in this chapter.

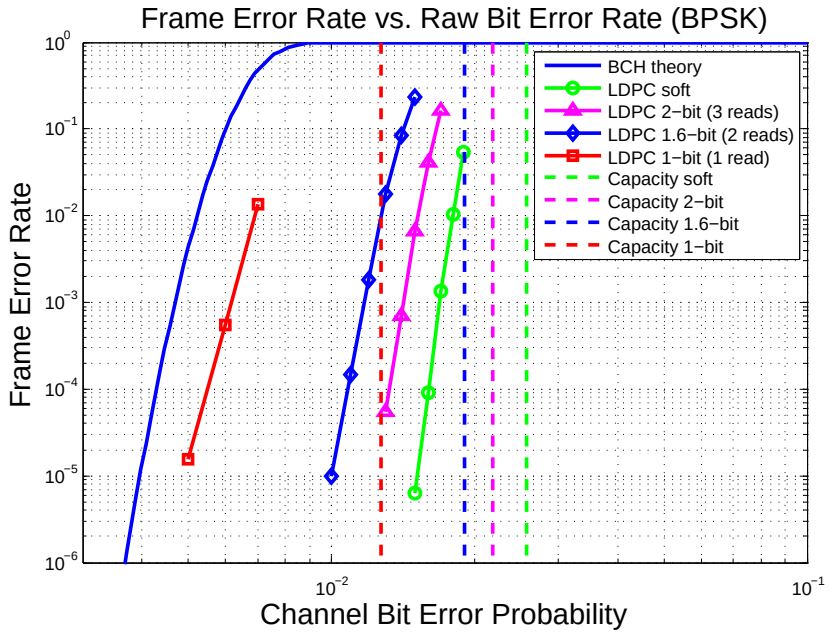


Figure 3.16: Simulation results showing FER vs. raw channel bit error probability for SLC using Code 2 and MMI quantization.

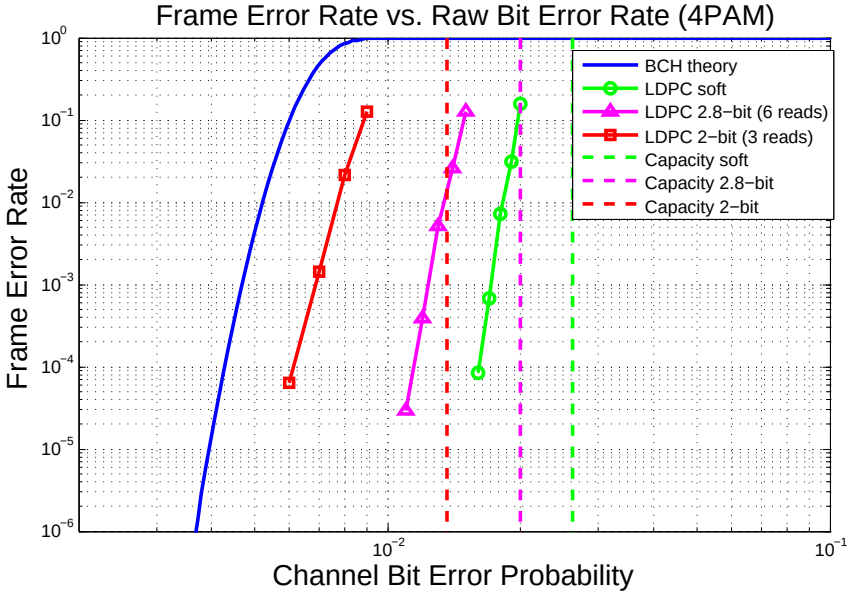


Figure 3.17: Simulation results showing FER vs. raw channel bit error probability for 4-level MLC using Code 2 and MMI quantization.

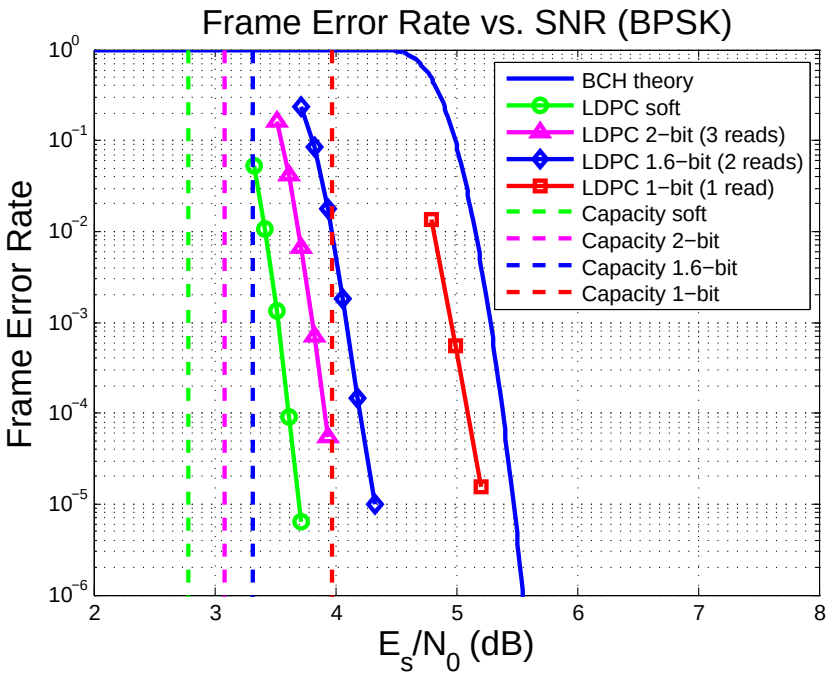


Figure 3.18: Simulation results showing FER vs. signal-to-noise ratio for SLC using Code 2 and MMI quantization.

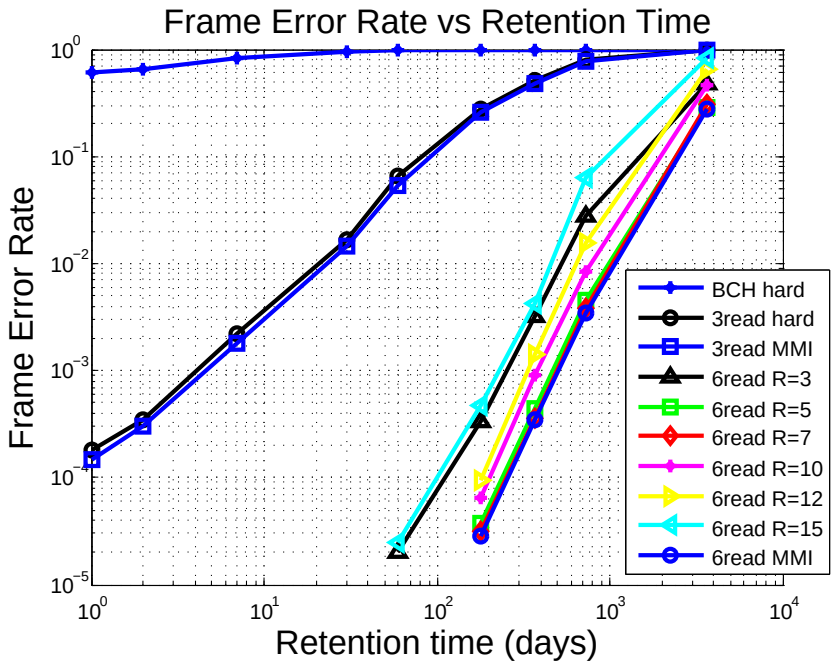


Figure 3.19: Simulation results of FER vs. retention time for LDPC Code 2 for 4-level MLC with the retention channel model using MMI and constant-R quantization with a variety or R values.

3.4.4 Comparison of quantization methods

Fig. 3.19 shows frame error rate (FER) plotted versus retention time for Code 2 under a variety of quantizations. As discussed above, Code 2 has an adjusted degree distribution (with no degree-3 nodes) to lower the FER for hard quantization. Since the noise model is not symmetric, the MMI approach with 3 reads has a slightly larger MI than the hard quantization with 3 reads, and thus performs slightly better.

This plot shows the behavior that was highlighted in Fig. 3.14: the performance of the LDPC codes is closely related to the MI of the equivalent channel given in Fig. 3.20. The $R = 7$ case has the closest MI to the MMI in Fig. 3.20 and its FER performance is also very close to that of the MMI approach in Fig. 3.19. For comparison, Fig. 3.21 shows similar results for six reads using the Gaussian model.

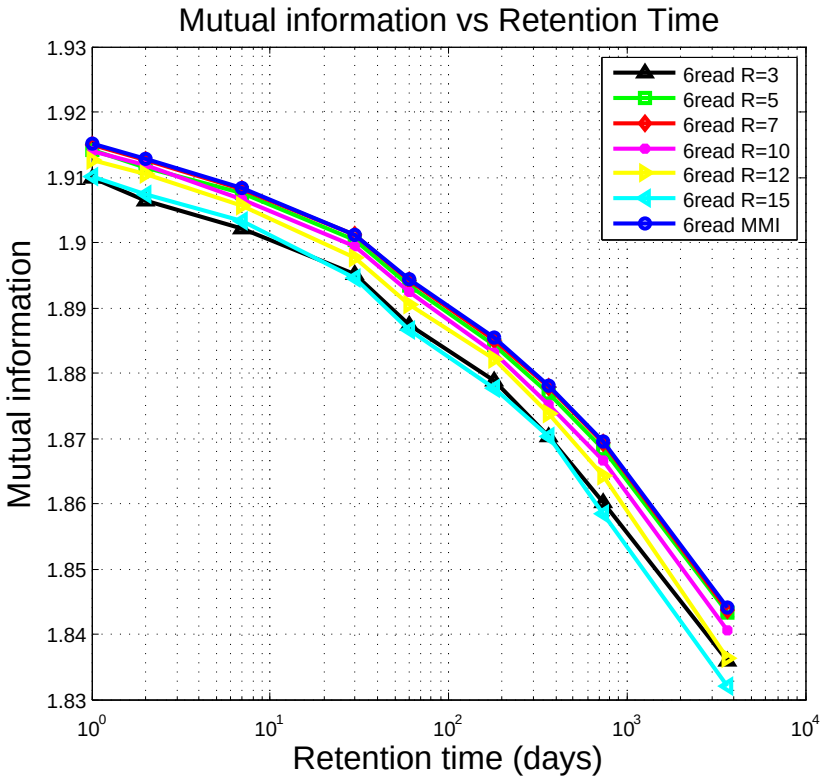


Figure 3.20: Mutual information vs. retention time for the MMI quantization and constant-ratio quantizations with a variety of R values.

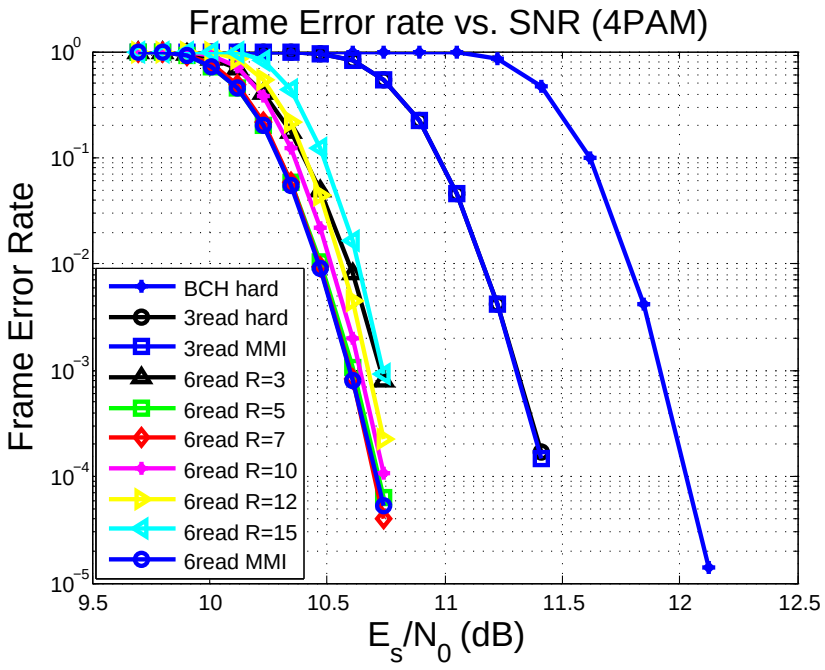


Figure 3.21: Simulation results of FER vs. SNR for LDPC Code 2 for 4-level MLC with the Gaussian channel model using MMI and constant-R quantization with a variety or R values.

Figs. 3.22 and 3.23 are analogous to Figs. 3.19 and 3.21 but for Code 1, which was not adjusted for the quantized setting. Thus, Figs. 3.22 and 3.23 present the detailed simulations that were highlighted in 3.15. In this case the constant ratio method with $R = 15$ slightly outperforms the MMI approach. Figs. 3.22 shows results for six reads per cell for four-level MLC using the retention channel model, and Fig. 3.23 shows similar results for six reads per cell for four-level MLC using the Gaussian model. This example reflects our experience that the only cases in which the constant ratio method (slightly) outperforms the MMI method are cases in which the LDPC degree distribution is not well-matched to the channel. In other words, when one has identified a good code for the channel, MMI will give the best quantization. The degree distribution for which MMI was not optimal was also not the best choice of degree distribution.

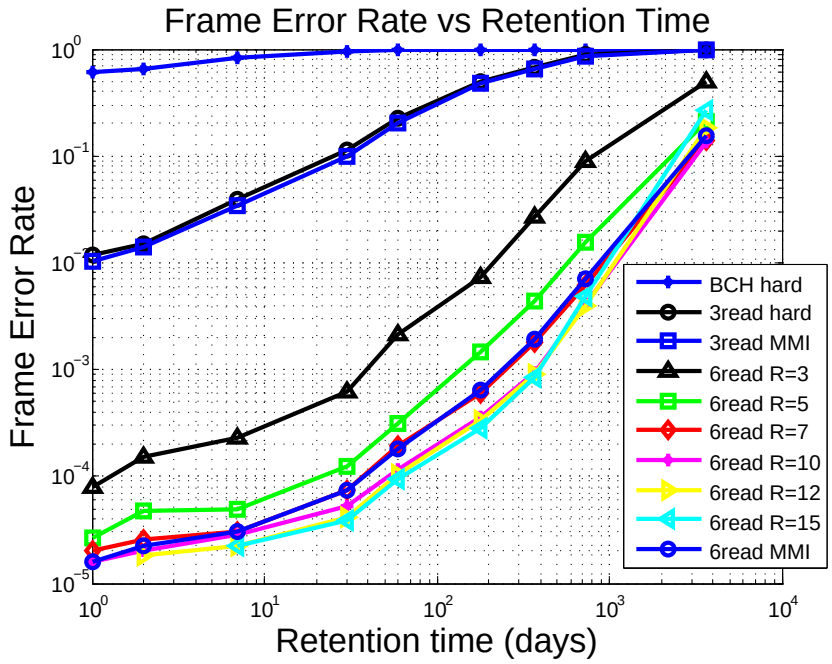


Figure 3.22: Simulation results for 4-level MLC using MMI and constant R with retention data and Code 1.

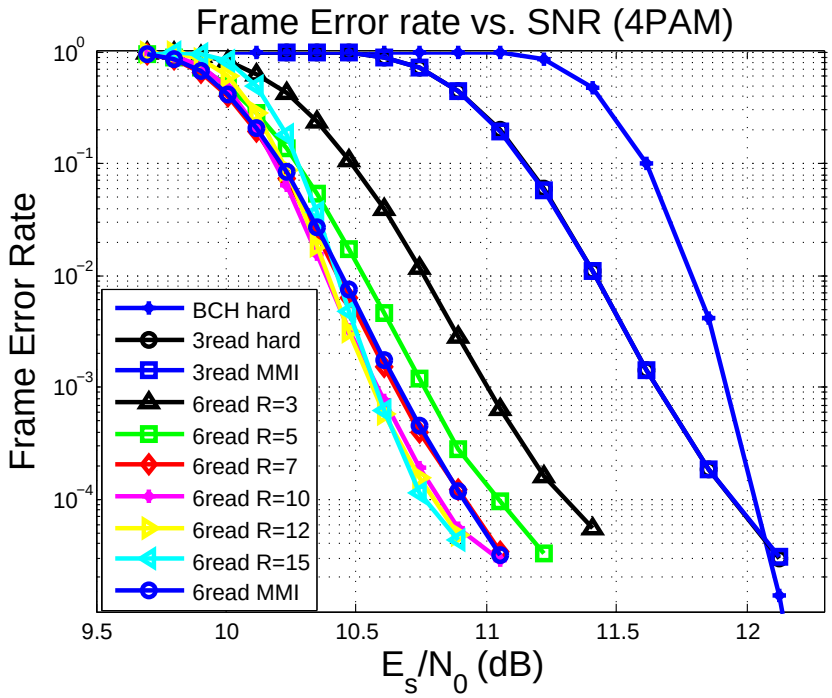


Figure 3.23: Simulation results for 4-level MLC using MMI and constant R with the Gaussian model and Code 1.

3.5 Conclusion

This chapter explores the benefit of using soft information in an LDPC decoder for flash memory. Using a small amount of soft information improves the performance of LDPC codes significantly and demonstrates a clear performance advantage over conventional BCH codes. In order to maximize the performance benefit of the soft information, we present a method that selects the word-line voltages to maximize the MI between the input and output of the equivalent read channel. This method can be applied to any channel model and provides an effective and efficient estimate of the word-line voltages compared to other existing quantization techniques. Possible directions for future research include the design of better high-rate LDPC codes for flash memory, and the analysis of the corresponding error-floor properties.

CHAPTER 4

Conclusion

This dissertation tackles two problems in LDPC codes: how to construct better codes to lower the error floor and how to obtain soft information from flash memory to have better LDPC decoding performance. To explore the first problem, this dissertation proposes the SR-SCB and SSR-SCB code families based on circulant matrices that are suitable for applications operating at low FER levels. A novel approach introduces the cycle consistency matrix to describe each dominant absorbing set structure. This approach establishes the necessary and sufficient conditions for a given absorbing set structure to exist. Our analysis shows that this deterministic method can provably eliminate certain absorbing sets in SCB codes without sacrificing code properties such as girth, rate, and implementation complexity. The software and hardware simulation results support our analysis in various cases and show significant improvement in the low BER region over a range of decoding algorithms. A useful feature of the proposed approach from the implementation standpoint is that it can be easily combined with efficient decoding algorithms for the maximum improvement of the overall performance.

There are still several challenges to be addressed on this topic. The current CCM approach is based on SCB codes. How to extend this approach to other classes of quasi-cyclic LDPC codes or even irregular codes is still an open problem. A possible direction to solve the irregular case is using the mask technique to zero out some circulant matrices in the parity-check matrix so that code properties such as degree distribution, girth and the absorbing set spectrum can be improved.

To tackle the second problem, this dissertation investigates LDPC design and decoding for flash memory. We show that degree distributions optimized for the AWGN channel do not give

the best possible performance in quantized channels. Altering the degree distribution to avoid small absorbing sets can significantly improve performance. We also show that a small amount of soft information improves the performance of LDPC codes significantly and demonstrates a clear performance advantage over conventional BCH codes. For a given number of reads, we propose an MMI approach which selects word-line voltages to maximize the mutual information between the input and output of the equivalent read channel. This method can be applied to any channel model and provides an effective and efficient estimate of the word-line voltages. The constant-ratio method adds a constraint to the MMI approach that simplifies the optimization process.

One direction for future research is an investigation of the mechanism that causes absorbing sets to be more important for quantized channels than unquantized channels. Other areas for future investigation include the design of better high-rate LDPC codes for flash memory, optimization techniques for larger numbers of reads and the analysis of the corresponding error-floor properties.

REFERENCES

[ABA11a] R. Asvadi, A. H. Banihashemi, and M. Ahmadian-Attari. "Design of finite-length irregular protograph codes with low error floors over the binary-input AWGN channel using cyclic liftings." *http://arxiv.org/pdf/1102.0424v1*, Feb. 2011.

[ABA11b] R. Asvadi, A. H. Banihashemi, and M. Ahmadian-Attari. "Lowering the error floor of LDPC codes using cyclic liftings." *IEEE Trans. Inform. Theory*, **57**(4):2213–2224, Apr. 2011.

[BCM03] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti. "Introduction to Flash Memory." *Proc. IEEE*, **91**(4), April 2003.

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon limit errorcorrecting coding and decoding: Turbo codes." In *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993.

[CGW10] A. I. Vila Casado, M. Griot, and R. D. Wesel. "LDPC decoders with informed dynamic scheduling." *IEEE Trans. Comm.*, **58**(12):3470–3479, Dec. 2010.

[CRU01] S. Chung, T. Richardson, and R. Urbanke. "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation." *IEEE Trans. Inform. Theory*, **47**:657–670, Feb. 2001.

[CXD04] L. Chen, J. Xu, I. Djurdjevic, and S. Lin. "Near-Shannon-limit quasi-cyclic low-density parity-check codes." *IEEE Trans. Comm.*, **52**(7):1038–1042, Jul. 2004.

[DM98] M. C. Davey and D. J. C. MacKay. "Low-density parity check codes over GF(q)." *IEEE Commun. Lett.*, **2**:165–167, June. 1998.

[Dol10] L. Dolecek. "On absorbing sets of structured sparse graph codes." In *Proc. Info. Theory and Applications (ITA) Workshop*, San Diego, CA, Feb. 2010.

[DWZ10] L. Dolecek, J. Wang, and Z. Zhang. "Towards improved LDPC code designs using absorbing set spectrum properties." In *Proc. of 6th International symposium on turbo codes and iterative information processing*, Brest, France, Sept. 2010.

[DXZ11] G. Dong, N. Xie, and T. Zhang. "On the Use of Soft-Decision Error-Correcting Codes in NAND Flash Memory." *IEEE Trans. Circ. and Sys.*, **58**(2):429–439, Feb. 2011.

[DZW10] L. Dolecek, Z. Zhang, M. J. Wainwright, V. Anantharam, and B. Nikolic. "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes." *IEEE Trans. Inform. Theory*, **56**(1):6261–6268, Jan. 2010.

[Fan00] J. L. Fan. "Array codes as low-density parity-check codes." In *Proc. of 2nd Int. Symp. on Turbo Codes*, Brest, France, Sept. 2000.

[For01] G. D. Forney. "Codes on graphs: normal realizations." *IEEE Trans. Inform. Theory*, **47**:520–548, Feb. 2001.

[Fos01] M. Fossorier. "Iterative reliability-based decoding of low-density parity-check codes." *IEEE J. Select. Areas Commun.*, **19**:908–917, May. 2001.

[Fos04] M. P. C. Fossorier. "Quasi-cyclic low-density parity-check codes from circulant permutation matrices." *IEEE Trans. Inform. Theory*, **58**(8):1788–1793, Aug. 2004.

[Gal63] R. G. Gallager. *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[HDL11] Q. Huang, Q. Diao, S. Lin, and K. Abdel-Ghaffar. "Cyclic and quasi-cyclic LDPC codes: new developments." In *Proc. Info. Theory and Appl. Workshop*, San Diego, CA, Feb. 2011.

[HEA01] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold. "Progressive edge-growth Tanner graphs." In *Proc. IEEE GLOBECOM*, San Antonio, TX, Feb. 2001.

[ICV08] M. Ivkovic, S. K. Chilappagari, and B. Vasic. "Eliminating trapping sets in low-density parity-check codes by using Tanner graph covers." *IEEE Trans. Inform. Theory*, **54**(8):3763–3768, Aug. 2008.

[KFL01] F. R. Kschischang, B. J. Frey, and H. Loeliger. "Factor graphs and the sumproduct algorithm." *IEEE Trans. Inform. Theory*, **47**:498–519, Feb. 2001.

[KKH09] S. Kim, K. Ko, J. Heo, and J. Kim. "Two-staged informed dynamic scheduling for sequential belief propagation decoding of LDPC codes." *IEEE Comm. Letters*, **13**(3):193–195, Mar. 2009.

[KLF01] Y. Kou, S. Lin, and M. Fossorier. "Low-density parity-check codes based on finite geometries: a rediscovery and new results." *IEEE Trans. Inform. Theory*, **47**:2711–2736, Nov. 2001.

[KY] B. M. Kurkoski and H. Yagi. "Quantization of Binary-Input Discrete Memoryless Channels, with Applications to LDPC Decoding." *Submitted to IEEE Trans. Inform. Theory. Available http://arxiv.org/abs/1107.5637.*

[LHC02] J.-D. Lee, S.-H. Hur, and J.-D. Choi. "Effects of floating-gate interference on NAND flash memory cell operation." *IEEE Electron Device Letters*, **23**(5):264–266, May 2002.

[LHM06] S. Laendner, T. Hehn, O. Milenkovic, and J. Huber. "When does one redundant parity-check equation matter?" In *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, San Francisco, CA, Nov. 2006.

[LLa08] Y. Li, S. Lee, and et al. "A 16 Gb 3b/cell NAND Flash Memory in 56nm With 8MB/s Write Rate." In *Proc. of ISSCC*, pp. 506–632, Feb. 2008.

[LM05] S. Laendner and O. Milenkovic. "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes." In *Proc. of Wireless Comm.*, Honolulu, HI, Jun. 2005.

[LMS01] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman. "Improved low-density parity-check codes using irregular graphs." *IEEE Trans. Inform. Theory*, **47**:585–598, Feb. 2001.

[LT05] J. K.-S. Lee and J. Thorpe. "Memory-Efficient Decoding of LDPC Codes." In *Proc. IEEE Int. Symp. on Info. Theory (ISIT)*, Adelaide, Australia, July 2005.

[LZ10] S. Li and T. Zhang. "Improving Multi-Level NAND Flash Memory Storage Reliability Using Concatenated BCH-TCM Coding." *IEEE Trans. VLSI Systems*, **18**(10):1412–1420, Oct. 2010.

[LZT07] L. Lan, L. Zeng, Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar. "Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: a finite field approach." *IEEE Trans. Inform. Theory*, **53**(7):2429–2458, Jul. 2007.

[Mac99] D. J. C. MacKay. "Good error-correcting codes based on very sparse matrices." *IEEE Trans. Inform. Theory*, **45**:399–431, Mar. 1999.

[MB01] Y. Mao and A. H. Banihashemi. "Decoding low-density parity-check codes with probabilistic scheduling." *IEEE Commun. Lett.*, **5**:414–416, Oct. 2001.

[MH09] Y. Maeda and K. Haruhiko. "Error Control Coding for Multilevel Cell Flash Memories Using Nonbinary Low-Density Parity-Check Codes." In *24th IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems*, Chicago, IL, Oct. 2009.

[MKL06] O. Milenkovic, N. Kashyap, and D. Leyba. "Shortened array codes of large girth." *IEEE Trans. Inform. Theory*, **5**(8):3707–3722, Aug. 2006.

[MS03] M.M. Mansour and N.R. Shanbhag. "High-throughput LDPC decoders." *IEEE Trans. on VLSI Systems*, **1**(6):976 – 996, Dec. 2003.

[NCM10] D. V. Nguyen, S.K. Chilappagari, M. Marcellin, and B. Vasic. "LDPC Codes from Latin squares free of small trapping sets." *http://arxiv.org/pdf/1008.4177*, Aug. 2010.

[NVM10] D. V. Nguyen, B. Vasic, M. Marcellin, and S.K. Chilappagari. "Structured LDPC codes from permutation matrices free of small trapping sets." In *Proc. IEEE Info. Theory Workshop (ITW)*, Dublin, Ireland, Sept. 2010.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Morgan Kaufmann, 1988.

[R 06] R. Diestel. *Graph Theory*. Springer, 2006.

[Ric03] T. Richardson. "Error-floors of LDPC codes." In *Proc. 41st Annual Allerton Conf.*, Monticello, IL, Oct. 2003.

[RSU01] T. Richardson, M. Shokrollahi, and R. Urbanke. "Design of capacity-approaching irregular low-density parity-check codes." *IEEE Trans. Inform. Theory*, **47**(2):616–637, Feb. 2001.

[RU01] T. Richardson and R. Urbanke. "Efficient encoding of low-density parity-check codes." *IEEE Trans. Inform. Theory*, **47**:638–656, Feb. 2001.

[RW04] A. Ramamoorthy and R. D. Wesel. "Construction of Short Block Length Irregular LDPC Codes." In *Proc. IEEE Int. Conf. on Comm. (ICC)*, Paris, France, June. 2004.

[Tan81] R. M. Tanner. "A recursive approach to low complexity codes." *IEEE Trans. Inform. Theory*, **27**:533–547, Sept. 1981.

[TJV04] T. Tian, C. Jones, J. D. Villasenor, and R. D. Wesel. "Selective Avoidance of Cycles in Irregular LDPC Code Construction." *IEEE Trans. Comm.*, **52**(8):1242–1247, Aug. 2004.

[TSa09] C. Trinh, N. Shibata, and et al. "A 5.6MB/s 64 Gb 4b/Cell NAND Flash Memory in 43nm CMOS." In *Proc. of ISSCC*, p. 246, Feb. 2009.

[TSS04] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello. "LDPC block and convolutional codes based on circulant matrices." *IEEE Trans. Inform. Theory*, **50**(15):2966 –2984, Dec. 2004.

[TVW04] T. Tian, J. D. Villasenor, and R. D. Wesel. "Selective avoidance of cycles in irregular LDPC code construction." *IEEE Transactions on Communications,*, **52**(8):1242–1247, Aug. 2004.

[VF06] N. Varnica and M. Fossorier. "Improvements in belief-propagation decoding based on averaging information from decoder and correction of clusters of nodes." *IEEE Comm. Letters*, **10**(12):846–848, Dec. 2006.

[WC07] Z. Wang and Z. Cui. "Low-complexity high-speed decoder design for quasi-cyclic LDPC code." *IEEE Trans. on VLSI Systems*, **15**(1):104 – 114, Jan. 2007.

[WCS11] J. Wang, T.A. Courtade, H. Shankar, and R.D. Wesel. "Soft Information for LDPC Decoding in Flash: Mutual-Information Optimized Quantization." In *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, Houston, TX, Dec. 2011.

[WDW11a] J. Wang, L. Dolecek, and R. D Wesel. "LDPC absorbing sets, the null space of the cycle consistency matrix, and Tanner's constructions." In *Proc. Info. Theory and Appl. Workshop*, San Diego, CA, Feb. 2011.

[WDW11b] J. Wang, L. Dolecek, and R.D. Wesel. "Controlling LDPC absorbing sets via the null space of the cycle consistency matrix." In *Proc. IEEE Int. Conf. on Comm. (ICC)*, Kyoto, Japan, June. 2011.

[WDZ11a] J. Wang, L. Dolecek, Z. Zhang, and R. D Wesel. "Absorbing set spectrum approach for practical code design." In *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, Saint Petersburg, Russia, Jul. 2011.

[WDZ11b] Q. Wu, G. Dong, and T. Zhang. "Exploiting Heat-Accelerated Flash Memory Wear-Out Recovery to Enable Self-Healing SSDs." In *USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage)*, June 2011.

[Wib96] N. Wiberg. *Codes and decoding on general graphs*. Linkping Univertisy, 1996.

[XB04] H. Xiao and A. H. Banihashemi. "Improved progressive-edge-growth (PEG) construction of irregular LDPC codes." *IEEE Communication Letters,*, **8**(12):715–718, Dec. 2004.

[ZDN06] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. Wainwright. "Investigation of error floors of structured low-density parity-check codes by hardware emulation." In *Proc. IEEE Global Comm. Conf (GLOBECOM)*, San Francisco, CA, Nov. 2006.

[ZDN08] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. Wainwright. "Lowering LDPC error floors by postprocessing." In *Proc. IEEE Global Comm. Conf (GLOBECOM)*, New Orleans, LA, Nov. 2008.

[ZDN09] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. J. Wainwright. "Design of LDPC decoders for improved low error rate performance: quantization and algorithm choices." *IEEE Trans. Comm.*, **57**(11):3258–3268, Nov. 2009.

[ZHL10] L. Zhang, Q. Huang, S. Lin, K. A. Ghaffar, and I. Blake. "Quasi-cyclic LDPC codes: an algebraic construction, rank analysis, and codes on Latin squares." *IEEE Trans. Comm.*, **58**(11):3126–3139, Nov. 2010.