The Design of Binary Shaping Filter of Binary Code

by

Shao-Lun Huang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science and Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
February 2009

© Massachusetts Institute of Technology 2009. All rights reserved.

Author .................................................................
Department of Electrical Engineering and Computer Science
January 29, 2010

Certified by ............................................................
Lizhong Zheng
Associate Professor
Thesis Supervisor

Accepted by ...........................................................
Arthur C. Smith
Chairman, Department Committee on Graduate Theses
The Design of Binary Shaping Filter of Binary Code

by

Shao-Lun Huang

Submitted to the Department of Electrical Engineering and Computer Science
on January 29, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

In this thesis, I designed and implemented a compiler which performs optimizations
that reduce the number of low-level floating point operations necessary for a specific
task; this involves the optimization of chains of floating point operations as well
as the implementation of a “fixed” point data type that allows some floating point
operations to be simulated with integer arithmetic. The source language of the compiler
is a subset of C, and the destination language is assembly language for a micro-floating
point CPU. An instruction-level simulator of the CPU was written to allow testing
of the code. A series of test pieces of codes was compiled, both with and without
optimization, to determine how effective these optimizations were.

Thesis Supervisor: Lizhong Zheng
Title: Associate Professor
Acknowledgments

This is the acknowledgements section. You should replace this with your own acknowledgements.
Contents

1 introduction .................................................. 13

2 The Encoding and Decoding Structure ............... 17
   2.1 The Encoder and Decoder ............................ 17
   2.2 Sphere Packing ........................................ 19
   2.3 The Modification of The Encoder ................. 20

A Tables ......................................................... 23

B Figures ........................................................ 25
List of Figures

1-1 The encoding scheme .............................................. 15
1-2 Sphere packing .................................................. 16
2-1 Decoder ......................................................... 18
2-2 Modified encoder ............................................... 21
B-1 Armadillo slaying lawyer. ..................................... 25
B-2 Armadillo eradicating national debt. ...................... 26
List of Tables

A.1 Armadillos ................................................. 23
Chapter 1

introduction

The goal of this paper is to develop an efficient error correcting encoding scheme for generating channel codes with arbitrary distributions.

In information theory, it has been shown that for a given channel, the optimal channel codes should have the empirical distribution matching to the one, which maximizes the mutual information. In addition, because of the necessity in maximizing the conditional mutual information, the basic network information theory results often rely on codes with arbitrary distribution. Particularly, the unbalanced binary codes, which have distribution Bern($p$) where $p \neq \frac{1}{2}$.

However, practical codes including LDPC (low density parity check) codes, and PSK/QAM constellation for AWGN channel are based on Bern($\frac{1}{2}$) distributed code. Therefore, we would like to invent an efficient scheme to generate channel codes with arbitrary distribution.

There exists several ways to do this. First of all, the random code in information theory can generate codes with arbitrary given distribution. Nevertheless, random code can not be used in practice since the encoding and decoding complexity grows exponentially with the codeword block length. Moreover, the block length of the codeword, which is needed to achieve good performance, is usually very long. Another simple way to generate codes with specific distribution is the following. The information bits are first partitioned into small blocks, and then these block are mapped to the codewords through the cumulative function of the specific distribution. This method has
low complexity as long as the partitioned block length is small. Nevertheless, Forney in ?? showed that this kind of concatenated code typically reduces the error exponent in decoding. Thus the block decoding error rate will be increasing. Furthermore, small number of errors in one block of the concatenated code will induce a large number of errors in decoding. Consequently, the increased block error rate implies that group of errors occur in a large number of blocks in the concatenated codes after decoding because of a small number of errors in the received signals. This is a serious deficiency of the concatenated codes.

On the contrary, LDPC codes encode and decode with long block lengths. Since the codewords of block length $n$ are in the $n$ dimensional space $\mathbb{F}_2^n$, compared to concatenated codes, LPDC codes can be thought of as encoding in high dimensions. In particular, the parity check bits are the sum of information bits that spreads out of the whole information bits block. Therefore, LDPC codes can recover the small number of errors occurred in the received signals very well. In addition, these codes all have efficient decoding algorithm thus can be implemented in practical systems. Because of these advantages of LDPC codes, it is natural to develop an encoding scheme based on LDPC codes to generate arbitrarily distributed codes.

In this paper, we would like to invent a novel scheme, which exploits the high dimensional non-linear shaping. The key idea is to first encode the information bits by a linear encoder $C_1$ and to use the decoder of another linear code as the non-linear shaping operator. This is illustrated in Fig. 1-1. In the first stage, the linear code $C_1$ is used to encode the information bits $d$ into the codeword $u$. In the second stage, the codeword $u$ of $C_1$ is decoded by another linear code $C_2$, and the decoding error $x$ is obtained by substracting the decoded codeword $\hat{u}$ to $u$. Finally, we take $x$ as our encoded codeword for the information bits $d$. Particularly, note that $x$ can be viewed as shaping the codeword $u$ of $C_1$ by $C_2$.

There is a geometric way to interpret this encoding scheme, namely, the sphere packing. In Fig. 1-2, suppose that the large outer ball is $\mathbb{F}_2^n$, the vector space of codewords with length $n$. Consider the codewords of $C_2$ and the corresponding maximal likelihood decoding region in the large ball, then the decoding regions are small balls.
packed into the large ball $\mathbb{F}_2^n$, and the codewords are the center of the balls. Since $C_2$ is a linear code, the decoding region of all codewords have the same shape, which is the $n$ dimensional ball. Hence we can map all balls to the one located in the origin. Suppose that the codeword of $C_1$, $u$, is in the ball centered at $\hat{u}$, then the difference of $u$ and $\hat{u}$ is the codeword of our scheme. If we map all of them to the ball centered at the origin, and assume that $\hat{u}$ maps to $x$, then $x$ is our encoded codeword. Therefore, by this mapping, the set of codewords in our scheme is just the points in the ball centered at the origin, which is the decoding region of $\overline{0}$.

![Diagram](image)

Figure 1-1: The encoding scheme
Figure 1-2: Sphere packing
Chapter 2

The Encoding and Decoding Structure

In the following of this paper, we assume that that all the channel models are binary and the information bits are independent and identity distributed (i.i.d.) $\text{Bern}(\frac{1}{2})$. Our goal is to generate codewords with probability distribution of $\text{Bern}(p)$ for some $p < \frac{1}{2}$. In this section, an efficient encoding and decoding structure, which achieves this goal will be introduced. The information bits is first encoded by a linear code $C_1$, and then the encoded cordword is decoded by another linear code $C_2$. The decoding error $x$ is treated as our encoded codeword of our encoding scheme, which can be viewed as shaped by $C_2$. Moreover, we modify the original encoder to a new form, which facilitates the designs and implementations. The geometric interpretation of our encoding scheme will also be presented based on the sphere packing picture.

2.1 The Encoder and Decoder

The information bits $d$ is assumed to be a $k$ vector with i.i.d. $\text{Bern}(\frac{1}{2})$ elements. In order to encode $d$, we first encode $d$ by a linear code $C_1$ with an $n \times k$ generating matrix $G_1$ and an $(n - k) \times n$ parity check matrix $H_1$. Let the $n$ vector $u$ be the encoded codeword of $d$ by codebook $C_1$, that is, $u = G_1d$. The second step is to decode $u$ by another linear code $C_2$ with an $n \times (n - n_2)$ generating matrix $G_2$ and an $n_2 \times n$. 


parity check matrix $H_2$. The decoded codeword of $\tilde{u}$ by codebook $C_2$ is $\hat{u}$. Finally, the decoding error

$$x = \hat{u} - u,$$  \hspace{1cm} (2.1)

is our encoded codeword of $d$ through our encoding scheme. The block diagram of the encoder is shown in Fig. 1-1, and note that since $x$ is an $n$ vector, the rate of this channel code is $r = \frac{k}{n}$.

\[\begin{array}{c}
\tilde{x} \\
\text{Multiplication} \\
H_2 \\
\tilde{s} \\
\text{Decoding} \\
C_3 : \{G_3 = H_2G_1, H_3\} \\
\hat{d}
\end{array}\]

Figure 2-1: Decoder

After passing the encoded codeword $x$ through the channel, we receive $\tilde{x}$, which is the noisy version of $x$. To decode $d$ from $\tilde{x}$, one would try reversing the procedure in the encoder, which first map $\tilde{x}$ back to an estimate $\tilde{u}$ of $u$, and then decode $\tilde{u}$ to $\tilde{d}$ by linear code $C_1$. However, there is no bijective map between $x$ and $u$. In fact, many different $u$'s can give rise to the same $x$. To solve the problem, observe that

$$H_2\tilde{x} = H_2u = s,$$ \hspace{1cm} (2.2)

is the syndrome of $u$ to the linear code $C_2$. Moreover, we have the relation

$$G_1d = u,$$ \hspace{1cm} (2.3)

Combining these two equation, we have the following,

$$s = H_2\tilde{x} = H_2G_1d.$$ \hspace{1cm} (2.4)

If we define the linear code $C_3$ such that its generator matrix is $G_3 \triangleq H_2G_1$, then the decoder can be described as follows. First we multiply $\tilde{x}$ by $H_2$, which gives an
estimate $\tilde{s}$ of the syndrome $s$, and then decode $\tilde{s}$ by the codebook $C_3$. The decoded codeword $\tilde{d}$ of $C_3$ is then the estimate of $d$ by the decoder. The decoder structure is shown in Fig. 2-1.

Note that in order to guarantee that the linear code $C_3$ is a valid channel code, since its generating matrix $G_3$ is an $n_2 \times k$ matrix, we have the constraint

$$n_2 > k. \quad (2.5)$$

Because the generating and parity check matrix of good linear channel codes are full rank, we can assume that $G_1$ and $H_2$ are full rank. Hence from (2.4), there is a surjective map between $s$ and $d$. By the syndrom decoding, the map between $s$ and $x$ is bijective, thus there is a surjective map between $x$ and $d$. This indicates that under the constraint, $n_2 > k$, the encoder in Fig. 1-1 is a valid channel encoder.

### 2.2 Sphere Packing

The encoding scheme in Fig. 1-1 can be interpreted geometrically by the sphere packing picture. In Fig. 1-2, assume the large outer ball is the $n$ dimensional vector space $F_2^n$, which is centered at the zero vector $0$. Every binary $n$ vector is a point in this large ball. Consider the codewords of $C_2$ and the corresponding maximal likelihood decoding regions in the large ball. Since the rate is smaller than one, the decoding regions can be visualized as small $n$ dimensiona balls packed into the large ball $F_2^n$. Moreover, each decoding region ball is centered at the corresponding codeword.

Let us consider the codeword $u$ of $C_1$ in this picture. Since $u$ is decoded to $\hat{u}$, $u$ has to locate in the decoding region of $\hat{u}$. Therefore $u$ must be in the small ball centered at $\hat{u}$.

Since $C_2$ is a linear code, the shape and volume of all small balls are the same, there is a bijective map which maps every small ball to the one centered at the origin, the zero vector $0$. Assume that by this map, $\hat{u}$ and $u$ are mapped to $0$ and $x$ correspondingly.
Then since
\[ x = x - 0 = u - \hat{u}, \]  \hspace{1cm} (2.6)
we know that \( x \) is just the codeword of our encoding scheme in Fig. 1-1. Therefore, the codewords of our encoding scheme can be viewed as the points in the small ball centered at origin, namely, the maximal likelihood decoding region of the zero codeword \( 0 \).

Note that since the decoding regions of the codewords are not vector subspaces of \( \mathbb{F}_2^n \), the code by our encoding scheme is not a linear code. Furthermore, the condition \( n_2 > k \) derived in (2.5) guarantees that no two different codeword \( u \) and \( u' \) of \( C_1 \) map to the same point in the small ball centered at \( 0 \). This implies the validity of our encoding scheme.

Suppose that the codewords we generated have the distribution Bern(\( p \)). Since the set of codewords are a subset of the decoding region, the volume of the decoding region is at most \( 2^{nH(p)} \), where \( H(\cdot) \) is the entropy. Moreover, because the information bits are \( k \) vectors, we know that the volume of the set of possible codewords is \( 2^k \). Note that the rate \( r = \frac{k}{n} \), we have
\[ k \leq nH(p) \Rightarrow \frac{k}{n} \leq H(p) \Rightarrow r \leq H(p) \]  \hspace{1cm} (2.7)
This is the fundamental restriction of the rate and the distribution of the codewords.

### 2.3 The Modification of The Encoder

When implementing the encoder and decoder in Fig. 1-1 and Fig. 2-1 to communication systems, the linear codes \( C_1, C_2, \) and \( C_3 \) should be chosen such that the channel codewords have good distance distribution as well as efficient encoding and decoding algorithm. In order to achieve both properties, the linear code \( C_2 \) and \( C_3 \) have to be chosen as good linear codes with efficient decoding algorithm such as convolutional code and LDPC code. Since \( G_3 = H_2G_1 \), the linear code \( C_1 \) can be determined by \( C_2 \)

*Throughout this paper, the entropy is always assumed to be base 2*
and $C_3$. Nevertheless, because $G_1$ is determined by $H_2$ and $G_3$, even though there are efficient algorithms in encoding and decoding for $C_2$ and $C_3$, $C_1$ may still not have an efficient encoding algorithm. This increases the total encoding complexity in Fig. 1-1. Due to the fact that we do not have the freedom in choosing $C_1$ arbitrarily, it motivates us to transform $C_1$ to a simple combination of $C_2$ and $C_3$, and makes the whole encoding scheme invariant after this transformation.

Observe that in the encoder in Fig. 1-1, we can split out the decoding process of $C_2$ to two parts: first multiplies $u$ by $H_2$ to calculate the syndrome, and then do the syndrome decoding. This is shown in the upper diagram of Fig. 2-2, here we substitute the subscript of $C_2$, $G_2$, and $H_2$ from "2" to "s"\(^\dagger\). Then we can associate $G_1$ and $H_s$ and give birth to a new linear code $C$ with generating matrix

$$G = H_s G_1,$$

(2.8)

which encodes $d$ to the syndrome $s$. Because of $G_3 = H_2 G_1$, $G$ is identical to $G_3$. Therefore the new encoding scheme in the lower part of Fig. 2-2 is essentially the

\(\dagger\)The "s" represents “shaping”, since $C_s$ acts as a shape operator to the codeword $u$ of $C_1$. 

Figure 2-2: Modified encoder
reverse of the original decoder. Thus in order to implement this new encoding and decoding scheme, we only need to choose two good code $C$ and $C_s$ such that they can be encoded and decoded easily, and $C_s$ has efficient syndrome decoding algorithm. In next section, we will consider practical issues of implementing this scheme.
Appendix A

Tables

Table A.1: Armadillos

<table>
<thead>
<tr>
<th>Armadillos</th>
<th>are</th>
<th>our</th>
<th>friends</th>
</tr>
</thead>
</table>
Appendix B

Figures

Figure B-1: Armadillo slaying lawyer.
Figure B-2: Armadillo eradicating national debt.