Design and VLSI Implementation of a Convolutional Encoder and Majority Logic Decoder for Forward Error Correction in Intrabuilding Power Line Communications

by

David Friedman

B. Sc. Computer Eng., Kiev Polytechnical Institute, 1986
M. Sc. Computer Eng., Kiev Polytechnical Institute, 1986

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in
THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF ELECTRICAL ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

September 1992

© David Friedman, 1992.
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature)

Department of Elect. Eng.

The University of British Columbia
Vancouver, Canada

Date Sept 23, 1992.
Abstract

The need for simple and effective forward error correction (FEC) schemes for use in modern low-cost communication systems continues, especially for intrabuilding power line (IPL) communications. In particular, the use of short random error correcting convolutional codes with a moderate degree of interleaving has been shown to be a viable FEC option that enhances the performance of communications over the power line channel.

In this thesis, a VLSI convolutional encoder and threshold decoder (codec) chip for use in intrabuilding power line communications was successfully designed, fabricated and tested. The chip implements the rate 1/2 (2, 1, 6) self-orthogonal convolutional code together with programmable degrees of interleaving (λ = 1, 3, 5, 7). This code provides random and burst error correcting capabilities. Threshold or majority–logic decoding was selected as the algorithm to be used, due to the ease of its implementation and its appropriateness for burst error channels. Interleaving and deinterleaving can easily be included in the encoder and decoder, respectively.

The codec is a semi-custom design that uses standard library cells, and was fabricated using 1.2 μm CMOS technology. A maximum throughput of 50 Mbps is feasible. Built-In Self-Test was incorporated into the design providing 100% single stuck-at fault coverage, with under 10% of silicon area overhead.

To evaluate the effectiveness of the VLSI codec, the chip was integrated onto two existing power line modems. Performance tests were completed for coded and uncoded data transmissions under conditions of varying channel quality. This study successfully demonstrates that the use of this FEC chip is effective in increasing the error–free throughput of intrabuilding power line communications. The chip provides the possibility of maintaining reliable communications over channels that could otherwise not be used for data communications at 19.2 Kbps.
Contents

Abstract ................................................................. ii
List of Figures .......................................................... v
Acknowledgments ....................................................... viii

Chapter 1 Introduction .................................................. 1
Section 1 Background and Motivation ............................... 1
Section 2 Thesis Outline ............................................... 4

Chapter 2 Forward Error Correction Coding in Power Line Communications .................................................. 7
Section 1 Data Communication Systems ........................... 7
Section 2 Intrabuilding Power Distribution Circuits ............... 9
Topic 1 Building Wiring Plans ....................................... 10

Chapter 3 Convolutional Coding for Power Line Data Communications .................................................. 21
Section 1 Encoding of Convolutional Codes ....................... 21
Section 2 Decoding of Convolutional Codes ....................... 23
Section 3 Self-Orthogonal (2, 1, 6) Code ......................... 26
Section 4 Properties and Performance of Threshold Decoding .... 30
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>General Description</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Encoder</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Decoder</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Support Blocks</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Built-In Self-Test</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Test Procedure</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>Design Verification</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Codec Performance Tests</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>System Level Implementation</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Test Parameters</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Performance in Controlled Environments</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>AWGN Channel</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Burst Error Channel</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>Performance over the Power-Line Channel</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>Local Area Communications</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>Extensive Area Communications over the General Sub-System and across Sub-Systems</td>
<td>72</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>Summary</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Concluding Remarks</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Suggestions for Future Research</td>
<td>86</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td></td>
<td>88</td>
</tr>
<tr>
<td>Appendix A</td>
<td></td>
<td>Codec Schematic and Layout Representations</td>
<td>93</td>
</tr>
<tr>
<td>Appendix B</td>
<td></td>
<td>Codec Bonding Diagrams</td>
<td>99</td>
</tr>
<tr>
<td>Appendix C</td>
<td></td>
<td>Enhanced PLM Schematic Diagrams</td>
<td>103</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Block diagram of a data communication system.</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Block diagram of a simplified communication system.</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Residential power delivery scheme.</td>
<td>11</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Commercial/Industrial three-phase power delivery scheme.</td>
<td>12</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Block diagram of a (2, 1, 6) convolutional encoder.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Block diagram of a (2, 1, 6) threshold decoder.</td>
<td>29</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Syndrome register (feedback links marked a-d)</td>
<td>31</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Schematic diagram of a (2, 1, 6) convolutional encoder.</td>
<td>38</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Schematic diagram of a (2, 1, 6) majority-logic decoder.</td>
<td>40</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Variable interleaving implementation</td>
<td>43</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Schematic diagram of 7Cell block.</td>
<td>44</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Schematic diagram of 3Cell block.</td>
<td>45</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Built-In Self-Test schematics.</td>
<td>47</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Built-In Self-Test simulation (without faults in the CUT).</td>
<td>49</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Built-In Self-Test simulation (with a single stuck-at fault in the decoder).</td>
<td>51</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Example of STL program.</td>
<td>53</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Packet Format</td>
<td>57</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Enhanced modem performance for AWGN channel: (a) BER, (b) BLKER.</td>
<td>63</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Channel BER vs (a) Decoded BER, and (b) BLKER for AWGN Channel.</td>
<td>65</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Channel BER vs Decoded BER (a), and BLKER (b) for a Bursty channel.</td>
<td>67</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Channel BER vs Decoded BER (a), and BLKER (b) for X to Y channel.</td>
<td>69</td>
</tr>
</tbody>
</table>
Figure 22 Channel BER vs Decoded BER (a), and BLKER (b) for Y to Y channel. ............................................................... 70

Figure 23 Channel BER vs Decoded BER (a), and BLKER (b) for Z to Y channel. ............................................................... 71

Figure 24 BER as a function of time for transmissions from Rm 458 to Rm 214. (a) Transmissions over the General Power Sub-system, (b) Transmissions across Power Sub-systems. . 74

Figure 25 Throughput as a function of time for transmissions from Rm 458 to Rm 214. (a) Transmissions over the General Power Sub-system, (b) Transmissions across Power Sub-systems. . 75

Figure 26 BER as a function of time for transmissions from Rm 458 to the 3rd floor stairwell. (a) Transmissions over the General Power Sub-system, (b) Transmissions across Power Sub-systems. . 76

Figure 27 Throughput as a function of time for transmissions from Rm 458 to the 3rd floor stairwell. (a) Transmissions over the General Power Sub-system, (b) Transmissions across Power Sub-systems. . 77

Figure 28 Average BLKER (over 24 hours) for each of the tested channels. ............................................................... 79

Figure 29 Schematic diagram of the (2,1,6) convolutional encoder and threshold decoder with fixed interleaving degree ($\lambda=7$). ...... 93

Figure 30 Layout representation of the (2,1,6) convolutional encoder and threshold decoder with fixed interleaving degree ($\lambda=7$). Design identification name BCCOD. (Only the metal layers are shown). ............................................................... 94

Figure 31 Schematic diagram of the (2,1,6) convolutional encoder and threshold decoder with programmable interleaving degrees ($\lambda=1, 3, 5, 7$). ............................................................... 95
| Figure 32 | Layout representation of the (2,1,6) convolutional encoder and threshold decoder with programmable interleaving degrees ($\lambda=1, 3, 5, 7$). Design identification name BCMAJ. (Only the metal layers are shown). ............................................. 96 |
| Figure 33 | Schematic diagram of the (2,1,6) convolutional encoder and threshold decoder with programmable interleaving degrees ($\lambda=1, 3, 5, 7$) and BIST. ............................................. 97 |
| Figure 34 | Layout representation of the (2,1,6) convolutional encoder and threshold decoder with programmable interleaving degrees ($\lambda=1, 3, 5, 7$) and BIST. Design identification name BCBST. (Only the metal layers are shown). ............................................. 98 |
| Figure 35 | 68–Pin PGA (a) pin numbering scheme, and (b) bonding diagram. ............................................. 99 |
| Figure 36 | BCCOD bonding diagram. ............................................. 100 |
| Figure 37 | BCMAJ bonding diagram. ............................................. 101 |
| Figure 38 | BCBST bonding diagram. ............................................. 102 |
Acknowledgments

I would like to express my sincere gratitude to my supervisors Dr. R. W. Donaldson, and Dr. A. Ivanov for their support, guidance and encouragement throughout the course of this research.

I am very grateful to Dave Gagne for his support during the initial stages of the chip's design. I would also like to thank Barry Buternowsky for his assistance in the construction and testing of the enhanced modem prototypes, as well as for his helpful suggestions and enlightening discussions.
Chapter 1
Introduction

1.1 Background and Motivation

Throughout the previous decade the utilization and application of computer equipment has become increasingly widespread, causing an increase in the demand for more efficient and reliable communication technologies. This trend has affected the development of Local Area Networks (LANs) used for factory and office automation, smart homes, data communications, remote sensing, security and access control, and energy management.

The common element of all these network systems is the need for intrabuilding computer communications, for which different types of physical communication media can be used. Such media includes twisted pair wires, coaxial cables, infrared links and fiber optic lines. However, the practical application of these networks has been limited to a great extent by the costs, inconveniences, time and complexities of installing communications wiring in existing buildings and production sites. Scheduling delays, dislocations during installation, preservation of building appearances, and the logistics of allocating installation personnel are a few of the problems encountered. Furthermore, with installed cabling the network has limited flexibility, reconfigurability, and expandability [1].

The use of the already installed and virtually universal power lines as a communication media has long been recognized as a possible alternative to a dedicated channel. The evident advantages include:

1. Presence of an already existing and widespread physical communication media in the form of power electric distribution lines at no incremental cost.
2. Easily accessible interface in the form of a standard wall plug.
3. Universal coverage over the entire building.
4. Reconfigurable, expandable and flexible power line LANs have no need for time consuming and costly rewiring.
Several commercial power line communication systems have been developed recently, including modems from Signetics, National Semiconductor, NONWIRE, BSR and ExpertNet. However, the performance of these commercial systems is still limited, even though the demand for more efficient and reliable systems steadily grows.

Power lines were not designed as a communication media and provide an inhospitable environment for data communications; noise and frequency dependent signal attenuation are common impairments. At low data rates, power line impulse noise is relatively small; energy per data bit is high; and, reliable performance is achievable with low transmitter power levels. As transmission speed increases the detrimental effects of impulse noise and fading similarly increase. At data rates above 4.8 Kbps impulse noise tends to become a dominant factor in determining the performance of the communication channel [2].

Increases in transmission speeds of power line modems generates a series of new problems. The most significant of these is a steady growth in the Bit Error Rate (BER) of the transmitted data. It has been shown in previous studies [2] that an effective way to lower the BER while maintaining high data rates is the use of a method which detects and corrects errors which occur during data transmissions due to the inevitable impairments of the power line channel. Such a technique is termed Forward Error Correction (FEC).

Much work has been done to study and characterize the power line channel; some of this work has been used as a platform for this thesis. Chan and Donaldson showed in [2, 3], that rate 1/2 convolutional codes are effective and robust codes for the power line channel (PLC). Specifically, the (2,1,6) self-orthogonal convolutional code with a moderate degree of interleaving has been shown to be a viable FEC option that will reduce the adverse effects of using the PLC as a communication channel. However, in some very noisy environments (channel BER>10^{-2}) the use of FEC is not always sufficient to ensure reliable communications. In such cases FEC is used together with packet retransmission and code combining which enable reliable communications even over very hostile channels [4].

The general structure of convolutional codes is relatively straight forward and the
Encoders are simple to implement. However, the decoding of such codes is a much more complex procedure. Viterbi [5] decoders must perform $2^k$ computations (where $k$ is the number of inputs of the encoder) for each bit that is to be decoded, in order to maintain high data rates the complexity and cost of these decoders grows exponentially with $k$. For this reason it is infeasible for use in low cost applications such as power line modems. In such cases an alternative decoding scheme, such as majority logic decoding (MLD) may be employed, providing a less complex and cost-effective solution (where cost is primarily measured by the gate count of the design). Majority logic or threshold decoding, developed by Massey [6, 7], decodes the value of each bit according to the data of only one constraint length. This limitation considerably simplifies the decoding procedure, while minimizing the coding gain penalty.

After system performance, the most important considerations when developing power line communication equipment is cost, complexity, reliability, and size. The decision to include FEC depends primarily on the affordability of the different schemes under discussion, since the overall cost of a high speed power line modem should not exceed a few hundred dollars to be considered a commercially attractive solution. Threshold decoding, because of its simplified decoding procedure, is an excellent candidate for a low cost power line communications FEC scheme.

Due to practical limitations regarding cost, complexity, reliability, and size, the use of FEC in existing PLMs has been limited to prototype devices. FEC on these prototypes has been successfully implemented using primarily medium scale integration (MSI) integrated circuits (ICs), but cannot be considered a viable solution due to its large size and low reliability. The need of a convolutional encoder and threshold decoder with variable degrees of interleaving, all on one chip, is clear. The design and implementation of such a chip, using very large scale integration (VLSI) technology, is motivated by the following factors:

1. Convolutional encoder / majority-logic decoder chips (codecs) are not commonly available and existing Viterbi codecs are overly expensive for this type of application, costing $20$ to $100$ per chip.
2. Commercially available convolutional codecs do not have variable interleaving capabilities integrated on the chips.

3. The decoding algorithm is very similar to the encoding process. Because of this similarity certain structures are repeated in both blocks, making threshold decoding an attractive alternative for VLSI implementation.

4. The number chips needed to implement the FEC scheme using MSI technology is exceedingly high (over 50 ICs), making the dimensions of the complete PLM to large for use in many applications. The reliability of the completed device also decreases due to the large number of ICs and corresponding interconnections. The use of a single chip VLSI codec provides minimum size and maximum reliability.

5. The use of VLSI technology enables further integration of the PLM. Future designs may integrate the entire modem (such as the modem implemented by Lee in [8]) and FEC modules onto a single chip.

The testing of complex VLSI circuits is currently one of the most pressing problems in VLSI design, due to the complexity of this task. Because of this fact and to comply to a rapidly growing industry wide standard testing technique, Built-In Self-Test (BIST) was incorporated into the design. BIST has become one of the most popular design for testability (DFT) techniques, due to the fact, that it allows thorough testing of VLSI circuits at reasonable cost, since the chip overhead due to BIST can be minimized by a proper choice of the implementation techniques. The inclusion of BIST on the codec will also be of great importance in future designs, when the entire modem and FEC modules are integrated onto one chip, which could then have separate BIST circuitry for each of its functional blocks.

1.2 Thesis Outline

The purpose of this thesis is to design, implement and evaluate a VLSI convolutional encoder and threshold decoder for forward error correction on intrabuilding power line communication channels. Design simplicity and the use of reliable low cost solutions are of concern and are greatly emphasized.
The coder/decoder (codec) circuit was designed and fabricated using 1.2 \, \mu m \, CMOS technology, and was evaluated on a variety of different power line communication environments with transmission rates up to 19.2 Kbps. The data collected from the tests shows the benefits of using FEC coding to combat power line channel impairments.

After this introductory chapter, this thesis is organized as follows:

Chapter 2 describes the power line communication channel, the importance and potential benefits obtainable using FEC coding, as well as possible difficulties that can arise when codes that do not match the channel conditions or requirements are used. A summary of previous work (by others) comparing the performance of various random and burst error correcting codes used on the power line channel (PLC) is presented.

Chapter 3 presents several general considerations and comparisons regarding convolutional encoding and decoding and the use of interleaving in such cases, as well as a detailed theoretical description of the \((2,1,6)\) self-orthogonal convolutional code and its majority-logic decoding. Specific aspects are emphasized, such as the effective constraint length and the maximum error-correcting capability of the code. These aspects are important for understanding some of the design solutions and the BIST scheme implemented.

Chapter 4 describes the codec hardware design and VLSI implementation. A general description of each of the functional blocks, as well as a detailed overview of each of the hardware blocks is provided. Some particular codec features are presented and discussed, including the variable degree of interleaving and the BIST scheme employed. Also described are the test and verification processes involved in the different steps of VLSI chip design and fabrication. These processes include design simulation during the pre-fabrication period and different post-fabrication tests performed on the codec using specialized automated test equipment (ATE) and bit error analyzers.

Chapter 5 details the hardware aspects related to the interfacing of the codec to the power line modem. Actual performance results obtained from a series of tests completed during power line communications with and without the use of FEC are summarized. Several different parameters were recorded in each case such as bit error rate (BER), block error rate (BLER), percent of lost packets and throughput (C). The tests were
repeated while varying the following: (1) degrees of interleaving, (2) channel conditions, (3) period of the day, and (4) transmitter power level.

Chapter 6 provides a summary of the research results obtained. Details of the codec chip are given, which show that the VLSI implementation of the (2,1,6) convolutional encoder and threshold decoder with variable degrees of interleaving is 10 to 50 times more efficient, with respect to silicon area, than other codecs of similar characteristics. The performance results demonstrate that the utilization of suitable, inexpensive FEC coding can foster efficient and reliable data transmission over otherwise unreliable power line channels. Suggestions for further research are also included in this chapter.
Chapter 2
Forward Error Correction Coding in Power Line Communications

The theory and practice of error-correction coding is concerned with the protection of digital information against errors that occur during data transmission. Error detecting and correcting techniques have a broad range of capabilities and are widely used in a series of diverse and important applications. The recurrent problem with any high speed data communication system is providing a method to control the errors that occur during data transmission through a noisy channel. In order to achieve reliable communications effective codes and efficient decoding algorithms must be developed and implemented. The use of error control techniques has grown substantially in the last decade and will continue to grow considerably with the use of anticipated advances in integrated circuit technology.

2.1 Data Communication Systems

In a communication system data is transmitted from an information source through a physical channel to a destination. Because a channel is normally subject to different types of noise, distortion, and interference, the output of the channel often differs from its input due to the errors that result from impaired transmissions. The major concern in designing FEC codes is the control of errors such that reliable reproduction of information can be obtained.

To efficiently design and implement FEC codes a well-known model of data communication systems is used extensively. The block diagram in figure 1 illustrates the basic system elements [9].

Since the focus of FEC techniques are the channel encoder and channel decoder a simplified model of a coded data transmission system is generally used for their analysis. Such a model is represented in figure 2. Within this model the information source and source encoder are combined into a digital source with output u; the modulator, channel
and demodulator are combined into a coding channel with input $v$ and output $r$; and, the source decoder and destination are combined into a digital sink with input $\hat{u}$. Quantities $u$, $v$, $e$, $r$, and $\hat{u}$ may be either scalars or vectors.

To properly design and implement an encoder/decoder pair, as we have set out to do in this thesis, the following issues must be considered: information must be transmitted in a noisy environment at high speeds and with low overhead; reliable reproduction of the information should be obtainable at the decoder output; and, the complexity of the codec should be kept as low as possible.

One of the requirements in the design of a communication system is extensive knowledge of the characteristics of the communication channel. Signal attenuation and channel noise behavior must be understood to obtain high performance. Such
understanding is needed, particularly in designing or selecting FEC coding schemes. The use of an inappropriate code that does not match the channel conditions and impairments will result in inefficient or even incorrect error handling. The result is an unsatisfactory system error rate, which causes unacceptable delays and consequent degradation of system throughput.

2.2 Intrabuilding Power Distribution Circuits

The way electrical power is distributed within a building is an essential point that must be understood in order to properly design systems that can effectively make use of the intrabuilding power lines as a communication channel.

The network that distributes electric power from the generating plants to the end consumers is a large and complex structure. Three-phase high voltage lines transmit
power from generating plants to the substations, from where transmission to distribution transformers occurs [10]. The secondary side of the distribution transformer connects to circuit panels in the buildings. A single distribution transformer may provide power for several residential buildings, or for single buildings in industrial/commercial sectors. From a circuit panel power is delivered to electrical loads via intrabuilding branch circuits [2].

2.2.1 Building Wiring Plans

In residential housing or apartment units, the secondary side of the distribution transformer delivers split-single-phase power to circuit panels by two 120 V lines 180° out of phase and a neutral conductor as shown in figure 3. The neutral conductor is normally connected to the grounded circuit panel. Electrical power is distributed throughout the building on general purpose branch circuits which consisting of a 120 V line and a neutral conductor deliver power to small loads using standard wall plug-in sockets.

Large appliances, such as refrigerators, washers, freezers, and dishwashers, are connected to the distribution system using a special individual dedicated branch circuit. Appliances with large heating elements such as electric stoves, water heaters and dryers usually require a 240 V branch circuit. These appliances are connected to both 120 V line, and the neutral [2].

Commercial and industrial buildings are typically supplied with three-phase electrical power as shown in figure 4. In large buildings, each floor or floor group may be supplied by separate three-phase transformers. Standard branch circuits consisting of a 120 V and neutral line supply small loads. Larger loads are supplied by circuits that deliver either single-phase or polyphase power [3].

On each phase, the hot wire directly connects all loads. In addition, electric loads provide signal transmission paths between a 120 V line and a neutral line. Loads connected across two 120 V lines provide signal transmission paths from one phase to another, together with the signal transmission path provided by the inherent capacitive coupling across the supply transformer secondary.
Figure 3 Residential power delivery scheme.
Figure 4 Commercial/Industrial three-phase power delivery scheme.
All the electrical components and loads connected to a power distribution circuit for part of the PLC, and affect signal transmission. Altogether, the differences in physical distance between nodes, together with constantly varying electrical loads causes the power line network to be a hostile and complex communication environment [3].

2.3 Intrabuilding Power Line Communication Channel Characteristics

It has been demonstrated through a series of commercial and industrial applications [11–13] that power lines may be used as an efficient and reliable communication channels. However, power line channels remain as an inhospitable environment for data communications. Substantial noise and frequency attenuation are found on most power lines. Without well–designed error control coding means, bit errors in power line environments occur at unacceptably high rates. Hence, the actual throughput could represent a fraction of the raw data rate [1].

2.3.1 Power Line Error Sources

To encourage successful use of PLC channels, extensive studies of its characteristics have been performed [2, 14, 15]. These studies have shown that the signal transmission characteristic and noise structure can be extremely complex, highly variable, and load dependent. Communication signals on power lines are subject to channel attenuation, distortion, intersymbol interference, frequency and time fading, and both impulse and random noise. Of these error sources impulse noise is the most dominant.

Attenuation is the amount by which a signal decreases in amplitude during transmission. The number and type of loads attached to any power line circuit will vary over time; consequently, the signal attenuation will be highly variable and unpredictable. Such variations can exceed 20dB [2], even when the transmitter and receiver use the same phase. These attenuation values tend to be higher if the transmitter and receiver are not on the same phase, and tend to increase on long transmission paths with many loads. Attenuation is frequency dependent and increases with frequency, although such increases are not always monotonic. The signal may also exhibit periodic 6 Hz and 120
Hz fades that will cause periodic degradation in received signal to noise ratio which in turn causes periodic burst errors in digital transmissions [2].

Variations in signal attenuation indicate that a signal transmission level which is adequate for a PLC with moderate attenuation may not be sufficient for networks with large attenuation. The use of the “worst case” approach is not a practical solution, since in some cases, transmission may be subject to large attenuation of up to 50 dB [3]. Large transmitter power levels to combat high attenuation would bring attached increases in cost, power consumption, and interference.

There are a series of approaches to solve this problem which include: application of suitable error control coding, use of appropriate data link protocols, use of repeaters in LANs with severe attenuation, and signal bypassing between different power phases of the power system.

Because attenuation may greatly vary on different links of a power line LAN, the bit error rate (BER) performance for such links will also vary. Some links may have relatively good BER values while others can have very poor performance at any given moment.

In contrast to attenuation, noise tends to decrease as frequency increases, since noise levels together with the data signals show increased attenuation with frequency; noise sources nearest to a receiver are a primary cause of bit errors [11]. Consequently the choice of signalling frequency is a compromise between a relatively high noise level with low signal attenuation and a high signal attenuation with reduced noise power.

**Noise.** Power line noise contains both background and impulse components. Such noise requires an adequate description if we are to counteract its detrimental effects on data transmissions. Previous studies contain estimates of the probability distributions for the amplitude, width, and interarrival time of noise impulses [2], and noise spectral density estimates [16]. Such knowledge is essential in the design of effective error control codes and data link protocols.

Power line noise is composed of continuous, relatively low-level background noise punctuated by relatively strong noise impulses. The impulse noise characteristics are of primary concern since this noise is the major cause of channel errors.
Noise impulses are typically of more than 10 dB above the background noise level and in some cases can exceed 40 dB. The strength varies according to the noise source and its proximity to the receiver. The frequency for the dominant impulse train is 120 Hz in synchronism with the cycles of the 60 Hz power voltage. In some cases, impulses will occur in proximity to each other, forming long error bursts [3].

In addition to periodic impulse noise, high-level random noise impulses can result from random load switching, such as thermostat switching, lightning strikes, and switching of power factor correction capacitors. The effect of sources close to the receiver will have greater effect on the received noise structure due to the fact that noise impulses also suffer from attenuation.

The presence of periodic impulse noise on the PLC can cause unacceptably high bit error rates that may seriously affect the system throughput since a large portion of the transmitted packets would be subject to retransmission. Actual values of the bit error probability during noise impulses depend on the impulse amplitudes and widths relative to the received signal level and data bit duration.

To overcome the harmful effects of impulse noise, various methods have been developed which involve the use of adaptive, on-line filtering techniques. The success of such methods greatly depends on the continuous accurate estimation of the fluctuating power line noise. The work done using this approach, shows the obtained BER to be in the order of 10^{-3} even for low data rates under 1 Kbps [17]. Other methods are based on the use of FEC and are discussed in section 2.4.

**Fading.** Power line fading is normally periodic at double the 60 Hz power voltage frequency or its submultiples and varies with time in accordance with the electrical load profile. Under severe periodic fading, digital transmissions are subject to periodic burst errors. The effect of fading is analogous to periodic impulse noise impairment, since the occurrence of either signal drop-out or impulse noise results in signal-to-noise ratio degradation. For this reason, the same FEC coding techniques used in combating impulse noise may be used in handling periodic signal fading.

Severe signal fading is rare and over 90% of all fading occurrences are less than a
few dB from the median level. Nevertheless, in the cases when deep signal fades do occur, they cause large numbers of periodic burst errors [3].

After performing a detailed analysis of all the sources of PLC errors. One concludes that the PLC is not a "random error channel". In most cases the PLC is not a simple "burst error channel" either. The PLC is not a "random error channel", due to the overwhelming presence of periodic error burst sources. A simple "burst error channel" has two states: a "good state" in which transmission errors occur infrequently with the probability of error approximately 0 (these error-free spans are called guard spaces); and a "bad state" in which transmission errors are highly likely. The channel is in the good state most of the time, but occasionally shifts to the bad state, during which errors occur in clusters or bursts that never exceed a certain length [9]. The PLC does not meet these conditions; the guard spaces or good states are often corrupted with random errors, and the length of the error bursts are often difficult to predict. For such reasons the PLC may be described more accurately as a "messy" channel [3].

2.4 Evaluation of Forward Error Correction Codes for High Speed Power Line Data Communications

The selection of a specific code and an appropriate decoding algorithm for a given communication system requires the consideration of many factors. These include: channel characteristics and parameters, decoding performance and complexity, coding efficiency and delay, storage requirements, and decoder speed, cost and complexity. As well as these direct considerations that are, other more general ones, including: message size and format, nature of data traffic, transparency of the codec during data transmissions, and, the connectivity of such devices to existing systems.

Cost and complexity are our major concerns with regard to the implementation requirements. From a system design point of view, the codec's efficiency and performance will be the most important factors.

The two different types of codes in common use today are block codes and convolutional codes. In previous work Chan and Donaldson [2, 14] provided a detailed study of
the performance of a wide range of convolutional and block codes in high speed power line data communications. In their studies they used interleaving techniques to disperse clustered errors. For block codes, interleaving may be implemented by arranging $\lambda$ code words from the original code into $\lambda$ rows of a rectangular array and subsequently transmitting them in column order. The parameter $\lambda$ is the interleaving degree. The effect of interleaving is such that adjacent bits of a codeword are actually separated in the transmission by $\lambda-1$ bits. At the receiving end codewords must be deinterleaved into the original order for decoding.

Interleaving is also applicable to convolutional codes. Interleaving requires the multiplexing of the outputs of $\lambda$ separate encoders before transmission over the channel. The received bits are then demultiplexed and sent to $\lambda$ separate decoders. An error burst of length $\lambda$ will look like single errors to each of the separate decoders. In practice it is not necessary to use $\lambda$ separate encoders and decoders, but to implement one pair in such a way that their operation is equivalent to that of $\lambda$ pairs.

Interleaving does not involve additional redundancy but does add decoding delay and storage costs. For interleaving to be effective, an adequate interleaving degree must be chosen, which has low overhead (regarding both delay and costs), and disperses the errors that occur during impulse noise and signal fading into intervals between these impairments.

2.4.1 Block Codes

For evaluation purposes, Chan [3] chose two types of block codes: a set of random error correcting block codes was selected from the class of BCH codes with lengths ranging from 15 to 63 and with different error correcting capabilities. Secondly, for burst error correction some very efficient burst error correcting cyclic and shortened cyclic codes were analyzed. The burst error correcting codes were selected with parameters similar to those of the random BCH codes, thereby allowing a comparative analysis of code performance.
The evaluation was performed for different types of channel impairments at different BER with and without interleaving [2, 3]. The effects of interleaving in each case were thoroughly investigated. The principal effects of parameter variation in code selection are summarized as follows:

1. For a fixed code rate $R$ and variable block length $n$:
   a. Interleaved short random error codes were the most effective and reliable means of improving the PLC performance.
   b. Interleaving periods should not be comparable in length to the period of the impulse noise disturbance, since errors tend to be periodic. For this reason a short code with moderate interleaving or a longer but more powerful code with a small degree of interleaving would be among the best choices.

2. For a fixed block length $n$ and a variable code rate $R$:
   a. The effect of interleaving increases with the error correcting capability $t$ of the code, which also causes a decrease in throughput $R$.
   b. For codes with inadequate error correcting capability, the use of interleaving is virtually useless.
   c. For codes with nearly adequate error correcting capability, interleaving noticeably upgrades their performance.
   d. Interleaving may reduce the error correction performance at some multiples of the error period when the interleaving block contains additional periodic noise impulses.
   e. For codes with high error correcting capability, error correction performance improves monotonically with interleaving. Unfortunately the loss of throughput due to the excessive overhead of the code, in most of these cases is prohibitive.

2.4.2 Convolutional Codes

In contrast to block encoding where the data is processed as codewords that are basically independent of each other, convolutional encoders process an input bit stream
continuously. The specifics of encoding and decoding convolutional codes, as well as their characteristics are detailed in Chapter 3. Here, we summarize some of the results obtained in previous studies [2], as a point of comparison.

For testing purposes three different convolutional codes were selected from the well known classes of self-orthogonal and diffuse codes [7, 9]. These were all systematic rate 1/2 convolutional codes including: the self-orthogonal (2, 1, 6) code, the self-orthogonal (2, 1, 35) code, and, the diffuse (2, 1, m) code with m=3d+1, where d is any positive integer greater than 1. In every case, threshold decoding was used. It is the most attractive option for low cost applications, due to its ease of implementation.

The error correction effectiveness of the previously mentioned codes against periodic impulse noise, random noise, signal distortion, phase distortion, and other impairments encountered on the PLC, are summarized as follows:

1. The use of a simple short constraint length convolutional code with threshold decoding and a moderate degree of interleaving can reduce the decoded BER several orders of magnitude below uncoded transmissions, thus proving that the use of a suitable, inexpensive FEC technique makes feasible data communications over an otherwise unreliable PLC.

2. Significant improvements in performance were obtained by introducing moderate amounts of interleaving (λ ≤ 7), after which the improvements became much more gradual. Large interleaving degrees are also less economical and impractical to implement.

3. Due to the constant variations in the PLC, interleaving a random error correcting code that is able to combat both burst and random errors, was found to be a much more robust and reliable FEC technique, than was trying to select a particular code to correct a fixed number of errors.

4. The initial delay and overhead due to coding is negligible compared with the delay of retransmitting packets with errors, since packets are generally over a thousand bits long. The packet reception rate with coding increased from close to 0% to over 90%, drastically improving the throughput of any retransmission scheme.
5. Retransmission of those packets with uncorrected errors, under an appropriate ARQ strategy, should be used to assure highly reliable data link communications.

As stated previously the selection of a suitable FEC code for the PLC is a complex issue involving many factors. A remarkably simple, reliable and inexpensive scheme to combat the impairments of the PLC, is to use a short random error correcting code with a moderate degree of interleaving. Due to the simplicity of the structure of the decoder and its inherent suitability its for interleaving, convolutional coding and threshold decoding was chosen as the most appropriate. The self-orthogonal (2, 1, 6) convolutional code with variable degrees of interleaving (λ=1, 3, 5, 7) was selected for implementation.
Chapter 3
Convolutional Coding for Power Line Data Communications

In section 2.4 a summary of previous work done by Chan and Donaldson [2, 3] was presented. In their studies the performance of two types of codes, block and convolutional, was practically evaluated within a PLC environment. The use of a short random error correcting code with a moderate degree of interleaving was shown to be the most appropriate solution. The (2, 1, 6) convolutional code with variable degrees of interleaving was selected as a code complying with these requirements. Threshold decoding was chosen due to the simplicity of the decoding algorithm and its appropriateness for use with code interleaving.

3.1 Encoding of Convolutional Codes

The information sequences of block codes are grouped into k-bit blocks which are independently encoded into n-bit code words. In this manner the coded sequence becomes a sequence of fixed-length independent code words. Convolutional codes on the other hand greatly differ from this approach, since the convolutional encoder contains memory and the encoder outputs depend not only on the k inputs but also on m previous input blocks. Therefore, convolutional codes have a more complex structure than block codes, which makes them harder to analyze but sometimes easier to decode.

A convolutional encoder code accepts k-bit blocks of the information sequence u and produces an encoded sequence v of n-symbol blocks. Each encoded block depends not only on the corresponding k-bit message block at that time unit, but also on m previous message blocks; and hence the encoder has a memory order of m. The set of encoded sequences produced by a k-input, n-output encoder of memory m is called an \((n, k, m)\) convolutional code [9].

The ratio \(R = k/n\) is called the code rate; redundant bits for combating channel impairments can be added to the information sequence when \(k < n\) in which case \(R < 1\). Typically,
Figure 5 Block diagram of a (2, 1, 6) convolutional encoder.

$k$ and $n$ are small integers and more redundancy is added by increasing the memory order $m$ of the code while keeping the other parameters constant. The memory order of a code determines its constraint length which is generally accepted as $n_A = (m+1) \times n$ [6]. Thus, the constraint length of a convolutional code will increase when redundancy is increased by augmenting the memory order $m$ of the code.

Next, we describe the encoding process for a $R=1/2$ systematic convolutional code will be described. A system block diagram of such an encoder is shown in figure 5.

The information sequence $u=(u_0, u_1, u_2, \ldots)$ enters the encoder one bit at a time. Since the encoder is a linear system, the two encoder output sequences $v^{(1)}=(v_0^{(1)}, v_1^{(1)}, v_2^{(1)}, \ldots)$ and $v^{(2)}=(v_0^{(2)}, v_1^{(2)}, v_2^{(2)}, \ldots)$ can be obtained as the convolution of the input sequence $u$ with the two encoder "impulse responses". Since the encoder has an $m$-time unit memory, the impulse responses can last at most $m+1$ time units, and are written $g^{(1)}=(g_0^{(1)}, g_1^{(1)}, \ldots, g_m^{(1)})$ and $g^{(2)}=(g_0^{(2)}, g_1^{(2)}, \ldots, g_m^{(2)})$. The impulse responses $g^{(1)}$ and $g^{(2)}$ are called generator sequences of the code.

The generator sequences of a $R=1/2$ systematic convolutional code, are described as: $g^{(1)}=(1 \ 0 \ 0 \ \ldots)$, $g^{(2)}=(g_0^{(2)} \ g_1^{(2)} \ g_2^{(2)} \ \ldots)$ and generator matrix $G$ as follows:
For an information sequence $u$, the encoding equations are given by:

$$v^{(1)} = u * g^{(1)} = u$$

$$v^{(2)} = u * g^{(2)}$$

where $*$ denotes convolution. The transmitted code word is $v = uG$. If $v$ is transmitted, the binary received sequence $r$ can be written as

$$r = \left( r_0^{(1)}, r_1^{(1)}, r_2^{(1)}, ... \right) = v + e,$$

where the binary sequence $e$ is called the channel error sequence. The received sequence $r$ can be divided into a received information sequence

$$r^{(1)} = \left( r_0^{(1)}, r_1^{(1)}, r_2^{(1)}, ... \right) = v^{(1)} + e^{(1)} = u + e^{(1)},$$

and a received parity sequence

$$r^{(2)} = \left( r_0^{(2)}, r_1^{(2)}, r_2^{(2)}, ... \right) = v^{(2)} + e^{(2)} = u * g^{(2)} + e^{(2)},$$

where $e^{(1)}$ is the information error sequence, and $e^{(2)}$ is the parity error sequence. The received information sequence (5), and the received parity sequence (6) are used by the decoder to estimate the values of the information bits.

### 3.2 Decoding of Convolutional Codes

The three different methods that are widely used for decoding convolutional codes: Viterbi decoding [5], sequential decoding [18], and threshold decoding [6]. These decoding methods have different characteristics, and the selection of any particular scheme.
depends to a great extent on the particular application. Important parameters used for comparisons include: probability of bit errors, implementation complexity, and decoding speed.

Viterbi decoding is an optimum decoding algorithm, (i.e. it achieves the lowest probability of bit errors). However, the complexity of the decoder grows exponentially with code constraint length and the number of encoder states. For this reason Viterbi decoding is generally used for short constraint codes. The complexity of sequential decoding is essentially independent of constraint length, which allows it to decode long constraint length codes. The performance of sequential decoding is slightly suboptimum (regarding the probability of decoded bit errors), compared to Viterbi decoding, because the number of computations needed to decode a frame of data is random. Although most frames are decoded very quickly, in some cases long searches may result, and cause occasional erasures.

Threshold decoding (or majority-logic decoding) is a practical decoding technique originally proposed by Massey [6] which can achieve moderate coding gains with relatively simple implementations. Threshold decoding has received much attention due to its simplicity and its suitability to internal interleaving. Majority-logic decoding rules, definitions and theorems are given in detail in [6, 9].

The performance of a threshold decoder is inferior to Viterbi or sequential decoder regarding the bit error probability of the decoded bit stream. Viterbi decoders delay decision making until the entire received sequence is processed, basing its decisions on the total or free distance between sequences. In contrast, threshold decoders require a delay of only one constraint length. Due to the simplicity of the algebraic approach used and that decisions are based on the minimum distance over just one constraint length, majority-logic decoders are capable of higher speeds than that of Viterbi\(^\text{1}\) or sequential decoders. For example Viterbi decoders perform \(2^k\) computations per decoded bit, with a decoding delay, \(d\), of \(L+m\) cycles (where \(L\) is the length of the entire frame, and \(m\) is the

\(^1\) Presently a wide range of high-speed Viterbi decoders have been implemented taking advantage of parallel architectures, but this greatly increases the cost of the decoder.
memory order of the decoder). A majority-logic decoder only requires one computation per decoded bit, and the decoding delay, \( d \), is of \( m \) data cycles.

Threshold decoders are inherently simpler to implement since the decoder only contains a replica of the encoder, several modulo-2 adders, a syndrome register, and a majority-logic gate. These modest requirements make threshold decoders very attractive for low cost applications such as power line LANs, where a reasonable amount of coding gain is desired at the minimum cost.

As stated previously interleaving of the code symbols effectively randomizes the errors as they appear at the decoder. Interleaving can be implemented by substituting each single shift register cell by a \( \lambda \) stage cell in its place causing the code symbols to be spaced \( \lambda \) symbols apart. Deinterleaving is accomplished by the \( \lambda \)-stage cells in the decoder registers. If channel bursts are less than \( \lambda \) symbols long, the performance, in terms of the bit error probability, of such a encoder-decoder will be identical to that of one without interleaving [19].

The syndrome sequence \( s \) is defined as

\[
s \triangleq rH^T, \tag{7}
\]

where \( H \) is the parity check matrix given by:

\[
H = \begin{bmatrix}
g_0^{(2)} & 1 \\
g_1^{(2)} & 0 & g_0^{(2)} & 1 \\
g_2^{(2)} & 0 & g_1^{(2)} & 0 & g_0^{(2)} & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
g_m^{(2)} & 0 & g_{m-1}^{(2)} & 0 & g_{m-2}^{(2)} & 0 & \ldots & g_0^{(2)} & 1 \\
g_m^{(2)} & 0 & g_{m-1}^{(2)} & 0 & g_{m-2}^{(2)} & 0 & \ldots & g_1^{(2)} & 0 & g_0^{(2)} & 1 \\
g_m^{(2)} & 0 & g_{m-1}^{(2)} & 0 & g_{m-2}^{(2)} & 0 & \ldots & g_2^{(2)} & 0 & g_1^{(2)} & 0 & g_0^{(2)} & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
\end{bmatrix} \tag{8}
\]

The condition \( vH^T = 0 \) will be true if and only if \( v \) is a code word. Since \( r = v + e \), we
can rewrite (7) as

\[ s = (v + e)H^T = vH^T + eH^T. \]  \hspace{1cm} (9)

Since \( vH^T = 0 \), the syndrome depends only on the channel error sequence and not on the code word transmitted, i.e.,

\[ s = eH^T. \]  \hspace{1cm} (10)

For decoding purposes, knowing \( s \) is equivalent to knowing \( r \). Hence, the decoder can be designed to operate on \( s \) rather than on \( r \). Such a decoder is called a syndrome decoder [9]. After substituting (8) in (7), the syndrome sequence at the receiver is formed as:

\[ s = r^{(1)} * g^{(2)} + r^{(2)}. \]  \hspace{1cm} (11)

Substituting the appropriate received sequences \( r^{(1)} \) and \( r^{(2)} \) for their corresponding expressions we obtain:

\[ s = [u + e^{(1)}] * g^{(2)} + u * g^{(2)} + e^{(2)} = e^{(1)} * g^{(2)} + e^{(2)}. \]  \hspace{1cm} (12)

Any syndrome bit, or any sum of syndrome bits, represents a sum of channel errors and is called a parity-check sum. If the received sequence is a code word, all syndrome bits, thus all check sums, must be zero. If the received sequence is not a code word, some check sums will not be zero. The next section describes in detail how this property is used in majority-logic decoding.

### 3.3 Self-Orthogonal (2, 1, 6) Code

The self-orthogonal (2, 1, 6) systematic \( R=1/2 \) convolutional code is characterized by the generator sequences:

\[ g^{(1)}(D) = 1 \text{ and } g^{(2)}(D) = 1 + D + D^4 + D^6. \]  \hspace{1cm} (13)

A block diagram of a (2, 1, 6) convolutional encoder was given in figure 5. The constraint length of this code is expressed as,

\[ n_A = (m + 1) \times n = (6 + 1) \times 2 = 14. \]  \hspace{1cm} (14)
From (10) we can obtain an expression for the truncated syndrome sequence,
\[ [s]_6 = [e]_6 [H^T]_6, \]  
where \([s]_6 = (s_0, s_1, \ldots, s_6).\) Transposing both sides of (15),
\[ [s^T]_6 = [H]_6 [e^T]_6 \quad \text{or}, \]
\[ (15) \quad (16) \]
\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1
\end{bmatrix}
= 
\begin{bmatrix}
e_0^T \\
e_1^T \\
e_2^T \\
e_3^T \\
e_4^T \\
e_5^T \\
e_6^T
\end{bmatrix}. \]  
(17)

Since the even numbered columns of \(H^T\) form an identity matrix, (17) can be rewritten in the following form,
\[
[s^T]_6 = \begin{bmatrix}
s_0 \\
s_1 \\
s_2 \\
s_3 \\
s_4 \\
s_5 \\
s_6
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
1 & 1 \\
0 & 1 & 1 \\
0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
e_0^{(1)} \\
e_1^{(1)} \\
e_2^{(1)} \\
e_3^{(1)} \\
e_4^{(1)} \\
e_5^{(1)} \\
e_6^{(1)}
\end{bmatrix}
+ 
\begin{bmatrix}
e_0^{(2)} \\
e_2^{(2)} \\
e_4^{(2)} \\
e_6^{(2)}
\end{bmatrix}. \]  
(18)

The matrix which multiplies the information error sequence is called the parity triangle of the code. Its first column is \(g^{(2)}\) and the \(k^{th}\) column is a shifted down version of column \(k-1\). From (18) the orthogonal check sum equations are obtained and given by:
\[
\begin{align*}
s_0 &= e_0^{(1)} + e_0^{(2)} \\
s_1 &= e_0^{(1)} + e_1^{(1)} + e_1^{(2)} \\
s_2 &= e_0^{(1)} + e_2^{(1)} + e_2^{(2)} \\
s_3 &= e_0^{(1)} + e_3^{(1)} + e_3^{(2)} \\
s_4 &= e_0^{(1)} + e_4^{(1)} + e_4^{(2)} \\
s_5 &= e_0^{(1)} + e_5^{(1)} + e_6^{(1)} + e_6^{(2)} \\
s_6 &= e_0^{(1)} + e_6^{(1)} + e_6^{(2)}
\end{align*} \]  
(19)

Because the error bit \(e_0^{(1)}\) is the only bit that appears in every check sum, we may say that (19) is a set of check sums orthogonal on \(e_0^{(1)}\), and that the code is self-orthogonal.
Since there are a total of four (symbolized as $J$) orthogonal check sums, this code will correctly estimate $e^{(1)}_0$ whenever $t_{ML} \triangleq \lceil J/2 \rceil = 2$ or less of these 11 noise bits are channel errors [9].

The total number of distinct noise bits checked by the set of check sums is called the effective constraint length of the code and will be denoted $n_E$; for this code $n_E=11$. We must note that $n_E \leq n_A$, since not all of the bits of the parity error sequence $e^{(2)}$, in the given constraint length $n_A$, form part of the set of $J$ check sums.

However, the set of check sums given by (19), only estimates $e^{(1)}_0$ from the first constraint length of syndrome bits $s_0, s_1, ..., s_m$. In general, we must estimate the values of all the other information error bits as well. The procedure to implement this is relatively simple; the estimated $e^{(1)}_0$ is subtracted from each syndrome equation it affects to form a new modified syndrome set $s_0', s_1', ..., s_m'$. Together with the new syndrome bit $s_{m+1}'$, this modified set is used to estimate $e^{(1)}_1$. If we assume that $e^{(1)}_0$ was correctly estimated, a new set of orthogonal $e^{(1)}_1$ check sums may be formed which will replicate the previous ones. Thus, the same decoding rule may also be used. The $i^{th}$ iteration yields an estimate of $e^{(1)}_i$. Since each estimate must be fed back to modify the syndrome register before the next estimate is made, resulting in a feedback decoder. Each estimate depends only on one constraint length of error bits because the effect of previously estimated error bits has been removed by the feedback.

Assuming that previous estimates of errors are correct the syndrome equations from (19) can be expressed in a more general manner

\[
\begin{align*}
    s_i' &= e^{(1)}_i + e^{(2)}_i \\
    s_{i+1}' &= e^{(1)}_i + e^{(1)}_{i+1} + e^{(2)}_{i+1} \\
    s_{i+4}' &= e^{(1)}_i + e^{(1)}_{i+3} + e^{(1)}_{i+4} + e^{(2)}_{i+4} \\
    s_{i+6}' &= e^{(1)}_i + e^{(1)}_{i+2} + e^{(1)}_{i+5} + e^{(1)}_{i+6} + e^{(2)}_{i+6}
\end{align*}
\]

The same error correcting capabilities will apply for this set of syndrome equations; that is, $e^{(1)}_i$ will be correctly estimated if there are $t_{ML}=2$ or fewer errors among the $n_E=11$ error bits checked. A block diagram of such a decoder is shown in figure 6.
In general, the operation of a threshold decoder for a (2, 1, 6) systematic code can be divided into the following steps:

1. The syndrome bits for the first constraint length are calculated.
2. From these previously calculated syndrome bits a set of four checksums on $e_0^{(1)}$ is formed.
3. The results from the four check sums are fed into a majority-logic gate. This gate produces a 1 at its output if and only if more than half of its inputs are 1. The presence of a 1 at its output indicates the presence of an error. Thus, the received information sequence must be corrected (by adding the output of the majority gate to the corresponding bit of the received sequence). This output is also fed back and subtracted from each syndrome it affects.
4. The estimated information bit $\hat{u}_2 = r_0^{(1)} + \hat{e}_0^{(1)}$ is shifted out of the decoder. The syndrome registers are shifted once to the right. The next 2 bits are shifted into the decoder, and finally, the next syndrome bit is calculated and shifted into the leftmost stage of the syndrome register.

Figure 6 Block diagram of a (2, 1, 6) threshold decoder.
5. The syndrome register now contains the modified syndrome bits, together with the new syndrome bit. Steps 2, 3 and 4 are repeated to estimate the next information bit. All the following information error bits are estimated in the same manner.

It should be noted once again that each independent bit that is estimated depends only on one constraint length of the of error bits, since the effect of the previously estimated bits are removed by the feedback to the syndrome register. In the next section we will discuss the extent to which this feedback affects the performance and error correcting capabilities of the decoding scheme itself.

3.4 Properties and Performance of Threshold Decoding

Because estimates of the error bits are subtracted from the syndrome register, errors will propagate when their estimate is incorrect, hence generating the same effect as having additional channel errors. This effect is known as the error propagation effect [20], and will be present in any decoder which feeds back the estimated values.

There are a number of approaches that can be taken to limit the error propagation effect, such as resynchronizing the decoder with a “zero sequence” that “flushes” all the possible post-decoding errors, alternatively a code with automatic resynchronization properties may be used, such that if the channel is error free for more than a constraint length, the effects of previous errors will be removed from the decoding circuitry. Self-orthogonal codes possess this automatic resynchronization properties with respect to error propagation. Further discussion and proof of this property may be found in [20].

The elimination of the use of feedback in the decoder completely avoids the problem of error propagation, as Robinson suggested [21]. However this approach, known as definite decoding, in most cases, will not provide better performance than feedback decoding, since the effects of previously estimated channel error bits are not removed from the syndrome register and will continue to affect following bit estimates, thus increasing the probability of decoding errors.

Definite decoding is of great interest because certain single “stuck at” faults in the circuit can transform (or partially transform) the feedback decoder into a definite one.
The absence of a link (which would in effect be equivalent to a “stuck at zero” fault) in the feedback of the syndrome register would yield the same decoding estimate as would the feedback decoder with all the links. In figure 7 the positions of such faults that could transform a feedback decoder to a definite decoder are marked a–d.

During the analysis that follows, we will only consider the effects of stuck at zero faults, since the effect of any stuck at one fault will clearly generate incorrect estimates that can be easily detected during the test procedure. In what follows stuck at zero faults will be referred to simply as faults.

Case A: If a fault were to occur at point a, there would be no feedback at all to the syndrome registers, a definite decoder would result, for which the checksums of (20) would become

\[
\begin{align*}
\bar{s}_i' &= e_{i-6}^{(1)} + e_{i-4}^{(1)} + e_{i-1}^{(1)} + e_i^{(1)} + e_i^{(2)} \\
\bar{s}_{i+1}' &= e_{i-5}^{(1)} + e_{i-3}^{(1)} + e_i^{(1)} + e_{i+1}^{(1)} + e_{i+1}^{(2)} \\
\bar{s}_{i+4}' &= e_{i-2}^{(1)} + e_i^{(1)} + e_{i+3}^{(1)} + e_{i+4}^{(1)} + e_{i+4}^{(2)} \\
\bar{s}_{i+6}' &= e_i^{(1)} + e_{i+2}^{(1)} + e_{i+5}^{(1)} + e_{i+6}^{(1)} + e_{i+6}^{(2)} \\
\end{align*}
\]
The checksums in (21) show that definite decoding still forms a set of four orthogonal checksums on $e^{(1)}$. However, the effective constraint length of the code $n_E$ is no longer 11, as in the previous feedback decoder, but has increased to 17. The decoder in this case will only correctly estimate $e^{(1)}$ if 2 or fewer of these 17 error bits are 1. This clearly shows the manner in which the error correcting capability of the code decreases. Whenever there are more channel error bits present in the decoding of any given bit, the possibilities of incorrectly estimating the bits value increases.

In cases such as this one where the resulting effective constraint length is larger than the constraint length of the code ($n_E > n_A$), it is relatively simple to detect the presence of a fault in the decoder. The error correcting capability of the code is weakened, causing the previous maximum error correcting capability of the code to be no longer obtainable. Such is the case if there is a fault at $a$.

Case B: If a fault were to occur at point $b$ then the checksums from (20) would become

\[
\begin{align*}
\tilde{s}_1 &= e_{1-1}^{(1)} + e_1^{(1)} + e_1^{(2)} \\
\tilde{s}_{i+1} &= e_i^{(1)} + e_{i+1}^{(1)} + e_{i+1}^{(2)} \\
\tilde{s}_{i+4} &= e_i^{(1)} + e_{i+3}^{(1)} + e_{i+4}^{(1)} + e_{i+4}^{(2)} \\
\tilde{s}_{i+6} &= e_i^{(1)} + e_{i+2}^{(1)} + e_{i+5}^{(1)} + e_{i+6}^{(2)}
\end{align*}
\]

In this case only one more term is added to the equation set, increasing the effective constraint length by one to $n_E=12$. Similar results can be obtained for such cases when faults occur at locations $c$ or $d$. The effective constraint length in each case will be $n_E=13$ and $n_E=14$, respectively. Decoders with these faults are still able to maintain the error correcting capability of the code, thus correcting any two errors in any 14 consecutive channel bits.

However, as can be seen from the expression given by Lin in [9], the difference in the effective constraint lengths in these cases will affect the decoded bit error probability, $P_b(E)$. For channels with bit error probabilities for which $n_p < 1$ (i.e. $p < 10^{-2}$) the
decoded bit error probability is expressed as follows:

\[ P_b(E) = \left( \frac{1}{k} \right)^{n_E} (t_{ML}+1) p^{b_{ML}+1} \]  \hspace{1cm} (23)

From (23) it is clear that an increase in the effective constraint length \( n_E \) will in effect also increase the bit error probability. A comparison of the actual effect of such faults on \( P_b(E) \)

For a feedback majority-logic decoder,

\[ P_t(E) \approx \binom{11}{3} p^3 = 165 p^3. \] \hspace{1cm} (24)

For a decoder with fault \( b \),

\[ P_t(E) \approx \binom{12}{3} p^3 = 220 p^3. \] \hspace{1cm} (25)

For a decoder with fault \( c \),

\[ P_t(E) \approx \binom{13}{3} p^3 = 286 p^3. \] \hspace{1cm} (26)

For a decoder with fault \( d \),

\[ P_t(E) \approx \binom{14}{3} p^3 = 364 p^3. \] \hspace{1cm} (27)

From the syndrome equations in (20), it was shown that the effective constraint length of the code using a feedback decoder is \( n_E = 11 \), that means that there are three channel error bits which are not used to compute the values of the syndromes in a given moment. This fact allows the decoder to achieve correct estimates in the presence of certain patterns of triple errors during the transmission of 14 consecutive channel bits. For example the estimated information bit, \( \bar{U} \), will be correctly estimated if the following triple error patterns occur: \( e_1^{(1)} + e_4^{(1)} + e_3^{(2)} \), \( e_0^{(1)} + e_6^{(1)} + e_5^{(2)} \) or \( e_0^{(1)} + e_6^{(1)} + e_5^{(2)} \)

This knowledge will be used in the BIST scheme, since from the sets of syndrome equations for the cases with faults at points \( b, c \ or \ d \), it is clear that the effective constraint length, \( n_E \), increases with any of these faults. Thus it will always be possible to find a
certain set of triple channel errors that can be correctly estimated by the decoder with no faults, but will generate an incorrect estimate if used with a decoder with faults $b$, $c$ or $d$. The first pattern shown previously will cause incorrect decoding for a codec with faults $b$ or $d$, while the second error set will generate an incorrect estimate for decoders with faults $b$ or $c$. In the next chapter a detailed description of the realized BIST procedure is provided.
Chapter 4
VLSI Design and Implementation

4.1 General Description

In this chapter the issues related with the design and implementation of a VLSI codec are discussed, such as: schematic capture, layout generation, schematic and layout simulations, as well as post-fabrication tests of the chip. A detailed description of the functional blocks and signals that form part of the chip is also presented.

Before the process of designing a chip starts, several technological factors and possible trade-offs between them must be taken into consideration, to ensure that the IC meets its given set of specifications while expending minimal resources within a given constraint of time [22]. The most relevant of these factors are: the choice of the minimum feature size of the fabrication process, and the selection of the design scheme.

In this case the selection of the minimum feature size of the fabrication process was not difficult, since the choice was limited to those processes offered by Northern Telecom Electronics (NTE) through Canadian Microelectronics Corporation (CMC) for university research. Both the CMOS3DLM 3 \( \mu m \) process and the CMOS4S 1.2 \( \mu m \) processes were adequate for the fabrication of the codecs. The CMOS4S 1.2 \( \mu m \) process was selected taking into account the future possibility of integrating both the modem and codec modules onto one chip, and due to its favorable fabrication schedules.

Three possible alternatives of design schemes were analyzed: full–custom design, semi–custom design, and the use of a field programmable gate array (FPGA). The use a full-custom design approach includes the design of each of the basic cells (NAND, NOR, flip-flops and others). The process of designing, simulating and extensively testing each of the independent cells is time consuming and costly. Since the codecs' designs are composed entirely by commonly used standard cells, redesigning and optimizing the existing library cells would not bring a substantial decrease in the silicon area of the chips, making the use of a semi-custom design approach the most suited alternative.
The last of the design schemes considered was the use of an FPGA based design of the codecs. Unfortunately, the use of FPGAs for low-cost, large volume applications, such as the PLMs is not yet feasible due to their cost ($300–$400 for a suitable Xilinx™ FPGA). FPGAs at this stage could be used for prototypes, but cannot be seen as a commercially viable solution.

All the schematics and layout generations were done using EDGETM (CADENCE) CAD tools [23, 24], in the VLSI laboratory of the University of British Colombia. The chip design was based on the use of standard cells from the Canadian Microelectronics Corporation's (CMC) CMOS4S Library [25], which contains all the basic low level cells necessary to realize a hierarchical bottom-up design. Simulations of the designs were run with SILOS II® Logic and Fault Simulator [26]. In this manner, verification of the correctness of the circuit was performed before the generation of the layout.

The codecs were fabricated using the Northern Telecom Electronics 1.2μm double-poly-silicon, double-metal CMOS process, and packaged in a 68–pin ceramic pin grid array (PGA). Of the actual 68 pins only a small portion is used in each case, but the 68–pin PGA is the only packaging used for prototypes at Northern Telecom Electronics (NTE). Three chips were designed and manufactured, each time increasing in complexity and available features. Each of these chips were:

• This chip is a (2, 1, 6) convolutional encoder & threshold decoder with fixed interleaving degree (λ=7). This chip has 10 input/output pins, 1 power and 1 ground pin, and a total area of 6.0 mm² (2760 μm x 2190 μm). The core area is of 1.73 mm², and has approximately 1,260 gates.

• This chip is (2, 1, 6) convolutional encoder & threshold decoder with variable interleaving degree (λ=1, 3, 5, 7). This chip has 14 input/output pins, 1 power and 1 ground pin, and a total area of 6.98 mm² (2850 μm x 2450 μm). The core area is of 2.11 mm², and has approximately 1,480 gates.

• This chip is (2, 1, 6) convolutional encoder & threshold decoder with variable interleaving degree (λ=1, 3, 5, 7) and built-in self-test. This chip has 15 input/output pins,
1 power and 1 ground pin, and a total area of 7.57 mm$^2$ (2910 µm x 2600 µm). The core area is of 2.32 mm$^2$, and has approximately 1,620 gates.

The VLSI codec designed, fabricated and tested as part of this thesis work consist of two major blocks as the name suggests: the encoder and the decoder. Both blocks were designed to be fabricated on one chip, thus reducing the total size and cost of the final modem and this way supporting half and full duplex transmissions using a single chip. Integration of the encoder and decoder onto one chip, this provided an ideal situation for the use of BIST [27]. A detailed description of the BIST scheme used in this case, and its implementation, which provides 100% fault coverage for single stuck-at faults [28], is given in section 4.5.

From the initial stages of the design, the necessary steps were taken to provide a simple approach towards the use of variable degrees of interleaving ($\lambda$ = 1, 3, 5, 7) in the encoder and decoder blocks. The use of an interleaved code make it necessary to introduce a few additional support blocks which provide proper synchronization and structure to the encoded bits being multiplexed for transmission over the channel.

In what follows, the notation "-" denotes the presence of a negative edged or active low signal. The pin number for each of the signals varies in each case for the three versions of the chip. In appendix A the schematic and layout representations of each version of the codec are shown. In appendix B the bonding diagrams of the different chips are given.

### 4.2 Encoder

The structure of a convolutional encoder is shown in the previous in figure 5 of the previous chapter. In this case we set out to design a self-orthogonal (2, 1, 6) systematic $R=1/2$ convolutional encoder characterized by the generator sequences in (13).

In this section a description of the encoder signals and their functions is given. The schematic diagram of the encoder is shown in figure 8.
Figure 8 Schematic diagram of a (2, 1, 6) convolutional encoder.
Chip pin descriptions are as follows:

U1 (Input): Information sequence input. This is a negative edge triggered synchronous serial input. An extra flip-flop (150) is placed in front of the actual encoder to avoid the propagation of undesirable asynchronous changes on the input that could alter the values of the V1 and V2 outputs in at any time other than after the negative edge of the clock cycle.

V1, V2 (Output): Output sequences, which are the encoded outputs of the information sequence U1, after passing through each of its respective encoding equations, characterized by the generator sequences: $g^{(1)}(D) = 1$ and $g^{(2)}(D) = 1 + D + D^4 + D^6$. Both of these signals are negative edge triggered.

C1, C2 (Input): These control signals select the interleaving degree which is to be used ($\lambda = 1, 3, 5, 7$).

–CLK (Input): Clock Signal. Negative edged clock, with a maximum frequency, $f_{\text{max}}$, of 50 MHz.

SET, –RESET (Input): Set (active high) and Reset (active low) signals for initializing the encoder. The hold time, $t_{\text{hold}}$, for these signals is 1.5 ns in both cases.

4.3 Decoder

From chapter 3, the received information sequence, $r^{(1)}$, and the parity sequence $r^{(2)}$ at the decoders were expressed by (5) and (6). These signals are received at the R1, R2 inputs of the decoder, from which the syndrome sequence at the receiver is formed as in (11) and (12).

A schematic diagram of the (2, 1, 6) majority logic decoder with variable degrees of interleaving is shown in figure 9.

The decoder is split in two separate functional blocks, the syndrome generator and the syndrome register. The syndrome generator is formed by a replica of the encoder. It takes the convolution of $r^{(1)}$ and $g^{(2)}$ and adds it (mod 2) to $r^{(2)}$, thus forming the syndromes such as in (11). The output of gate 128 in figure 9 will be the resulting syndrome bit $s_1$, which will be zero unless an error has occurred during the transmission.
Figure 9  Schematic diagram of a (2, 1, 6) majority–logic decoder.
The output from the syndrome generator goes to the input of the syndrome register, (input B of gate 129 in figure 9), to be used in the estimation of $e_1^{(1)}$. The syndrome register is designed to realize the set of orthogonal check sums described in (20). The syndrome bits $s_i$, $s_{i+1}$, $s_{i+4}$, $s_{i+6}$ are used to estimate the error bit $e_1^{(1)}$, using the majority-logic gate (136). This gate produces an estimate of “1”, if more than two of the syndrome bits are “1’s”. The output from the majority-logic gate is fed back to the syndrome register and subtracted from each syndrome bit that it affects, in order to generate the correct next set of syndrome equations.

The estimate of $e_1^{(1)}$ is used as well to calculate the estimate of information bit $u_1$. This is accomplished by adding (mod2), the received information bit $i^{(1)}$ with the estimated error bit. The result is obtained at the output of gate 140. An additional flip-flop (144) was added to clean the output signal of small spikes, that are created due to differences in the response times of the various components.

Chip pin description are as follows:

R1 (Input) : Received Information Sequence. This is a negative edge triggered synchronous serial input.
R2 (Input) : Received Parity Sequence. This is a negative edge triggered synchronous serial input.

u1 (Output) : Estimated Information Bit. This is a negative edge triggered synchronous serial output.

C1, C2 (Input) : These control signals select the interleaving degree which is to be used ($\lambda = 1, 3, 5, 7$).

–CLK (Input) : Clock Signal. Negative edged clock, with a maximum frequency, $f_{\text{max}}$, of 50 MHz.

SET, –RESET (Input) : Set (active high) and Reset (active low) signals for initializing the encoder. The hold time, $t_{\text{hold}}$, for these signals is 1.5 ns in both cases.

4.4 Support Blocks

The use of a $R=1/2$ code generates 2 symbols per clock cycle. These symbols are
multiplexed onto one transmission line, demultiplexing is performed at the receiver. For this reason two additional blocks were implemented, a multiplexer and a demultiplexer.

The multiplexor (MUXL) takes the two outputs from the encoder block and alternately outputs them onto the channel. The information sequence V1 is output during the low level of the clock, and parity sequence V2 is output during the high level of the clock.

The demultiplexor (DMUXL) accepts two symbols from the channel and places them on the appropriate inputs of the decoder, R1 and R2. Two important issues must be noted regarding the demultiplexor. First, the values at the outputs must be generated such that both outputs are valid at the negative edge of the clock signal. For this reason an extra delay must be inserted in the DMUX cell which assures data synchronization. Second, great care must be taken to ensure that the transmitter clock and the receiver clock are co-phased, making sure that one is not the inversion of the other. With inversion, the demultiplexed values of the received information sequence and that of the received parity sequence would be the wrong inputs of the decoder, thus resulting in incorrect decoding of the whole data stream.

The clock signal must be distributed to over 200 cells on the chip. The driving capacity of the input pads cannot assure this distribution without degradation of the signal. Consequently, a tree was implemented for clock distribution in order to avoid clock skews at different places of the codec. Buffers were used to drive the clock signal from its “root” to its “branches” in an organized and balanced manner. Each distribution path has the same delay between the clock signal at the input pad and any of its destinations within the chip core.

A block diagram of the process involved for accomplishing the interleaving is shown in figure 10. This procedure is described in further detail in the previous chapter as well as in [3] and [19]. The realization of variable interleaving is accomplished by means of two separate blocks, each of which is a shift register of variable length. The first block, 7Cell, has programmable lengths of 1, 3, 5 and 7 units. The second block, 3Cell, can vary in length from 0 to 3 units. The structure of these cells appear in figures 11 and 12.
4.5 Built-In Self-Test

The complexity of testing a VLSI circuit can be converted into costs associated with the testing process which include: the cost of test pattern generation, the cost of generating fault location information, the cost of the automatic test equipment (ATE), and the cost of the testing process itself. Because these costs can be high, it is important that they be kept within reasonable bounds. Design for testability (DFT) is one of the ways this may be accomplished, by means of using design procedures that ensure that a device is testable [27].

Many design for testability techniques, such as: level-sensitive scan design (LSSD), boundary-scan, and other scan path-based techniques, are widely used in many electronic systems [29]. However, the increasing complexity of VLSI circuits leads to harder accessibility of internal logic nodes, and to greater complexity in the process of generating test patterns.

Built-In Self-Test (BIST) emerged in the last decade as a technique that deals not only with the limited accessibility problem, but the costly test generation and tester problems as well. BIST can be defined as the capability of a circuit (chip, board or system) to test itself without requiring any external test equipment [29]. BIST has become increasingly widespread because of its great potential in VLSI testing, due to the following:
Figure 11  Schematic diagram of 7Cell block.
• The need for a cost effective testing approach that can be applicable to all categories of testing, from production to field maintenance.

• The increasing time, cost, and complexity of test pattern generation, due to the growing density of VLSI circuits.

• The cost of ATE has steadily increased, as well as the fact that the performance and speed of the circuits under test has become comparable to that of the testers themselves, making the use of very costly ATE much less productive.

• Substantial increases in the time and cost involved in the testing process itself, due to the size and complexity of the circuits that need to be tested.

However, incorporating BIST to a design is accompanied by some additional costs, which are reflected through: increased silicon area, extra pins, decreased reliability due
to silicon area overhead, performance penalties due to additional circuitry, and additional
design time and cost.

Since the BIST procedure designed for this case is based on the fact that both the
encoder and decoder are situated on the same chip, a minimal amount of additional
circuitry is needed (less than 10% area overhead and only 1 extra pin) to achieve 100%
fault coverage of single stuck–at faults. In this scheme there is no performance penalty,
since no components have been introduced in the critical paths of the codecs.

The primary logic components that form part of the BIST scheme are the following:

- **Circuit Under Test (CUT):** This block is formed by both the encoder and decoder,
  which in test mode are interconnected in a loop-back fashion. The output of the
  encoder is fed back into the input of the decoder directly. The input sequence at U1
  (Encoder Input) should be exactly the same as the output sequence at u1 (Decoder
  Output) after decoding.

- **BIST Control:** This block generates all the appropriate signals that control the self-test
  procedure, whenever the BIST signal is “high”.

- **Automated Test Pattern Generator (ATPG):** The generation of input test patterns is
  accomplished with a linear feedback shift register (LFSR) that serially outputs finite
  pseudorandom sequences (16 bits long) repeatedly. Two such blocks are used,
  since the output of the CUT must be compared, after the encoding/decoding process
  is complete, with a sequence identical to the input one.

- **Comparator (COMP):** The decoded output sequence (u1) is compared with the
  “delayed” test pattern. In the case of a mismatch the output will be set to “high”,
  indicating the presence of a circuit fault. In the initialization of the test, this output
  will be toggled to ensure that the block itself does not contain a “stuck at” fault.

- **Multiplexors (MUX):** A series of multiplexors, blocks I1, I10, and I24, are used to
  multiplex the normal inputs/outputs of the CUT with their BIST inputs/outputs.

A schematic diagram of the BIST circuits is given in figure 13. A description of the
test procedure and control signals used by the BIST follows.
Figure 13 Built-In Self-Test schematics.
4.5.1 Test Procedure

The test procedure developed for this BIST scheme was designed to provide 100% fault coverage (for single stuck at faults). The scheme is based upon the fact that both the encoder and decoder circuits are on the same chip, and that in test mode the output of the decoder will be directly attached to the input of the decoder. These two blocks together will form the circuit under test (CUT). In this manner we can obtain a decoded output sequence that should be identical to the input sequence, after a certain decoding delay, \( d \), where \( d = \lambda m + (\lambda - 1)/2 + c \). As before \( \lambda \) is the interleaving degree, \( m \) is the memory order of the encoder; and \( c \) is a constant related to the delays that are due to the need to multiplex encoded bits over the channel (in this case \( c=3 \)). The sequential characteristics of the codec design guarantee that all the gates and interconnections must be fault-free to obtain the correct result at the output.

To test the syndrome register and its components it was necessary to introduce channel errors in the connection between the encoder and decoder. This will force the codec to work at the maximum error-correcting capability of the code. The self-test is divided into two separate cycles, \( t1 \), where \( \lambda=1 \), and \( t2 \), where \( \lambda=7 \). This test will ensure that all the interconnections are free of faults. In the \( t1 \) cycle specific error patterns were introduced that will also check for faults in the feedback of the syndrome register, as was discussed previously in section 3.4. During the \( t2 \) cycle, the burst error correcting capabilities of the codec are tested. In this case 14 channel-bit burst errors are introduced, which test the circuits at their highest capabilities. Figure 14 shows simulation results that illustrate the BIST test procedure. The timing of each of the signals involved in the BIST routine can be observed in detail.

The BIST routine as shown in figure 14, is formed by these steps:

1. Check the bist input. If bist=0, continue in normal operation, if bist=1 start BIST procedure.

2. The signals –rst1 and –bst1 which were resetting the CUT and the ATPG blocks, go high, enabling the start of the test. The control block outputs a toggle pulse, which
Figure 14 Built-In Self-Test simulation (without faults in the CUT).
makes the comp output go high for 1 test clock cycle, tclk, in this way testing the comparator block itself.

3. The signal \texttt{tt2} (l6.t2 in figure 14) of the BIST control block goes low, indicating that the first phase of the test is in progress, with \texttt{A} set to \texttt{1}. The input (U1) of the CUT block is switched from external data to the internal test pattern generated by the ATPG block (l28.tp in figure 14). At the same time that the ATPG block outputs the test pattern, the control block generates the corresponding error pattern, \texttt{el} (l6.e1 in figure 14), to force the decoder to work at its maximum error-correcting capability.

4. After a delay of \(d=9\) tclk cycles, the valid signal becomes active, which enables the output ATPG block (l29.tp in figure 14) to start generating the sequence which will be compared with the decoded output of the CUT (l30.u in figure 14).

5. The COMP block continues comparing both outputs for 64 tclk cycles. If an error is detected its output, (l21.out in figure 15) is set high, thus indicating the presence of a fault.

6. The second phase of the test starts when \texttt{tt2}=1, setting \texttt{A} to \texttt{7}. The \texttt{rst1} and \texttt{bst1} signal reinitialize the CUT and ATPGs. During the period in which the ATPG (l28.tp in figure 14) is outputting test patterns, the channel error signal, \texttt{el} (l6.e1 in figure 14), outputs error bursts of maximum correctable length.

7. The valid signal becomes active after a delay \(d=48\) tclk cycles, after which the output ATPG block (l29.tp in figure 14) starts generating the sequence which will be compared with the decoded output of the CUT (u1). The COMP block will analyze both outputs. If an error is detected its output is set high, thus indicating the presence of a fault. The self-test will continue in this state while the \textit{bist} pin is kept high. This provides the possibility of using this feature for when prolonged periods of testing are desired, such as burn-in tests.

4.6 Design Verification

During the process of designing and fabricating a chip various steps must be taken to ensure the correct functionality of its components and eventually of the whole circuit. The
following tests were performed in order to confirm and validate the actual implementation of each of the design stages:

1. Schematic Capture Simulations are used as a tool to verify the correctness of the initial schematic designs developed with the EDGE™ design entry tools [23]. The simulation program used was SILOS II® Logic and Fault Simulator [26], obtaining as a result of each simulation the corresponding waveforms. Previous to each simulation it is necessary to create an STL (Simulation and Test Language) program, which defines the parameters to be used as well as describing the test vectors. An example of such a file is given in figure 16.

As stated previously, the codec was designed using a bottom-up approach, with simulations at each level:

   a. Independent block simulations evaluate the correctness and functionality of each of the logic blocks.

   b. Simulations of the completed design evaluate the correctness of the whole design and determine possible timing and/or functional discrepancies between blocks.

2. Physical Design Verification is employed as a means to countercheck the physical and functional parameters of the layout generated from the schematics. This process is done by two separate and distinct programs:

   a. Design Rule Checker (DRC), checks that the technology rules (CMOS4S, Northern Telecom’s 1.2 μm process) have not been violated in the generation of the completed layout.

   b. Place and Route Verification. To verify the placement and the routing of the layout it is necessary to generate a netlist of the layout and then compare it with the previous netlist that was generated from the schematics. This process ensures that the interconnections of the layout are the same as the ones described in the schematics.

3. Post-Fabrication Functional and Parametric Tests. Prior to fabrication the designed chip passes a series of tests, all of which are based on simulation models of the cells
and their properties. One cannot be sure that the fabricated chip will actually realize its functions properly, due to the limits of the simulation models as well as problems encountered due to the fabrication process.

For this reason a series of tests must be realized to certify its adequacy. Some of these might include:

a. Automated Test Equipment (ATE) is used to do both functional and parametric testing of the chip. The facilities at Simon Fraser University have an IMS XL100™
tester, which was used in this case, that allows the user to control by software the in/out signals from each of its probes. Functional tests are performed in a similar manner to the SILOS II™ simulations. The vectors from the STL programs are used on the actual chip. Instead of applying the test vectors to a schematic model they can be directly linked through the tester to the appropriate pin of the codec. The responses from the chip are read directly into the test file for its further analysis.

Parametric tests are executed with the joint use of the IMS tester and a Tektronix 2246A™ 100MHz oscilloscope. For this purpose the input vectors are set in a loop and the clock frequency is gradually incremented. At the same time the output responses should be observed through the oscilloscope. In this manner the rise and fall times of the output waveform can be accurately measured. The maximum clock frequency tested was 50MHz.

b. Data Error Analyzer tests are used to evaluate the performance of the codec over extended periods of time. For this procedure a Hewlett Packard™ Data Error Analyzer (HP 1645A™) was used. Pseudorandom data was generated by the HP 1645A™ and used as an input to the encoder. The decoded data was then compared to the input pattern. Any errors detected during this process are counted, obtaining in this manner an upper bound on the BER of the encoding/decoding process. In all cases the tests were run until a BER of at least $10^{-9}$ was obtained.

c. Built-In Self-Test (BIST) can be used as a substitute for some of the previous functional tests, avoiding the use of costly ATE and other such devices as well. The tests in this case are more exhaustive than the ones performed formerly, since the BIST introduces errors on the internal channel during its execution. A simple testbed was designed to denote the detection of a fault in a chip being tested.

4. Application Specific Functional Tests provide benchmark results for the use of the codec in different environments, thereby determining the chip’s performance under
diverse channel conditions. In contrast to the previous steps where the chip was tested independently, here the codec is interworked with the power line modem (PLM) for which it was designed. A detailed report on these tests is given in the next chapter, since it is a focal point of this research. A distinction is made for the following cases:

a. Controlled environment tests were completed for channels where Additive White Gaussian Noise (AWGN) and Burst Errors were added in a supervised manner, thereby obtaining the statistics of the actual error rate value of the chip under such conditions.

b. Practical channel performance tests were done for data transmissions over actual power line communication channels. In each case a comparative analysis was done for each of the different degrees of interleaving.
Chapter 5
Codec Performance Tests

In this chapter a complete report on the results of performance measurements of the codec during its operation on real communication channels is given. The chip was employed as a FEC device for high speed power line communications, by coupling it with two existing power line modems (PLMs). Each modem was developed in the Department of Electrical Engineering at the University of British Columbia.

The objective of this study was to evaluate and gain in-depth knowledge of the actual benefits of the use of error-control on PLC data communications. A series of tests was performed that covered a wide range of operating conditions, from relatively clean channels to very noisy ones. In every case various of parameters were measured, including: Bit Error Rate (BER), Block Error Rate (BLKER) and Throughput.

5.1 System Level Implementation

To realize practical communication tests over the PLC, two sets of high speed PLMs were enhanced with FEC. One modem uses Binary Phase Shift Keying (BPSK) and the other Minimum Frequency Shift Keying (MFSK). Each modem provides an opportunity to compare the performance of these different modulation schemes together with error-control techniques. To interface the modems with the CODEC, minor changes had to be performed to the modems' hardware and firmware.

The enhanced PLMs (EPLMs) were designed to allow software control of the data transmission speed (2400, 4800, 9600, 19200 Bauds) and the encoding parameters (Bypass Coding, $\lambda=1, 3, 5, 7$). The values of these parameters are set in the software that is used to analyze the received data which is called A Bit Error Tester for Power Line Modems [30].

The format of a transmitted packet is shown in figure 17. The preamble is used by the data link layer of the receiver to facilitate the recognition of a high percentage of the incoming packets and to provide the receiver with a synchronization sequence on which
the receive clock (RxClk) can lock. Since the RxClk is not properly synchronized with the received bit stream until after the preamble has been processed, no data is clocked into the decoder before the moment of synchronization acquisition. For this reason, the preamble is not encoded before it is transmitted, hence creating the need for a procedure that will enable the EPLMs to bypass the codec when sending or receiving the preamble, and encoding or decoding immediately thereafter the packet header, data and trailer fields.

![Figure 17 Packet Format](image)

The preamble bypass procedure was realized by inserting an extra multiplexor (U91, appendix C) and reconfiguring one of the control pins (P1.2) of the microcontroller (U24), such that this signal is high during the duration of the preamble and low during the transmission of the rest of the packet. A D flip-flop (U93) was added at the data input of the encoder to synchronize edges the TxClk with the data signal.

The use of a $R=1/2$ code implies that for every data bit two channel bits will be transmitted, or conversely, for every two channel bits received only one decoded data bit will be produced. For this reason it is necessary to generate two clocks. The data clock (CLK) is half the frequency of the channel clock (2CLK) for both the encoder and decoder blocks. The flip-flops in U92 generate CLK from 2CLK. In future designs these flip-flops (U92, U93) may be included as part of the codec.

Similar to other convolutional decoding techniques, one of the most delicate problems that must be dealt with is that of synchronization, i.e., the necessity of distinguishing
between information and parity symbols. Since these two types of symbols are sent alternately on a single channel, the receiver must decommutate these bits in the proper order. For this purpose, pin P1.6 of the micro-controller (U24) has been configured as Clock Set (CLK SET), which enables the clock (CLK) at the transmitter and the receiver to be set high immediately after the preamble is finished. In this way, the start of the data is identified and the data clocks can be adjusted such that the receiving data clock phase agrees with the transmitting data clock phase and there is no uncertainty as to the location of information and parity bits.

Apart from these minor changes, the use of the codec is transparent to the user, since the operational conditions of the modems remain unchanged. The schematic diagrams of the enhanced BPSK modem are given in appendix C. The performance results shown in the following sections correspond to tests executed using the enhanced BPSK power line modem.

5.2 Test Parameters

To provide a better understanding of the performance results that will be provided later in this chapter, a detailed description of the test parameters is given first. The software used to obtain the data for statistical analysis is called A Bit Error Tester for Power Line Modems [30]. For test purposes, packets such as the one shown in figure 17 are sent from the transmitting host to the receiving host. The receiving host analyzes the packets using this software [30], which outputs the performance parameters.

During the reception of a packet, the preamble is stripped off by the data link layer, after which the host begins to examine the header. The header marks the beginning of a packet and is itself checked and terminated by a cyclic redundancy check (CRC) number, formed by its last two bytes. Following FEC decoding if no errors are detected in the header or in the trailer then the packet is analyzed; otherwise the packet is flagged as damaged. A detailed description of the parameter calculations that follow can be found in [31].
For test purposes a data field of 125 eight-bit characters was used. The test pattern is user selectable, in this case a string of ASCII characters was employed. The BER is computed by counting the number of error bits detected within the data field of the packets, divided by the number of received bits:

\[
BER = \frac{\text{# of Bits in Error}}{\text{# of Received Bits}}.
\]  

(28)

The BLKER is the number of blocks received with one or more errors in the data field, relative to the total number of received blocks:

\[
BLKER = \frac{\text{# of Blocks Received with Errors}}{\text{# of Received Blocks}} \times 100\%.
\]  

(29)

BLKER is one of the most important parameters. The BLKER indicates the number of packet retransmissions that an actual system would have to sustain, in order to achieve an error-free communication link. The number of retransmissions (RET) is given by:

\[
RET = \frac{BLKER}{1 - BLKER}.
\]  

(30)

For a valid comparison regarding the benefits of applying FEC, we must point out that with the use of a \( R=1/2 \) code, the actual data rate will be half of the actual transmission rate without encoding. Therefore in most of what follows we will relate encoded 19200 bps transmissions to unencoded 9600 bps transmissions, since in both cases the data rate after decoding, \( R_D \), will be the same.

The Overhead involved in the transmission of a data packet is the number of data bytes in an uncoded packet relative to the total number of bytes in the packet:

\[
\text{Overhead} = 1 - \left( \frac{\text{Data Bytes}}{\text{Preamble} + \text{Header} + \text{Data Bytes} + \text{Trailer} + \text{Decoding Delay}} \right)
\]  

(31)

The value of each of the elements that form part of (31) is:

- **Data Bytes**: A packet consists of 125 data bytes (1000 bits).

2 Although a string of ASCII characters was used, a set of pseudorandom data could have also been selected, previous tests have shown that the results using different data types are virtually identical.
• Preamble: 9 bytes in length. For communications with FEC the rate at which it is transmitted, is double that of the data rate, since it is the only part of the packet which is not encoded. For this reason the effective length in an encoded packet will be equivalent to 4.5 bytes.
• Header: 11 bytes long.
• Trailer: 2 bytes.
• Decoding Delay for each case is:
  I. Unencoded (Bypass FEC): 0 bytes.
  II. Encoded $\lambda=1$: 1 byte.
  III. Encoded $\lambda=3$: 3 bytes.
  IV. Encoded $\lambda=5$: 5 bytes.
  V. Encoded $\lambda=7$: 6 bytes.

Thus, the value of the Overhead factor for each instance is:

1. Unencoded : Overhead $= 1 - \frac{125}{147} = 1 - 0.8503 = 0.149$.
2. Encoded $\lambda=1$: Overhead $= 1 - \frac{125}{143.5} = 1 - 0.871 = 0.129$.
3. Encoded $\lambda=3$: Overhead $= 1 - \frac{125}{145.5} = 1 - 0.859 = 0.141$.
4. Encoded $\lambda=5$: Overhead $= 1 - \frac{125}{147.5} = 1 - 0.847 = 0.153$.
5. Encoded $\lambda=7$: Overhead $= 1 - \frac{125}{148.5} = 1 - 0.841 = 0.159$.

The differences in the values of the Overhead, is at most of 3%, which translates to a decrease in the throughput of up to 288 bps (9600 bps x 3%). It should also be noted that, surprisingly, the Overhead of encoded transmissions (for $\lambda=1$) is lower than that of unencoded transmissions by 2%, which will account for an increase in the throughput for coded data of up to 200 bps over unencoded data.

Throughput, $C$, is calculated as follows:

$$ C = \text{Bit Rate} \left(1 - \text{BLKER}\right) \left(\frac{\# \text{ of Received pkts}}{\# \text{ of Transmitted pkts}}\right) \left(1 - \text{Overhead}\right). \quad (32) $$

The error-free throughput is the most complete of all the indicators observed, since it takes into consideration all the different variables. On the other hand, since several
factors are considered for its calculation, variations of independent parameters may not be always easily detected.

To give a proper evaluation of the effectiveness of FEC, an explicit and complete analysis can be derived from examining the Throughput, $C$, of each system. From (32), one sees that all major factors involved in evaluating the network performance, are included when calculating its Error-Free Throughput. These parameters include: Data Rate, BLKER, % of Lost Packets, and Overhead.

Lost Packets are those which are not analyzed by the software (Bit Error Tester for Power Line Modems) [30]. One or more errors in the preamble, header or trailer of a packet, will result in a lost packet. The packet will not be recognized as valid by the receiver, and will not be processed, even though its data field might not contain any errors. To indicate the rate at which packets are lost the indicator % of Lost pkts is calculated as follows:

$$\text{% of Lost pkts} = \left(1 - \frac{\text{# of Received pkts}}{\text{# of Transmitted pkts}}\right) \times 100\%$$

A major issue that affects lost packet performance, is that the preamble, comprised of nine bytes, is not encoded. Consequently the percent of lost packets for coded transmissions at 19.2 Kbps is higher than for lower rates. This is one area for further work, to enhance the performance of EPLMs at high data rates.

Since the data rate of a system with FEC, using a $R=1/2$ code, is half of the actual channel transmission speed, encoded transmissions at 19.2 Kbps are compared to unencoded communications at 9.6 Kbps. However, the results obtained for uncoded transmissions at 19.2 Kbps are also shown in all the graphs, in this way providing a better understanding of the channel’s characteristics and the amount and types of errors which must be corrected to achieve reliable levels of error-free throughput.

5.3 Performance in Controlled Environments

In this section, the performance of the EPLM in two types of controlled channels is summarized. The improvement obtained by the use of FEC for each type of channel is
described. The channel consists of a 6 foot long copper cable, to which certain "error signals" are added. All tests were run for at least a 1000 packets. In some low BER cases, 10,000 packets were transmitted.

5.3.1 AWGN Channel

To simulate the effects corresponding to an AWGN channel, a white noise generator (Bruel & Kjaer 1405™ Noise Generator) is used. The white noise is added in a directed manner to the transmitted signal.

Figure 18 outlines the results obtained in these trials for: (a) BER and (b) BLKER versus $E_b/N_o$, where $E_b$ is the energy per bit and $N_o$ is the noise spectral density. The equations for determining these parameters are:

$$E_b = \frac{V_s^2}{R_D} ,$$

(34)

where $V_s^2$ is the received signal power and $R_D$ is the data rate after decoding, and

$$N_o = \frac{V_n^2}{W_R} ,$$

(35)

where $V_n^2$ is the received noise power and $W_R$ is the receiver noise bandwidth (in this case $W_R=80$ KHz). Both $V_s$ and $V_n$ in (34, 35) are true rms voltage values. To collect these measurements a Fluke 45™ Dual Display Voltmeter was used.

Figure 18 (a) shows that a coding gain of 1–4 dB can be obtained in AWGN channel (2dB for channel BER of $10^{-5}$). Using 1000 bit data packets and rate 1/2 convolutional coding, virtually no communication can be achieved on channels with BER approaching $10^{-2}$, since for channels with BER near $10^{-2}$ the preamble and header have a high probability of being corrupted, and hence, very few packets will be processed by the receiver. However, code combining and packet retransmission can provide for significant throughput even for channels with BER of $10^{-2}$. In part (b) of figure 18, the relationship with the BLKER is plotted. On average the BLKER of the encoded signal with interleave 7 will decrease by 33%. In both cases the encoded signals are only shown for $\lambda=1, 7$ in order to provide a less cluttered graph. There is no real loss of information, since the results for $\lambda=3, 5$ fall between the ones shown in figure 18. This statement applies
Figure 18 Enhanced modem performance for AWGN channel: (a) BER, (b) BLKER.
throughout most of this chapter, unless some specific regarding codec performance requires consideration of all degrees of interleaving.

The results in figure 18, are not explicit in regards to the effects of FEC on the BER, BLKER and throughput of a received bit stream. For this purpose the clearest relationship is shown in graphs where these values are displayed versus the uncoded channel BER. Since the data rate for coded 19.2 Kbps is 9.6 Kbps, the BER of uncoded transmissions at 9.6 Kbps is taken as the channel BER. However, the BER of uncoded transmissions at 19.2 Kbps is also shown for reference. The solid line in figure 18 represents the theoretical decoded bit error probability as expressed by (23). The theoretical decoded BLKER is calculated as follows:

$$BLKER_{Dec} = \left(1 - (1 - p)^L\right)100\% \approx (Lp)100\%,$$

(36)

where $L$ is the length of the data frame, and $p$ is the theoretical decoded bit error probability.

The graph in figure 19 shows the relationship between the channel BER and the decoded BER and BLKER. From graph 19 (a) and (b) it is evident that for channel BER $< 10^{-3}$ the decoded BER decreases noticeably, ranging from 1 to over 2 orders of magnitude in improvement. The BER for the uncoded 19.2 Kbps channel on the other hand is 1 to 2 orders of magnitude larger than that of the 9.6 Kbps channel BER. The BLKER is also reduced substantially, being on the average three times lower than without coding.

The theoretical decoded bit error probability for a random-error channel (represented by the solid line in figure 19 (a)), falls slightly below the practical decoded BER for $\lambda = 1$ (no interleaving), due to the fact that the theoretical decoded bit error probability as expressed by (23), is for a truly random channel, and considers all the previous bit estimates feedback into the syndrome register of the decoder to be correct. This is not always the case in practical realizations of the AWGN, where the generated noise patterns are not completely random. The spectrum of the white noise generator used is band-limited to 100 KHz, generating pseudorandom white noise. The practical decoded BER for $\lambda = 7$ on the other
Figure 19 Channel BER vs (a) Decoded BER, and (b) BLKER for AWGN Channel.
hand is very close to the theoretical value, since the interleaving further randomizes the errors generated by the pseudorandom AWGN generator.

5.3.2 Burst Error Channel

To evaluate the burst error correcting capabilities of the CODEC, a burst error channel was implemented, where the burst lengths and interarrival times could be controlled. The channel was designed in such a manner that the bursts would be periodic (this is a common effect seen on the power lines). For the duration of a burst the signal was severely attenuated (over 30 dB). A moderate level of background white noise was added constantly to the bit stream to generate conditions that would approximate those on an actual communication channel. After completing a wide range of tests, where both the interarrival times and the burst lengths were varied, it was concluded that the actual burst handling capabilities of the chip corresponded to its theoretical capacity.

The experiments done with maximum correctable length bursts were found to be the ones of most interest. In figure 20, the results of an experiment done on a burst error channel with 14 channel-bit (19.2 Kbps) burst errors are presented. It can be seen from figure 20 (a), that the codec with interleaving degree 7, starts correcting all the errors as soon as the channel BER drops below $2.0 \times 10^{-2}$, which in this case represents the maximum correctable error-burst frequency (196 Hz). This value may be obtained analytically by using the following arguments:

- A $(2, 1, 6)$ threshold decoder with $\lambda=1$, can correct 2 out of 14 channel-bit errors.
- If interleaving techniques are used and $\lambda=7$, such a decoder can correct 14 out of 98 channel-bit errors.
- For a 19.2 Kbps channel transmission speed, 98 bits represent $19200 \text{ KHz}/ 98 = 196 \text{ Hz}$.

Figure 20 also illustrates that for error bursts of length 14, neither $\lambda=1$, 3 or 5 is sufficient to eliminate all of the bursts. There is a considerable improvement for case $\lambda=5$, since in many cases a burst of length 14 will not necessarily corrupt all the bits it attenuates.
Figure 20 Channel BER vs Decoded BER (a), and BLKER (b) for a Bursty channel.
5.4 Performance over the Power-Line Channel

A wide series of tests were completed over various channels in the Electrical Engineering building. In general, the power lines in this building constitute a harsh communication environment. Very diverse types of loads are encountered, ranging from industrial motors and machinery to office equipment and computers.

The basic issues that are of concern for an appropriate understanding of the results obtained through our experiments are:

- The general power distribution system is divided into two three phase sub-systems, one for all the laboratories and machine shops and the other for all lighting and wall outlets. Each of these systems has an independent distribution bus. For ease in cross referencing we will use the terminology used in [31]; the first system will be referred to as Lab Sub-System and the second as General Sub-System.
- Power flows from the building’s three phase transformer to the General Sub-System, and from there, across a link, to the Lab Sub-System.
- The three phases of each of the systems will be denominated X, Y and Z.

Three different types of tests were completed on this network, providing ample range of channel quality:

I. Local Area communications within the Lab Sub-System, in this case communication links are limited to a room.

II. Extensive Area communications throughout the General Sub-System.

III. Extensive Area communications, with transmissions across the link interconnecting the Lab Sub-System and the General Sub-System.

5.4.1 Local Area Communications

The environment selected to run the local area communication tests was the Communications Laboratory (Rm. 458), where a large number of electronic equipment is constantly in use. The tests were performed over three different representative channels, X-Y, Y-Y and Z-Y, having in this way same phase and cross phase transmissions. Figures 21, 22 and 23 show the results obtained in this study.
Figure 21 Channel BER vs Decoded BER (a), and BLKER (b) for X to Y channel.
Figure 22 Channel BER vs Decoded BER (a), and BLKER (b) for Y to Y channel.
Figure 23 Channel BER vs Decoded BER (a), and BLKER (b) for Z to Y channel.
To obtain distinct channel BER and BLKER over the same channel, the transmitter's output voltage is varied accordingly, thus changing the signal to noise ratio of the signal. However, since the object of this research is to examine the effects of FEC on power-line communications, graphs that emphasize transmitter power are not shown, but it must be noted that to obtain the same BER over different links, the transmitter voltage must be increased for cross phase transmissions. For these trials the transmitter voltages varied in the following ranges: for X-Y from 61 to 72 dBmV, for Y-Y from 35 to 43 dBmV, and for Z-Y from 57 to 71 dBmV.

Comparing the results obtained in figures 21, 22 and 23 with the corresponding plots in figure 19, we can see that the total effect of all the channel impairments on these links is similar to that of a AWGN channel. In all cases, the results obtained show the positive effects that FEC has on the reliability of the data transmissions. On average the BER can be reduced from one to two orders of magnitude (for uncoded channel BER of $10^{-3}$ and $10^{-5}$, respectively). These graphs indicate decreases in the BLKER ranging from 30% to over 100% of the original channel BLKERS for unencoded transmissions. By reducing the need for packet retransmissions drastically in these cases, a substantial increase in the general throughput of the system results.

5.4.2 Extensive Area Communications over the General Sub-System and across Sub-Systems

It should also be pointed out that there are great variations in the behavior of each of the channels during different times of the day. For this reason the extensive area communication tests were done in a different manner. The fluctuations of the characteristics of each channel over 24 hours, as well as the effect of error control techniques in each case, was observed and recorded. The results obtained from these of tests provide a practical indication of the performance of intrabuilding power line communications over a wide range of different communication links. As stated earlier the communications links were established over channels of a single power sub-system, as well as connections across both power sub-systems.
**Test Procedure.** The tests were conducted in the following manner: the transmitting modem was connected to either the wall outlet (*General Sub-System*) or to the *Y* phase of the power bars (*Lab Sub-System*) in room 458. The receiver was moved among different locations across the building, in this way providing a wide range of channels with varied characteristics. Since the PLC is a highly time-variant channel, the trials were executed for 24 hours in each site, and averages over this period were taken for evaluation purposes.

The locations where the receiver was situated were the following:

- Room 113 (*Z Phase*).
- Room 214 (*Y Phase*).
- 3rd Floor Stairwell (*X Phase*).
- 4th Floor Stairwell (*X Phase*).
- Room 402 (*Y Phase*).

Due to the routing of the wiring within the building [32], physical proximity of a transmitter-receiver pair does not necessarily imply a shorter communication channel. Actually, since the main distribution buses are located in the basement, proximity between transmitter and receiver often results in very long communication paths. Examples that illustrate these cases can be found in [31]. A summary of the structure of this power-line network is given in [31], and detailed plans of the wiring layout may be found in [32].

Figures 24, 25, 26, and 27 were selected as being the most representative of the results obtained during the extensive area transmissions. These results represent performance of the “best” and “worst” channels. In each case the tests were duplicated, once for transmissions within the *general sub-system* and the other for communications across *sub-systems*.

It can be seen from the graphs in figures 24–27, that the channels that have links across *power sub-systems* are of considerably lower quality than those that are limited to just one of the *sub-systems*. For unencoded transmissions at 19.2 Kbps the BER is relatively constant, fluctuating between $10^{-2}$ and $10^{-3}$, even for very “clean” channels. An example of this is the one between Rm 458X and the 3rd Floor stairwell, where the $BER_{9600}$ was under $10^{-7}$ the $BER_{19200}$ rarely dropped below $10^{-3}$.
Figure 24 BER as a function of time for transmissions from Rm 458 to Rm 214.

(a) Transmissions over the General Power Sub-system,
(b) Transmissions across Power Sub-systems.
Figure 25 Throughput as a function of time for transmissions from Rm 458 to Rm 214.

(a) Transmissions over the General Power Sub-system,

(b) Transmissions across Power Sub-systems.
Figure 26 BER as a function of time for transmissions from Rm 458 to the 3rd floor stairwell.

(a) Transmissions over the General Power Sub-system,
(b) Transmissions across Power Sub-systems.
Figure 27 Throughput as a function of time for transmissions from Rm 458 to the 3rd floor stairwell.

(a) Transmissions over the General Power Sub-system,

(b) Transmissions across Power Sub-systems.
For very noisy links, FEC with interleaving degree 7 proved to be the most effective scheme. For channels such as those between Rm 458Y–Rm 214 and Rm 458Y–3rd Floor stairwell (Figures 25 and 27) a considerable increase in the throughput can be obtained in this way, achieving an average throughput of over 1600 bps over links that previously were virtually unusable.

Since the PLC is a highly variable channel, the BER between two links may have fluctuations anywhere in the range from $10^{-2}$ to $10^{-8}$, over a 24 hour period. For this reason the average BER is not a parameter by which the effectiveness of FEC can be judged, since the weight of one high BER sample (e.g. $10^{-2}$) will out weigh a series of low BER samples (e.g. $10^{-8}$) obtained throughout the rest of the day. The average BLKER will give a much better idea of the real performance of the system throughout the test period. The average BLKER over a 24 hour period for each of the trial channels is shown in figure 28. The results show that on average the BLKER of encoded 19.2 Kbps transmissions (9.25%) is three times lower than that of uncoded 9.6 Kbps transmissions (30.18%), and seven times lower than that of uncoded 19.2 Kbps transmissions (71.67%).

Table 1 provides a summary of the average throughput for each communication link. Table 1 shows the error-free throughput of the system taking into account the percent of lost packets, which is due primarily to the use of an uncoded preamble even with the use of FEC. In table 2 the resulting error-free throughput is shown, considering the percent of lost packets to be zero.
X-Denotes transmissions over the general power sub-system.

Y-Denotes transmissions across power sub-systems.

Figure 28 Average BLKER (over 24 hours) for each of the tested channels.
<table>
<thead>
<tr>
<th>Mode/Location</th>
<th>Mode/Location</th>
<th>Mode/Location</th>
<th>Mode/Location</th>
<th>Mode/Location</th>
<th>Mode/Location</th>
<th>Mode/Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>9600 Bypass</td>
<td>9600 Bypass</td>
<td>9600 Bypass</td>
<td>9600 Bypass</td>
<td>9600 Bypass</td>
<td>9600 Bypass</td>
<td></td>
</tr>
<tr>
<td>19200 Bypass</td>
<td>19200 Intlv=1</td>
<td>19200 Intlv=3</td>
<td>19200 Intlv=5</td>
<td>19200 Intlv=7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rm 113*</td>
<td>6124</td>
<td>1102</td>
<td>5813</td>
<td>5495</td>
<td>5633</td>
<td>5998</td>
</tr>
<tr>
<td>Rm 113‡</td>
<td>7863</td>
<td>4057</td>
<td>8077</td>
<td>7936</td>
<td>7808</td>
<td>7797</td>
</tr>
<tr>
<td>Rm 214*</td>
<td>6815</td>
<td>2437</td>
<td>7904</td>
<td>7736</td>
<td>7720</td>
<td>7677</td>
</tr>
<tr>
<td>Rm 214‡</td>
<td>5412</td>
<td>2506</td>
<td>7222</td>
<td>7162</td>
<td>7150</td>
<td>7081</td>
</tr>
<tr>
<td>Stairs 3rd*</td>
<td>8273</td>
<td>8455</td>
<td>8407</td>
<td>8281*</td>
<td>8188</td>
<td>8130</td>
</tr>
<tr>
<td>Stairs 3rd‡</td>
<td>441.1</td>
<td>2.366</td>
<td>1479</td>
<td>1847</td>
<td>2515</td>
<td>2803</td>
</tr>
<tr>
<td>Stairs 4th*</td>
<td>8246</td>
<td>6729</td>
<td>8365</td>
<td>8224</td>
<td>8128</td>
<td>8071</td>
</tr>
<tr>
<td>Stairs 4th‡</td>
<td>2256</td>
<td>67.28</td>
<td>5043</td>
<td>4706</td>
<td>4625</td>
<td>4312</td>
</tr>
<tr>
<td>Rm 402*</td>
<td>8274</td>
<td>15297</td>
<td>8246</td>
<td>8166</td>
<td>8019</td>
<td>8023</td>
</tr>
<tr>
<td>Rm 402‡</td>
<td>0.000</td>
<td>0.000</td>
<td>1097</td>
<td>1037</td>
<td>1700</td>
<td>1651</td>
</tr>
<tr>
<td>Average</td>
<td>5370</td>
<td>4065</td>
<td>6165</td>
<td>6059</td>
<td>6148</td>
<td>6155</td>
</tr>
</tbody>
</table>

Table 1 Average Error-Free Throughput for communications from room 458.
Transmissions over the General Sub-system are marked *.
Transmissions across Sub-systems are marked ‡.
<table>
<thead>
<tr>
<th>Mode/Location</th>
<th>Location</th>
<th>9600 Bypass</th>
<th>19200 Bypass</th>
<th>19200 Intvl=1</th>
<th>19200 Intvl=3</th>
<th>19200 Intvl=5</th>
<th>19200 Intvl=7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rm 113*</td>
<td></td>
<td>6829</td>
<td>2036</td>
<td>7815</td>
<td>7819</td>
<td>7588</td>
<td>7702</td>
</tr>
<tr>
<td>Rm 113‡</td>
<td></td>
<td>7912</td>
<td>4985</td>
<td>8478</td>
<td>8359</td>
<td>8245</td>
<td>8190</td>
</tr>
<tr>
<td>Rm 214*</td>
<td></td>
<td>7062</td>
<td>3266</td>
<td>8369</td>
<td>8253</td>
<td>8215</td>
<td>8157</td>
</tr>
<tr>
<td>Rm 214‡</td>
<td></td>
<td>6446</td>
<td>3666</td>
<td>8322</td>
<td>8261</td>
<td>8176</td>
<td>8080</td>
</tr>
<tr>
<td>Stairs 3rd*</td>
<td></td>
<td>8275</td>
<td>9243</td>
<td>8480</td>
<td>8362</td>
<td>8247</td>
<td>8191</td>
</tr>
<tr>
<td>Stairs 3rd‡</td>
<td></td>
<td>889</td>
<td>2.4</td>
<td>4297</td>
<td>5562</td>
<td>5472</td>
<td>5598</td>
</tr>
<tr>
<td>Stairs 4th*</td>
<td></td>
<td>8250</td>
<td>7714</td>
<td>8480</td>
<td>8361</td>
<td>8247</td>
<td>8191</td>
</tr>
<tr>
<td>Stairs 4th‡</td>
<td></td>
<td>3813</td>
<td>170</td>
<td>7725</td>
<td>7527</td>
<td>7472</td>
<td>7175</td>
</tr>
<tr>
<td>Rm 402*</td>
<td></td>
<td>8275</td>
<td>15787</td>
<td>8480</td>
<td>8362</td>
<td>8247</td>
<td>8191</td>
</tr>
<tr>
<td>Rm 402‡</td>
<td></td>
<td>0.000</td>
<td>0.000</td>
<td>3006</td>
<td>3330</td>
<td>4744</td>
<td>4852</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>5775</td>
<td>4686</td>
<td>7345</td>
<td>7420</td>
<td>7465</td>
<td>7432</td>
</tr>
</tbody>
</table>

Table 2. Average Error-Free Throughput for communications from room 458. % of Lost Packets = 0. Transmissions over the General Sub-system are marked *. Transmissions across Sub-systems are marked ‡.
Summarizing the results obtained in the experiments that were completed as part of this research, the following conclusions may be drawn:

I. Forward Error Correction proved to be an effective way to increase the Error-Free Throughput of Intrabuilding Power Line Communications, providing in this manner the possibility of maintaining reliable communications over channels with highly variable characteristics.

II. The application of FEC is extremely effective for very hostile environments. In such cases, the burst error-correcting capabilities of the CODEC are utilized extensively, making the employment of such features indispensable for obtaining reliable communications.

III. Even for relatively “clean” channels it is more efficient to transmit at higher speeds with FEC, rather than at unencoded lower data rates. The overall throughput will be higher due to the reduction in overhead by 2%, providing in this way an increase of approximately 200 bps.

IV. The throughput of the EPLM increased by an average of 15% with the use of error-control techniques for the 10 test locations. If methods that would allow external synchronization were to be used, (thus permitting the preamble to be encoded), an additional 15% increase could be achieved, obtaining a total increment of up to 30% in the error-free data rate.

V. The use of coding with higher levels of interleaving, (λ=5, 7), in many cases did not result in superior performance over that of FEC with λ=1. This puts in evidence the fact, that in many cases, the compounded effect of all the error sources on the PLC will be similar to that of random noise. However, in the cases where there were a large percentage of burst-errors, interleaving increased considerably the performance of the system.

VI. An increase of the channel transmission speed by a factor of 2, will result on average, in an increase in the BER of 10 to 100 times for unencoded transmissions, resulting also in a substantial increase in the BLKER.
VII. The packet throughput (the amount of packets that are received and processed) decreases by 10% to 70% when 19.2 Kbps communications are used. The use of FEC limits this loss of packets, but does not eliminate the problem completely, since the preamble is transmitted without encoding. The use of encoded preambles would increase the packet throughput by another 15% to 20%.

VIII. The effect of local error sources, close to the receiver, can be substantial because the received signal can be severely attenuated and faded; consequently local noise of relatively low power (such as the host computer) can be the origin of a large number of errors.
Chapter 6
Summary

6.1 Concluding Remarks

This thesis summarizes the research involved in the design, implementation, fabrication, testing and evaluation of a VLSI convolutional encoder and threshold decoder. The focus throughout the development of the codec was to achieve a design with relatively low complexity that could provide the necessary performance for its application in intrabuilding power line communications.

The codec consists of a rate $1/2 (2, 1, 6)$ convolutional encoder and threshold decoder, with programmable degrees of interleaving. Threshold (majority-logic) decoding was selected due to the ease of its implementation, and because the algorithm is very well suited for use with interleaving, making it easily extendable to correct error bursts. Performance capability and implementation simplicity make threshold decoding an attractive alternative to more complex and expensive decoders, such as Viterbi decoders.

The main features of the developed FEC chip are:

1. Random and burst error correcting capabilities (for $\lambda=1, 2$ errors of every 14 channel bits can be corrected; or for $\lambda=7$, error bursts of up to 14 channel bits can be corrected).
2. Automatic error resynchronization properties with respect to error propagation.
3. Programmable degrees of interleaving ($\lambda=1, 3, 5, 7$).
5. Small decoding delay compared to other methods.
6. Independent encoder and decoder circuits on one chip, supporting full duplex communications.
7. Built-In Self-Test that provides 100% single stuck-at fault coverage, with under 10% of area overhead.
8. Total core area of 2.32 mm$^2$, which represents less than 10 to 50 times the area of Viterbi decoders of similar characteristics [33].

The VLSI codec was fabricated using 1.2$\mu$m CMOS technology and is a semi-custom design that uses standard CMOS4S CMC library cells. Its total complexity is 1,600 gates (5K transistors), with a core area of 2.32 mm$^2$. This area includes the BIST and interleaving circuitry.

To evaluate the benefits of FEC in power line communications, the codec was interfaced to two existing PLMs. A wide range of tests were performed to determine the modem’s performance with and without FEC. Trials were carried out over channels with controlled error sources, and over actual operational power line channels. Tests were completed for local area communications as well as for extensive area transmissions. Original data was collected which permitted a statistical analysis of the BER, BLKER, percent of lost packets, and throughput in every case.

The results of this study show that the use of forward error correction is an effective method of increasing the Error-Free Throughput of intrabuilding power line communications, providing in this manner the possibility of maintaining reliable communications over channels that could otherwise not be used for data communications at 9.6Kbps and 19.2Kbps. The positive effects of FEC are prominent during transmissions over very hostile environment, where error bursts are very frequent due to channel impairments. Even for communications over “clean” channels, the use of a higher data rate combined with FEC proved to be slightly more efficient than transmitting at a lower data rate.

Throughout this research a wide range of channels with different characteristics were used. A common factor in all circumstances was that the increase of the channel transmission speed by a factor of 2, will result on average, in an increase in the BER by 1 to 2 orders of magnitude for uncoded transmissions. However, with the use of this type of error-control technique an increase in the throughput of the system, ranging from a minimum of 10% to several times that of uncoded transmissions, can be obtained. This is primarily due to a steady decrease in the BLKER for coded communications, which lowers the number of packet retransmissions due to corrupted data.
The test results also indicate that the frequent presence of error bursts is not a common factor to all power line channels. For this reason the use of higher degrees of interleaving (\(\lambda=5, 7\)) will not always provide the best results. This is proof of the need of a codec that can handle variable interleaving degrees.

### 6.2 Suggestions for Future Research

Enhanced power line modems can reliably support high speed power line communications at 19.2 Kbps in relatively hostile environments. This is due in part to two recent, significant developments: namely the design and fabrication of an all-digital MSK modem chip, and of a low cost FEC codec chip. At present both these chips, together with a micro-controller form the EPLM, which at this stage is already small enough to be implemented as an internal expansion card of a personal computer (PC).

The reliability of power line communications has increased substantially due to the fact that FEC is now an integral part of the modem, and also because the number of external interconnections (wire-wrapping or soldering) has drastically decreased. Many of the modem functions are implemented by these three major chips.

There are several enhancements which can be performed to the existing EPLMs. The first and most noticeable would be the unification of the modem and the codec onto one single chip, increasing further the level of integration and component reliability.

The second enhancement would be to find alternatives that would allow the encoding of the preamble. This could be achieved by using some type of external synchronization technique, and would generate an increase in the throughput of the system, probably by more than 15%.

A third possible improvement would be the use of a faster microcontroller to enable the EPLM to double its bit rate to 38.4 Kbps. However, the use a higher speed could be accompanied by other detrimental effects, such as unreasonably high BER and BLKER, which would decrease the overall throughput of the system. A complete evaluation of the system's performance, together with the effects of FEC in different operational environments would have to be repeated for this case.
From a statistical and practical point of view, further studies should be done which could define the limits (worst cases), for which the EPLMs can be used. Focal points of such work could be directed towards finding maximum in-building distances for reliable communications, with the use of the maximum allowable transmitter signal power, and evaluation of the performance of coded 38.4 Kbps communications versus uncoded 19.2 Kbps transmissions.

These above suggestions are not exhaustive. Much work could be done in related areas. Examples include development of means for eliminating the local noise generated by the host computer, the development of the higher level protocol layers, also conduct a wider and more extensive series of tests, in different buildings with different characteristics. Use of some sort of adaptive modulation/FEC coding is also of interest, to optimize throughput performance under conditions of varying channel quality. Recent work reported in [4] is a viable approach, in which the use of FEC together with packet retransmissions and code combining is proposed, as a mean to obtain reliable communications even in very hostile environments.
Bibliography


  Automatic Place and Route, Vol. I & II.

  1990.


[29] R. M. Sedmak. Built-In Self-Test Pass or Fail? IEEE Design & Test of Computers,

  Unpublished In-House Document, University of British Columbia, Dept. of Electrical

  Architecture for Power Line Communications. Master's thesis, University of British

  Pratt Architects, Structural, Electrical, Plumbing, and heating Plans for the Electrical

[33] G. Fettweis and H. Meyr. High Speed Parallel Viterbi Decoding: Algorithm and VLSI-

[34] V. Cappellini. Data Compression and Error control Techniques with Applications,


Appendix A  Codec Schematic and Layout Representations

PRUEBA CODEC 1  
ENCODER

DECODER

Figure 29  Schematic diagram of the (2,1,6) convolutional encoder and threshold decoder with fixed interleaving degree ($\lambda=7$).
Figure 30 Layout representation of the (2,1,6) convolutional encoder and threshold decoder with fixed interleaving degree ($\lambda=7$). Design identification name BCCOD. (Only the metal layers are shown).
VAR INTERLEAVE CODEC2
ENCODER

DECODER

Figure 31 Schematic diagram of the (2,1,6) convolutional encoder and threshold decoder with programmable interleaving degrees ($\lambda=1, 3, 5, 7$).
Figure 32 Layout representation of the (2,1,6) convolutional encoder and threshold decoder with programmable interleaving degrees ($\lambda = 1, 3, 5, 7$).

Design identification name BCMAJ. (Only the metal layers are shown).
Figure 33: Schematic diagram of the (2,1,6) convolutional encoder and threshold decoder with programmable interleaving degrees (λ = 1, 3, 5, 7) and BIST.

BISTCODEC CODEC3
ENCODER

DECODER
Figure 34 Layout representation of the (2,1,6) convolutional encoder and threshold decoder with programmable interleaving degrees ($\lambda = 1, 3, 5, 7$) and BIST. Design identification name BCBST. (Only the metal layers are shown).
Appendix B  Codec Bonding Diagrams

Figure 35  68-Pin PGA (a) pin numbering scheme, and (b) bonding diagram.
no substrate connections required, all pins should be floating

<table>
<thead>
<tr>
<th>Wire Alloy</th>
<th>Dia.</th>
<th>Elong.</th>
<th>T.S.</th>
<th>W/B Method</th>
<th>U.S.</th>
</tr>
</thead>
<tbody>
<tr>
<td>99% Al/1% Si</td>
<td>.001&quot;</td>
<td>1.5 - 4%</td>
<td>14-16 gms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>99% Al/1% Si</td>
<td>.00125&quot;</td>
<td>1.5 - 4%</td>
<td>18-22 gms</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

D/A PREFORM Alloy 98% Au/2% Si RECOMMENDED SIZE

NOTES:
1. Die attach pad size: .400 x .400
2. Zero ground

Figure 36 BCCOD bonding diagram.
Figure 37 BCMAJ bonding diagram.
Figure 38 BCBST bonding diagram.
Appendix C  Enhanced PLM Schematic Diagrams

The schematic diagrams of the EPLM given in this appendix are the sections in which changes were made to the BPSK PLM developed by Buternowsky. Schematics of other functional blocks not related to the encoding/decoding process may be found in [31].

The schematic diagrams that are shown on the following pages are:

1. Functional block diagram of the EPLM.
2. Enhanced power line modem control block.
3. Enhanced power line modem PSK modulator block.
4. Enhanced power line modem FEC encoder/decoder block.