Analog Viterbi Detection

for

Partial-Response Signaling

by

Mohammad Hossein Shakiba

A thesis submitted in conformity with the requirements for
the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

University of Toronto

1997

0-612-27719-4

Abstract

Analog Viterbi Detection for Partial-Response Signaling
Mohammad Hossein Shakiba, Ph.D. Dissertation, 1997

Along with the growth of signal processing capabilities, sequence detection of digital signals transmitted over noisy channels has become the preferred choice in many applications. Consequently, researchers accelerated their efforts toward addressing the implementation issues of such detectors. Naturally, almost all of the solutions were developed in a digital realization environment, mainly because digital signal processing has been shown to be powerful and flexible. At the same time, the idea of analog implementations of sequence detectors was introduced by a small dedicated minority in the hope of finding areas where digital solutions fail to fulfil some of the system requirements. This hope became a reality when analog Viterbi decoders outperformed their digital counterparts in the exceptionally demanding saturated magnetic storage application. The challenge was to realize the Viterbi algorithm such that the ever increasing requirements of small size, low power, and high speed are satisfied.

Described in this thesis, is another attempt for realizing the Viterbi algorithm in the analog domain. Partial-response sequence detectors with application to magnetic recording and data transmission over cables are of special interest, however, other subjects are also addressed. Although the essence of an analog realization is to eliminate the power-hungry analog-to-digital converter, here it is shown that additional savings may also be accomplished if the algorithm is carefully examined from an analog implementation perspective. In particular, in this thesis, an analog architecture for realizing a class-IV partial-response Viterbi decoder is introduced, integrated circuit implementation of this decoder is described, and experimental results are presented. The operating speed and savings in the silicon and power consumption are well beyond the reach of any reported digital decoder, even in more advanced technologies in many cases. Furthermore, and supported by experiments, it is shown that more complicated signaling schemes can also benefit from an analog implementation. The idea was to alleviate the transistor-level obstacles and encourage analog designers to explore new territories for analog sequence detectors. Finally, some implementation issues of reduced-state sequence decoders are addressed and analog solutions are introduced.

Acknowledgments

Many thanks to my supervisors Prof. David Johns and Prof. Ken Martin for their genuine support and encouragement. Without their dedication, this work wouldn't have received this much interest and recognition. Thanks for giving me the freedom to follow the paths of my choice. Your gracious manner has made an enduring impression on me.

To Prof. Glenn Gulak and Dr. Mohammad Nejad, the internal and external appraisers, and other members of my doctoral committee, Profs. W. Ng, B. Francis, M. Van de panne, and P. Martin for their advice and guidance.

I am grateful to have been a part of the EA104 lab at U of T from when it started. Thanks to the "gang", past and present, for creating such a wonderful atmosphere. I'll always remember the friendly chats we had, specially those long ones with Greg whose sad departure coincided this writing. Greg, rest in peace. My sincere thanks to Steve Jantzi, Ralph Duncan, Bryn Owen, Karen Kozma, Ayal Shoval, and Khoman Phang for all those technical and non-technical sharings. To Bob Richens for his valuable help. Bob, that HP plotter worked right on time.

I am indebted to Prof. Frank Kschischang for his invaluable inputs throughout this research, to Dr. Tirdad Sowlati for revising the manuscript, and to Jasmine Cheng for the FPGA implementation.

To my parents who are so much a part of me. Although it is a long time I have left home, I still feel their support and care I received growing up. Thanks for devotedly giving me your continual support. I don't know where I'd be without it. I just hope I deserve it. So, thanks Ma and Pa for everything. To my Grandpa who shaped me and set me off on my life journey to stay strong. Grandpa, from your little kid, thank you. I miss you a lot. To my sister for her pure love. Thanks for always being there and looking after most of my responsibilities. I've always wished I could've been as much of a brother for you as you've been a sister for me.

To my wife, Narges, my daughter, Nika, and my son, Kia, for being with me all the time. You received all the downsides of my Ph.D. work. Nika and Kia, thanks for making our home moments full of joy, hope, and happiness. It was very early for you to experience the difficulty of achieving a high goal in the life, and you kids were fantastic. I thank you for your amazing understanding. I hope I can spend more time with you now. Narges, I'll never forget the motivation you gave me to

enter this new territory, the price you sincerely paid for it, and the amount of suffer you delightfully accepted during the past difficult years to let our family successfully finish the journey. If it wasn't because of you, I would've never been able to initiate the move and if it wasn't because of your intense care and compassionate support I would've never been able to go this far. You indeed shared with me the good and the bad. I hope sometime I'll have the opportunity to do the same for you. I owe you everything. So, from the bottom of my heart, thank you.

Table of Contents

CHAPTER3

Maximum-Likelihood Sequence Detection 24

CHAPTER4

Class-IV Partial-Response Analog Viterbi Decoder 43

List of Tables

List of Figures

Glossary of Terms

A/D	Analog-to-Digital converter
ACS	Add-Compare-Select
AWGN	Additive White Gaussian Noise
BER	Bit-Error Rate
CM	Common Mode
CMC	Canadian Microelectronics Corporation
CMFB	Common-Mode FeedBack
DFE	Decision-Feedback Equalizer
DUT	Device Under Test
EPR	Extended Partial Response
FCC	Federal Communications Commission
IC	Integrated Circuit
ISI	Inter-Symbol Interference
ML	Maximum Likelihood
MLSD	Maximum-Likelihood Sequence Detection
NEXT	Near-End cross-Talk
NT	Northern Telecom
pdf	probability density function
PR	Partial Response
PR4	class-IV PRS
PRML	Partial-Response Maximum Likelihood

PRS	Partial-Response Signaling
QPRIV	Quaternary class-IV PRS
RSSD	Reduced-State Sequence Detection
S/H	Sample and Hold
SER	Symbol-Error Rate
SNR	Signal-to-Noise Ratio
UTP	Unshielded Twisted Pair
V/I	Voltage-to-current converter
VA	Viterbi Algorithm

Chapter 1

Introduction

The traditional approach in detecting a digital signal transmitted over a noisy channel is the well-known symbol-by-symbol detection technique. In addition to the simplicity of implementation, in many cases this scheme results in optimum performance. However, if each received symbol contains some information about others, a symbol-by-symbol approach will no-longer lead to the optimum detector. The optimum detector must consider the entire segment of the sequence which conveys helpful information, in detecting each individual symbol. The drawbacks are, however, increased complexity and delay in the detection process.

Digital communication systems employing error-correction coding are examples of systems in which the optimum decoders are sequence detectors. These sequence detectors are usually implemented by realizing different variations of the Viterbi algorithm. The Viterbi algorithm is a special application of dynamic programming to communication theory. The idea is to determine the most-likely transmitted sequence having the minimum distance from the received signal, as compared to all other possibly-transmitted sequences. The objective function to be minimized depends on the error criterion and is commonly the squared-Euclidean distance.

The Viterbi algorithm was first proposed for decoding convolutional codes. Later on, it was extended to the detection of signals transmitted over linear inter-symbol interference channels. Consequently, partial-response systems, being interference signaling schemes in nature, received renewed attention. A partial-response system is a signaling system in which the signal undergoes some known inter-symbol interference before being transmitted. The idea is to shape the spectrum of the signal by relaxing the highly-demanding zero-interference condition. Spectrally shaping makes better use of the available bandwidth possible, since the transmitted energy can be more concentrated around the frequency bands in which the channel has better characteristics.

One illustrative example of a partial-response system is a system which pushes the energy of the signal away from DC, where many channels fail to establish a reliable link. A single spectral null at DC can be realized by differentiating the signal prior to transmission. In the discrete-time domain, this corresponds to subtracting each symbol from its previous one. The resulting system is called a *dicode* in the literature. If, in addition, high-frequency concentration of the dicode system is not desirable, a complementary low-pass block can be employed as well. A first order realization of this block, namely *duobinary*, is obtained by adding two successive input symbols. The overall signaling scheme is classified as *class-IV* in classification of the partial-response systems. One can take advantage of the high-frequency null to maximize the symbol rate in the given bandwidth in practice. Due to the aforementioned advantages, the class-IV system has been one of the most-widely-used partial-response signaling schemes so far.

Partial-response systems are multi-level communication systems, in the sense that they increase the number of levels at their outputs. This increase was the reason for the initial reputation of their poorer noise performance compared to Nyquist systems. Later on, it was shown that this poorer performance is not an inherent drawback, but, is due to the non-optimality of symbol-by-symbol detection. In fact, if the coding property of these signaling schemes is exploited by the detector, the loss in the signal-to-noise ratio can be combatted.

Not as obvious as their spectrally-shaping behavior, is the coding nature of the partial-response systems. Availability of more levels at the output, without an increase in the symbol rate, is equivalent to having some redundant levels. Utilizing this redundancy can significantly improve the performance. This improvement is maximized by employing a sequence detector. The price paid, however, is mainly more complexity in the receiver. The decoding delay associated with sequence detection can be tolerated in many applications.

Partial-response signaling applied to magnetic recording has drawn a lot of attention nowadays. Traditionally, the written information was retrieved by detecting the peaks of the read signal. To keep the error rate below a certain level, the adjacent symbols had to be far enough apart not to interact significantly. This restriction had imposed a limit on the density of the storage system. Viewing the read signal as a partial-response signal allows much more interference between the adjacent transitions. Moreover, if the coding nature of the read signal is exploited, a much better noise performance can be achieved. The better performance can be translated to the capability of increasing the density, since more interference is tolerated. Employing partial-response sequence detection techniques in the new generation of hard disks is one reason for the drastic increase observed in the amount of memory offered by these devices.

Magnetic recording is not the only promising application of partial-response systems with sequence detection these days. High-rate data transmission over band-limited channels is considered as well. Apart from achieving the Nyquist rate (which is not practically feasible in Nyquist systems), more increase in the symbol-rate is still possible by using M-ary partial response schemes. Here, again, a sequence detector substantially improves the performance of the communication system compared to that of systems with conventional symbol-by-symbol detectors.

The above arguments have motivated researchers to search for simple implementations of the Viterbi algorithm in general, and for partial-response systems in particular. These efforts have led to several implementation techniques. Coincidence of these efforts with the growing digital signal processing methods directed almost every effort toward a digital solution. However, for applications demanding high speed, low power, and small size, and not requiring a high degree of accuracy, the possibility of implementing the algorithm in the analog domain has also been investigated. Analog Viterbi detectors are extremely suitable if the signal processing prior to the Viterbi detection is relatively simple and could be done in the analog domain as well. In these cases, the savings become significant, as there would be no need for an analog-to-digital converter in the detector. Magnetic recording and high-rate data transmission over unshielded twisted-pair cables are two such examples. Recently, the magnetic-recording industry has shown a lot of interest in pursuing this viable alternative and many state-of-the-art computer disk drives now employ analog Viterbi detectors in their read channels[1]. Utilizing an analog Viterbi decoder has also been

[1] The first commercial implementation of an analog Viterbi detector within a magnetic channel was reported during the course of this thesis work.

considered for small size, low-power, low-cost, and high-rate transceivers for transmitting data over $100m$ of unshielded twisted-pair cables. These communication links are proposed to distribute information in a fiber-distributed data interface system.

The present work is a continuation of the attempts made for realizing the Viterbi algorithm in the analog domain. Since analog implementations seem to benefit today's most important partial-response applications, special attention has been directed toward these systems. However, other systems have not been forgotten. This thesis is organized as follows:

In the next chapter, the general idea behind partial-response signaling and its differences with the more-familiar Nyquist systems are described. The system model used throughout this thesis is introduced and some practically-important partial-response schemes are explained. Detection techniques for decoding partial-response signals are discussed with more emphasis on the decision-feedback equalization, leaving the sequence detection for a separate chapter. A decision-feedback equalizer is categorized as a symbol-by-symbol detector and its performance is frequently compared with that of the sequence detector. An expression for the symbol-error rate in a first-order system[1] is derived which includes the effect of the error propagation associated with the feedback. Being a good candidate for the magnetic-recording read channels and possessing a sequence detector well-suited for an analog implementation, a class-IV scheme is emphasized throughout the chapter. At the end of the chapter, two important applications of partial-response signaling, namely magnetic recording and high-rate data transmission, are explained.

Chapter 3 describes the maximum-likelihood sequence detection technique realized by the Viterbi algorithm. It then considers the detection problem in a two-state Viterbi decoder and introduces a general difference-metric approach. The approach is applied to the detection of a first-order partial-response signal. A special version of this latter algorithm and its digital realizations have already been known for decoding a dicode signal. Our approach, however, is different in the sense that an analog implementation has been desired from the first steps of derivations. From the analog-implementation stand-point, the dicode difference-metric algorithm is further developed and a new derivation is obtained. The proposed algorithm is referred to as the "*input-interleaved algorithm*", as it results in an input-interleaved structure when implemented in the analog domain. The implementation issues will be explained later in the next chapter. Also, as will be shown in the next chapter, the input-interleaved decoder results in a high-speed realization. Chapter 3 proceeds with explaining a useful approach to determine the conditions under which partial-response codes

[1] A first order partial-response system is a system which involves only one inter-symbol interference term.

satisfy the full recovery of the associated losses in the signal-to-noise ratios. These are named "*maximal-distance codes*" in this thesis. Although not all of the maximal-distance codes are useful in practice, the concept is used to show that partial-response systems do not inherently perform worse than the uncoded systems. The idea also helps us to illustrate why not all of the codes proposed to be used in the magnetic-recording systems seem helpful in practice. The chapter ends with discussing some of the important practical issues a Viterbi decoder faces. The specific analog imperfections are left to the next chapter, where the analog realization is exclusively targeted.

Chapter 4 is one of the main contributions of the thesis. In this chapter, the issue of an analog implementation of a class-IV Viterbi decoder is discussed. Similar to digital realizations, the decoder can be realized by time-interleaving two dicode decoders. Since only 6-bit accuracy is required, simple and fast analog circuits can be employed. The difference-metric and the input-interleaved algorithms can be used to implement each dicode decoder. It is shown that the difference-metric algorithm results in a structure very similar to the threshold device. The differences are adaptive adjustment of the threshold levels and existence of the path memory in the sequence detector. Due to this similarity, the detector is called an "*adaptive-threshold detector*". Consequently, it is shown that the complexity of the decoder is much less than that of the analog-to-digital converter, by itself, in a digital realization. The structure is fast and can be realized in small silicon area. Also, it is shown that the input-interleaved algorithm, proposed in the previous chapter, can increase the speed while keeping the size and power consumption of the decoder almost unchanged. The new Viterbi decoder is named an "*input-interleaved decoder*" and is the basis for our integrated-circuit class-IV decoder. It is known that an analog design can be adversely affected by offsets, mismatches and charge injections. The performances of both of the aforementioned structures in the presence of these imperfections are evaluated and their robustness is illustrated. To show the feasibility of the proposed analog realizations, two Viterbi decoders were implemented. The adaptive-threshold detector was first implemented by constructing a discrete prototype. In the second step, an integrated version was designed based on the input-interleaved structure. The design was fabricated in a $0.8\mu m$ BiCMOS process. Both of these decoders were tested and the results, reflected in the chapter, confirm the validity of the approaches in practice. The integrated decoder occupies only $0.5mm^2$ of area and consumes $30mW$ from a single $3.3V$ power supply while operating at $200Mb/s$.

To extend the advantages of an analog realization to any kind of Viterbi decoder (including other partial-response decoders), a general implementation technique was required. The goal was to derive a generic approach which can be applied to various applications. Simplicity was a basic requirement, since speed, size, and power consumption were the major concerns. Chapter 5 starts with explaining the proposed technique followed by describing the required building blocks. Circuit-realization issues and design methodologies are described next. The implementation method was applied to two different decoders. Despite being an inefficient method for decoding a simple dicode signal, a two-state dicode decoder was chosen to prove the concept. Also, a more complicated partial-response decoder, with eight states, was designed to illustrate the extendibility of the approach. The latter scheme is called *EPR4* and is predicted to very soon find its first application in the disk-drive industry. The decoders were implemented on a common silicon core and the chip was fabricated in a $0.8\mu m$ BiCMOS process. To save design time, digital path memories were not included on the chip. By the time of writing, the dicode decoder has undergone experimental tests up to $80Mb/s$. However, with an on-chip path memory, simulations indicate that speeds in the order of a few hundreds of megahertz can be achieved. The power consumption of the decoder is estimated to be about $15mW/state$ drawn from a $5V$ single power supply. It should be emphasized here that the approach proposed in this chapter is not restricted to partial-response systems and can be used in other systems such as digital communication employing error-correction codes. Also, the technique can be used to implement programmable (adaptive) sequence detectors.

Implementation issues related to reduced-state sequence detection is the subject of chapter 6. This detection approach has been suggested to reduce the complexity of a partial-response decoder at the expense of some degradation in the performance. The idea is to combine the symbol-by-symbol decision-feedback detector with the sequence detector to trade off some of the complexity of the sequence detector with the simplicity of the symbol-by-symbol detector. A circuit realization most benefits this complexity reduction if the link between the reduced-state detector and the decision-feedback equalizer is fully exploited. This is represented through two illustrative examples at the beginning of the chapter. Recently, the detection method applied to quaternary class-IV has drawn some attention since this signaling scheme has been proposed for transmitting $125Mb/s$ in a bandwidth of about $30MHz$ over a piece of unshielded twisted-pair cable. The major portion of chapter 6 is devoted to developing the Viterbi algorithm in the presence of the complexity reduction resulted from the decision-feedback mechanism for this application. It is shown that the detector can be implemented in the analog domain by employing a programmable version of either one of the adaptive-threshold or the input-interleaved detectors introduced in chapter 4. As well, it is predicted that the proposed algorithm results in some reduction in the complexity even in a digital realization.

The thesis concludes with a final chapter on general conclusions of the present work and suggested directions for future research in this exciting area. These future works cover a variety of system-level and circuit-level aspects such as CMOS implementations, circuit realization of the decoders which were addressed without actual realizations, automatic-layout generation of analog Viterbi decoders, soft-output Viterbi decoders, combined equalization and sequence detection, and adaptive Viterbi decoders.

Chapter 2

Partial-Response Signaling and Applications

Partial-response signaling (PRS) [1], also known as correlative level-coding, is a signaling scheme first proposed for data communications [2, 3]. In contrast to a conventional pulse-amplitude modulation system, in which no inter-symbol interference (ISI) is allowed, a PRS system introduces a controlled amount of ISI to the signal. This ISI is known and can be removed at the receiver. By relaxing the condition of zero ISI, certain beneficial effects can be attained through convenient spectrally shaping. Two examples of these effects are providing more similarity between the spectrum of the transmitted signal and the frequency response of the channel and realizing minimum-bandwidth transmission systems in practice.

The operation of a PRS system can be understood by the model shown in figure 2.1. In this model, an FIR filter is employed to introduce the ISI, whereas a low-pass filter band-limits the resulting signal. To have the ISI exclusively controlled by the FIR filter, the low-pass filter, $H(\omega)$, should be designed such that it preserves the relative sample values.

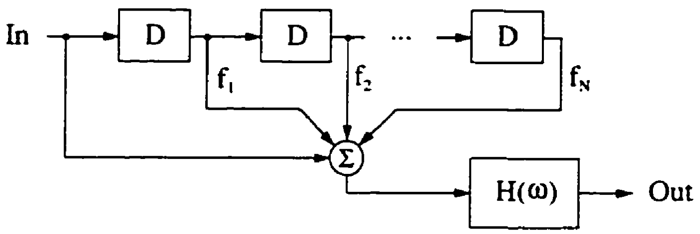


Figure 2.1: A partial-response encoder model.

Note that in figure 2.1, the operator D reflects a time-step delay of $T = 1/f_s$, where f_s is the sampling frequency. Using this notation, the resulting transfer function of the FIR filter becomes a polynomial of D

$$F(D) = 1 + \sum_{i=1}^{N} f_i D^i = 1 + F_1(D) \tag{2.1}$$

where $f_N \neq 0$ is assumed. We shall refer to the above polynomial as the *"coding polynomial"* of the PRS system.

From (2.1) it is clear that a PRS system results in an increase in the number of output levels. In addition to a drop in the power of the signal, the implementation complexity will increase as the number of these levels increases. As a result, this number should be kept to a minimum. Without any loss of generality, we shall normalize the peak values of the multi-level partial-response (PR) signal to ±1 by considering a low-frequency gain equal to

$$h = \frac{1}{1 + \sum_{i=1}^{N} |f_i|} \tag{2.2}$$

for the low-pass filter $H(\omega)$ in figure 2.1.

2.1. Minimum-Bandwidth Communication Systems

Efficient use of the available bandwidth has always been one of the most important goals in designing a communication system. However, achieving this goal requires a compromise which may not necessarily be in favor of a minimum-bandwidth solution. Other usual components are feasibility of the system in practice and its complexity. In sections below, the role of PRS systems in this compromise will be briefly explained.

2.1.1. Nyquist Criterion for Zero ISI

The basic principle underlying a Nyquist system [4] is that the received sample values should be functions of their corresponding data values, and not the adjacent symbols. In other words, the whole communication system (including transmitter, channel, and receiver) should behave like a memoryless channel. This zero-ISI condition, known as the Nyquist criterion, necessitates that the sampled values of the impulse response of the system be zero at the non-corresponding data instants. This criterion, expressed in the frequency domain, results in a constant value if the frequency response of the system is repeated with a period equal to the sampling frequency [5]. Figure 2.2 illustrates the above criterion in the time and frequency domains.

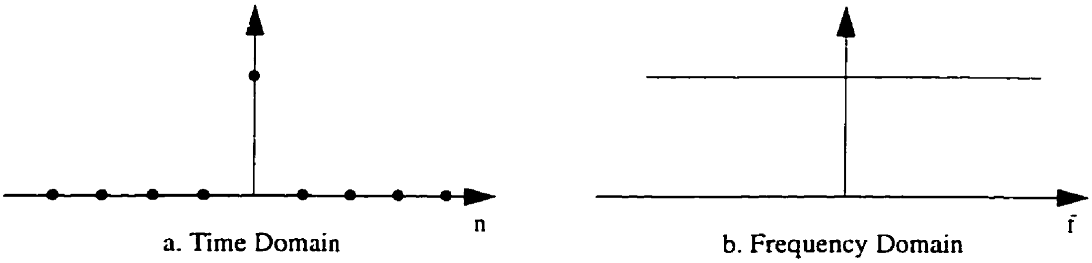


Figure 2.2: Nyquist zero-ISI criterion.

2.1.2. Nyquist Systems

To maximize the symbol rate in a given bandwidth, and eliminate the ISI, one is interested in the minimum-bandwidth solution to the Nyquist system described above. From figure 2.2 it is clear that such a solution exists, is unique, and equals an ideal low-pass filter with a brick-wall cut-off frequency of $f_s/2$. In the time domain, this corresponds to a *sinc* impulse response, as shown in figure 2.3.

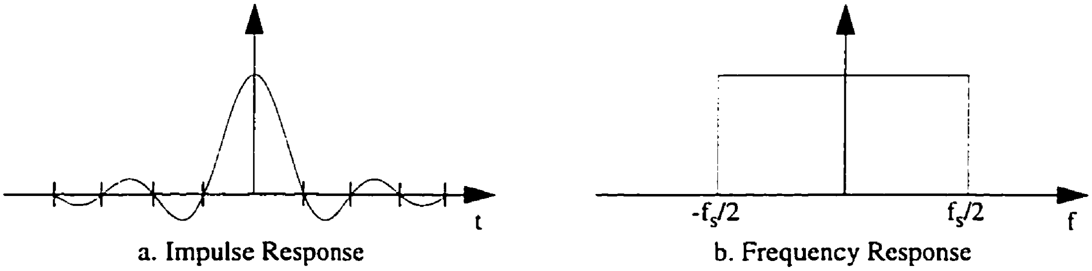
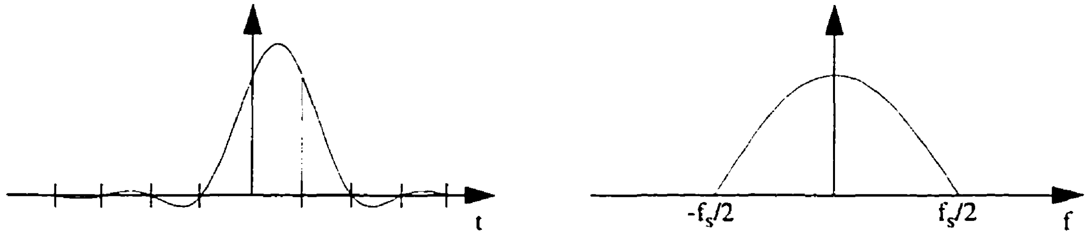


Figure 2.3: The minimum-bandwidth Nyquist system.

Unfortunately, this system is not feasible in practice. The problem arises from the slowly-decaying tails of the impulse response which cause excessive ISI if any timing perturbation occurs. It has been shown that the impulse response decays asymptotically as $1/|t|^{K+1}$ if the frequency response and its first $K-1$ derivatives are continuous and the K'th derivative is not [6]. The above problem can be overcome by relaxing the minimum-bandwidth constraint and using some excess bandwidth to achieve filters with smoother transition edges. Raised-cosine filters have been the most popular filters used in practice [5].

2.1.3. Minimum-Bandwidth PRS Systems

The question of the feasibility of a minimum-bandwidth communication system was not answered until the *duobinary* signalling scheme was introduced [2][1]. The basic idea is to relax the zero-ISI constraint by violating the Nyquist criterion. By controlling the ISI, the discontinuities in the frequency response of the system (and its derivatives) can be avoided, leading to perturbation-tolerant filters in practice. The duobinary technique is described by $F(D) = 1 + D$ and has one null at $f_s/2$. The impulse response of such a system decays as $1/|t|^2$, resulting in a bounded (and reasonably low) unwanted ISI if any timing jitter occurs. The frequency response and the impulse response of the duobinary signaling scheme are illustrated in figure 2.4. Note the fast decay of the impulse response compared to that of the minimum-bandwidth Nyquist system shown in figure 2.3.



a. Impulse Response

b. Frequency Response

Figure 2.4: The duobinary PRS system.

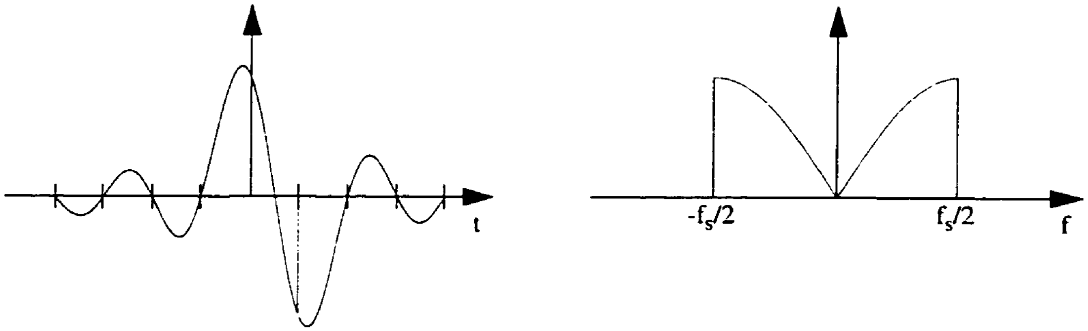
The communication system can be made more jitter-tolerant by widening the null around $f_s/2$. In general, it can be shown that the first $K - 1$ derivatives of the frequency response are continuous iff $F(D)$ has a factor of $(1 + D)^K$ [1].

2.2. DC Nulls

A DC null, often desirable in a communication system, can be easily accomplished in a PRS system. Such a null is created by including at least one $1 - D$ factor in the coding polynomial of the system. Although multiple zeros at DC widen the null and cause a more gradual roll-off in the low-frequency content of the signal, a single zero is sufficient in many cases. A single zero at DC is generated by the coding polynomial $1 - D$, known as *dicode*. Figure 2.5 shows the frequency

[1] This concept was then generalized [3], leading to the signalling technique named partial-response.

response and the impulse response of the dicode PRS system.



a. Impulse Response

b. Frequency Response

Figure 2.5: The dicode PRS system.

As the above figure shows, the ISI is introduced by subtracting the adjacent symbol from the main symbol. This corresponds to differentiating the sampled-value signal.

2.3. Class-IV PRS Scheme

Both of the aforementioned advantages of the duobinary and dicode systems, namely the feasibility of implementing a minimum-bandwidth system and the DC null required in many applications, can be achieved by using the system described by the following coding polynomial

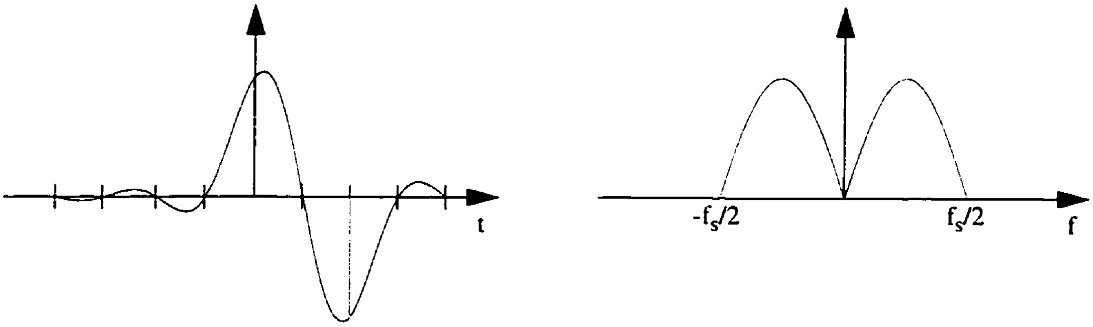
$$F(D) = 1 - D^2 \tag{2.3}$$

Beside the communication applications, the above system, known as *class-IV* (also *PR4*), can be used to resemble the signal spectrum at the output of a read head in a magnetic recording media[1]. We will address this issue in some detail later on. Here, note that the ISI is intentionally introduced by subtracting symbols two bit-intervals apart. The impulse response and the frequency response of the class-IV system are depicted in figure 2.6.

In addition to the usefulness of the spectrally shaping attained from this signaling scheme, it is also attractive from an implementation point-of-view. A class-IV system results from time-interleaving two independent dicodes. This is simply based on the fact that the coding polynomial given by (2.3) is exclusively a function of D^2, which results in a time-interleaved structure without any cross-coupled branches [7]. Figure 2.7 illustrates the idea.

Note that while the original class-IV system must be clocked at f_s, each dicode in the time-

[1] Throughout this thesis, always saturated magnetic-recording systems are considered.



a. Impulse Response

b. Frequency Response

Figure 2.6: The class-IV (PR4) system.

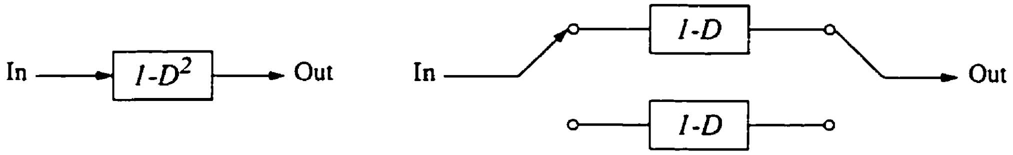


Figure 2.7: The class-IV system and its time-interleaved structure.

interleaved structure will be clocked at $f_s/2$. This decrease in the internal operating rate will be more appreciated if one considers the relatively sophisticated signal processing needed in the sequence detector. Sequence detection of PRS schemes will be considered in detail throughout this thesis.

Finally, the above approach clearly reveals that in a class-IV system the increase in the number of levels is equal to that in a single dicode and is minimum among all the first order PRS schemes.

2.4. Detection of Partial-Response Signals

The problem of detecting a PR signal is equivalent to removing the introduced ISI from the signal. It is well known that either a linear filter or a decision-feedback equalizer (DFE)[1] can be utilized to do the job [8]. However, to achieve better noise performance, one might decide to choose a more complicated detection technique. The decision mainly depends on the trade-off between the complexity of the receiver and its performance. In what follows, we shall briefly elaborate on the detection problem of a PR signal contaminated by additive white Gaussian noise (AWGN).

[1] This DFE should not be confused with the adaptive DFE usually used in a communication system to equalize the channel. Although, they may be combined.

2.4.1. Symbol-by-Symbol Detection

Traditionally, a symbol-by-symbol PRS detector employs a DFE to remove the ISI from the received signal. In this approach, the noise-enhancement of a linear filter is avoided by using estimates of the data in the feedback path of the equalizer. Also, pre-coding can be used to overcome the error propagation associated with the DFE [8]. With the coding polynomial given by (2.1), the symbol-by-symbol detector shown in figure 2.8 results. Note that with an M-ary original data, the slicer shown in this figure outputs one of the M possible levels at each symbol time.

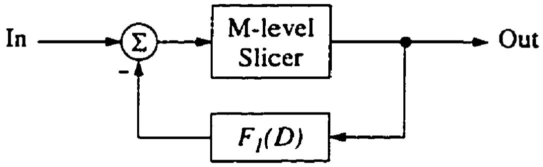


Figure 2.8: Symbol-by-symbol PRS detector based on a DFE.

The noise performance of the above symbol-by-symbol detector can be evaluated by considering the model shown in figure 2.9. In this figure, the AWGN represented by n is the channel noise and h is given by (2.2).

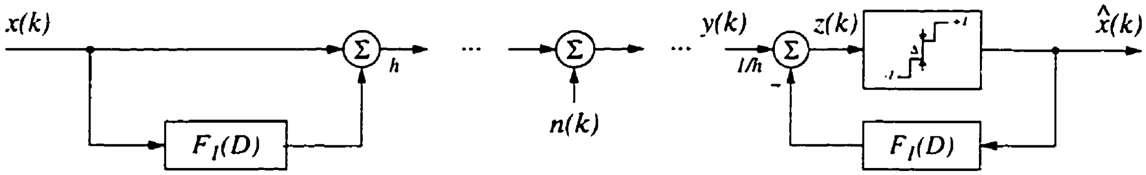


Figure 2.9: PRS-system model for noise-performance analysis.

Appendix A contains a brief review of the error analysis when the error-propagation noise is neglected. To include the effect of error propagation, the probability density function (pdf) of the contributed noise should be derived. This noise is a discrete random variable and can take up to $(2M-1)^N$ distinct values. This makes an analytic solution rather unwieldy for large M and N. In what follows, we consider PRS schemes with only one ISI term, however, with arbitrary number of input levels[1]. The coding polynomial of these systems is given by

$$F(D) = 1 + f_N D^N \tag{2.4}$$

For the above systems, error-propagation noise given by (A.2) reduces to

[1] This category includes multi-level duobinary, dicode, and class-IV schemes which are frequently addressed in this thesis.

$$e(k) = f_N(x(k-N) - \hat{x}(k-N)) \tag{2.5}$$

which will take $2M-1$ values of

$$2f_N(1 - \frac{i}{M-1}) \qquad ,i = 0, \ldots, 2(M-1) \tag{2.6}$$

From the above values, the first and the last $M-2$ are very unlikely, since the probability of their occurrence is the same as the probability of shifting more than two levels at the output of the slicer. Also, note that the most-probable value, the median, has a probability of happening equal to the probability of making no error in detecting $x(k)$. As a result, the pdf of the error-propagation noise can be approximated by

$$f_{e(k)}(u) = \frac{SER}{2}\delta(u + \frac{2f_N}{M-1}) + (1 - SER)\delta(u) + \frac{SER}{2}\delta(u - \frac{2f_N}{M-1}) \tag{2.7}$$

where $\delta(\ldots)$ is the unit impulse function.

Expanding (A.3) based on the total probability relationship [9] and using the above pdf yields

$$SER = \frac{Q(\frac{h\Delta}{2\sigma})}{\frac{1}{2}\frac{M}{M-1} + Q(\frac{h\Delta}{2\sigma}) - \frac{1}{2}Q(\frac{h\Delta}{2\sigma}(1 - 2f_N)) - \frac{1}{2}Q(\frac{h\Delta}{2\sigma}(1 + 2f_N))} \tag{2.8}$$

which is a general expression for symbol-error rate (SER) in M-ary PRS schemes of the type given by (2.4). Note that for binary signals, (2.7) is exact and there will be no approximation in deriving (2.8). Also, comparing (2.8) with (A.5) shows an increase by at most a factor of M in the SER due to error propagation. This is in agreement with [1].

It is useful to express SER as a function of signal-to-noise ratio (SNR) at the input of the detector. It can be shown that the power of the original signal consisting of M equi-probable equally-spaced symbols within the interval [-1, 1] is equal to

$$S_x = \frac{1}{3}\frac{M+1}{M-1} \tag{2.9}$$

and that it results in a PR signal transmitted through the channel with a power of

$$S_y = \frac{1}{3}\frac{M+1}{M-1}h^2\left(1 + \sum_{i=1}^{N} f_i^2\right) \tag{2.10}$$

This expression can be used to calculate the SNR of the PR signal.

To examine the validity of our approach, the dicode PRS system was simulated[1]. Figure 2.10 illustrates the SER performance as a function of SNR at the input of the DFE detector. Both the binary and quaternary schemes were considered. These plots show that the theoretical predictions, based on (A.5) and (2.8), are in very good agreements with the simulation results. Note that these results are directly applicable to class-IV schemes as well.

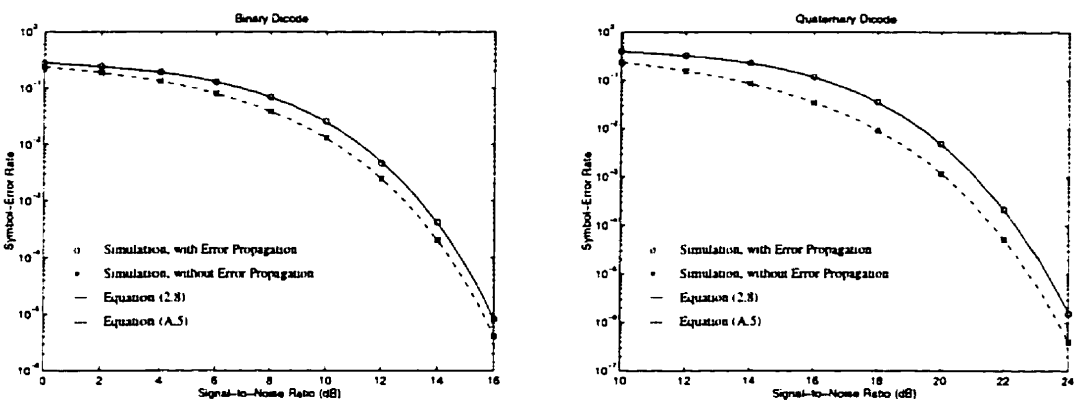


Figure 2.10: Noise performances of binary and quaternary dicode PRS schemes with DFE detectors.

In addition to the conventional decision-feedback equalization, the noise enhancement of a linear filter can be overcome if the symbol-by-symbol detector is preceded by a slicer. This quantization does not degrade the performance if the threshold levels are optimally spaced between the noiseless levels of the PR signal. As an example, with equi-probable equally-spaced symbols and L levels for the PR signal, the slicer consists of $L-1$ threshold levels equally spaced in the interval [-1, 1]. This symbol-by-symbol detector is depicted in figure 2.11. Note that the limiter in the feed-forward path of the filter can be absorbed in the summer.

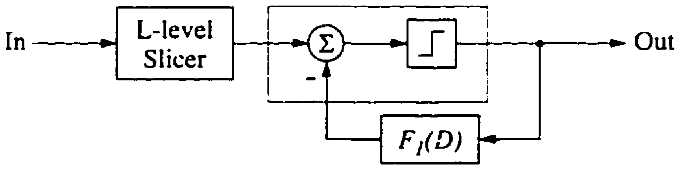


Figure 2.11: An alternative to the conventional DFE-based symbol-by-symbol PRS detector.

The advantage of the above structure is that it relaxes the design constraints on the summer, since both of its inputs can take only some discrete values. Compared to the conventional DFE, the overall complexity of the detector may show a reduction, especially if L is not much larger than M.

[1] For the system-level simulations throughout this thesis, C-code behavioral models were used.

2.4.2. Sequence Detection

It is known that because of the increased number of levels in a PRS system, a traditional PRS detector requires a higher SNR to perform as well as the conventional pulse-amplitude-modulation system [10]. However, it has been shown that this poorer performance is due to non-optimality of the detection technique and is not an inherent drawback of the PRS system. In fact, almost all the SNR loss can be recovered if the more complicated detection scheme known as maximum-likelihood sequence detection is employed [11, 12]. As an example, the $3dB$ loss in the power of a binary dicode PR signal (equation (2.10)) can be totally recovered by a sequence detector. Sequence detectors and their realizations will be investigated in detail throughout this thesis.

2.5. Magnetic Recording

Mass storage of information has been dominated by magnetic-recording systems mainly due to providing high capacity at a much lower cost than that of semiconductor memories. In a rotating magnetic-storage device, such as a computer hard disk, the areal density is the product of linear density and track density. While both of these densities have been increasing by improvements in the design of heads and disks, the linear density has been showing a substantial growth since the communication theory was practically applied to magnetic-recording systems. As a result, the storage capacity has continued to double every two or three years for the past thirty years [13].

Being one of the most promising applications of PRS, magnetic-recording systems will be explained in some detail in this section.

2.5.1. System Model

A typical data-storage system, shown in figure 2.12, can be considered as a data-transmission system. The goal, however, is to reliably store as much information as possible in a given area. In analogy to a communication system, this model consists of three basic parts: transmitter (write channel), receiver (read channel), and communication channel (magnetic media). The communication channel, like any other transmission media, has its own pulse-shaping behavior and adds some noise to the signal.

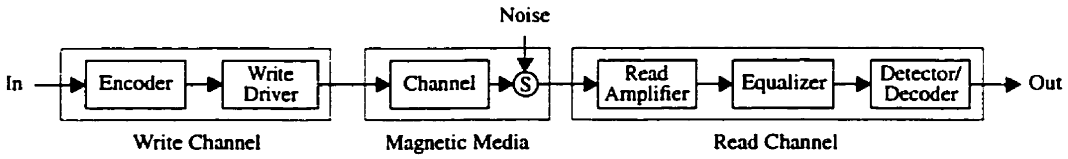


Figure 2.12: A typical magnetic data-storage system.

In the write channel, coding operations such as error-correction coding and run-length-limited coding are usually performed on the input data to achieve high immunity against noise, reliable timing extraction, and reduction in ISI. However, some other coding schemes may be applied as well [14]. Prior to applying to the write head, the encoded data may undergo a pre-compensation operation to immunize the signal against shifting in the position of the peaks [15].

In the read channel, after filtering and amplification, the signal is equalized before detection. The goal of equalization is to combat the ISI introduced by the media by forcing it either to zero or to a known value, depending on the detection scheme. The output data is finally obtained by decoding the detected signal. The decoder may be combined with the detector in some cases [16].

In addition to thermal noise, in a magnetic recording media other sources of interference are present. Interferences from adjacent tracks (off-track noise) and previously-written information (overwrite noise) are two components of the overall noise which are usually considered to be random signals [17, 18]. The combination of thermal with interference noise results in typical channel SNR values around $17dB$. The desired BER is typically 10^{-7} without any higher-layer error-correction coding.

2.5.2. Lorentzian Model of the Magnetic-Recording Channel

The magnetic media tends to introduce significant amount of ISI to the signal. This will be illustrated by approximating the step response of the channel by the following Lorentzian characteristic

$$s(t) = \frac{1}{1 + \left(\frac{2t}{PW_{50}}\right)^2} \tag{2.11}$$

in which PW_{50} is the width of the step response between its two 50% of the peak value. Figure 2.13 shows the Lorentzian pulse.

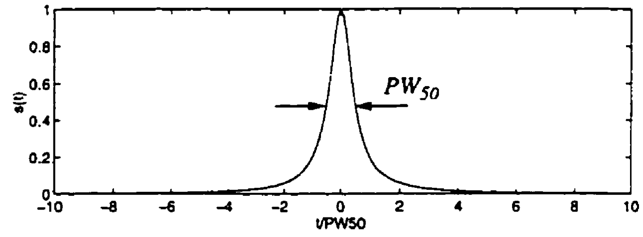


Figure 2.13: The Lorentzian pulse.

At low densities, where the adjacent pulses are far enough apart, there will be no noticeable interference between them and the channel is almost independently responding to the transitions of the write signal. However, as the density increases the tails of these pulses interfere, resulting in a potentially significant amount of ISI. Figure 2.14 illustrates a typical write current and its corresponding read signal for two cases of very-low and relatively-high densities.

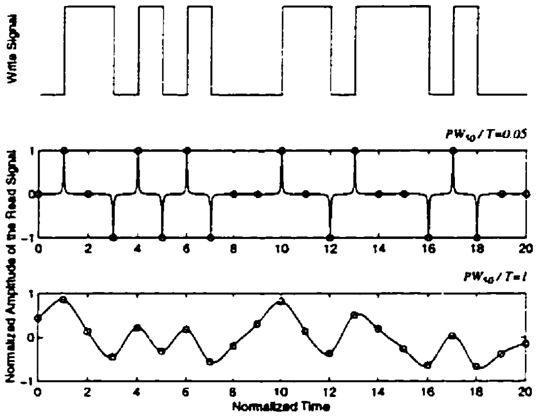


Figure 2.14: Typical write and read signals for low and high densities.

In general, interfering the adjacent transitions causes a reduction in the amplitude of the signal and deviates its samples from their nominal values. This can be translated in a drop in SNR.

It is also interesting to look at the above phenomenon in the frequency domain. Using the Lorentzian model, and after some manipulations, one can show that at a clock rate of $1/T$ the pulse-response spectrum of the magnetic media can be given by

$$S(\omega) = 2\frac{\pi\frac{PW_{50}}{2T}}{sinh(\pi\frac{PW_{50}}{2T})} sin(\frac{\omega}{2}) cosh(\frac{PW_{50}}{2T}(\omega-\pi)) \qquad , 0 \le \omega \le \pi \qquad (2.12)$$

which is plotted in figure 2.15 for different values of PW_{50}/T.

2.5.3. Detection Techniques

It has been a common practice to employ peak detectors to retrieve the recorded information in a magnetic-recording system [15]. These peak detectors suffer from ISI at increased densities and are usually accompanied by high-frequency boost stages (also known as pulse slimmers) to compensate for the low-pass effect shown in figure 2.15 [19]. DFE-based detectors are shown to outperform peak detectors by avoiding the noise enhancement caused by pulse slimming [20]. After introducing the application of partial-response signaling to magnetic recording [21], different PR

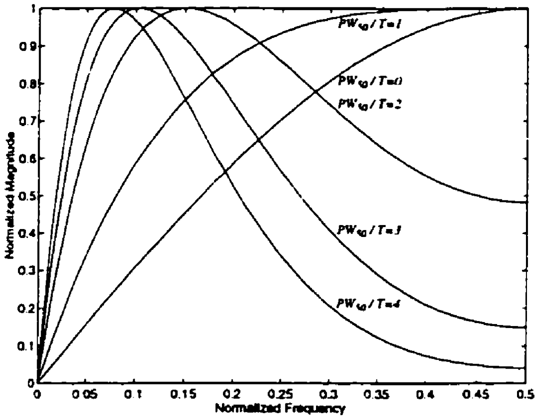


Figure 2.15: Pulse-response spectra of a Lorentzian magnetic channel.

detection techniques were considered as well [22-25]. Among different detection techniques [26-29], sequence detection based on partial-response interpretation of the media has drawn a lot of attention nowadays. Giga-bit density magnetic-recording systems with partial-response maximum-likelihood (PRML) read channels have recently been reported [30, 31].

2.5.4. Partial-Response Read Channel

Despite the fact that at low densities there would be no considerable interference between adjacent transitions, the magnetic media can still be considered as an ISI channel. Comparing the input and output signals of the channel in figure 2.14 reveals that at very low densities the media can be modeled as a dicode PRS system. At higher densities, other PRS models with higher-order polynomials can be used. In fact, a class-IV system was the first PRS system proposed to approximate the spectrum of the read signal in a magnetic media [21]. Recently, a more general category has been considered to model the magnetic channel at different densities [32]. These schemes, known as *extended partial-response (EPR)* in the magnetic-recording literature [33], are characterized by

$$F(D) = (1-D)(1+D)^n \qquad ,n = 0, 1, \ldots \qquad (2.13)$$

and result in signal spectra shown in figure 2.16. Note that $n = 0$ and $n = 1$ correspond to dicode and class-IV systems, respectively. For $n \geq 2$, we shall follow the notation in [33] (i.e. the system corresponding to $n = 2$ will be called *"EPR4"*, since four symbols are involved.).

Comparing figure 2.16 with figure 2.15 shows that EPR schemes indeed resemble the magnetic-media in a wide range of densities. These short-memory approximations result in significant reductions in the amount of equalizations needed and hence their noise contributions [33-35].

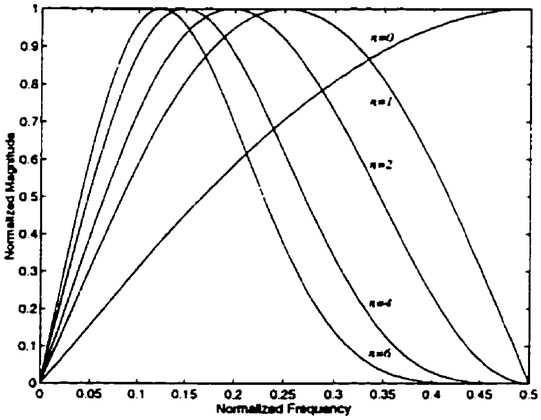


Figure 2.16: Output signal spectra of the EPR systems.

However, it should be mentioned that at extremely high densities the nonlinearity of the media becomes severe and the Lorentzian model starts to collapse. This, in conjunction with the complexity of the pre-compensation circuit and the unrecoverable SNR loss during detection, imposes an upper bound on n in (2.13) in practice. We elaborate more on this issue in the next chapter.

Recent developments in the signal-processing techniques (analog and digital) have convinced the disk-drive industry to seriously look at partial-response read channels with sequence detectors as the best alternative in their high-density products [36].

2.6. Data Transmission over Band-Limited Channels

As it was mentioned earlier, data transmission has always been one of the major applications of PRS. In addition to the capability of increasing the baud rate to near the Nyquist rate, a further increase in bit rate is possible by employing an M-ary signaling scheme. In fact, a quaternary class-IV system was recently proposed for high-rate data transmission over unshielded twisted-pair (UTP) cables [37].

Since this work has been partly motivated by the above application, it will be briefly explained in the following section.

2.6.1. UTP Cables in High-Rate Data Transmission Systems

The cost of coaxical cables and optical transmission links and their installation have motivated researchers to look for low-cost alternatives to the distributing cables in fiber-distributed data interface systems [38] and local area networks. Consequently, UTP copper cables are receiving renewed attention to provide the required connections [37, 39-44]. The transceivers should communicate at rates of over $100 Mbits/s$, however, to achieve compliance with federal communications commission (FCC) radiation limits, spectral components of the transmitted signal should be well suppressed above $30 MHz$ [45].

In a promising proposal, the use of a quaternary class-IV PRS (QPRIV) scheme has been suggested to achieve a data rate of $125 Mbits/s$ [37, 39]. At this rate, the spectrum of a QPRIV signal is limited to $31.25 MHz$, which facilitates satisfying the aforementioned FCC requirement. Both voice-grade and data-grade cables can be utilized to carry the required baud rate. The data-grade cable eliminates the need for a near-end cross-talk (NEXT) cancellation at the expense of a higher cable price (Provided that good terminations are also available.).

2.7. Summary

In a partial-response signaling system, a controlled amount of inter-symbol interference is intentionally introduced to spectrally shape the signal. Transmitting over AC-coupled channels and achieving minimum-bandwidth communication are possible by inserting spectral nulls at DC and the Nyquist frequency. The resulting signaling schemes, known as dicode and duobinary, can be combined leading to the class-IV system whi.h is one of the most widely used partial-response signaling schemes.

The problem of extracting the original data from a partial-response signal is equivalent to removing the introduced inter-symbol interference. This interference is known and can be removed by a decision-feedback equalizer with fixed coefficients, in a symbol-by-symbol detection fashion. However, the noise performance of the detector can be substantially improved if the more complicated maximum-likelihood sequence detection scheme is employed. In fact, this detection technique combats the signal loss resulted from increasing the number of transmitted levels, compared to uncoded systems. For a symbol-by-symbol detector, an expression for symbol-error rate in a category of partial-response systems including dicode, duobinary, and class-IV is derived. The detailed description and analysis of the sequence detector is postponed to the next chapter, where it will be shown that it outperforms the symbol-by-symbol detector by $3dB$ for the aforementioned signaling schemes. This amount is equal to the loss in the power of the encoded signal. The above subjects comprise the bulk of the present chapter, however, some other implementation issues were addressed as well.

The chapter was concluded with describing two important applications of partial-response signaling, namely magnetic recording and data transmission. Not being as obvious as data transmission, partial-response signaling applied to magnetic recording has been considered in some detail. The basic idea is to approximate the read signal of a magnetic-recording system with a partial-response signal. The traditional peak detector can then be replaced with a sequence detector to improve the performance. This improvement can be translated to an increase in the recording density. The increase is significant and has motivated researchers to look for low-complexity realizations of sequence detectors for partial-response read channel applications.

Chapter 3

Maximum-Likelihood Sequence Detection

In detecting a digital sequence, it is well known that if one received symbol contains some information about other symbols, symbol-by-symbol detection will no-longer lead to an optimum detector [8]. The degradation arises when the symbol-by-symbol detector removes the effects of the other symbols in the detection process, hence, ignoring some useful information. The detection scheme in which the detector takes full advantage of the above information in estimating the transmitted sequence is called maximum-likelihood sequence detection (MLSD).

As a good example, one can consider transmitting a sequence over an ISI channel, which causes a time dispersion of the signal energy. Since the interference contains some information about the transmitted symbols, for optimum performance, the whole received sequence should be used to detect any symbol or group of symbols. Partial-response signaling schemes fall in this category. Sequence detection of PR signals will receive most of our attention in this chapter.

In the MLSD approach, the detector determines the most-likely transmitted sequence having the minimum distance from the received signal, as compared to all other possibly-transmitted sequences. The objective function to be minimized depends on the error criterion and is the squared-Euclidean distance in the case of additive-Gaussian noise.

Although some early work had been done on optimum sequential detection [46], it was not adopted in practice due to its computational complexity. The number of computations required in a brute-force approach grows exponentially with the length of the sequence. Furthermore, computations can not begin until the entire sequence has been received. Both of these obstacles have been removed in an algorithmic implementation of the MLSD, known as the Viterbi algorithm (VA). We start with a brief explanation of the algorithm and will focus on its application to PRS systems. Some practical issues in implementing an MLSD decoder, based on the VA, will be addressed as well.

3.1. The Viterbi Algorithm

The Viterbi algorithm was first proposed for decoding convolutional codes [47] (also see [48]). Later, it was shown that this algorithm is indeed a special application of dynamic programing [49] to digital communications [50]. The significance of dynamic programming is that the number of computations grows linearly with the length of the transmitted signal [51]. Also, the algorithmic nature of the approach enables the detector to start the computations as soon as the first sample is received. Although, ideally, no effective decision is made until the whole sequence is received.

The Viterbi algorithm was extended to the detection of signals transmitted over linear ISI channels [11, 12]. Since then, PRS systems have received more attention due to the recovery of the SNR loss, associated with this signaling scheme, by the MLSD detector. In fact, it was shown that MLSD is the optimum detection technique for decoding a PR signal.

The basic idea behind Viterbi detection is to consider the received sequence as a finite-state discrete-time Markov process contaminated by memoryless noise. A trellis diagram is conceptually constructed by unwrapping the state diagram in time. The detector assigns a metric to each branch of the trellis, proportional to the error signal between the received value and the ideal signal resulting from that transition. The maximum-likelihood (ML) sequence is the one which results in the minimum accumulated error throughout the trellis. This approach is algorithmic in the sense that at each time step, for each one of the states of the trellis, the accumulated error signal (also known as state metric) is calculated using the previous state metrics and the branch metrics at that time step.

In addition to state metrics, enough knowledge regarding the paths along which these optimum metrics have been obtained should also be saved. A block of memory, used with different managements [52], can be utilized to save the required information. This information, stored in the form of digital sequences and updated at the end of each iteration, enables the decoder to track back the optimum paths ending to each state. Following the literature, we shall refer to this memory as path memory and its contents as survivor sequences.

Assuming an M-ary input signal and a memory size of N for the channel (or the PRS encoder), the trellis diagram could have up to $L = M^N$ different states, with a maximum of M transitions initiating from and/or ending to each state. Figure 3.1 illustrates a two-state trellis diagram for an unconstrained binary signal, starting from a known initial state. Some sample branch metrics are chosen and the state metrics in subsequent iterations of the algorithm are shown. At each time step, the state metric of each state is calculated by adding the previous values to the branch metrics, comparing the resulting accumulated errors, and assigning the minimum error to that state as its new metric value.

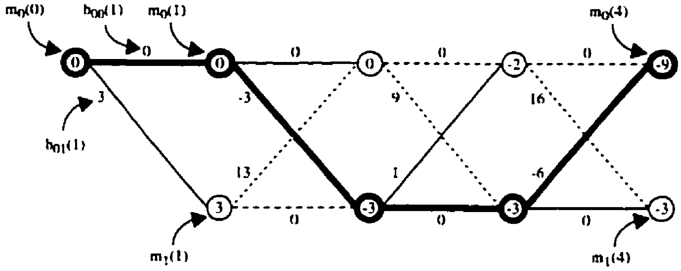


Figure 3.1: A two-state trellis diagram.

In figure 3.1, the minimum-error path corresponding to the ML sequence is shown in bold line, the paths which have been discarded in the competitions at each time step are shown in dotted lines and those which were candidates but have been discarded in next iterations are shown in solid lines. Note that the minimum-error paths of different states merge into a single path backward in time at some point. Up to this time, the decoded sequence will not be affected by future decisions.

From the above arguments it can be seen that the arithmetic involved in the VA is of the form of add-compare-select (ACS). These operations can be formulated as

$$m_i(k) = \min_j \{m_j(k-1) + b_{ji}(k)\} \qquad , \begin{matrix} i = 0, 1, \ldots, L-1 \\ j = 0, 1, \ldots, L-1 \end{matrix} \tag{3.1}$$

where $m_i(k)$ denotes the metric of state i at time step k and $b_{ji}(k)$ is the metric of the branch connecting state j at time step $k-1$ to state i at time step k. Note that for those values of i and j for which there does not exist a transition between states, branch metrics equal to infinity should be considered. This does not increase the implementation complexity and has been assumed for the generalization of the above mathematical expression.

The inputs to the algorithm are the branch metrics, which should be calculated based on the error criterion. Commonly, minimizing the square-Euclidean distance is the objective. In this case, branch metrics are of the quadratic form, but, can be reduced to linear combinations of the received sample and some constant values. The idea is to expand the quadratic terms and cancel out the common terms. Also, a fixed gain might be considered for all of the branches in the trellis.

3.2. Difference-Metric Algorithm in a Two-State Trellis

The ACS operations given by (3.1) applied to a two-state trellis result in direct calculation of two state metrics

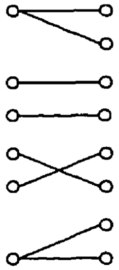
$$\begin{cases} m_0(k) = min\{m_0(k-1) + b_{00}(k), m_1(k-1) + b_{10}(k)\} \\ m_1(k) = min\{m_0(k-1) + b_{01}(k), m_1(k-1) + b_{11}(k)\} \end{cases} \tag{3.2}$$

However, by defining the difference between these state metrics as

$$\Delta m(k) = m_0(k) - m_1(k) \tag{3.3}$$

one can conclude that the difference signal can be updated equivalently. To show this, we subtract $m_1(k-1)$ from all of the four terms involved in (3.2). This subtraction does not affect the outcomes of the algorithm, however, the new state metrics are now expressed in terms of $\Delta m(k-1)$. Since the individual state metrics are no-longer required, there is no need to update them. The new value of the difference signal, $\Delta m(k)$, can be directly calculated by subtracting the expression for $m_1(k)$ from that for $m_0(k)$ in (3.2). This leads us to four possible update equations and survivor extensions given below

$$\Delta m(k) = \begin{cases} b_{00}(k) - b_{01}(k) & , \begin{cases} \Delta m(k-1) < b_{10}(k) - b_{00}(k) \\ \Delta m(k-1) < b_{11}(k) - b_{01}(k) \end{cases} \\ b_{00}(k) - b_{11}(k) + \Delta m(k-1) & , \begin{cases} \Delta m(k-1) < b_{10}(k) - b_{00}(k) \\ \Delta m(k-1) > b_{11}(k) - b_{01}(k) \end{cases} \\ b_{10}(k) - b_{01}(k) - \Delta m(k-1) & , \begin{cases} \Delta m(k-1) > b_{10}(k) - b_{00}(k) \\ \Delta m(k-1) < b_{11}(k) - b_{01}(k) \end{cases} \\ b_{10}(k) - b_{11}(k) & , \begin{cases} \Delta m(k-1) > b_{10}(k) - b_{00}(k) \\ \Delta m(k-1) > b_{11}(k) - b_{01}(k) \end{cases} \end{cases} \tag{3.4}$$



From the two middle decision regions specified in (3.4), one never occurs. This impossible region is determined by the relative values of $b_{10}(k) - b_{00}(k)$ and $b_{11}(k) - b_{01}(k)$. Once the impossible region is determined, it can be discarded. As a result, the decision regions and the path-memory extension alternatives will be reduced to three. We shall refer to this simplified algorithm as the "*difference-metric algorithm*[1]".

3.2.1. PRS Schemes and the Difference-Metric Algorithm

A two-state trellis diagram results if a binary signal is applied to the PRS encoder described by (2.4). Without any loss in generality, we only consider the system[2]

$$F(D) = 1 + f_1 D \tag{3.5}$$

The other systems result by time interleaving N independent systems given by (3.5).

After normalizing the peaks of the uncoded and encoded signals to ±1, the trellis shown in figure 3.2 results. The normalizing gain, h, is given by (2.2) and is equal to $1/(1 + |f_1|)$.

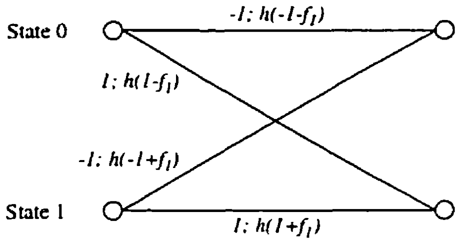


Figure 3.2: Trellis diagram of the system characterized by (3.5). Each branch is labeled with its corresponding pair of *uncoded: encoded* signals.

Denoting the input sample at time step k by $y(k)$, the branch metrics in the above trellis diagram will be equal to

$$b_{ji}(k) = (y(k) - h[(2i-1) + f_1(2j-1)])^2 \qquad , \begin{matrix} j = 0, 1 \\ i = 0, 1 \end{matrix} \tag{3.6}$$

which after cancelling the common term and applying a gain of $1/2$, reduces to

[1] The difference-metric algorithm described here can be considered as a generalization to the algorithm developed in [53] for decoding a class-IV PR signal.

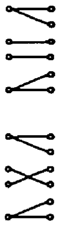
[2] We also restrict ourselves to $|f_1| \leq 1$. For $|f_1| > 1$, the coding polynomial can be written as $f_1 D(1 + (1/f_1) D^{-1})$, and the PRS system can be treated as a noncausal version of the system considered here.

$$b_{ji}(k) = -h[(2i-1)+f_1(2j-1)]\,(y(k)-\frac{h}{2}[(2i-1)+f_1(2j-1)]) \qquad , \begin{matrix} j=0,1 \\ i=0,1 \end{matrix} \tag{3.7}$$

Application of (3.7) to (3.4), results in the following difference-metric Viterbi algorithm for decoding a two-state PR signal

$$\Delta m(k) = \begin{cases} 2h(y(k)+f_1h) & , \qquad \Delta m(k-1) < -2f_1h(y(k)-h) \\ 2h(1+f_1)y(k)+\Delta m(k-1) & , -2f_1h(y(k)-h) < \Delta m(k-1) < -2f_1h(y(k)+h) \\ 2h(y(k)-f_1h) & , -2f_1h(y(k)+h) < \Delta m(k-1) \end{cases} \quad , f_l<0$$

$$\Delta m(k) = \begin{cases} 2h(y(k)+f_1h) & , \qquad \Delta m(k-1) < -2f_1h(y(k)+h) \\ 2h(1-f_1)y(k)-\Delta m(k-1) & , -2f_1h(y(k)+h) < \Delta m(k-1) < -2f_1h(y(k)-h) \\ 2h(y(k)-f_1h) & , -2f_1h(y(k)-h) < \Delta m(k-1) \end{cases} \quad , f_l>0 \tag{3.8}$$



It is interesting to note that for $f_1 < 0$, regardless of $y(k)$, $b_{10}(k) - b_{00}(k)$ is always larger than $b_{11}(k) - b_{01}(k)$, and hence, the third decision region of (3.4) never occurs. In the case of $f_1 > 0$, it is the second region which does not occur. This fact reduces the number of decision regions, update mechanisms, and path-memory extension alternatives to three in both cases.

Note that in the difference-metric approach, there is only one quantity propagating during the iterations[1]. In addition to simplicity, propagating only one differential signal offers other advantages, in general, and in an analog implementation, in particular. Avoiding the algorithmic growth of the state metrics and the accumulative error in calculating these values, are two of the most important advantages. We shall elaborate on these issues later.

3.2.2. Statistics of the Difference-Metric Signal

In realizing the VA, knowing the upper and lower bounds of the signals is necessary to determine the dynamic range of the circuits, in an analog realization, and the length of the registers and arithmetic units, in a digital realization [54]. Beside the received signal, which its statistics depend on the statistics of the transmitted signal and the channel noise, $\Delta m(k)$ is the only signal required to be calculated and stored in the difference-metric algorithm. In a noiseless situation, the survivor sequences always merge. Equation (3.8) and the branch labels shown in figure 3.2 can then be invoked to show that the difference metric signal will always be equal to

$$\Delta m(k) = \pm 2h^2 \tag{3.9}$$

In the presence of noise, we prefer to distinguish two mechanisms by which the above signal

[1] In general, it can be shown that the minimum number of propagating quantities is equal to the number of states minus one.

will be deviated from its nominal values based on the survivor extensions shown by (3.8). If at the end of the iteration two survivor sequences merge, the difference signal will be updated to $2h(y(k)\ \pm f_1 h)$. One can show that these values are, in fact, values given by (3.9), contaminated by noise components with deviations equal to $2h$ times the deviation of the channel noise. On the other hand, if survivor sequences do not merge, the updated signal will be equal to $2h(1-|f_1|)y(k)\ -\ (f_1/|f_1|)\Delta m(k-1)$, which can be easily shown to be bounded to (and at moderate-to-high SNR approximately equal to) $2h(y(k)\ \pm f_1 h)$. As a result, the pdf of the difference signal can be approximated by

$$f_{\Delta m(k)}(u) = \frac{1}{4h}\left[f_{n(k)}\left(\frac{u-2h^2}{2h}\right)+f_{n(k)}\left(\frac{u+2h^2}{2h}\right)\right] \qquad (3.10)$$

where $f_{n(k)}(\ldots)$ is the pdf of the channel noise. This expression shows that the difference signal has a zero mean and, although unbounded, can be effectively bounded in an actual realization.

Figure 3.3 illustrates distributions of the received and difference-metric signals for two cases of $f_1 = -1$ and $f_1 = 0.25$, obtained by simulations. The channel noise is considered to be an AWGN with a standard deviation of $\sigma = 0.1$, corresponding to SNR of 17 and $18.3dB$ respectively. Note that the deviation of the difference-metric signal around its nominal values is $2h$ times σ, while that of the received signal is equal to σ.

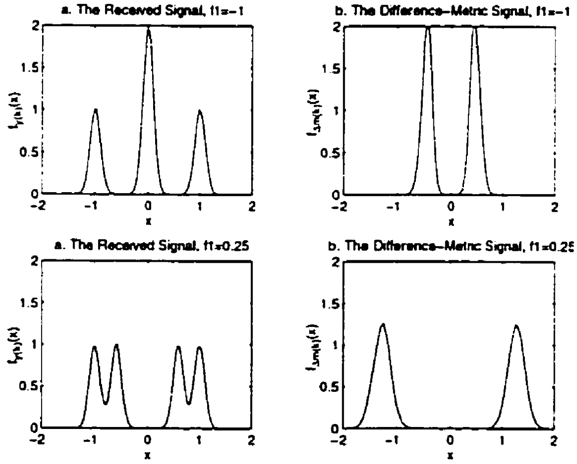


Figure 3.3: pdf's of the received and difference-metric signals obtained by simulations.

3.3. The Input-Interleaved Algorithm

Among the PRS schemes described by (3.5), we are most interested in the dicode system[1].

[1] Refer to chapter 2.

Setting $f_1 = -1$ in (3.8), results in the following algorithm

$$\Delta m(k) = \begin{cases} y(k) - 0.5 & , \quad \Delta m(k-1) < y(k) - 0.5 \\ \Delta m(k-1) & , y(k) - 0.5 < \Delta m(k-1) < y(k) + 0.5 \\ y(k) + 0.5 & , y(k) + 0.5 < \Delta m(k-1) \end{cases} \tag{3.11}$$

which is similar to the algorithm derived in [23].

Although an efficient analog realization based on (3.11) was recently proposed [55], a more interesting analog detector is obtained if the algorithm is further examined from an analog implementation perspective [56][1]. A closer look at the recursion given by (3.11) reveals that $\Delta m(k-1)$ is equal to either a positive or a negative DC-shifted version of a previously-sampled input signal. Specifically, if the previous sample is denoted by $y(j)$ then

$$\Delta m(k-1) = y(j) \pm 0.5 \tag{3.12}$$

which, combined with (3.11), leads us to two possible update equations given below

$$\begin{aligned} \Delta m(k) &= \begin{cases} y(k) - 0.5 & , y(j) + 1 < y(k) \\ y(j) + 0.5 & , \quad y(j) < y(k) < y(j) + 1 \\ y(k) + 0.5 & , \quad y(k) < y(j) \end{cases} \quad , \Delta m(k-1)=y(j)+0.5 \\ \Delta m(k) &= \begin{cases} y(k) - 0.5 & , \quad y(j) < y(k) \\ y(j) - 0.5 & , y(j) - 1 < y(k) < y(j) \\ y(k) + 0.5 & , \quad y(k) < y(j) - 1 \end{cases} \quad , \Delta m(k-1)=y(j)-0.5 \end{aligned} \tag{3.13}$$

By defining u as a DC offset which can take one of the two values of 1 and 0 (corresponding to two possible alternatives), the two above expressions can be combined. Also, note that as long as $y(j)$ and u are known, there is no need to calculate the difference signal. These two quantities can be propagated instead[2]. As a result, the iterations can equivalently proceed as follows

$$\begin{cases} y(j) = y(k), u = 0 & , \quad y(j) + u < y(k) \\ y(j) = y(j), u = u & , y(j) + u - 1 < y(k) < y(j) + u \\ y(j) = y(k), u = 1 & , \quad y(k) < y(j) + u - 1 \end{cases} \tag{3.14}$$

Expression (3.14) simply states that whenever $y(k)$ is in between the threshold levels, no update is required. However, if $y(k)$ falls outside this region the previously-sampled input signal should be updated to the current input, and the DC offset should be set either to 0 or 1, depending on $y(k)$ being more than the upper or less than the lower threshold level respectively.

We shall refer to the above algorithm as the "*input-interleaved algorithm*," for, as will be

[1] These implementations will be described in detail in the next chapter.

[2] Note that one of these quantities is only a single bit digital signal.

shown later, it can be realized in the analog domain by an input-interleaved architecture. The most important advantages of the input-interleaved architecture are its inherent high speed of operation and circuit simplicity. These implementation issues will be addressed in the next chapter. The input-interleaved algorithm seems to be a good alternative for a digital realization as well.

3.4. Probability of Error

Because we often refer to the noise performance of Viterbi detectors, a brief overview of the error analysis is given in the appendix. The analysis is based on the approach presented in [12]. Here, as an example, consider the binary PRS schemes given by (3.5)[1] and $f_1 = \pm 1$, with normalized input and encoded signal peaks. The minimum-distance error events are illustrated in figure 3.4[2].

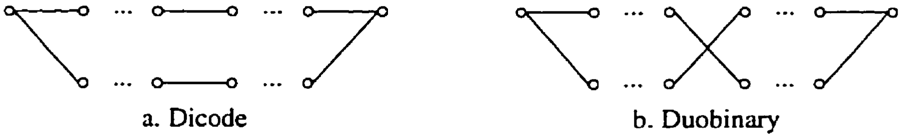


Figure 3.4: Minimum-distance error events in the Viterbi detection of two binary PRS schemes.

Using the branch labels in figure 3.2, one can show that the Euclidean distances of the error events shown in the above figure are equal to $\sqrt{2}$. Equation (B.2) can then be invoked to calculate the probability of symbol error. The result is[3]

$$SER = 4Q\left(\frac{1}{\sqrt{2}\sigma}\right) = 4Q\left(10^{\frac{SNR}{20}}\right) \qquad (3.15)$$

Comparing (3.15) with (2.8) (which for the PRS schemes considered here reduces to $2Q(1/(2\sigma)) = 2Q((1/\sqrt{2})\,10^{SNR/20})$ at high SNR) and with SER in an uncoded system (which is simply equal to $Q(1/\sigma) = Q(10^{SNR/20})$) reveals that the noise performance of the MLSD decoder asymptotically outperforms that of the DFE by $3dB$ and is the same as the performance of the ISI-free uncoded system. Figure 3.5 illustrates SER for the above three systems.

[1] The results are directly applicable to the systems described by (2.4) and $f_N = \pm 1$.

[2] Note that when $f_1 \neq \pm 1$, the minimum-distance error events have lengths of 2, however, as $f_1 \rightarrow \pm 1$ the Euclidean distances of some other error events approach d_{min} and the error events look more like those shown in figure 3.4.

[3] Note that when $f_1 \neq \pm 1$, the leading coefficient in the expression for SER is different from that in (3.15), however, as $f_1 \rightarrow \pm 1$, some of the other error events, which had been ignored, should be considered as well, and there would be no inconsistency between the results obtained from both approaches.

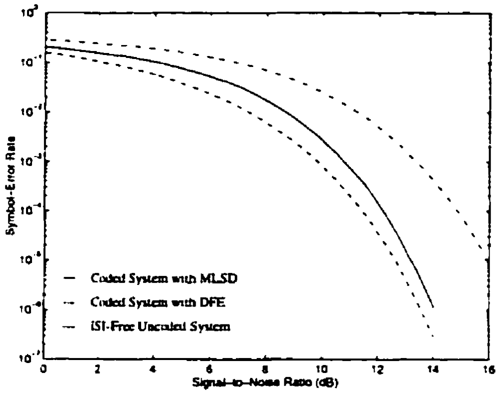


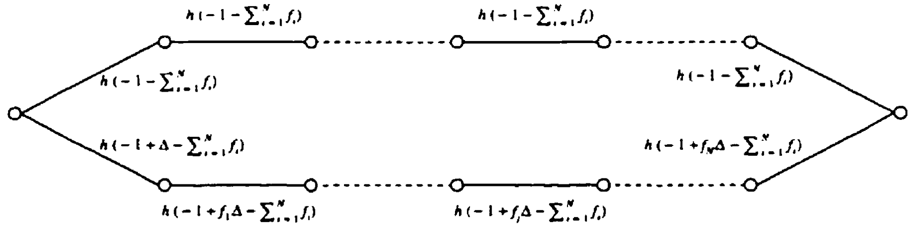
Figure 3.5: SER for binary $1 \pm D$ PRS systems with DFE and MLSD detections and the binary uncoded system.

3.5. Maximal-Distance Codes

In section B of the appendix it was shown that at high SNR, the SER of a PR-MLSD decoder can be expressed as

$$SER = KQ(\frac{d_{min}}{2\sigma}) \tag{3.16}$$

where K is a constant, σ^2 is the variance of the noise, and d_{min} is the minimum distance of the PR code, which depends on the coefficients of the coding polynomial. Let's consider the codes for which there exist minimum-distance error events that have the minimum length[1] (i.e. $N_e = N+1$). Figure 3.6 illustrates the minimum-length error event for an M-ary PRS scheme. The PRS system is characterized by (2.1) and Δ is given by (A.4). The transmitted PR signals are also shown to facilitate the calculation of d_{min}.



$h(-1-\sum_{i=1}^{N} f_i)$
$h(-1-\sum_{i=1}^{N} f_i)$
$h(-1-\sum_{i=1}^{N} f_i)$
$h(-1-\sum_{i=1}^{N} f_i)$
$h(-1+\Delta-\sum_{i=1}^{N} f_i)$
$h(-1+f_N\Delta-\sum_{i=1}^{N} f_i)$
$h(-1+f_1\Delta-\sum_{i=1}^{N} f_i)$
$h(-1+f_i\Delta-\sum_{i=1}^{N} f_i)$

Figure 3.6: The minimum-length error event in a PR-MLSD decoder.

Assuming the above event is among the minimum-distant error events, d_{min} is easily calculated to be

[1] Note that the reverse statement may not be true.

$$d_{min} = h\Delta\sqrt{1 + \sum_{i=1}^{N} f_i^2} \tag{3.17}$$

Combining (3.17) with (3.16) and expressing the result as a function of SNR, through (2.10), results in

$$SER = KQ\left(\sqrt{\frac{3}{M^2 - 1}}10^{\frac{SNR}{20}}\right) \tag{3.18}$$

SER in an uncoded system is equal to that of a DFE in a PRS system with $F(D) = 1$. This probability can be calculated by inserting $h = 1$ in (A.5) and the result can be expressed in terms of SNR by employing (2.9)

$$SER = 2\frac{M-1}{M}Q\left(\sqrt{\frac{3}{M^2 - 1}}10^{\frac{SNR}{20}}\right) \tag{3.19}$$

Comparing (3.18) with (3.19) depicts that the performance of the PR codes considered here is asymptotically equal to the performance of the uncoded systems. We shall refer to the PR codes which satisfy (3.17) as "*maximal-distance codes*". For these codes, the SNR loss resulted from increasing the number of levels can be almost totally recovered if an MLSD approach is taken.

3.5.1. First-Order Codes

A first-order coding polynomial results in an M-state trellis diagram, shown in figure 3.7.

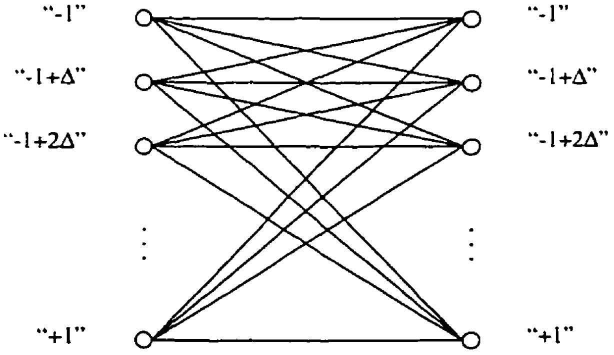


Figure 3.7: Trellis diagram of an M-ary first-order PRS scheme.

For the coding polynomial $1 + f_1 D$, the encoded signal associated with the transition from state "$-1 + i\Delta$" to state "$-1 + j\Delta$" is equal to $h(-1 + j\Delta + f_1(-1 + i\Delta))$, where i, j can take any integer values from 0 to $M - 1$. The trellis shown in figure 3.7 clearly shows that the minimum distance of the code is always equal to $h\Delta\sqrt{1 + f_1^2}$. In other words, the first-order code is a maximal-distance code regardless of f_1. This result is confirmed by figure 3.8, obtained from simulations. This figure illustrates that the noise performance of the first-order system relative to that of the uncoded system remains bounded with increasing SNR. As $f_1 \rightarrow \pm 1$, more shaping in the spectrum of the signal takes place, and the role of the MLSD detector in exploiting the redundancy introduced by coding becomes more critical.

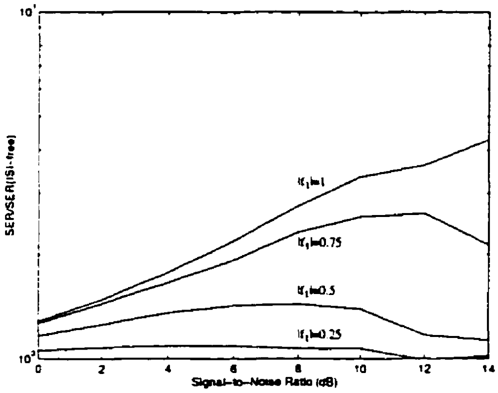


Figure 3.8: SER of different first-order PRS-MLSD systems relative to that of the ISI-free system.

3.5.2. Second-Order Codes

To further verify the validity of the concept presented here and also show how the maximal-distance codes can be derived, the second-order PRS system is considered as another example. The trellis diagram in this case is much more complicated than the one shown in figure 3.7. However, it is not difficult to recognize that the minimum distance of the code can only be equal to either one of the distances of the error events shown in figure 3.9, whichever is less.

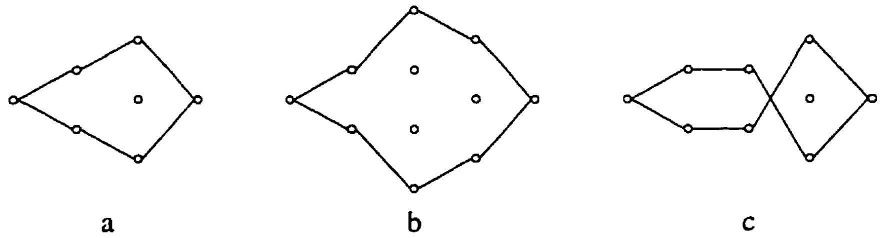


Figure 3.9: Different error events in a second order PRS-MLSD detector for calculating d_{min}.

Calculation of the distances of the error events shown in the above figure is straight-forward. Ignoring the details, the result is $d_a = h\Delta\sqrt{1 + f_1^2 + f_2^2}$, $d_b = h\Delta\sqrt{1 + (1 + f_1)^2 + (f_1 + f_2)^2 + f_2^2}$, and $d_c = h\Delta\sqrt{1 + (1 - f_1)^2 + (f_1 - f_2)^2 + f_2^2}$ for the events of a, b, and c, respectively. Based on the different values of f_1 and f_2, one of these three expressions will be minimum. The region in the

$f_1 - f_2$ plane in which d_a is minimum and the PR code is a maximal-distance code is plotted in figure 3.10. Here, we have restricted ourselves to $|f_2| \leq 1$, based on a similar argument we made in section 3.2.

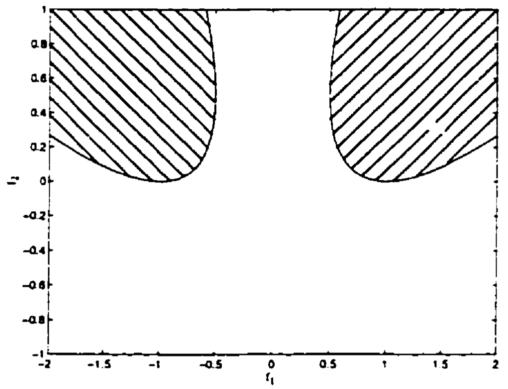


Figure 3.10: The region in the $f_1 - f_2$ plane in which the PR code is a maximal-distance code (unshaded area).

Simulation results of a second order MLSD-PRS scheme, with different values of f_1 and f_2, show that for the coefficient values outside the shaded region in figure 3.10, the SER performance relative to the performance of an uncoded system converges towards a constant value. Whereas for all the values inside this region, the relative performance diverges. Figure 3.11 illustrates some of the results.

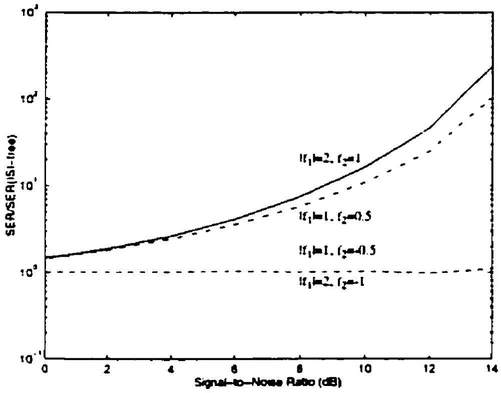


Figure 3.11: Noise performance of the binary second-order PRS-MLSD relative to that of the uncoded ISI-free system.

Also, note that for the codes which are not maximal distance, there will be an unrecoverable SNR loss of $min\ (d_b, d_c) / (h\Delta\sqrt{1 + f_1^2 + f_2^2})$, even when an MLSD technique is employed.

3.5.3. EPR Codes

In addition to the dicode PRS, of our special interest are the EPR codes given by (2.13). Apart from codes for $n = 0, 1$, shown to be maximal distance, $n = 2$ results in a trellis diagram which can also be shown to satisfy (3.17). This may not be true for other values of n. For example, $n = 3, 4$, and 5 lead to codes for which not all of the SNR loss can be recovered. There would be a loss of 2.2 dB, 3.7 dB, and 4.5 dB for each of these cases, respectively [32]. This is one of the reasons the widely accepted EPR codes for use in the magnetic read channels have been limited to $n \leq 3$ [57].

3.6. Practical Non-Idealities

It is often desired to trade the complexity of the MLSD decoder for some degradation in its performance. Finite precision in the digital calculations or of the analog circuitry, limiting the signal, and truncating the length of the path memory are of important related issues. Also, some other impairments in the receiver implementation, such as timing jitter, adversely affect the performance. These subjects are briefly explained in this section. The effects of other analog imperfections will be discussed later.

3.6.1. Path-Memory Truncation

It was mentioned earlier that MLSD, employing the VA, can be started as soon as the first transmitted symbol is received, however, it can not be completed until the whole sequence is available to the receiver. Apparently, in cases where the length of the transmitted sequence is too long, the decoding delay may not be tolerated. In this case, one might prefer to start detecting without receiving every transmitted symbol, at the expense of some degradation in the performance. The amount of degradation depends on the tolerable decoding delay and can be made as small as required by increasing the depth of the path memory. Usually, the path memory is truncated such that the additional probability of error is negligible compared to the MLSD error probability. Although some criteria such as truncating to 4-6 times the constraint length[1] has been suggested in some cases [58, 59], the actual length of the path memory depends on the minimum desirable error probability and varies from one application to another.

In PRS systems, the sensitivity of the MLSD performance to path-memory truncation depends

[1] Constraint length is defined as the size of the encoder memory plus one and is the length of the input data which affects the encoded symbol at each time.

on the interaction between the involved input symbols in the ISI mechanism. For the systems consider here ($1 + f_1 D$), this interaction is maximized if $f_1 = \pm 1$. As $|f_1| \rightarrow 0$, the PRS system moves towards an ISI-free system. Consequently, we expect that for the same amount of degradation, the $1 \pm D$ systems require deeper path memories. Figure 3.12 illustrates performance degradation of the decoder due to truncating the path memory for different values of f_1, obtained from simulations.

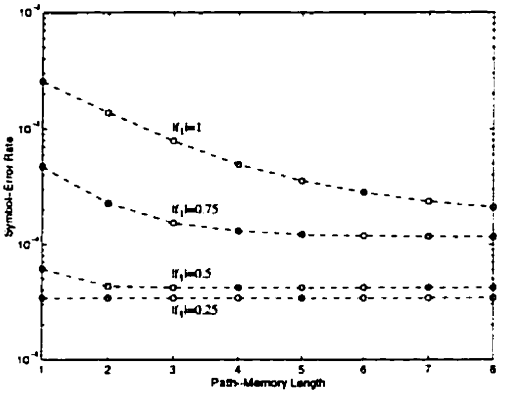


Figure 3.12: Performance degradation of a $1 + f_1 D$ PRS system due to path-memory truncation, at $SNR = 12dB$.

The above plots confirm our intuitive conclusion that as the interaction between the input symbols in generating the PR signal increases, and the spectrum of the signal undergoes more shaping, the decoder needs a deeper path memory to better construct the original sequence.

A related issue to the path-memory truncation, is the method by which the memory is tracked back. In some practical situations, to decrease the amount of signal processing in the decoder, it might be preferred to choose any arbitrary state and track its state-transition back. In general, this local-optimum track back results in a detected sequence which is different from that corresponding to a global-optimum track back. Toward the start, the local-optimum and global-optimum sequences are most likely the same, because of the merging which takes place in the histories of different state-transitions. However, toward the end of the sequences the depth of the path memory decreases and the probability of diverging increases. Even with complete merging, the last n bits in a 2^n-state decoder do not agree. As a result, if the path memory is deep enough, the local-optimum and global-optimum track backs yield the same detected bits. Figure 3.13 depicts typical SER performances of a binary dicode Viterbi decoder in two cases where the global and local optimum sequences have been tracked back.

As can be seen from this figure, the degradation associated with the local-optimum track back

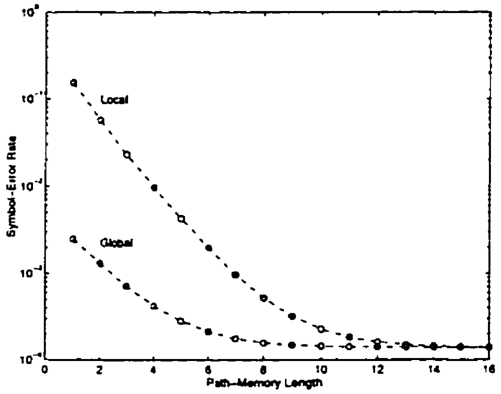


Figure 3.13: SER performance of a binary dicode Viterbi decoder at $SNR = 12dB$, as a function of the path-memory length, when local and global optimum sequences are tracked back.

can be compensated by an increase in the length of the path memory. In cases where the increase in the decoding delay is not a critical issue, the local-optimum track back might be preferable, since increasing the length of the path memory is straight-forward.

3.6.2. Level Limiting

In an actual realization, because of the limited dynamic range of the circuits, limiting the amplitude of the signals is unavoidable. Depending on the distribution of these signals, limiting levels should be set such that any degradation becomes negligible. In this section, we elaborate on the sensitivity of the VA to input-level limiting, in a binary dicode signaling scheme.

With the amplitude of the received signal, $y(k)$, limited to $\pm L$, (3.11) can be invoked to evaluate the performance of the decoder for different values of L. Without limiting, $y(k)$ would be concentrated around 0 and ± 1. So, one would expect a negligible degradation as long as $L \geq 1$. From the other side, (3.11) shows that if $L < 0.5$, there would be no updates during the iterations, and two survivors never merge. This corresponds to detecting either an all "0" or an all "1" digital sequence, which increases the SER to its maximum value of 0.5. As soon as L becomes slightly larger than 0.5, the update mechanism activates and the decoder starts to detect the signal. The number of errors suddenly drops and the error performance approaches its minimum value as L approaches infinity. In practice, values of L greater than 1 have negligible effect. The simulated performance of the decoder and its sensitivity to input-level limiting is plotted in figure 3.14.

It is also useful to look at the statistics of the difference signal in this case. With no updates for $L < 0.5$, $\Delta m(k)$ would always remain equal to zero. As L is increased beyond 0.5 up to 1, the dif-

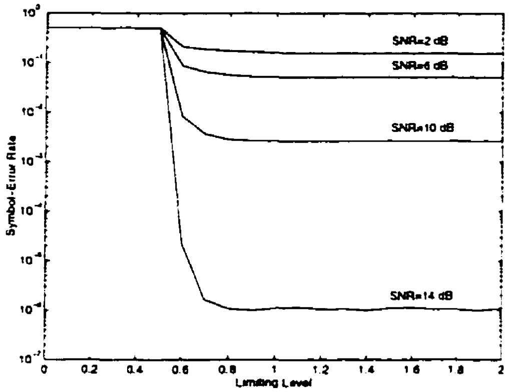


Figure 3.14: SER performance of the difference-metric dicode decoder with level-limited input signal.

ference signal will mostly take one of the values of $\pm(L-0.5)$, however, with some one-sided deviations due to the effect of the level-limited noisy input signal. For $L \geq 1$, where the nominal values of the signal is not affected by the limiter, $\Delta m(k)$ would be concentrated around ±0.5 with a one-sided deviation limited to $\pm(L-0.5)$. As a result, limiting the levels of the received signal to $\pm L$ always limits the difference signal of a dicode Viterbi decoder to $\pm(L-0.5)$. Figure 3.15 shows some typical distributions of the limited input signal and the difference signal developed by the VA.

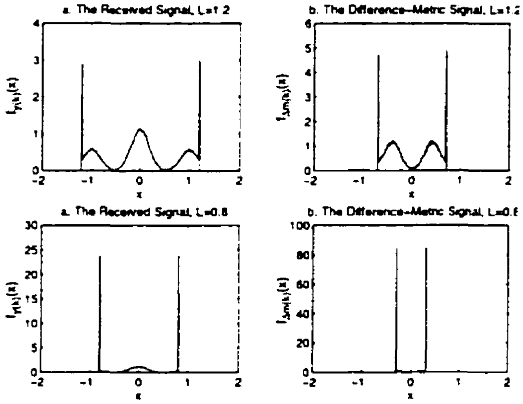


Figure 3.15: Distributions of the input and difference signals in a dicode difference-metric decoder with input-level limiting at $SNR = 12dB$.

3.6.3. Quantization

In a digital realization of the decoder, the limited number of bits used in binary representation of the signals adds another source of imperfection. The effect of this quantization is often considered as an independent additive white noise uniformly distributed over an interval equal to the minimum resolution of the digitized signal [60]. Assuming that the amplitude of the input signal is limited to ± 1, based on the above model the power of the quantization noise injected by a q-bit quantizer will be equal to

$$Q = \frac{1}{3\,(2^{q}-1)^{2}} \tag{3.20}$$

Depending on the relative power of this noise to the channel noise, and sensitivity of the noise performance of the decoder to an increment in SNR, the minimum-required number of bits is determined. The simulated SER degradation of the dicode difference-metric decoder, with an input signal quantized to q bits, is plotted in figure 3.16. The results are also shown for the case where the signal was not quantized and the quantization noise was taken into account as an additional independent noise component in the overall noise, with the power given by (3.20). Good agreement between two cases shows that the above approach in modeling the effect of quantization is applicable here, as well. Apparently, the model becomes more accurate as q increases. Also, it can be concluded that a minimum number of 6 bits is required at moderate-to-relatively-high SNR.

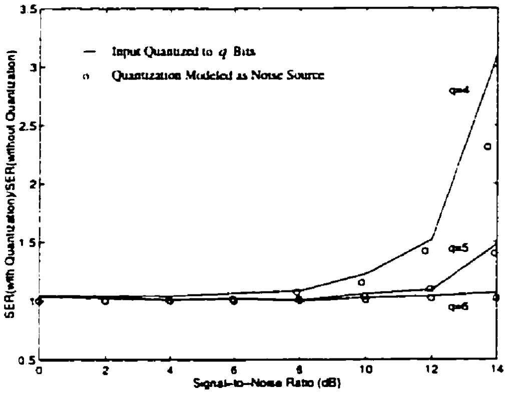


Figure 3.16: Noise performance of a dicode Viterbi decoder when the input signal is quantized.

The required number of bits in the binary representation of the signal, in a digital decoder, can be translated to the accuracy needed in the circuits of its analog counterpart. We shall elaborate more on this when we exclusively focus on the analog decoder.

3.6.4. Timing Jitter

It was mentioned in chapter 2 that any timing perturbation in sampling the signal in the receiver results in an excess ISI. The amount of this ISI depends on the transmitted sequence and the impulse response of the system. For a class-IV system this response is plotted in figure 2.6. From this figure, it can be observed that in a binary scheme, an alternating sequence maximizes the unwanted ISI. Also, it can be verified that in this situation, as an example, a timing phase error of 10° results in an error in the sampled value by about 10%. This corresponds to a maximum achievable SNR of $20dB$. This limit increases to $40dB$, $34dB$, and $26dB$ for the more realistic timing errors of $1°$, $2°$, and $5°$, respectively. In addition to demonstrating the sensitivity of the performance of the PRS system to a timing jitter, the results can be used to determine the required jitter performance of the sampler which enables the receiver to operate below the desired error rate.

3.7. Summary

Maximum-likelihood sequence detection of digital sequences has been shown to considerably outperform the traditional symbol-by-symbol detection technique. The difficulty is, however, the significantly higher volume of computations or, equivalently, the complexity of the detector circuits. Sequence detection, based on the Viterbi algorithm, has been applied to cases where the digital information experiences some inter-symbol interference during the transmission. Applying the Viterbi algorithm to a two-state trellis diagram, resulted from a first-order partial-response signaling scheme with binary input, reduced to a difference-metric algorithm. The significance of this algorithm was that it resulted in propagating only one signal, which could be easily updated. In addition to the simplicity of implementation by itself, the difference-metric algorithm applied to a dicode partial-response signal was further simplified. It will be shown later that the resulting algorithm, called the input-interleaved algorithm, realized in an analog fashion, outperforms any other detector reported so far.

Also, it was shown that almost all of the loss in the performance of the symbol-by-symbol partial-response detector, compared to the uncoded communication system, can be recovered by a sequence detector, if a maximal-distance code is used. Although all of the maximal-distance codes are not necessarily important in practice, the concept was used to prove that partial-response systems do not inherently perform worse than uncoded systems.

Finally, some practical issues in implementing a dicode sequence detector were investigated. As a result, some parameters, which are important in practice such as limiting levels of the circuits and minimum required number of bits (or needed accuracy) in representing the signals, were derived.

Chapter 4

Class-IV Partial-Response Analog Viterbi Decoder

Analog Viterbi decoders have recently been shown to be viable alternatives to their traditional digital counterparts. Analog detectors offer the advantages of reduced power and size primarily due to the elimination of the analog-to-digital converter (A/D). For the applications considered here, the saving is significant. For example, the 6-bit A/D required in a fast and small digital PRML disk drive is likely to dissipate more power than the entire read-channel power budget. Also, since in this application high accuracy is not needed, simple analog circuits can be employed to realize the VA. Using simple circuitry often translates to analog implementations which operate faster and lead to further savings in power and area in implementing the decoder itself. For example, 36 gates are required to perform a 4-bit digital summation, whereas two simple current sources perform the same function in the analog domain.

The above advantages of analog implementations have been investigated and demonstrated [55, 56, 61-71] and a commercial realization for magnetic read channels is currently available

[70]. Higher-rate analog products will likely appear in near future [72]. Although digital signal processing has been considered for, and employed in, most of the advanced read-channel chips (even with analog front-ends) [73-88], an industry-based survey indicates that analog realizations operate at considerably higher speeds and lower power consumptions when fabricated in similar processes [16]. Table 4.1 summarizes some of the results. Straight-forward design procedure and less dependency of the design to process parameters are the reasons most of the companies have pursued digital approaches so far. However, the fact that the required equalization and timing recovery can be also performed in the analog domain [66, 70, 79, 83, 89] and increasing demands on the size, power consumption, and speed seems to force most of the companies toward analog realizations [90].

Table 4.1: Summary of the advanced read-channel chips.

	IBM	Crystal Semiconductor/ Cirrus Logic	Silicon Systems	TI/ Quantum	IBM	ATT	IBM
Published in	1991	1994	1994	1994	1994	1994	1995
Channel	PR4	EPR4/ EPR6	PR4	PR4	PR4	PR4	PR4/ EPR4
Detector	Digital Viterbi	Digital Viterbi	Analog Viterbi	Digital Viterbi	Digital Viterbi	Digital Viterbi	Digital Viterbi
Process (μ)	1.2 BiCMOS	0.8 CMOS	1 BiCMOS	0.8 BiCMOS/ 0.8 CMOS	1 BiCMOS	0.6 CMOS	0.5 BiCMOS
Chip Area (mm^2)	30	51	24	? 2 Chips	56	25	25
Power (W)/ Rate (MS/s)	1/ 27	0.69/ 36	0.78/ 112	1.85/ 72	1.75/ 65	0.75/ 82	1.25/ 128

This chapter describes a high-speed low-power analog Viterbi decoder for class-IV PR signals. A new architecture based on input interleaving is introduced that substantially increases the speed of the analog circuitry. The decoder was fabricated in a $0.8\mu m$ BiCMOS process, consumes $30mW$ power from a $3.3V$ single supply while operating at $200MS/s$. The chip contains two time-interleaved dicodes each consisting of a fully differential analog processing core and a digital path memory. The total core area is only $0.5mm^2$.

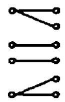
The chapter starts with deriving the analog algorithm and describes the designed decoder in detail from a top level down to layout. Experimental results of a discrete prototype and an integrated-circuit (IC) implementation of two different decoders are also given.

4.1. Analog Detector: An Adaptive Threshold Device[1]

It was mentioned in chapter 3 that a class-IV decoder can be realized by time interleaving two independent dicode decoders. In addition to the benefit of operating each dicode at half the symbol rate, in an MLSD decoder this decomposition is helpful in that it reduces the 4-state Viterbi decoder to two 2-state decoders. Furthermore, the difference-metric algorithm described in section 3.2.1 can be invoked to decode each dicode signal. This is particularly important in an analog implementation, since, as will be shown shortly, the difference metric algorithm (and our new derivation, the input-interleaved algorithm) can be efficiently realized in the analog domain.

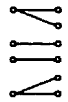
The difference-metric algorithm applied to a dicode scheme results by inserting $f_1 = -1$ in (3.8). This yields

$$\Delta m(k) = \begin{cases} y(k) - 0.5 & , \quad \Delta m(k-1) < y(k) - 0.5 \\ \Delta m(k-1) & , y(k) - 0.5 < \Delta m(k-1) < y(k) + 0.5 \\ y(k) + 0.5 & , y(k) + 0.5 < \Delta m(k-1) \end{cases} \tag{4.1}$$



Alternatively, the decision regions can be described by slicing the input signal to the decoder, $y(k)$. The result is

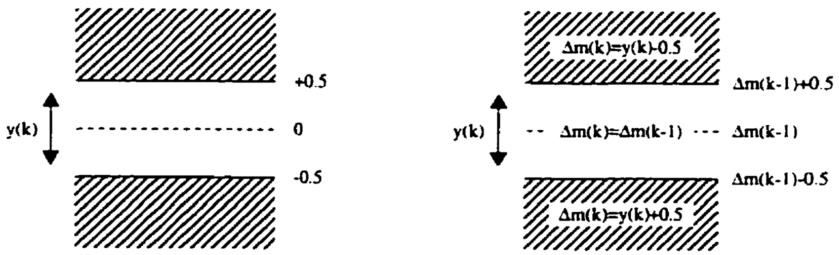
$$\Delta m(k) = \begin{cases} y(k) - 0.5 & , \quad y(k) > \Delta m(k-1) + 0.5 \\ \Delta m(k-1) & , \Delta m(k-1) + 0.5 > y(k) > \Delta m(k-1) - 0.5 \\ y(k) + 0.5 & , \Delta m(k-1) - 0.5 > y(k) \end{cases} \tag{4.2}$$



The latter representation is interesting in that it interprets the Viterbi detector as an "*adaptive-threshold device*" [55], and makes the comparison between the traditional symbol-by-symbol detector and the sequence detector more interesting from an implementation point-of-view. The symbol-by-symbol detector shown in figure 2.11 employs a threshold device with threshold levels at ± 0.5, for a ternary dicode signal. Comparing this detector with that described by (4.2), shows that the difference-metric Viterbi detector can be viewed as a threshold device which adaptively biases its threshold levels using $\Delta m(k-1)$. Figure 4.1 illustrates the decision regions used by these detectors.

One direct conclusion of the above comparison is that the amount of signal processing required in the Viterbi detector is not much more than that of the symbol-by-symbol detector. A sequence detector should be considered even when implementation complexity is an issue. Also, note that the path memory required by the Viterbi detector consists of a small number of simple

[1] The "adaptive threshold" referred to here, should not be confused with the "dynamic threshold" described in [91].



a. Symbol-by-Symbol Detector

b. Viterbi Detector

Figure 4.1: Decision regions in dicode detectors.

digital blocks.

4.2. The Adaptive-Threshold Architecture

The update mechanism given by (4.1) states that the new value of the difference metric signal is a hard-limited version of the previous value to the limiting levels $y(k) \pm 0.5$. Figure 4.2 depicts this relationship.

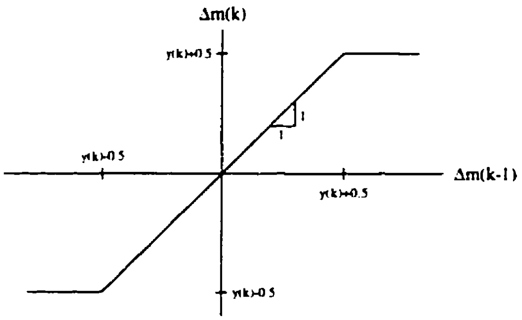


Figure 4.2: Limiter interpretation of the adaptive-threshold Viterbi detector for decoding a dicode signal.

In addition to updating the difference signal, hard information regarding the decision region in which the previous difference signal falls in, is also required. This information updates the contents of the path memory based on the survivor extensions shown in (4.1). The two aforementioned goals can be achieved by the simple block diagram shown in figure 4.3. In this block diagram, two DC-shifted versions of the input signal are generated and compared with the previous value of the difference signal stored by an intermediate multiplexing sample and hold (S/H). In the comparison phase, the outputs of the latched comparators are held at ground and both of the switches of the intermediate S/H are open, retaining the previously-stored voltage. Based on the comparison results, and during the latch phase, the stored value is either updated to one of the generated signals or is not updated at all. Note that the multiplexing S/H may track only one of its inputs at each time step, as two comparators can not output a “1” at the same time.

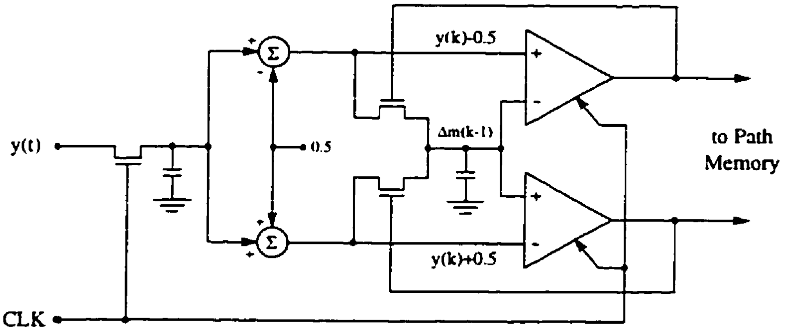


Figure 4.3: Block-diagram realization of the adaptive-threshold Viterbi detector for decoding a dicode signal (the adaptive-threshold architecture).

The precision of the analog circuits is modest, since typically only six bit accuracy is needed in this type of decoder[1]. Also, as seen in figure 4.3, the complexity of the decoder is estimated to be near to that of a 2-bit A/D. Comparing to a 6-bit A/D used in a digital realization, speed and power benefits are expected to be gained through this analog implementation.

In addition to updating the difference signal by properly controlling the multiplexing S/H, the outputs of the comparators are also used to update the path memory. For best performance, the final decision must be made only after all the paths in the path memory have merged. In practice, however, truncation is used to reduce the decoding delay and size of the required memory. The truncation length is determined such that the excess SER is negligible[2]. The register-exchange method is a commonly used technique for storage of the survivor sequences in Viterbi decoders with low number of states [52]. In this method, one shift register with a length equal to the truncated length of the path memory is used for each state. The different shift registers are then inter-connected according to the trellis diagram such that the optimum paths throughout the trellis are directly mapped into digital sequences stored by these shift registers. Applying this method to the dicode decoder results in two inter-connected serial/parallel in/out shift registers shown in figure 4.4. The serial/parallel loadings are controlled by the comparator outputs at the end of each iteration.

The adaptive-threshold architecture shown in figure 4.3 was the fastest architecture at the time it was proposed [55]. The key point was it avoids using any analog signal in the feedback path of

[1] See section 3.6.3.

[2] See section 3.6.1.

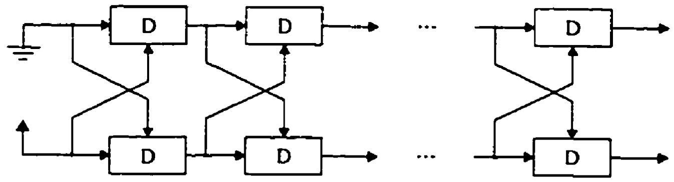


Figure 4.4: The decoder path memory based on the register-exchange method.

the decoder. In addition to reducing the sensitivity of the structure to analog imperfections (to be discussed later), absence of analog signals in the feedback path eliminates the need for a delay in the analog signal and increases the speed. The delay is necessary to prevent a destructive feedback and was implemented by using a master/slave S/H in [69]. Eliminating this S/H greatly improves the speed, as the S/H plays a major role in this regard [67]. Other minor advantages are also present, as the front-end circuit in the presented structure is as simple as a DC-level shifter. All these improvements have been achieved by fully exploiting the difference-metric algorithm in the proposed analog implementation[1].

4.3. The Input-Interleaved Architecture

Although the adaptive-threshold architecture can lead to a fast analog decoder, higher speeds can be obtained by modifying the Viterbi algorithm into what we refer to as the "*input-interleaved algorithm*". The basic idea is to eliminate the intermediate S/H in figure 4.3 to avoid latency due to its settling. This elimination is possible, as the value sampled by this S/H is a function of an already-sampled signal.

Recall from section 3.3 that the input-interleaved algorithm proceeds by updating two quantities $y(j)$ and u according to the iterations given by (3.14). This expression is repeated here for convenience

$$\begin{cases} y(j) = y(k), u = 0 & , \quad y(j) + u < y(k) \\ y(j) = y(j), u = u & , y(j) + u - 1 < y(k) < y(j) + u \\ y(j) = y(k), u = 1 & . \quad y(k) < y(j) + u - 1 \end{cases} \tag{4.3}$$

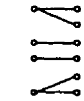


Figure 4.5 depicts a graphical sketch of the input-interleaved algorithm. If $u = 1$, the decoder sets its threshold levels to $y(j)$ and $y(j) + 1$. The previously-sampled input signal will be retained only if the input signal lies between two threshold levels, and will be updated to the current input if it falls outside these two levels. At the same time, the DC offset will not be changed if

[1] In fact, (4.1) and (4.2) were derived from scratch in the search for a suitable analog realization.

the input is less than $y(j) + 1$ and will be switched to 0 if it is more than $y(j) + 1$. The decoder uses the new values of $y(j)$ and u in the next iteration, however, if the DC offset is switched to 0, the new threshold levels will be $y(j) - 1$ and $y(j)$. The situation for $u = 0$ is similar, and is shown in figure 4.5.b.

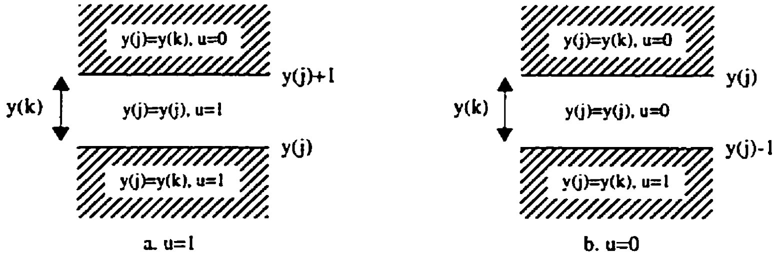


Figure 4.5: Graphical illustration of the input-interleaved algorithm.

The above algorithm can be implemented by the block diagram shown in figure 4.6. The front-end consists of two S/Hs. While the input signal is sampled and stored by one of the S/Hs, the previous input sample is held by the other S/H. The connections between these S/Hs resemble an interleaved structure, giving rise to the name "*input-interleaved*" for the algorithm and the architecture that realizes it.

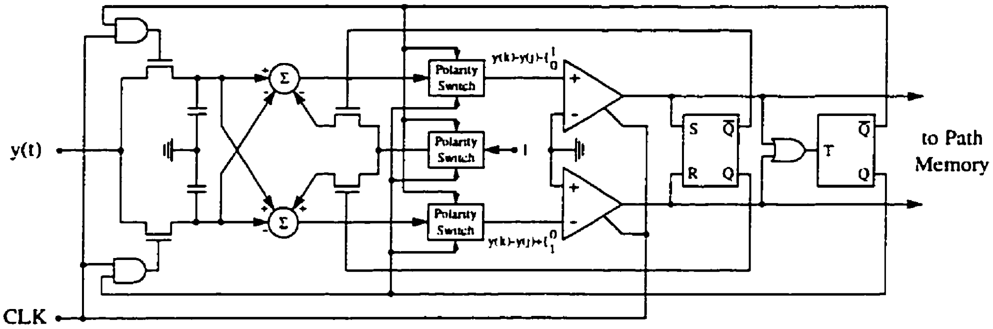


Figure 4.6: Block-diagram realization of the input-interleaved Viterbi detector (the input-interleaved architecture).

In figure 4.6, constructing the threshold levels and comparing them with the input signal are accomplished by properly combining the current input, the previously-sampled input, and the DC offset, and checking the polarities of the resulting combinations. The two input signals are combined together in two separate branches, with the DC offset introduced to only one of them. The branch to which the offset is added, is determined from the results of the previous iteration. Thus, adding the offset to the upper branch corresponds to having $u = 1$ in (4.3) and figure 4.5, whereas adding it to the lower branch corresponds to $u = 0$. The comparator outputs are utilized to generate the required switching signal, S (to switch the DC offset between two branches), and the toggling signal, T (to toggle the input S/Hs if an update in $y(j)$ is needed), as well as to update the contents of the path memory.

The update mechanisms illustrated in figure 4.5 clearly determine the rule for generating the signals S and T. The input S/Hs should toggle whenever an update in $y(j)$ is required (i.e. when the input signal exceeds the region between two threshold levels.). In figure 4.6, it can be easily verified that if the input signal lies in between these levels, none of the comparator outputs will be "1", and if $y(k)$ exceeds this region, then one (and only one) of the comparators will result in a "1". Consequently, a toggling should occur if either one of the outputs is "1". This is accomplished by employing a T flip-flop toggled by both of the comparators. If the DC offset is already added to the upper branch, it should be switched to the lower branch only if $y(k)$ is above the upper threshold, that is, if the output of the upper comparator is "1" (Note that the lower comparator outputs a "0".). The offset should only be switched back to the upper branch if $y(k)$ falls below the lower threshold level. In this case, the lower comparator will have a "1" at the output and the upper comparator outputs a "0". As a result, an SR flip-flop can be employed to switch the DC offset signal back and forth between two branches. This flip-flop should be set by the upper comparator and reset by the lower comparator. When $y(k)$ lies in the middle decision region, none of the comparators output a "1", and the SR flip-flop retains its state.

Note that in figure 4.6, in combining the sampled-input signals, a sign change results whenever a toggling occurs. These sign changes are compensated by utilizing polarity switches which are controlled by the toggling flip-flop.

Finally, it should be mentioned that the path memory of the input-interleaved decoder is identical to that of the adaptive-threshold decoder. This memory structure is illustrated in figure 4.4.

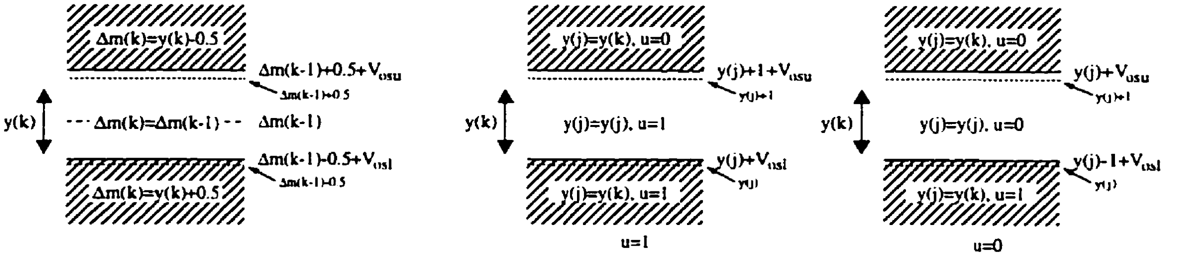
4.4. Imperfections

Analog implementations usually suffer from DC offsets, mismatches, and charge injections. Other imperfections such as finite precision in processing, truncating the path memory, limiting the input signal, and timing jitter in the sampling instances affect both digital and analog realizations. These effects were investigated in chapter 3. To examine the sensitivity of the proposed structures to analog imperfections, the derived algorithms were simulated in the presence of these impairments. In what follows, major sources of errors are considered and performance degradations of the adaptive-threshold and input-interleaved decoders are evaluated. It will be shown that these decoders show low sensitivities to these impairments.

Note that in our evaluations, the amounts of the impairments have been often exaggerated. This is to illustrate the robustness of the decoders against imperfections. This robustness makes the presented structures suitable for analog implementation. Finally, it should be mentioned that these non-ideal effects are expressed in percentage of the PR-signal peak-value.

4.4.1. Comparator Offset

Due to the fact that comparators are one of the most important sources of DC offsets in analog designs, this offset and its effects on the performances of the analog decoders are considered here in more detail. In addition, it will be shown that some other impairments can be modeled by equivalent comparator offsets, and the results presented here can be directly applied. In our analog decoders, an offset introduced by one of the comparators can be translated to a shift in the threshold level realized by that comparator. Figure 4.7 illustrates the concept when offsets equal to V_{osu} and V_{osl} are considered for the upper and lower comparators in figures 4.3 and 4.6 respectively.



a. Adaptive-Threshold Decoder

b. Input-Interleaved Decoder

Figure 4.7: Effects of comparator offsets on the threshold levels of the decoders (Figures 4.1.a and 4.5).

As can be seen from the above figure, the offsets do not affect the performance if the input signal does not lie in the regions between the original threshold levels and the shifted levels. Otherwise, an error equal to either V_{osu} or V_{osl} will occur in updating the threshold levels. Although the effect can be modeled as a noise added to the input signal, the fact that this additional noise is neither Gaussian nor uncorrelated makes simulations more appealing. The bit-error rate (BER) performances are plotted in figure 4.8. The Viterbi bound plotted in this figure is given by (3.15) at high SNR and obtained by simulations at low SNR. The threshold-device performance was evaluated in section 2.4.1 and is included here for comparison purposes.

Alternatively, the degradation can be expressed in the form of the loss in the coding gain. Figure 4.9 is a sample plot which illustrates the achievable coding gain as a function of the comparator offsets at a BER of 10^{-6}.

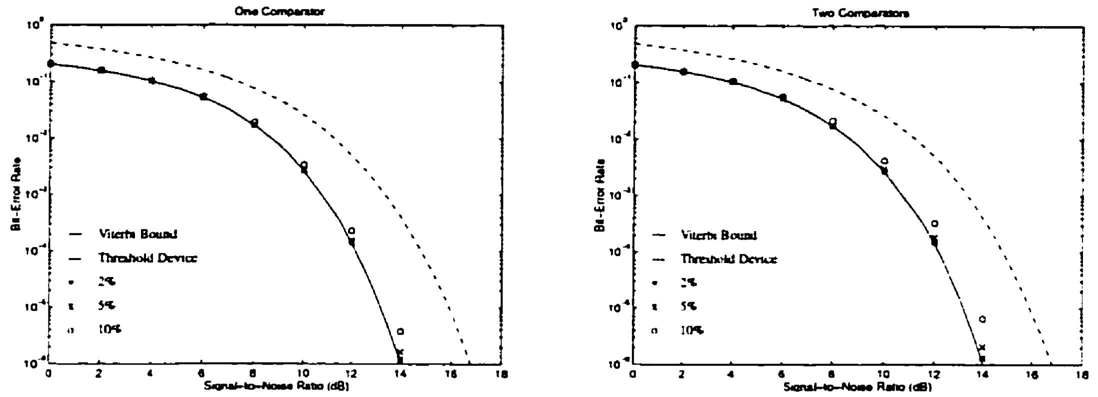


Figure 4.8: Performance degradation of the analog decoders due to the comparator offsets.

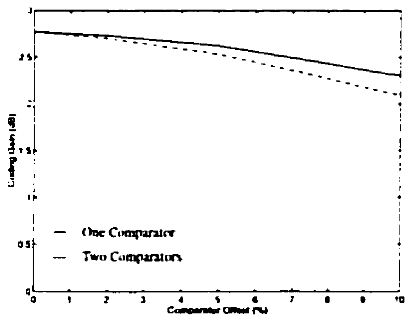


Figure 4.9: Performance degradation of the adaptive-threshold and input-interleaved decoders due to the comparator offsets at $BER = 10^{-6}$. The degradation is expressed in terms of the achievable coding gain.

These results show that the adaptive-threshold and the input-interleaved decoders are tolerant against comparator offsets and with a careful design (and for reasonable signal amplitudes) simple comparators without offset cancelation techniques can be employed.

4.4.2. Combiner Offset

Offsets produced at the outputs of the combiners in figures 4.3 and 4.6 also affect performance. For the input-interleaved decoder, it can be easily seen that these offsets are equivalent to offsets in the corresponding comparators. However, for the adaptive-threshold decoder, the situation is different. Figure 4.10 illustrates this decoder when offsets equal to V_{os1} and V_{os2} are present at the output of the combiners. These offsets can be represented by equivalent voltage sources shown in figure 4.10.b. If the last-closed switch is the upper switch, V_{os1} has been sampled and stored by the multiplexing S/H. This means that the output of the upper comparator will not be affected by the offset, but, the lower comparator, in conjunction with the equivalent sources, can be modeled by a comparator with an offset equal to $V_{os1} - V_{os2}$. However, if the lower switch is the last-closed switch, V_{os2} has been stored, the upper comparator models a comparator with offset $V_{os1} - V_{os2}$, and the lower comparator behaves offset-free.

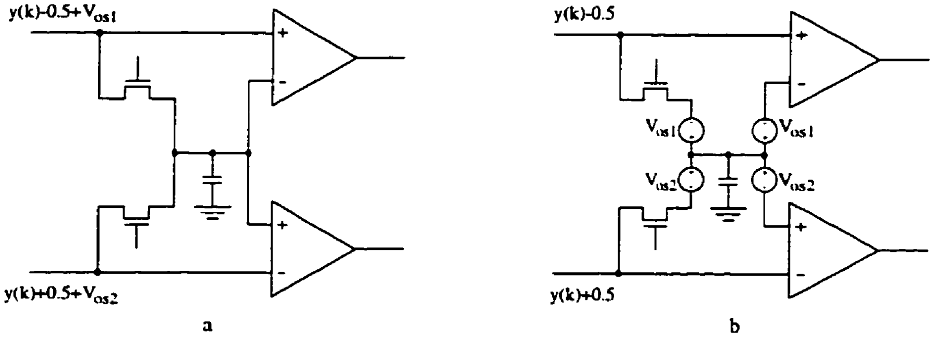
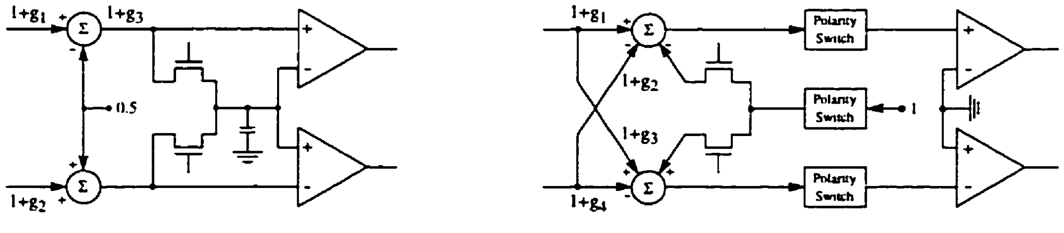


Figure 4.10: Offsets at the output of combiners (a) and their equivalent representation (b).

The above argument shows that a differential offset has the same effect on the performance as an equal offset introduced by only one of the comparators does. In addition, the adaptive-threshold decoder is insensitive to any common-mode offset. This insensitivity can be also explained by the common-mode rejection property of the structure, since any common-mode signal will be sampled by the multiplexing S/H and cancelled in the next iteration.

4.4.3. Gain Mismatches

Gain mismatches result if the relative weights associated with the inputs to the combiners deviate from their nominal values. In the input-interleaved decoder, however, there is less requirement on the gain matching. Two sets of weights, associated with two combiners, can be scaled independently without affecting the performance. The effect of gain mismatches can be quantified by considering gain deviation factors shown in figure 4.11.



a. Difference Metric

b. Input Interleaved

Figure 4.11: Gain deviation factors used to quantify mismatches in two types of analog decoders.

Due to the symmetry in the input-interleaved decoder, shown in figure 4.11.b, this structure exhibits identical sensitivities to all of the gain deviation factors. However, this is only true for deviation factors g_1 and g_2 in the adaptive-threshold decoder, in figure 4.11.a. Figure 4.12 depicts these sensitivities to different gain mismatches. Note that both of these decoders exhibit low sensitivities to each one of the gain mismatches.

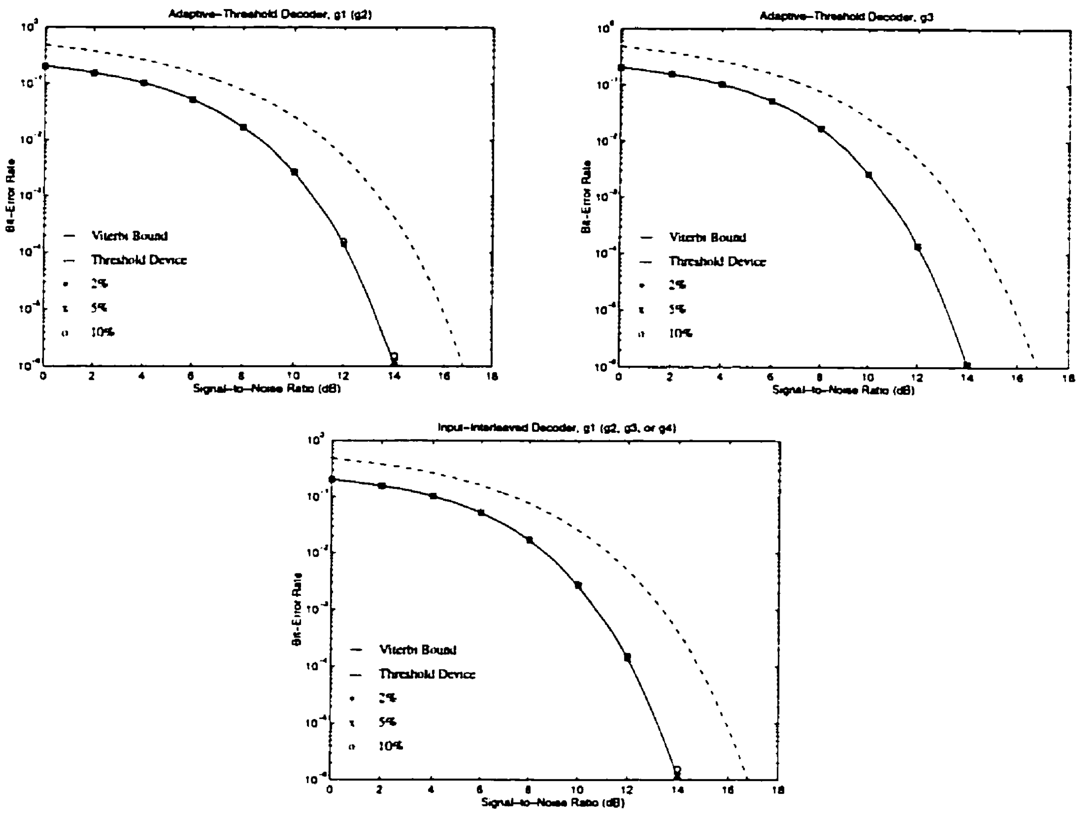


Figure 4.12: Effects of different gain mismatches on the performance of the decoders.

In figure 4.13, the overall effects of gain mismatches are shown when all of these deviations are present. From the different possible combinations, the worst case has been shown.

4.4.4. Sensitivity to the Reference Voltage

In an actual circuit, a reference voltage realizes the constant values used in the decoder block diagrams. Recall that these values were obtained from normalizing peaks of the PR signal, and based on other requirements (such as supply voltage) the scaling factor is determined.

Although it might not be clear from their block diagrams, the adaptive-threshold and the input-interleaved structures show identical sensitivities to changes in the reference voltage. A change in the reference voltage is equivalent to having a differential offset at the combiner outputs in the adaptive-threshold structure. In section 4.4.2 it was shown that the impact of such an offset is, in

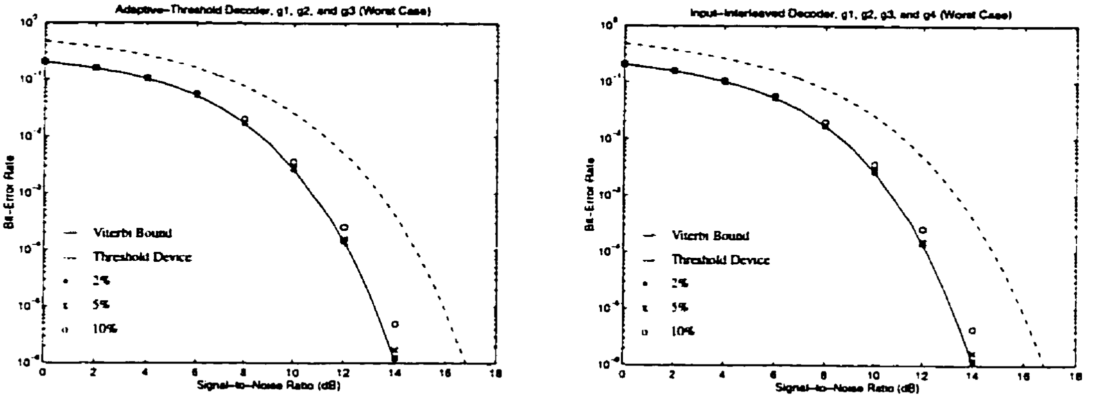


Figure 4.13: Worst-case performance degradation when all of the mismatches are present.

turn, equivalent to that caused by the same amount of offset introduced by either one of the comparators. If the input-interleaved decoder is examined from this point-of-view, the equivalency between a change in the reference voltage and the same amount of offset introduced by one of the comparators becomes evident. Every time the DC voltage is added to one branch, the impact is identical to that of an equal offset in the corresponding comparator.

Note that the above sensitivities are also applicable to the cases where the input signal undergoes either an unwanted attenuation or amplification. This is based on the fact that these decoders are only sensitive to the relative amplitudes of the input signal and the reference voltage. This also explains why the decoders have identical sensitivities to changes in the reference voltage.

4.4.5. Charge Injection and Clock Feedthrough

As any other S/H, the S/Hs used in the analog decoders contribute to the errors by partly injecting their channel charges and clock signals to the stored values. The change in the voltage value is a function of the input signal. However, if the input signal fluctuation is small, the input dependency of the injected voltage can be neglected. This is the case in the decoders described here, where the peak-to-peak of the input signal is only a fraction of a volt. Ignoring the signal-dependent part of the injected value, one can conclude that the input-interleaved structure is insensitive to charge injection, since two injected signals cancel each other in the combiner. In the adaptive-threshold structure, two sources of charge injection are present. The charge injected by the input S/H does not have any effect as it appears as a common-mode signal. In contrast to this, the charge injected by the intermediate S/H degrades the performance of the decoder and can be considered as equivalent offsets for the comparators. It should be emphasized here that the signal-independent charge injection and clock feedthrough will be further reduced in a fully-differential realization.

4.5. Integrated-Circuit Implementation

The adaptive-threshold structure was evaluated in practice by testing a discrete prototype. However, since achieving high speeds was the primary goal, the input-interleaved decoder was chosen for an IC realization. Note that the complexity of this decoder is only slightly more than that of the difference metric decoder, since the overhead circuitry is very simple and may be absorbed in the adjacent blocks as will be described later in this section.

A possible circuit implementation of the input-interleaved structure is illustrated in figure 4.14. All signals are differential to combat destructive effects such as common-mode noise and charge injection.

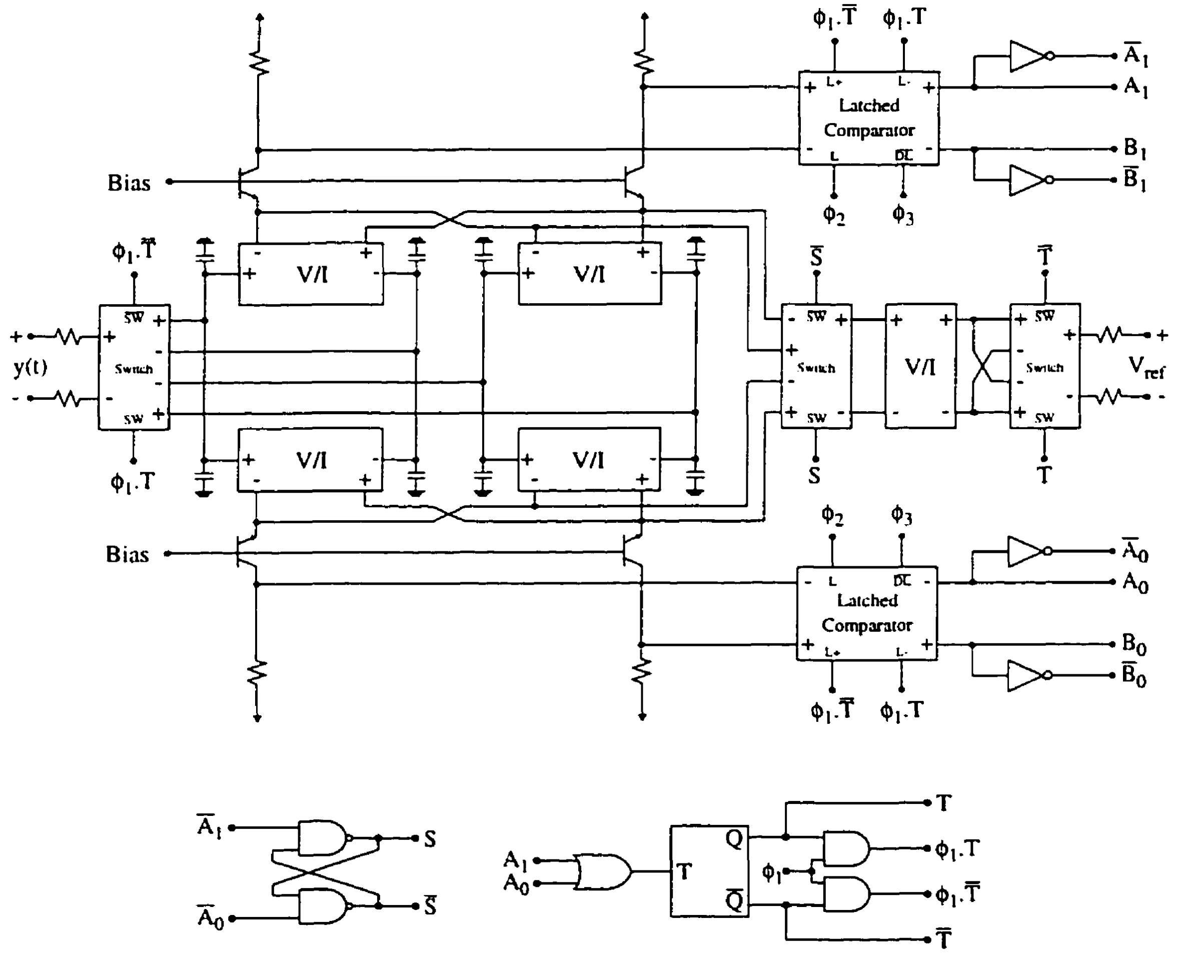


Figure 4.14: Circuit realization of the input-interleaved Viterbi detector.

In figure 4.14, the input S/Hs, consisting of a differential dual switch connected to holding capacitors, store the present and the previous input signals. These signals are converted to currents and combined with appropriate polarities by passing the currents through resistors via cascode transistors. The role of the cascode stages is to maximize the bandwidth when the voltage-to-current converters are connected to the load resistors. These stages are more helpful if a large gain is required and are not critical in our design. As shown in the block diagram in figure 4.6, a DC signal should be added to either one of the above combinations. A DC voltage obtained from an off-chip differential reference is also converted to current, appropriately switched, and added to one of the parallel branches. The series resistances connected to the signal and DC input terminals are small and are used as current-limit protections. Each holding capacitor is divided into two parallel capacitors to help achieving symmetry in the layout.

The resulting differential voltages are applied to two latched comparators which decide on the polarity of these signals. Comparison results are used to update the path memory, and also to generate the toggling signal, T, and the switching signal, S. These signals are fed back to possibly update the previously-sampled input signal, by toggling the input S/H, and the DC offset signal, by switching it from one branch to the other.

Φ_1 to Φ_3, shown in figure 4.14, are different phases of a clock signal. These phases are obtained from a master clock by a simple circuit which will be discussed later.

Based on the realizations shown in figures 4.4 and 4.14, an analog Viterbi decoder chip was designed and fabricated in a BiCMOS process. The chip contains two input-interleaved dicode decoders which were internally time interleaved to decode a class-IV PR signal. In what follows, the design issues of the different building blocks will be addressed and the circuits will be explained in detail. All the BJTs shown in the circuits have the minimum size and the gates of the MOSFETs have the minimum length of $0.8\mu m$ unless it is specified. Widths of the MOS devices (in μm) and specifications of other components are given in each figure.

4.5.1. Differential Dual Switch

A differential dual switch was used in a variety of locations. This switch consists of four NMOS transistors, connected as shown in figure 4.15, and has one differential input and two differential outputs. Two complementary digital signals determine which output the input signal should be directed to.

The above switch was utilized to implement the input S/H, to switch the offset signal back and

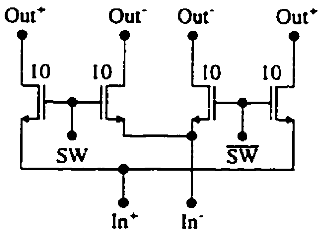


Figure 4.15: The differential dual switch used in figure 4.14.

forth between two branches, and to serve as a polarity switch for the reference voltage, all in a differential manner. Also, this switch was employed to perform the AND functions shown in figure 4.14, as will be described later.

Only NMOS transistors are used, since the voltage levels of the signals handled by the switches are considerably lower than the supply voltage minus the transistor threshold voltage.

4.5.2. Voltage-to-Current Converter

A BJT differential pair, linearized by emitter degeneration, was used to realize the voltage-to-current converter (V/I). Source follower input stages provide the required high input impedances as well as the necessary level shifts. As a result of these level shifts, the on-resistances of the input switches are minimized by reducing the input common-mode to near ground. The V/I circuit is depicted in figure 4.16.

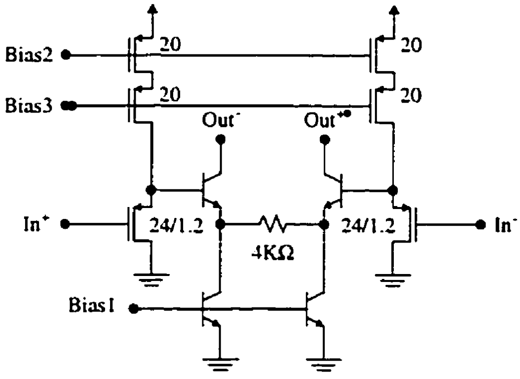


Figure 4.16: Circuit realization of the V/I block.

As shown in figure 4.14, the resulting currents, after combining, are converted back to voltage by resistors. The voltage gain is dominantly set by the ratio of the load resistors to the degeneration resistor in the V/I circuit. Good matching between poly resistors used in the IC implementation results in having very low gain mismatches and DC offsets when combining the signals. Note that mismatches between biasing current sources also contribute to generating the offset. By employing

BJTs and careful layout design, these mismatches can be kept very low as well.

4.5.3. Latched Comparator

In designing comparators, the use of a BiCMOS process aids a circuit designer. The resolution of a comparator is often limited by its offset voltage and becomes more severe if the input dynamic range is small. Offset analysis shows that, in general, CMOS latches exhibit much more offsets compared to bipolar latches [92]. This offset can be partly overcome by utilizing a high-gain preamplifier before the latch. Offset of this preamplifier should be kept small, as, in turn, it contributes to the overall input-referred offset voltage of the comparator. In a CMOS realization, large size transistors and/or offset cancelation techniques can help in reducing the offset, however, the speed of operation will be reduced. On the other hand, a bipolar latch has a lower offset and permits a smaller gain in the preamplifier, resulting in a correspondingly faster response. However, bipolar comparators do not have rail-to-rail output swings, required in many applications. All of the above advantages can be attained in a BiCMOS process, where both of these devices are available. The basic idea is to obtain a low input-referred offset voltage by amplifying the signal with a high-gain wide-band low-offset bipolar preamplifier prior to applying it to a CMOS latch [93]. The availability of bipolar transistors can further be appreciated if a bipolar latch is interposed between the preamplifier and the CMOS output latch [92]. This relaxes the constraints on the preamplifier and is particularly helpful in high-speed low-power designs. In fact, it has been concluded that to minimize the power-delay product, the amplification required in a comparator is best obtained by means of regeneration [94].

In the design presented here, as shown in figure 4.17, the differential signal is first amplified with a gain of less than 10, by activating one of the differential pairs of Q_1 and Q_2 or Q_3 and Q_4. Rather, the amplification is mostly done by incorporating Q_5 and Q_6 in a positive feedback configuration. Regeneration initiates at the beginning of the latch phase, L. Slightly after this positive feedback is started and a large-enough signal is developed, a CMOS latch is activated to produce a rail-to-rail swing output signal. The output CMOS latch is controlled by a delayed version of the latch signal, L. Both of the regenerative stages will be reset during the next amplifying phase.

Two cross-coupled differential pairs in the preamplifier provide the capability of reversing the polarity of the signal. This capability allows us to compensate for the sign change, which results each time a toggling between the input S/Hs occurs, by biasing one of the differential pairs at a time. Note that this polarity change is in conjunction with a change in the polarity of the DC signal, as shown in figure 4.14.

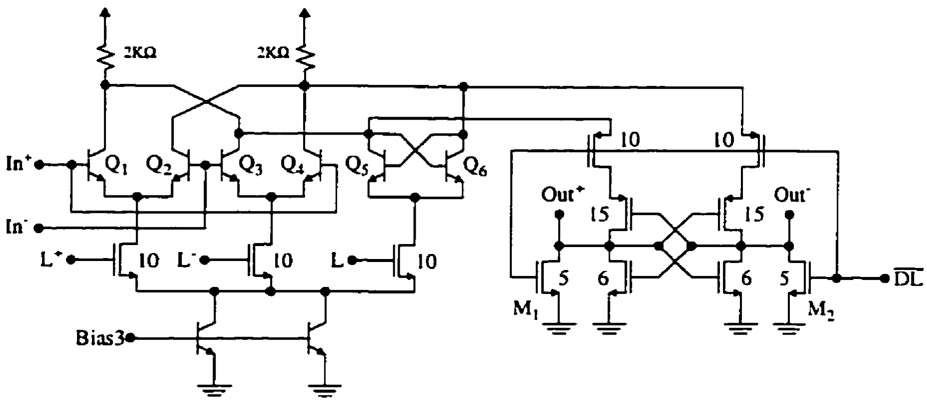


Figure 4.17: The latched-comparator circuit.

4.5.4. Multiplexed-Input D Flip-Flop

As shown in figure 4.4, the path memory of the decoder is composed of two interconnected strings of D flip-flops. Each flip-flop should be able to accept either a serial or a parallel input. This capability can be provided by using a regular latch with a 2-to-1 multiplexer at its input. Figure 4.18 depicts a typical circuit realization of the multiplexed-input D flip-flop.

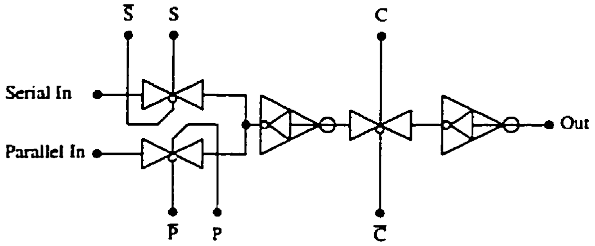


Figure 4.18: Multiplexed-input D flip-flop used to implement the path memory. Two small feedback inverters are shown inside the feed-forward inverters.

As the above figure shows, a dynamic latch is used to construct the flip-flop, however, this latch has been converted to a static latch by means of small feedback inverters [95]. These inverters are used to help the circuit to maintain its internal states, until a new load takes place, and should not have large driving capabilities. A large driving capability will prevent the new data from overwriting the old data once a loading command is issued. This can be achieved by employing small transistors in the inverter circuit.

Note that in the above realization, only one of the serial and parallel control signals, S and P, should become high at a time. The selected input will be transferred to the output at the positive edge of C. As will be seen shortly, the above flip-flop functions as a master-slave flip-flop in the path memory structure.

4.5.5. Path Memory

The path memory consists of 2-by-12 multiplexed-input D flip-flops utilized in the structure illustrated in figure 4.4. Figure 4.19 depicts the result.

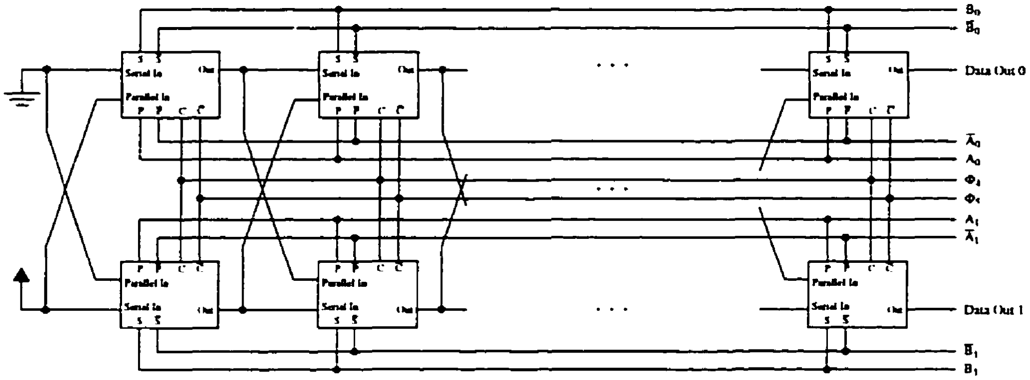


Figure 4.19: Path memory implementation using multiplexed-input D flip-flops.

In this figure, A_0, $\overline{A_0}$, B_0, $\overline{B_0}$, A_1, $\overline{A_1}$, B_1, and $\overline{B_1}$ are outputs of the latched comparators, and their complements, shown in figure 4.14. These signals control the serial and parallel loadings of the shift registers. Based on the decision regions sliced by the comparators (illustrated in figure 4.5 in conjunction with (4.3)), either a serial/parallel, a serial/serial, or a parallel/serial loading should occur in the contents of the upper/lower (corresponding to states "0/1") shift registers, in figure 4.19. From both of the outputs of each one of the comparators, which are initially pulled down to ground by transistors M_1 and M_2 in figure 4.17, one and only one will be pulled up during the latch phase. These positive transitions can be used to perform the required loadings. Table 4.2 summarizes the loading rules. This table demonstrates that the serial loading of shift register "0" should be controlled by complementary signals B_0 and $\overline{B_0}$, the serial loading of shift register "1" by B_1 and $\overline{B_1}$, the parallel loading of shift register "0" by A_0 and $\overline{A_0}$, and the parallel loading of shift register "1" by A_1 and $\overline{A_1}$.

Table 4.2: Loading rules for the shift registers of the path memory.

A_1	$\overline{A_1}$	B_1	$\overline{B_1}$	A_0	$\overline{A_0}$	B_0	$\overline{B_0}$	Upper ("0") Shift Register	Lower ("1") Shift Register
⎽⎽|⎺⎺	⎺⎺|⎽⎽	⎽⎽⎽⎽	⎺⎺⎺⎺	⎽⎽⎽⎽	⎺⎺⎺⎺	⎽⎽|⎺⎺	⎺⎺|⎽⎽	Serial	Parallel
⎽⎽⎽⎽	⎺⎺⎺⎺	⎽⎽|⎺⎺	⎺⎺|⎽⎽	⎽⎽⎽⎽	⎺⎺⎺⎺	⎽⎽|⎺⎺	⎺⎺|⎽⎽	Serial	Serial
⎽⎽⎽⎽	⎺⎺⎺⎺	⎽⎽|⎺⎺	⎺⎺|⎽⎽	⎽⎽|⎺⎺	⎺⎺|⎽⎽	⎽⎽⎽⎽	⎺⎺⎺⎺	Parallel	Serial

At each positive transition, one of the input sets of the corresponding shift register is transferred into its internal nodes by overwriting the first latch stages of the D flip-flops. The serial or parallel loading becomes complete at phase Φ_4, when the second latches of the flip-flops are forced to follow the internal contents of the shift register. The last D flip-flops in two chains output the decisions made by the detector. The final decision is the one which corresponds to the optimum path. However, If the path memory is deep enough, two survivor sequences will merge at some point and both of the shift registers output identical decisions. This implies that, one of the outputs can be arbitrarily selected as the final decision of the decoder, with negligible degradation.

4.5.6. SR and T Flip-Flops

In section 4.3, it was stated that an SR flip-flop is required to switch the DC-offset signal between the upper and lower branches of the decoder. As shown in figure 4.14, two cross-coupled NAND gates are employed to function as an SR flip-flop with $\bar{S}$ and $\bar{R}$ inputs connected to $\overline{A_1}$ and $\overline{A_0}$, respectively. This results in setting the flip-flop by the upper comparator and resetting it by the lower comparator, as expected.

A T flip-flop, also shown in figure 4.14, generates the toggling signal to control the input S/Hs. This flip-flop can be constructed from the D latch, depicted in figure 4.18, by inverting its output and feeding it back to both of the serial and parallel inputs. The serial and parallel loading controls can then be utilized to implement the required OR function, also shown in figure 4.14, without any need to further circuitry. The final signals used to toggle the S/Hs, are derived by use of two AND gates. Fast AND gates can be implemented by adding only two transistors to the switch shown in figure 4.15. Figure 4.20 illustrates the final circuit which generates the toggling signals.

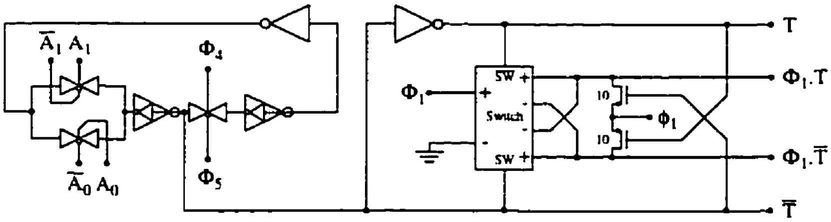


Figure 4.20: Circuit implementation of the T flip-flop and required gates shown in figure 4.14.

4.5.7. Clock Generator

The different clock phases were obtained by the clock generator circuit shown in figure 4.21. This circuit accepts a single-phase clock at its input and generates the appropriate phases Φ_1 to Φ_5 addressed in figures 4.14, 4.19, and 4.20.

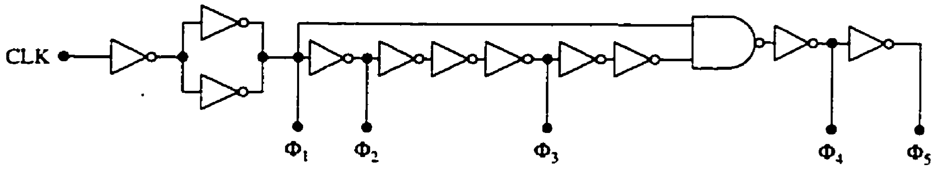


Figure 4.21: The clock generator circuit.

In the above circuit, the required delays are obtained through the use of inverters. Figure 4.22 illustrates a sample timing diagram.

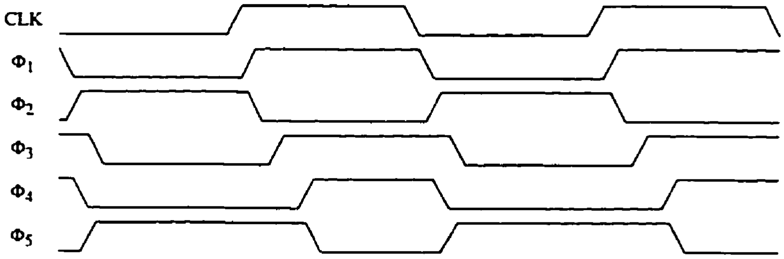


Figure 4.22: Timing diagram of the clock generator circuit.

4.5.8. Biasing Circuit

The analog circuits described so far are biased by an adjustable current generated off chip. On chip, this current is mirrored to generate the required biasing currents by NPN transistors. NPN transistors are preferred to NMOS devices to achieve better matching. However, to bias the source followers in figure 4.16 PMOS transistors are used due to non-availability of PNP transistors. These devices are employed in a cascode configuration with an improved swing capability [60]. Figure 4.23 illustrates the biasing circuitry. Note that a single input current generates all the biasing voltages for the NPN and cascode PMOS current sources. This current can be provided by either an off-chip current source or an off-chip resistor. The nominal bias current is considered to be $100\mu A$.

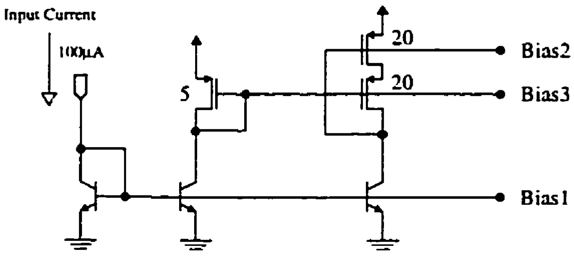


Figure 4.23: The biasing circuit.

4.5.9. Interleaving and De-interleaving

Time interleaving at the input was accomplished by applying the class-IV signal to both of the dicode decoders and using complementary phases for the second dicode. The complementary phases were obtained by a clock generator similar to figure 4.21 with the input inverter replaced with an RC low-pass circuit. The RC time-constant was chosen to accommodate the delay of the eliminated inverter to a first-order approximation.

De-interleaving the outputs was done by two 2-to-1 multiplexers. Each multiplexer accepts two corresponding outputs from each one of the path memories and combines them into one bit stream. A single address line, externally available, controls the multiplexers. If this line is clocked by the master clock of the decoder, two outputs of the path memories are de-interleaved, resulting in a full-rate class-IV decoded output with a decoding delay of 24 bits. However, by connecting the address line to either low or high, each individual dicode outputs its own decoded signal, which runs at half the rate with an effective decoding delay of 12. This capability was extensively used in testing the decoder in practice.

Figure 4.24 depicts the output multiplexers constructed from transmission gates and controlled by a common single-bit address line.

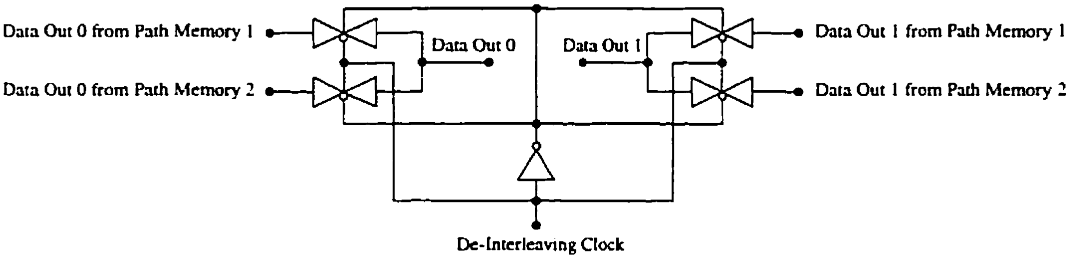

Data Out 0 from Path Memory 1
Data Out 0 from Path Memory 2
Data Out 0
Data Out 1
Data Out 1 from Path Memory 1
Data Out 1 from Path Memory 2
De-Interleaving Clock

Figure 4.24: The de-interleaving multiplexers.

4.5.10. Output Buffers

To be able to drive the pads and external loads, output buffers are necessary. These buffers can be very simple since linearity is not a requirement. Open-collector differential pairs were chosen, due to the high-current capability of bipolar transistors. To provide the required base currents, BiCMOS inverters were used. Open collectors can then be connected to the power supply by arbitrary off-chip pull-up resistors. This gives the opportunity of having either a 50Ω or a 75Ω output impedance, if a coaxial cable connection is required, and at the same time setting the output swing to any desired value by adjusting the biasing currents of the output buffers. As well, the output swing can be increased without increasing the current by choosing larger resistors if low output impedances are not needed. Figure 4.25 shows the circuit.

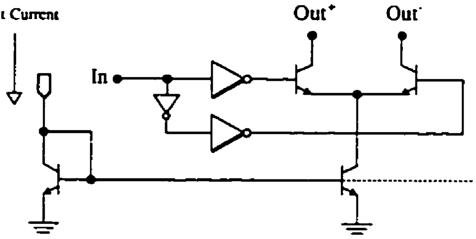


Figure 4.25: Open-collector output buffer.

Note that the biasing current can be adjusted independently from the decoder biasing current through use of separate current mirrors. By adjusting this current in conjunction with the pull-up resistors, any desired voltage swing (for example ECL compatible swing) can be obtained.

Also, it was decided to make the outputs of the comparators in figure 4.14 available for testing purposes. BiCMOS inverters, similar to those used in the output buffer, were utilized to provide enough driving capability to drive the pads. Figure 4.26 depicts the inverter.

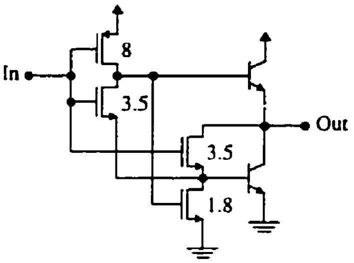


Figure 4.26: BiCMOS inverter capable of driving the output transistors of the buffer circuit as well as the pad.

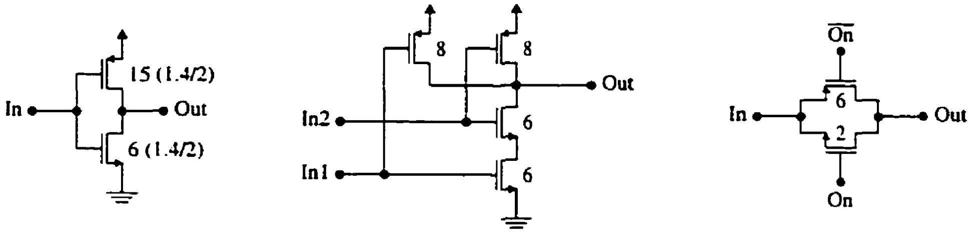
4.5.11. Input Protections

It is common to limit the voltage levels at the gates of MOSFETs which are directly connected to the input pads to protect the gate oxide from damages caused by electrostatic charges [95]. Reverse-biased diodes, formed from bipolar transistors, were employed to clamp the input voltages to one-diode drop below ground and above supply voltage. Small series resistors were also used to limit the currents through the diodes.

Also, small resistors ($150 - 200\Omega$) were inserted in series with input ports which connect the off-chip sources to the front-end circuits. These include the differential input signal and the DC reference signal. On-chip current-limiting series resistors are included in figure 4.14.

4.5.12. CMOS Gates

CMOS inverters, NAND gates, and transmission gates are the only basic digital building blocks used in the IC implementation of the decoder. Circuit schematics of these gates are illustrated in figure 4.27. Transistor sizes of the small inverters used in the feedback paths of the D flip-flop in figure 4.18 are given in brackets.



a. Inverter

b. NAND Gate

c. Transmission Gate

Figure 4.27: CMOS gates used in the decoder. The values inside brackets show the transistor sizes of the small feedback inverters in figure 4.18.

4.6. Experimental Results

This section contains the experimental results obtained from two decoders. The adaptive-threshold decoder was first prototyped using off-the-shelf components. No attempt was made to interleave two of these decoders, rather, a single dicode was built to illustrate the validity of the approach and its robustness against impairments present in practice. In the next phase, the input-interleaved decoder was chosen for an IC implementation. The chip is a full class-IV decoder containing two time-interleaved decoders and the interleaving and de-interleaving circuitry.

4.6.1. Discrete-Prototype

A discrete prototype was designed and built based on the block diagrams shown in figures 4.3 and 4.4. Figure 4.28 illustrates the measured BER performance of the decoder.

The measured BER shows good agreement with the Viterbi bound (also shown in the figure) and confirms the validity of the proposed approach in implementing the decoder in the analog domain. It is expected that the slight deviation seen is mostly related to the different sources of noise, which are present in a discrete prototype, particularly since the circuit is not differential. This degradation was greatly reduced in the IC implementation. The length of the path memory in the discrete prototype was chosen to be 16, which is deep enough not to affect the performance in the SNR range of measurement.

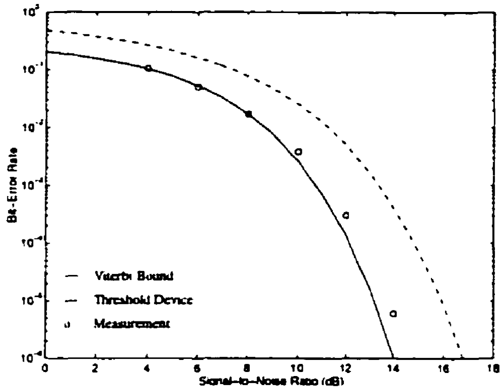


Figure 4.28: Measured BER performance of the discrete-prototype adaptive-threshold decoder.

4.6.2. Integrated Circuit

The decoder described in section 4.5 was fabricated and its BER performance was measured as a function of the input SNR. The class-IV decoder was tested at up to $100MS/s$. However, since each individual dicode was also tested at this speed, the class-IV decoder should be capable of operating at $200MS/s$. Direct experiments at this speed were not possible due to the test equipment limitations which limited the rate of the PR signal to a maximum of $100MS/s$. The BER performance of the class-IV decoder was very similar to that of each individual dicode, as expected. Figure 4.29 depicts the measurement results at two different speeds. BER was measured by counting the number of errors in a fixed period of time. Due to the high-speed operation of the circuit, thousands of errors could be counted in only few minutes even at the lowest BER. Such a large error count implies a high degree of confidence in the BER measurement results.

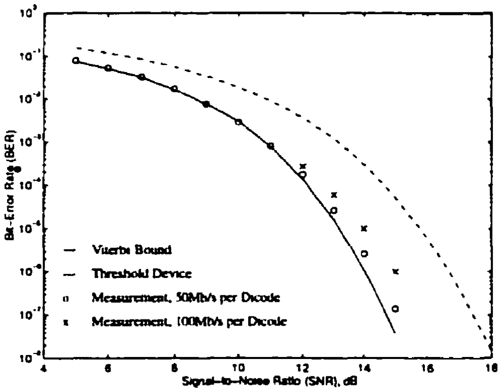


Figure 4.29: Measured BER performance of the integrated-circuit input-interleaved decoder.

The results follow the Viterbi bound, with some expected degradation at high SNR. From chapter 3, recall that not all of this degradation is specific to the present analog realization. Also, it is believed that a part of the degradation at high speeds is due to the input test signal which could not be generated as reliably as it could be generated at low speeds. Figure 4.29 shows that at an effective rate of $200Mb/s$ and at a BER of 10^{-6}, a coding gain of $1.7dB$ is achievable out of its $2.7dB$ upper bound. This increases to $2.4dB$ at $100Mb/s$.

Figure 4.30 illustrates the experimental setup. In this setup, a pseudo-random generator generates a binary sequence which is later encoded by the PR encoder. A controlled amount of white Gaussian noise is then added to the PR signal. The power of the generated noise could be accurately controlled in steps of $0.1dB$ and the amplitude of the PR signal could be precisely adjusted. These capabilities allowed a fine control on the input SNR. The output of the chip, after level adjustment, is compared against the original sequence and the number of mismatches are counted by an error counter. Before comparing, a delay equal to the total decoding delay of the decoder was introduced to the generated sequence to align the input and output signals in time. This delay was introduced by a chain of D flip-flops. To minimize the effect of timing errors on the performance, a global clock was used to drive both the encoder and the decoder. However, a relative phase adjustment on the clock signals is necessary to provide the required static phase difference between the encoder and decoder clock signals. In this setup, the phase of the decoder clock was manually adjusted for a minimum BER. All the off-chip logic was implemented by ECL devices.

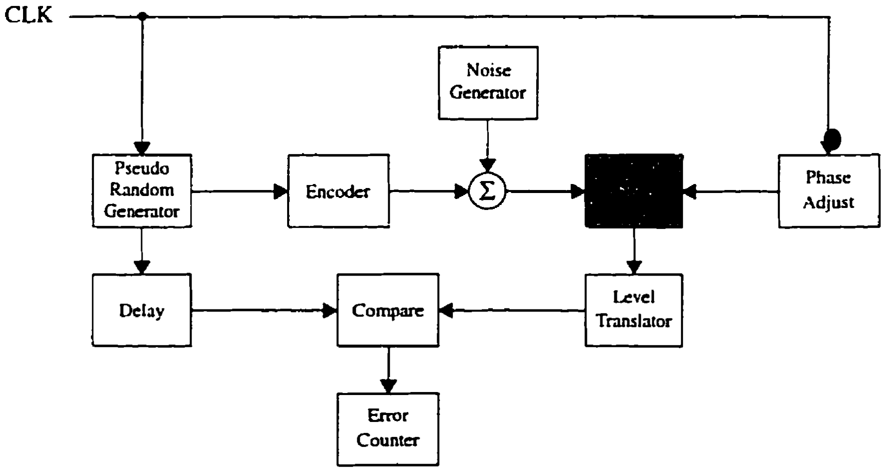


Figure 4.30: The experimental setup used to measure the BER of the decoder. The decoder is shown as the device under test (DUT).

In the present implementation, the path memory is truncated to a length of 12 bits. The excess BER due to truncating the path memory is not negligible compared to the decoder BER at the high end of the SNR range of measurement. Tracking back a local-optimum sequence extends this SNR range toward lower values. Thus, any direct measurement would have been affected by the excess

BER. To highlight the extremely low BER performance of the decoder, a different measurement technique was applied at high SNRs. The local track back was performed on both states of the dicode decoder. From the resulting local optimum sequences two bits were detected. The detected bits were compared against the corresponding original bit and an error was flagged only when both of the detected bits were not correct. Having two opposite detected bits is an indication that two survivor sequences have not yet merged. These sequences could have merged if a deeper path memory had been used. Note that even if these sequences had merged, still an error could have occurred with a probability equal to the BER of the decoder. Ignoring these errors results in setting a BER target that is, in general, below the BER of the Viterbi decoder. Any measurement now, should be compared to this fictitious target. However, simulations indicate that in the SNR and BER ranges of figure 4.29, and for the memory length of 12 bits, this target is hardly distinguishable from the Viterbi bound and the above technique can be used for low-BER measurements[1].

Figure 4.31 shows a typical pseudo-random binary signal (uncoded) and the decoded output at $100 Mb/s$ for one dicode. The decoded output shows a delay slightly more than the expected 13 bits (12 bits due to the length of the path memory plus one bit processing time). This extra delay is because of the latency introduced by the propagation time and was not observed at low speeds.

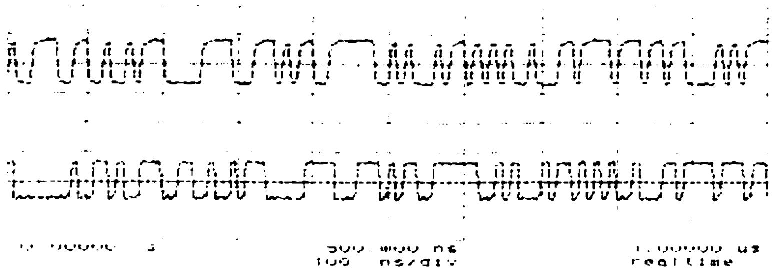


Figure 4.31: A typical uncoded signal at $100 Mb/s$ (upper trace) and its decoded output (lower trace).

Table 4.3 summarizes the specifications of the chip as well as some of the experimental results.

4.7. Layout and Related Issues

Figure 4.32 shows the layout of the chip, fabricated in a $0.8 \mu m$ BiCMOS process[2]. It contains

[1] Alternatively, straight measurements could have been compared to the performance of a Viterbi decoder with a truncated path memory. However, the above approach seems to better illustrate the functionality of the decoder at very low BER.

[2] The Northern Telecom (NT) BATMOS process, available through Canadian Microelectronics Corporation (CMC).

Table 4.3: A summary of the chip specifications and test results.

Chip	Analog Viterbi Decoder
Coding Scheme	Class-IV Partial Response
Process	$0.8\mu m$ BiCMOS
Data Rate	$200MS/s$
Power Consumption	$30mW$ at $200MS/s$, $26mW$ at $100MS/s$
Power Supply	$3.3V$
Size	$0.5mm^2$
Coding Gain ($BER = 10^{-6}$)	$1.7dB$ at $200MS/s$, $2.4dB$ at $100MS/s$

two dicode decoders operating in a time-interleaved fashion. Each dicode consists of an analog core, a digital path memory, and a control-signal generator. Note that the analog signal processor is very small (smaller than the path memory), demonstrating the efficiency of the proposed analog realization.

The analog and digital parts are separated by an N-well connected to the power supply via deep contacts. This barrier reduces digital noise, generated by the CMOS digital circuitry, from being injected into the analog part of the decoder. Also, separate supply and ground lines are used for analog and digital circuits. Isolating analog circuitry from its digital surroundings is crucial in mixed-mode processor designs [96]. Besides the common-mode rejection property of the fully-differential implementation of the structure, in transferring digital signals from one point of the layout to the other, complementary signals are also transferred. This approach cancels any feedthrough from digital lines to analog circuitry to a first-order approximation. Furthermore, an attempt was made to maintain symmetry to exploit the differential nature of the circuits in the layout level. To further reduce the noise injected to the analog signals, double-poly S/H capacitors and critical poly resistors were put inside reverse-biased N-wells. Also, poly capacitors were employed to bypass the power and bias lines in unused areas throughout the layout. In addition, MOS capacitors were used to bypass the power lines by putting MOS devices underneath these lines. This increases the bypass capacitance without occupying additional space.

4.8. Summary

Analog Viterbi decoders result in significant savings in power and size, while operating at higher speeds compared to their digital counterparts. These advantages are mostly achieved by

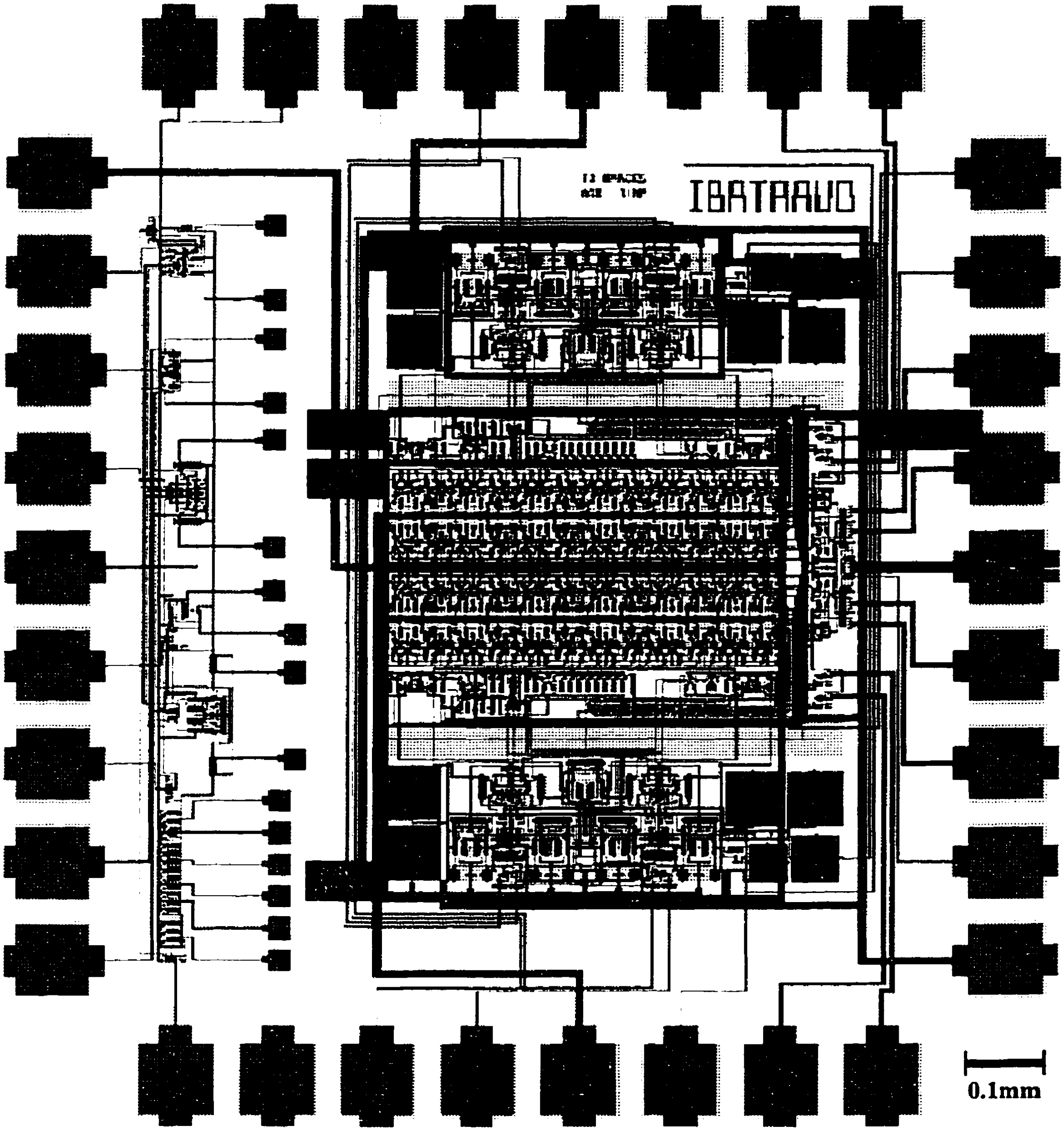


Figure 4.32: Layout of the class-IV analog Viterbi decoder, fabricated in the $0.8\mu m$ NT BiCMOS process (BATMOS).

eliminating the analog-to-digital converter. The problem of realizing a class-IV partial-response Viterbi decoder in the analog domain was addressed in this chapter. It was shown that such a decoder can be efficiently realized using a few simple building blocks. This goal was achieved by examining the difference-metric algorithm from an analog implementation point-of-view. The outcome was a new structure derived by exploiting the similarity between a traditional symbol-by-symbol threshold device and the difference-metric Viterbi detector. The analog decoder, named adaptive-threshold decoder was shown to be structure-wise faster than other reported decoders.

Since the speed of operation was our main concern, an attempt was made to further increase the speed before the analog decoder undergoes an integrated circuit fabrication process. However, to verify the validity of the proposed approach and its robustness in practice, a discrete circuit was prototyped. Experimental results were encouraging and an integrated version was constructed to demonstrate its high-speed operation. A substantial increase in the speed of the analog structure was made possible by developing a version of the difference-metric algorithm which eliminates the need of sampling and holding any intermediate signal. The new algorithm led to an input interleaved structure for which an integrated circuit was designed. The design was fabricated in a $0.8\mu m$ BiCMOS process, was tested, and achieved a speed of $100MS/s$ per dicode, corresponding to $200MS/s$ for the class-IV operation. Direct experiments on the class-IV decoder were limited to $100MS/s$ due to the test equipment limitations. The power consumption of the chip was $30mW$ from a $3.3V$ single power supply. The core area is $0.5mm^2$, from which only 25% is dedicated to the analog circuitry.

Also, it was shown that the proposed structures are robust against imperfections that an analog design faces in practice. The proven advantages make the analog detector an extremely attractive alternative in commercial products for high-speed, low-power and small-size applications such as magnetic disk-drive storage.

Chapter 5

Analog Viterbi Decoding: A General Approach

In the previous chapter, it was mentioned that analog Viterbi decoders have the potential of being small, fast, and power efficient. An analog class-IV partial-response Viterbi decoder was addressed in detail, experimental results from an IC implementation were presented, and it was shown that the decoder indeed achieves the aforementioned advantages. However, the analog class-IV decoder was made possible because of two simplifications: The simplified (difference-metric or input-interleaved) algorithm and time interleaving. These sort of simplifications can not be done in general, nevertheless, the benefits gained through analog implementations can still be maintained. To extend the idea of analog realizations to general Viterbi decoders, a new approach should be taken. It is predicted that very soon this approach will find its first application in the disk-drive industry, where an ever-increasing demand in the storage density calls for more sophisticated signal processing, and consequently sequence detection techniques with increased number of states. The Extended PRS Viterbi decoders are good examples of sequence detectors for which the aforementioned simplifications no-longer exist. Simplifications are possible, however, at the expense of a penalty in the performance [97].

Implementing Viterbi detectors in the analog domain has already been suggested [61-71]. The technique proposed in [63] is also applicable to decoding trellis codes [98]. However, in both cases the realization is practically limited to a hard decision detection. An attempt to extend this technique to a soft decision detection leads to a large number of diodes connected in series and requires an unreasonably high-voltage supply. Furthermore, it does not eliminate the need for an A/D, which is a basic motivation for an analog implementation.

The present chapter describes the attempts made towards exploiting the ability of simple analog circuitry in performing the required functions in the Viterbi decoder. Simplicity was a basic requirement since speed, size, and power consumption were the major concerns. The chapter starts with a general overview of the approach, followed by details of its circuit-level implementation. The IC implementation of two decoders will be explained and experimental results for one decoder will be given as a proof-of-concept. The decoders were implemented on a common silicon core and the chip was fabricated in a $0.8\mu m$ BiCMOS process. To save design time, digital path memories were not included on chip. As a result, experiments were carried up to $80Mb/s$. However, simulations indicate that speeds on the order of a few hundred megahertz can be achieved. The power consumption of the decoder is estimated to be about $15mW/state$ drawn from a $5V$ single power supply.

It should be pointed out here that the implementation approach is general and can be used in error-correction coding systems (including convolutional codes), M-ary communications, and irregular trellises. For example, application of the technique to a quaternary dicode Viterbi decoder appears to be a likely candidate. The quaternary dicode scheme will be considered in detail in the next chapter, where the issue of an analog realization for a near-to-optimum decoder for a quaternary class-IV system is addressed.

The technique can be also used to realize a programmable Viterbi decoder. In the magnetic-recording application, the decoder can be programmed to detect signals having more spectral similarities to the spectrum of the actual read signal. As a result, less equalization would be required. The overall performance of the read channel improves, as noise enhancement of the equalizer is minimized. As well, the decoder can be incorporated in a feedback loop resulting in an adaptive-sequence detector. This is similar to having an adaptive equalizer followed by a fixed sequence detector, however, with less noise enhancement. Although the idea of adaptive equalization does not seem to have met the industry requirements yet (most disk-drive companies presently make use of programmable equalizers), the idea is likely to be followed in future.

Finally, the implementation approach is extremely suitable for automated design of analog Viterbi decoders. The analog automation tool can extend down to a layout level and enables the analog design to be easily transferred from one technology to another [99].

5.1. General Overview

In an algorithmic realization of MLSD, the metric assigned to each state of the trellis is updated using the previous state metrics and the branch metrics. Each branch in the trellis corresponds to a transition between two states with its metric representing the distance of the received symbol from the noiseless signal resulting from that transition. In the case of additive white Gaussian noise, the error (distance) criterion is based on minimizing the squared Euclidean distance. The update mechanism takes place such that the accumulated error of the estimated sequence for each state is minimized. In a trellis with $L = M^N$ states, there are M transitions initiating from or ending to each state, where M is the size of the alphabet and N is the memory size of the encoder[1]. As an example, figure 5.1 shows the trellis diagram for $N = 2$ and $M = 2$.

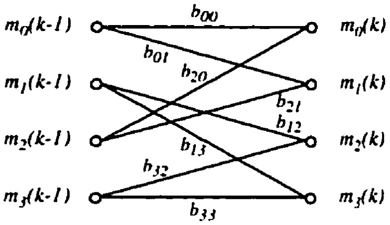


Figure 5.1: A typical trellis diagram.

From chapter 3, recall that the ACS performed by a Viterbi decoder can be described as[2]

$$m_i(k) = \max_j \{m_j(k-1) - b_{ji}(k)\} \qquad , \begin{matrix} i = 0, 1, \ldots, L-1 \\ j = 0, 1, \ldots, L-1 \end{matrix} \tag{5.1}$$

where $m_i(k)$ is the metric of the i'th state at time step k and $b_{ji}(k)$ is the metric of the branch connecting state j at time $k-1$ to state i at time k. Also, recall that the branch metrics can be expressed as linear combinations of the received sample and some constant values.

[1] Although only regular trellises fall in this category, the idea can be easily extended to irregular trellises as well.

[2] Note that here, equivalently, the *min* function has been converted to a *max* function with an inversion in the sign of the branch metrics.

The above mathematical operation can be realized by the simple circuit depicted in figure 5.2. In this circuit, there is a one-to-one correspondence between diode branches and the branches of the trellis diagram. In other words, each branch of the trellis is represented by a diode branch. The threshold voltage of each diode is equal to the metric of its corresponding branch. Note that for those values of i and j for which there exists no transition between the states, the diode branches should simply be omitted. The previous state metrics are translated to voltage sources, driving the bus lines. The new values of state metrics appear as voltages at nodes 0 to $L-1$.

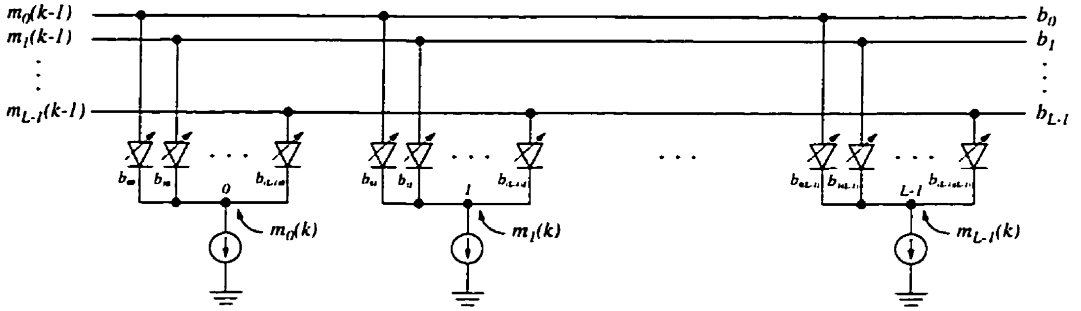


Figure 5.2: The diode network used to realize the ACS function.

Assuming sharp $i-v$ characteristics for the diodes, only one diode turns on in each set and conducts current. For junction diodes and the accuracy needed in a typical Viterbi decoder, this assumption seems reasonable. We shall return to this point when circuit implementation issues are addressed.

The threshold-programmable diode is shown in figure 5.3.a. This diode is composed of a diode-connected BJT and a voltage source placed in the loop. Note that the negative swing of the voltage source should be limited to prevent saturation. Also, note that with the aforementioned assumption, the v_{BE} drops of the transistors will have negligible effect on the operation of the circuit. The floating voltage sources can be implemented by resistors fed by current sources as illustrated in figure 5.3.b. The current sources are proportional to the branch metrics and can be generated by combining the input sample with some DC values.

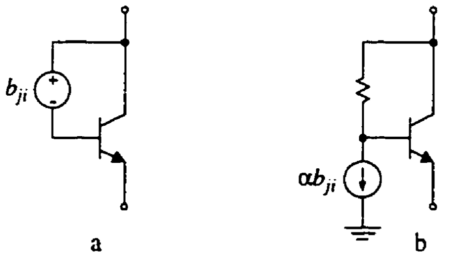


Figure 5.3: Threshold-programmable diode.

To avoid any destructive effects on the stored state metrics, two-stage S/Hs should be employed when feeding the new metrics back. While one stage is in track mode, sampling the new value of its corresponding state metric, the other is in the hold mode, holding the previous value across the bus line. Although master-slave S/Hs can also be used, ping-pong S/Hs are preferred, as they provide the potential of doubling the speed. This is particularly important as the speed of operation is mainly limited by S/Hs. Figure 5.4 illustrates the idea. Note that the output buffer should provide a very small output impedance as it will be used to drive one of the bus lines in figure 5.2. Also, note that one such circuit is required per state.

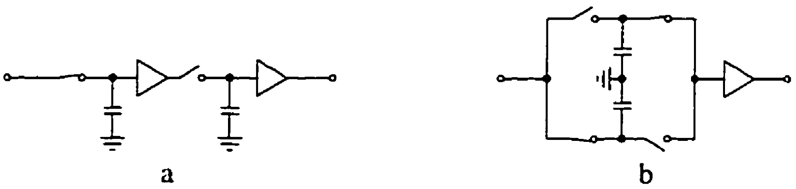


Figure 5.4: Two-stage S/H, a. master-slave, b. ping-pong.

As seen from (5.1), unbounded growth of the state metrics is an inherent problem of the Viterbi algorithm. To reduce the required dynamic range in internal calculations, several methods for normalizing the state metrics have been considered in a digital decoder [100]. Taking an average over the state metrics and subtracting it in each iteration, minimizes the required dynamic range. This optimum solution, not suitable for a digital realization, works well in our analog approach by employing a fast common-mode feedback (CMFB) circuitry. The CMFB operates on the new state metrics by continuously monitoring these metrics and trying to maintain a constant common-mode voltage for them.

The comparison results are the currents through the diodes. L conducting diodes, each from one group in figure 5.2, which carry the currents should be detected. To sense the branch currents, either mirror transistors can be used or the collector currents of the diodes can be directed rather than being drawn from the buses. Figure 5.5 depicts a programmable diode with current sensing.

In the mirror-transistor approach, the collector-emitter voltage variations of the different diodes are typically not large as in the other approach. This results in $v_{BE}-i_C$ characteristics less affected by the output circuits. However, this effect is usually negligible and the latter approach is almost always preferred.

The sensed currents should be used to update the contents of the path memory. Memory management is beyond the scope of this chapter, however, the register-exchanged method, described in

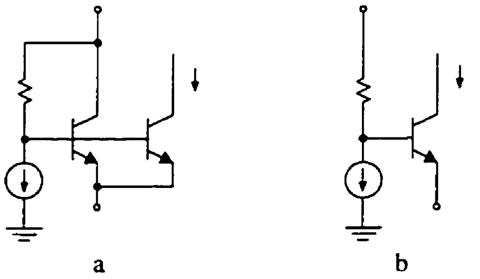


Figure 5.5: Sensing the current of a programmable diode by a. employing a mirror transistor and b. directing the collector current.

chapter 4, can be retained here as well. To guarantee that suitable digital levels will be developed from the sensed currents, the use of comparators (either current or voltage) is helpful. However, this approach is only suggested for binary signaling schemes, since, otherwise these comparators should compare more than two signals. Alternatively, in a binary scheme, the voltages at the two bases of the diode transistors can be compared to generate the required digital data, eliminating the need for sensing the currents.

Finally, it should be mentioned that each set of diodes in the diode network described here can be viewed as a generalized differential cell which compares the base voltages and directs the maximum input (with a v_{BE} drop) to the output taken from the common-emitter terminal. This equivalent approach has been taken in [71].

5.2. Circuit Realization

Having introduced a general overview of the analog implementation approach, more details of the functional blocks will be presented and circuit-level issues will be addressed in this section.

5.2.1. Ping-Pong S/H

The ping-pong S/H, shown in figure 5.4.b, consists of two basic S/Hs operating on a common input signal. A commutator directs the previous sample value, held by one of the S/Hs, to the output while the other S/H samples the new value of the input. The output buffer should exhibit a very high input and a very low output impedance. In a BiCMOS process, a combination of a MOS input-stage and a BJT output-stage is the best choice. Source follower and emitter follower stages were employed in our design. To keep the voltage swings within reasonable values, a PMOS source follower was chosen to alternate the level shifts introduced by two stages. As will be described shortly, the CMFB mechanism applies on the level shifts of the source-followers by adjusting their biasing currents. It should be pointed out here that instead of one source follower at the output, two identical stages were used at the inputs of the commutator. This prevents charge transfer from the holding capacitors to the input capacitance of the MOS transistor. Figure 5.6 depicts the S/H circuit. In this figure, Φ_1 and Φ_2 are two non-overlapping phases obtained from a divide-by-two version of the master clock. The clock-generator circuit will be discussed later in this section.

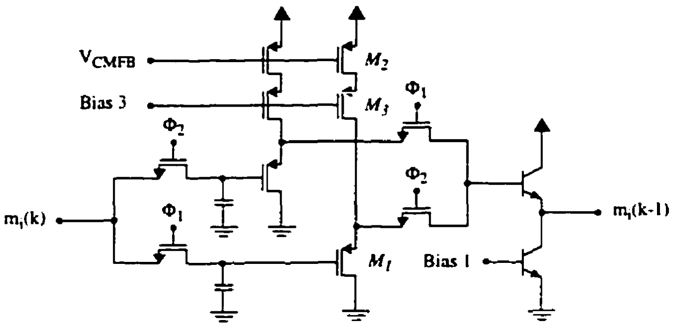


Figure 5.6: The ping-pong S/H circuit. Input and output are the new and previous state metrics of state i.

5.2.2. CMFB Circuit

The need for a CMFB control and its efficiency in minimizing the required dynamic range of the circuits was explained earlier. In our implementation, shown in figure 5.7, this mechanism is applied by continuously monitoring the common-mode (CM) signal of all of the state metrics and adjusting a value added to all of them to keep the CM signal equal to a CM reference voltage. This is accomplished by changing the level shifts of the source-follower buffers in the S/H circuits through adjusting their biasing currents in a continuous-time feedback loop.

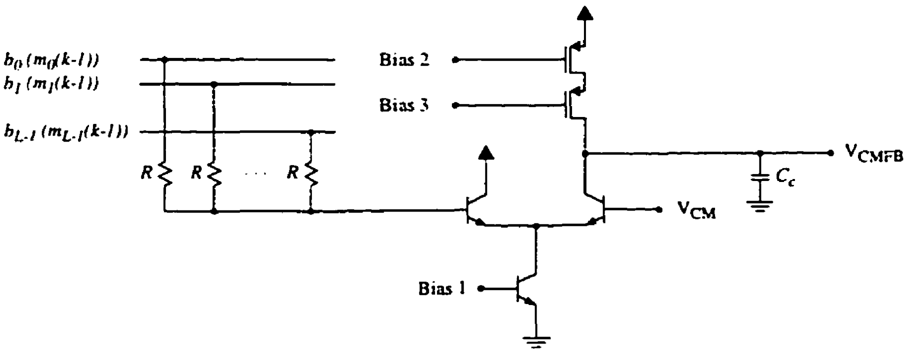


Figure 5.7: The CMFB circuit.

A simple small-signal analysis of the CMFB circuit is possible by connecting figure 5.7 to L S/H circuits of figure 5.6 in a loop and considering the single-input equivalent circuit shown in figure 5.8. Note that at each time, from $2L$ source followers, L fall outside the loop and their effects can be represented by their total gate capacitances. Also, note that in this circuit the sizes of transistors labeled "xL" are L times the sizes of their corresponding transistors in figure 5.6.

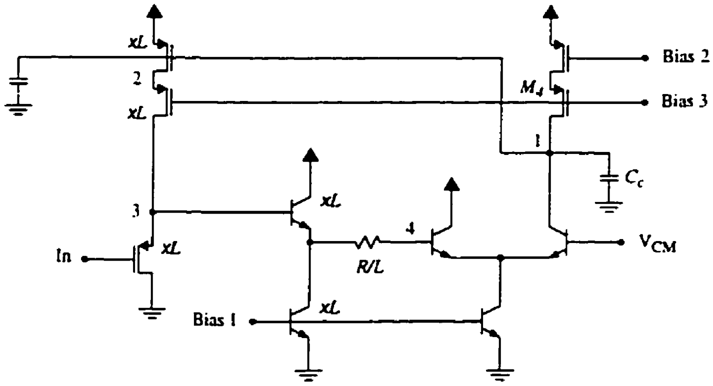


Figure 5.8: The equivalent circuit for analyzing the CMFB in presence of a change in the CM signal.

The small-signal dynamics of the above circuit can be better understood by breaking the loop and deriving a second-order approximation to the loop gain, G. The only high-impedance node of the circuit, node 1, gives rise to the dominant pole. Two other poles, associated with nodes 2 and 3, are predicted to be of equal order and their effect will be taken into account by introducing an equivalent pole, as described in [60][1]. The pole at node 4 is not expected to have a significant contribution to the loop gain in the frequency range of interest and its effect will be neglected for simplification. This is a reasonable presumption since the associated time constant is usually smaller than the time constants at nodes 2 and 3.

The dominant and two higher frequency poles are located at

$$\omega_{p_1} \approx 1/(r_o C_1) \tag{5.2}$$

$$\omega_{p_2} \approx g_{m_3}/C_2 \tag{5.3}$$

$$\omega_{p_3} \approx g_{m_1}/C_3 \tag{5.4}$$

where

$$C_1 \approx C_c + 2L(C_{g_2} + C_{gs_2} + C_{gd_2}(1 + g_{m_2}/g_{m_3})) \tag{5.5}$$

$$C_2 \approx C_{db_2} + C_{gd_2}(1 + g_{m_3}/g_{m_2}) + C_{gs_3} + C_{sb_3} \tag{5.6}$$

$$C_3 \approx C_{gs_1} + C_{sb_1} + C_{db_3} \tag{5.7}$$

[1] This approach is equivalent to approximating the third-order characteristic equation with a second order equation by neglecting the s^3 term. This is a valid approximation for frequencies well below $\omega_{p_2} + \omega_{p_3}$.

are the total capacitances from nodes 1, 2, and 3 to ground and r_o is the output impedance of the differential amplifier. In these expressions, indexes 1 to 3 refer to transistors M_1 to M_3 in figure 5.6[1].

The DC gain around the loop is

$$G_0 \approx g_m r_o g_{m_2} / (2g_{m_1}) \tag{5.8}$$

where g_m is the transconductance of the BJT in the differential amplifier.

Solving for $1 + G = 0$, results in the following characteristic equation

$$S^2 + \omega_{p_{eq}} S + G_0 \omega_{p_1} \omega_{p_{eq}} = 0 \tag{5.9}$$

where $1/\omega_{p_{eq}} = 1/\omega_{p_2} + 1/\omega_{p_3}$, $G_0 \gg 1$, and $\omega_{p_1} \ll \omega_{p_{eq}}$ have been assumed.

Equation (5.9) results in two poles which can be expressed in terms of their Q factor and pole frequency. After some manipulations, the result is

$$Q = \sqrt{\frac{g_m g_{m_2}}{2g_{m_1}^2} \frac{C_3 + (g_{m_1}/g_{m_3}) C_2}{C_1}} \tag{5.10}$$

$$\omega_0 = \sqrt{\frac{g_m g_{m_2}}{2C_1 (C_3 + (g_{m_1}/g_{m_3}) C_2)}} \tag{5.11}$$

Also, solving $|G(j\omega_t)| = 1$ for ω_t and plugging the result in $PM = 180° + \angle G(j\omega_t)$, yields an expression for the phase margin, PM. The result, expressed in terms of the Q factor, is

$$PM \approx tan^{-1}\left(\frac{1}{\sqrt{\sqrt{\frac{1}{4} + Q^4} - \frac{1}{2}}}\right) \tag{5.12}$$

5.2.3. Branch-Metric Generators

It was mentioned that the branch metrics (error signals) are usually expressed as linear combinations of the input sample and some DC values. As an example, (3.7) gives four metrics for a first order PRS Viterbi decoder. In constructing these combinations, both polarities of their generating components are usually required and the combinations are needed in the form of current signals. In

[1] Note that transistors M_1 and M_3 are subject to body effect. This can be taken into account by including the effect on the transconductances.

an efficient realization, differential transconductors might be used to convert the input voltages to differential currents. The resulting signals, with appropriate polarities, can be combined by simply interconnecting the outputs of the transconductors. Since high linearity is usually not a requirement, differential pairs with degeneration are suggested because of their simplicity.

The error currents program the thresholds of the diodes shown in figure 5.2 by developing proportional voltages across the resistors depicted in figures 5.3 and 5.5. To reduce the unnecessary DC voltage drops across these resistors (and even eliminate them altogether), constant current sources can be added to the combinations. These sources partially (entirely) bypass the biasing currents of the differential cells from the resistors. The technique reduces the operating-voltage requirement of the ACS circuitry and is suggested in low-voltage applications. Figure 5.9 illustrates the idea.

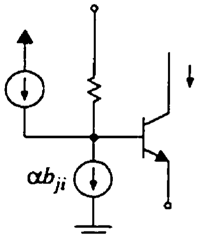


Figure 5.9: Reducing the DC threshold of the diodes by partially (entirely) bypassing the DC components of the error signals.

5.2.4. Clock Generator

A symbol-rate synchronous clock is needed to update the path memory at the end of each iteration. Such a clock is also required if comparators are employed after the ACS circuitry. Also, a divide-by-two version of the clock should be used to control the ping-pong S/Hs. Two phases of the latter clock are required as shown in figure 5.6. Non-overlapping is essential to guarantee prevention of the destructive feedback, explained in section 5.1.

If comparators are used, their outputs should change only after the path memory is updated. Figure 5.10 shows a typical timing diagram in which the path memory should be updated at the falling edges of the clock. Track and Latch are used by the comparators which are assumed to be latched comparators. The non-overlapping phases, Φ_1 and Φ_2 are also shown. These signals toggle the S/Hs after outputs of the comparators are latched.

The circuit depicted in figure 5.11 can be used to generate the required clocks from a single-phase clock. In this circuit, a T flip-flop, formed by feeding the complementary output of a D flip-

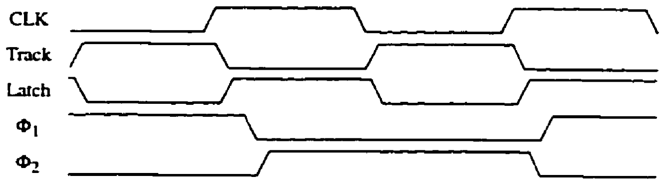


Figure 5.10: A typical timing diagram for the clock generator.

flop back to its input, divides the frequency of the clock by two. The divided-by-two output is then converted to phases Φ_1 and Φ_2 by a a non-overlapping clock generator. To drive the D-flip-flop, two symbol-rate non-overlapping signals and their complementary overlapping signals are required. These signals are generated by another non-overlapping clock generator which works at the symbol rate. The symbol-rate clock signals required by the comparators are obtained from those outputs which satisfy the above timing diagram.

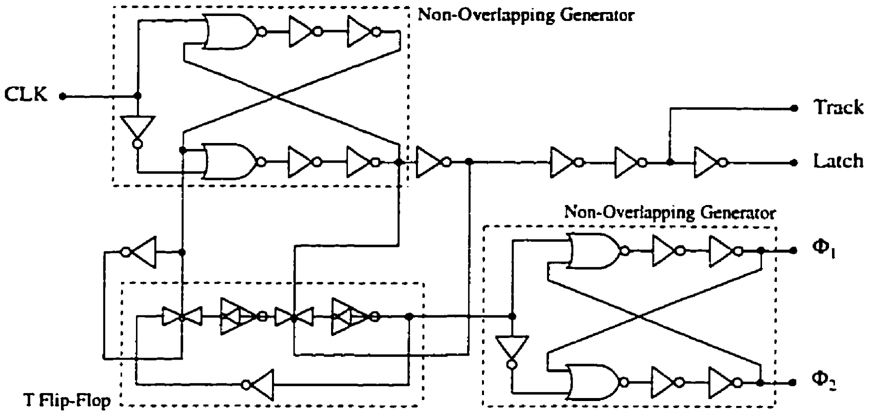


Figure 5.11: The clock generator circuit.

5.3. Design Examples

To investigate the feasibility of the proposed implementation approach in practice, two PRS Viterbi decoders were designed and fabricated on a common silicon core. Due to availability of the experimental setup[1], a dicode decoder was chosen. Although this approach is not efficient in this special case[2], it was fabricated to prove the concept. As well, an extended PRS decoder was designed to show that the approach can be easily extended to decoders with considerably higher number of states. The EPR4 scheme was chosen due to its future application in the read channels of magnetic recording systems. This scheme was described in section 2.5.4.

[1] This setup was already used to test the decoder described in chapter 4.

[2] For a preferred analog implementation of a dicode Viterbi decoder refer to chapter 4.

5.3.1. Binary Dicode: A Proof-of-Concept

Recall from chapter 3 that the trellis diagram of a dicode PRS scheme is a simple butterfly with four branch metrics given by (3.7) for $f_1 = -1$ and $h = 0.5$. This trellis is shown in figure 5.12. The implementation approach presented here results in the diode network also shown in figure 5.12. The threshold voltage of each diode is proportional to its corresponding branch in the trellis diagram. Furthermore, a constant value is added to prevent the threshold voltages from becoming negative.

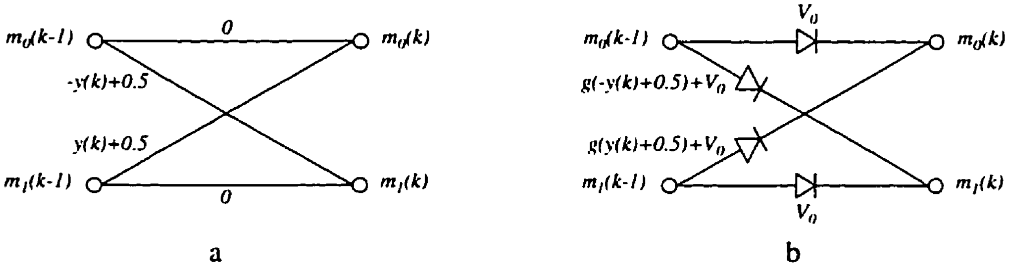


Figure 5.12: A dicode trellis diagram (a), and its corresponding diode network (b).

Applying the implementation method to the above decoder, the circuit depicted in figure 5.13 results. In this decoder, the diode shown in figure 5.3.b is used (i.e. without current sensing). Error signals are obtained by first converting the input signal, $y(t)$, and the DC signals to currents and then combining the currents with appropriate polarities. The biasing currents of the transconductance cells are absorbed by the bypass current sources, to prevent flowing excess DC currents in the resistors. As a result, the threshold voltages of two cross-coupled diodes fluctuate around the voltage generated by the DC components of the error signals plus the v_{BE} drops of the transistors. Two outer diodes have fixed threshold voltages of v_{BE}. The transconductance cells and the bypass current sources connected to these diodes result in a net current of zero and are included to maintain matching.

An intuitive explanation was given earlier that the v_{BE} drops introduced by the "on" diodes will not have a significant effect on the performance of the decoder. To investigate the effect, consider the generic circuit shown in figure 5.14. Assuming the exponential characteristic $i_E = I_S exp(v_{BE}/V_T)$ for the transistors, straight-forward analysis of the circuit yields the following expression for v_e

$$v_e = v_1 - e_1 - V_T ln\left(\frac{I}{I_S}\right) + V_T ln\left(1 + e^{\frac{v_2 - v_1 + e_1 - e_2}{V_T}}\right) \tag{5.13}$$

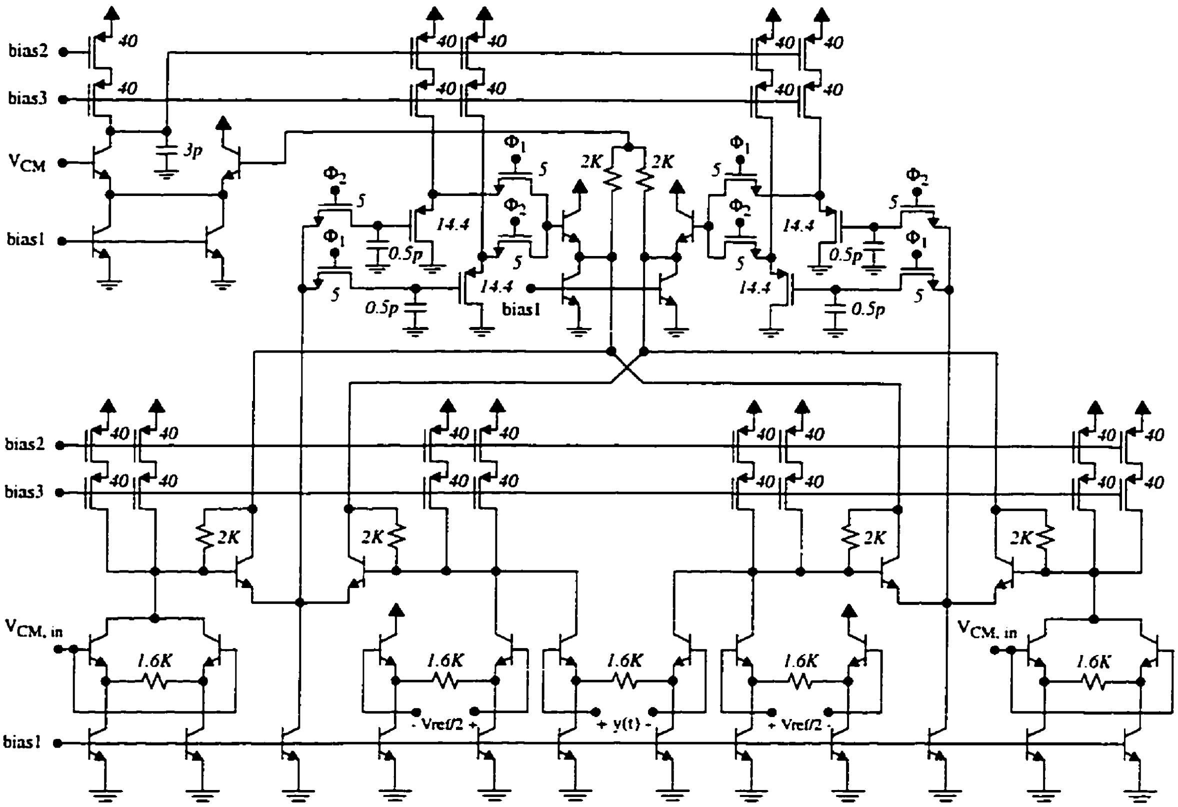


Figure 5.13: The binary dicode decoder.

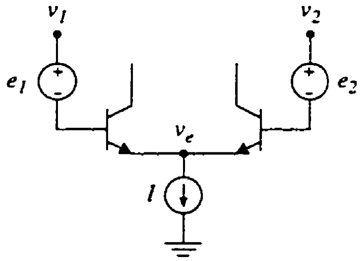


Figure 5.14: The generic circuit for generating the new state metrics.

Equation (5.13) can be applied to two metric generators in the dicode decoder[1] and the results can be adopted by simulations to illustrate the effects of $i - v$ dependency of the diodes on the performance of the decoder. Figure 5.15 depicts the simulation results in which the BER performance is plotted for different branch gains. As shown in figure 5.12, this gain is the proportionality coefficient in translating the branch metrics to threshold voltages of the diodes and is set by the transconductances of the V/I circuits and the resistors in the threshold-programmable diodes.

[1] Note that I_S does not have any effect since the third term is constant and can be discarded in calculations.

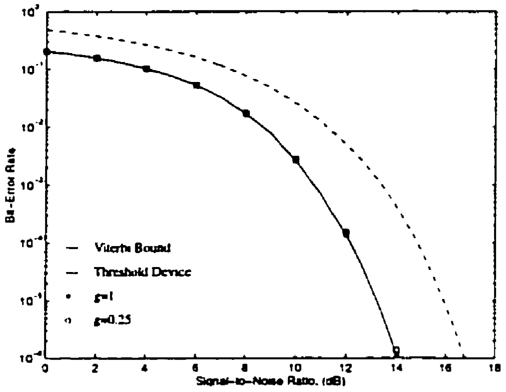


Figure 5.15: Effect of the non-ideal $i - v$ characteristics of the diodes on the performance of the dicode decoder for two branch gains.

This figure shows that performance degradation due to the differences in the v_{BE} of different transistors (which result if the "on" diodes are partially on and do not carry all of the tail currents) is indeed negligible. However, this degradation becomes considerable as the branch gain is reduced. This is expected, since decreasing the gain results in having v_{BE} differences comparable to threshold voltages of the diodes. Figure 5.16 illustrates the BER of the decoder as a function of the branch gain at $SNR = 12dB$.

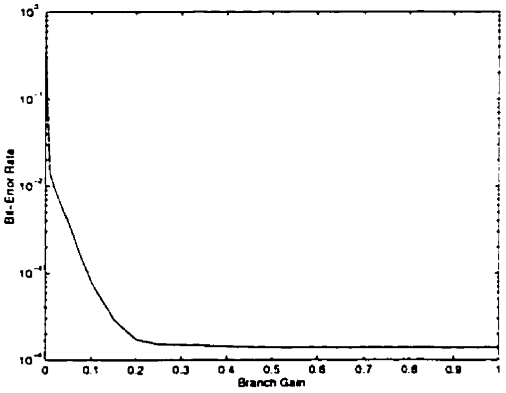


Figure 5.16: BER performance of the dicode decoder as a function of the branch gain at $SNR = 12dB$.

As seen from figure 5.16, for very small gains the performance suddenly drops, however, even for practically-small gains the decoder nearly achieves its optimum performance. A gain of 0.2 develops threshold voltages in the range of -0.1 to 0.3 V, whereas much higher swings can be handled even with the transistors in diode-connected configuration. Had the collector of the transistors been connected to higher potential levels, higher-swing threshold voltages could have been established.

In the two-state dicode decoder, two S/Hs are used to feed the new state metrics back to the buss lines. The CMFB circuit keeps the CM signal of these lines equal to a CM reference voltage. This feedback loop was analyzed in section 5.2.2. A $3pF$ compensation capacitor was used. From (5.10), a Q factor of 1 and a pole frequency of $290MHz$ is anticipated for the closed-loop poles. Consequently, from (5.12), a phase margin equal to 52° should be obtained. Figure 5.17 illustrates the Bode plot of the actual open-loop circuit, obtained by SPICE. It can be seen that a 50° phase margin and more than $30dB$ gain margin has been achieved.

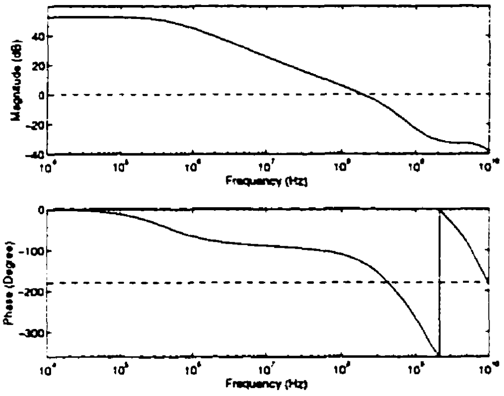


Figure 5.17: Bode plot of the CMFB circuit in the binary dicode decoder obtained by SPICE.

Also, frequency and time domain SPICE simulations were carried on the actual CMFB circuit. Figure 5.18 depicts the results when the output is considered to be V_{CMFB} in figure 5.7. From this figure, a Q factor of 1.15 and a pole frequency equal to $300MHz$ are derived. Note that the fast roll-off in the frequency response, resulting from high frequency poles, could indeed be neglected in the frequency range of our interest (up to ω_0).

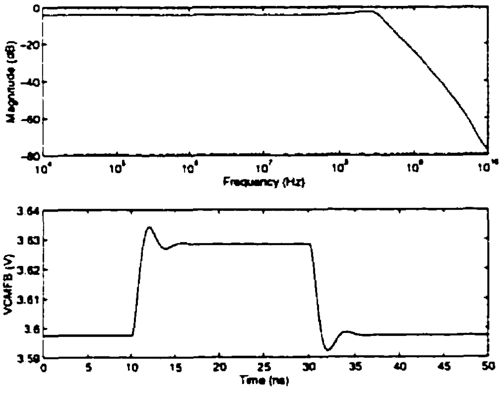


Figure 5.18: Frequency and time domain SPICE simulation results of the actual CMFB circuit.

The comparison results can be taken from the bases of the diode transistors. Two comparators should be employed to compare the signals and output the digital information to update the path memory. Figure 5.19 depicts the comparator circuit. The circuit consists of a preamplifier stage, which amplifies the input in the track mode, and a bipolar latch which develops two complementary outputs, when the comparator is switched to the latch mode.

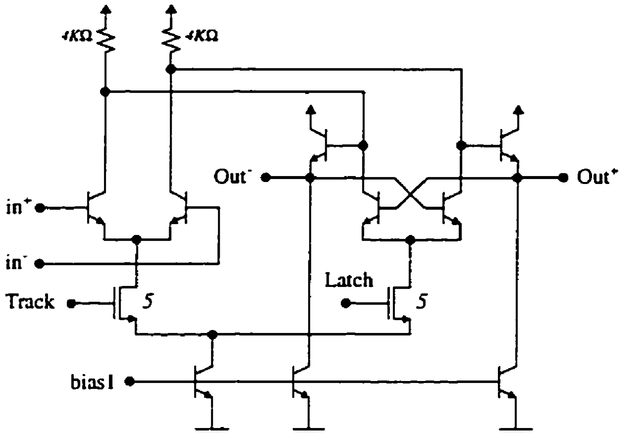


Figure 5.19: The latched-comparator circuit.

5.3.2. Binary EPR4: A Real Application

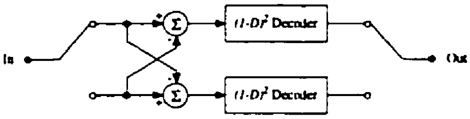
In an EPR4 signaling scheme, the coding polynomial is represented by

$$F(D) = (1-D)(1+D)^2 \tag{5.14}$$

Although the above polynomial can be realized by a time-interleaved structure [7], the parallel branches in this structure will not be independent due to the existence of cross-coupled connections. This implies that in the decoder side a linear filtering should be performed prior to the sequence detection[1]. The linear filter enhances the noise and degrades the performance.

Yet another implementation method for the EPR4 sequence detector is possible by viewing the signaling scheme as two concatenated systems. Consequently, the EPR4 signal can be decoded by two concatenated $1+D$ and $1-D^2$ decoders. However, to achieve the optimum performance, the

[1] In fact, it can be easily shown that time interleaving by two results in the following decoder



which consists of a $1-D$ filter and two independent $(1-D)^2$ detectors. The filter can be viewed as an equalizer which equalizes the received signal to a $(1-D)^2(1+D)^2 = (1-D^2)^2$ encoded signal. Since this polynomial is a function of D^2, the decoder can be realized by time interleaving two independent $(1-D)^2$ decoders.

first decoder should provide the second decoder with enough information. For example, if the duobinary decoder is the first one, it should decode the five-level EPR4 signal to a ternary signal. Furthermore it should output soft decisions, otherwise, the following class-IV decoder functions as a hard Viterbi decoder. Soft-output Viterbi decoders and their digital realizations have been studied in literature [101-104].

The complexity of the aforementioned implementations are not necessarily low. In fact, if the implementation approach introduced in this chapter is taken, one might conclude that the eight-state EPR4 decoder is more efficient than any one of the above solutions. Toward this end, consider the trellis diagram of the EPR4 signaling scheme shown in figure 5.20. In this trellis, each branch is labeled with its metric value. The branch metrics are obtained by applying the linearization method described in chapter 3.

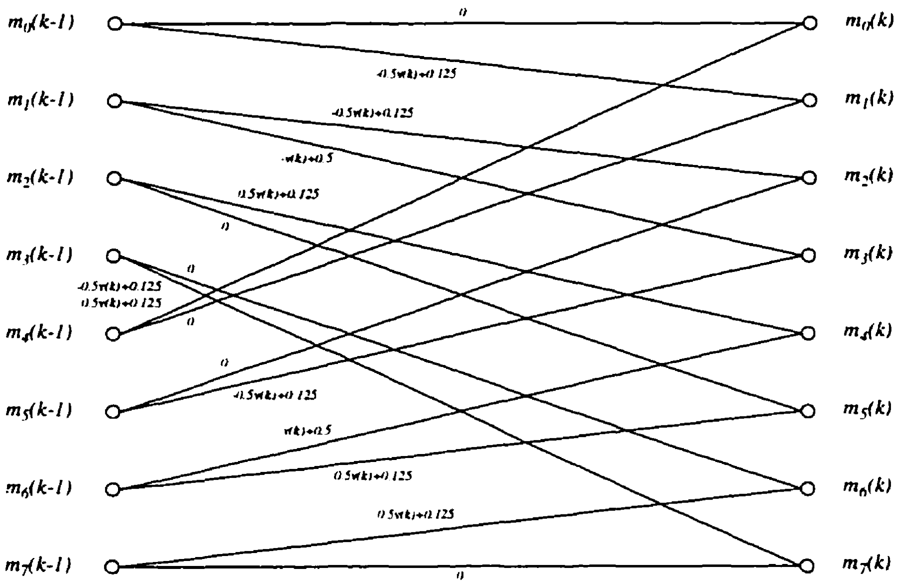


Figure 5.20: The trellis diagram of the binary EPR4 signaling scheme.

Application of the implementation technique to the above trellis is an extension to the dicode decoder design. The procedure is straight-forward and will not be repeated here[1]. Eight binary outputs update the contents of the path memory which, in a register-exchange configuration, consists of eight interconnected general shift registers. The decoded signal can be obtained by tracking the contents of one of the shift registers back to its very first stage. This shift register is selected by comparing eight state metrics and determining the maximum one. Alternatively, and in favor of reducing the design complexity, any one of the shift registers can be selected for a track back if a

[1] For the same compensation capacitor, however, a slightly more damped CMFB loop results.

deeper path memory is available[1].

5.4. Integrated-Circuit Implementation

An IC containing both the dicode and the EPR4 decoders was designed and fabricated. To save design time, the digital path memory was not included on the chip. Rather, comparators were employed to provide the serial and parallel loading controls for updating the off-chip path memory. To be able to drive the pads and the external circuitry, the comparator outputs were driven off-chip through open-collector differential pairs. Similar to chapter 4, the idea was to maintain flexibility in adjusting the output impedances, the driving capabilities, and the signal swings at the outputs of the chip.

The inputs to the decoders are the signals needed to generate the branch metrics. These are $y(t)$ and $V_{ref}/2$ for the dicode and $y(t)$, $y(t)/2$, $V_{ref}/2$, and $V_{ref}/8$ for the EPR4 decoder. Obviously, $y(t)/2$ and $V_{ref}/8$ can be obtained from $y(t)$ and $V_{ref}/2$ in a real situation. The common-mode reference voltage was also input to the decoders.

The bias voltages, addressed in figure 5.13 and 5.19 were obtained from the biasing circuit shown in figure 5.21. Except for the transistor sizes and the off-chip current source, the circuit is similar to the circuit described in section 4.5.8.

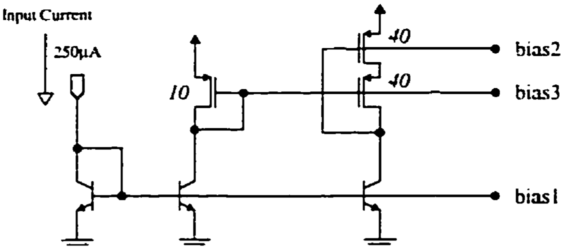


Figure 5.21: The biasing circuit.

The clock generator, depicted in figure 5.11, was used to generate the internal clocks from a single-phase off-chip master clock. This master clock, after phase adjustment, was also used to drive the path memory. The schematic of the inverter and transmission gates and the transistor sizes are given in figure 4.27. Those of the NOR gate is illustrated in figure 5.22.

It should be mentioned that all the MOS transistors in figures 5.13, 5.19, 5.21, 5.22, and 4.27 are minimum-length, except indicated, and have widths in μm specified in the figures. All the

[1] This approach was taken for the decoder described in chapter 4.

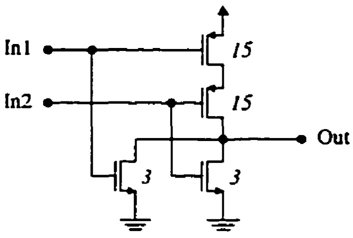


Figure 5.22: The schematic of the NOR gate used in the clock generator circuit.

BJTs are minimum size transistors available in the process.

5.5. Experimental Results

As it was mentioned earlier, due to the availability of the experimental setup, only the dicode decoder was tested. However, since the digital path-memory was left off-chip, the experiments were conducted at speeds much lower than what could have been achieved with a fully-integrated decoder. The path memory was implemented by wire-wrapping ECL shift registers. The local-optimum track back method, described in section 3.6.1, was followed. Figure 5.23 illustrates the experimental results at a speed of $50 Mb/s$. From this figure, it can be seen that the decoder performance indeed follows that of a Viterbi decoder. Almost similar results were obtained with a bit rate as high as $80 Mb/s$. However, with the off-chip path memory, careful adjustment of the phase of the memory clock relative to the master clock was required. The reference performance curve plotted in figure 5.23 was obtained from simulations. The sudden drop in the performance is caused by the truncated path memory and the local-optimum track back.

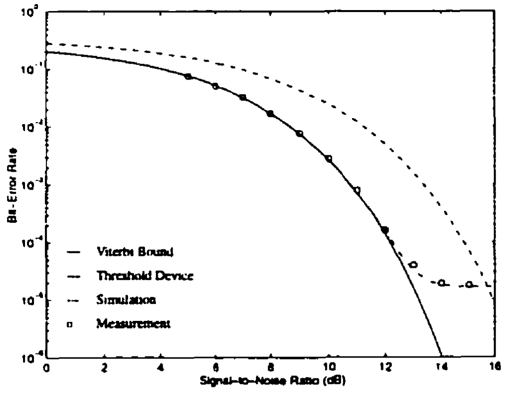


Figure 5.23: Measured BER performance of the dicode decoder at $50 Mb/s$. Note that a relatively-short path memory with the local-optimum track-back method has been used.

5.6. Layout

The dicode and EPR4 decoders were fabricated in a $0.8\mu m$ BiCMOS process[1]. The layout is illustrated in figure 5.24. The layout issues which improve the performance, and were addressed in section 4.7, were considered here as well.

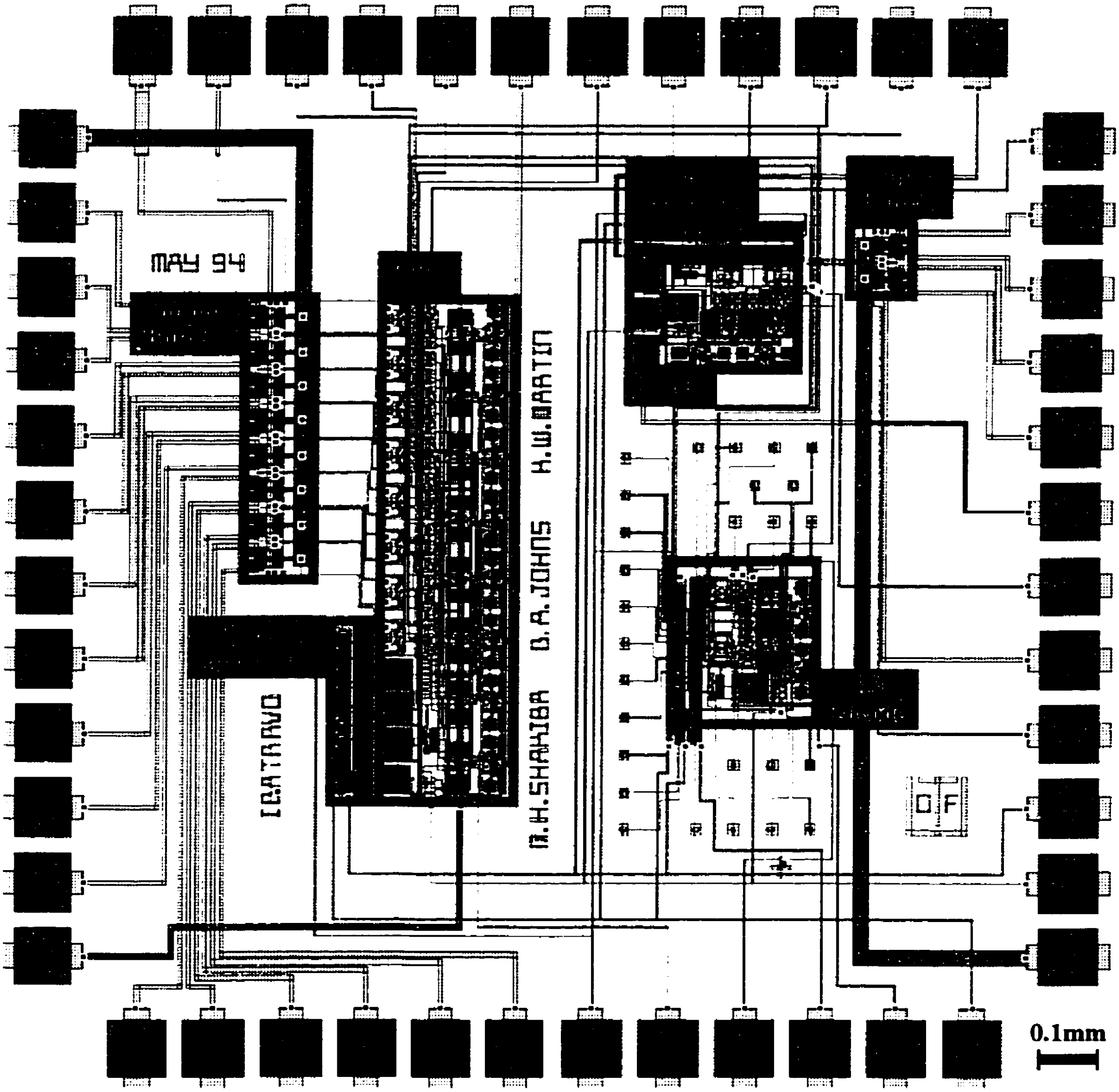


Figure 5.24: Layout of the dicode and EPR4 analog Viterbi decoders. A test dicode with test pads connected to some critical nodes is included in this layout as well.

The dicode decoder consists of a two-state analog processing core, a clock generator, and two

[1] The NT BATMOS process, available through CMC.

bipolar comparators. Two differential buffers are used to connect the outputs of the comparators to the pads. Also, a second dicode, with some test pads connected to its critical nodes, is included in the layout. This dicode could be used during the tests if more investigation of the internal operation of the decoder was necessary.

As evident from the layout, the eight-state EPR4 decoder is a straight-forward extension to the dicode decoder. Eight comparators and output buffers connect the outputs of the analog processor to the outside world. The clock generator is a duplicate of that of the dicode decoder.

5.7. Summary

In the previous chapter it was shown that a class-IV partial-response Viterbi decoder could save area and power and operate faster when implemented in the analog domain. To extend the aforementioned advantages to other Viterbi decoders, a general approach to implementing Viterbi decoders in an analog fashion seems promising. The generality of the realization technique is essential, as simplifications can not always be done. This chapter started with proposing a new implementation approach. After describing the approach, its basic building blocks were considered in detail and some circuit-level issues were discussed.

To illustrate the feasibility of the realization technique, two design examples were given. A two-state dicode decoder was used to address the implementation issues. The decoder was fabricated in a $0.8\mu m$ BiCMOS process. To save design time, the digital path memory was not included. The decoder, with an off-chip path memory, was successfully tested and it was observed that speeds in the order of tens of Mb/s can be easily achieved. Had the path memory been fabricated as well, higher speeds should have be attained. This was confirmed by simulations, where speeds in the order of few hundreds of Mb/s were achieved. The power consumption of the decoder was measured to be about $15mW/state$ drawn from a $5V$ power supply.

More complicated decoders are straight-forward extensions to the above simple decoder. To show how this extension could be done, an extended partial-response scheme was chosen. This scheme was chosen since its Viterbi decoder was a relatively complicated decoder with eight states and could be used to illustrate the extensibility of the approach. Furthermore, it was predicted that very soon the scheme will find its first application in the disk-drive industry. This second Viterbi decoder was briefly described and also fabricated in the same chip. However, experimental results are yet not available.

Finally, it should be mentioned that the implementation approach is general and can be used in error-correction coding systems, M-ary communications, and irregular trellises. The quaternary dicode scheme, which will be described in the next chapter, is a good example where application of the present approach in realizing its Viterbi decoder seems promising.

Chapter 6

Reduced-State Sequence Detection

The Viterbi algorithm provides a practical means of realizing an MLSD at much less complexity than a brute-force approach. However, the complexity of a Viterbi decoder is usually much more than that of a symbol-by-symbol detector. For example, in an N'th-order PRS scheme with an M-ary input signal, the implementation complexity of the Viterbi decoder is roughly M^{N+1} times that of a DFE [105]. In general, in detecting a PR signal, DFE and VA are at the two extremes of the spectrum from a complexity point-of-view.

There has been a significant amount of work to achieve a near-to-optimum performance in detecting a signal which has been subjected to severe ISI at a reduced complexity [105]-[115]. In the reduced-state sequence detection (RSSD) approach, the complexity reduction is achieved by reducing the number of states in the trellis diagram. The idea is to group the states of the original trellis and replace each group with a single hyper-state. A decision-feedback mechanism is then incorporated to resolve the ambiguities resulted from these state groupings. This mechanism causes the decoder to perform as a hybrid between a full-state sequence detector (with no state grouping) and a decision-feedback equalizer (when all of the states are grouped into one hyper-state). The sub-optimum degradation is minimized if the states are grouped in such a way that the minimum distance of the error events is maximized [116].

We start this chapter by establishing a strong connection between RSSD and DFE techniques. We will then show how this connection can be exploited to simplify the implementation problem of the RSSD decoder, through an illustrative example. The RSSD applied to quaternary class-IV PRS scheme will receive most of our attention in this chapter. Our primary motivation has been the search for a near-to-optimum, simple, fast, and low-power decoder for the QPRIV signal. It will be shown that all of these features can be achieved with an analog detector, similar to the one described in chapter 4. The results, however, can be used in digital realizations as well.

6.1. One-State Decoding: The DFE

RSSD can be viewed as an intermediate detection technique between two extremes of MLSD and DFE. To make this point clear, we start this chapter by analyzing the performance of the MLSD when all of its states are combined into one single hyper-state. The trellis diagram consists of only one state, connected to the same state at the previous time step with $2^N M$ parallel transitions. However, this number can be reduced to M if a decision feedback mechanism is used to give an estimate of the combined states from the previous time step. Assuming such an estimate exists and is represented by the vector $[\hat{x}(k-N)\ldots\hat{x}(k-1)]$, where $\hat{x}(j)$ denotes the decoded output at time step j, the trellis diagram shown in figure 6.1 results.

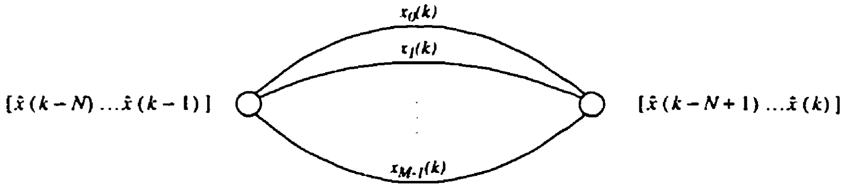

$[\hat{x}(k-N)\ldots\hat{x}(k-1)]$
$x_0(k)$
$x_1(k)$
$x_{M-1}(k)$
$[\hat{x}(k-N+1)\ldots\hat{x}(k)]$

Figure 6.1: The trellis diagram of a one-state sequence detector. The branch labels are the inputs leading to the corresponding transitions.

For the PRS system characterized by (2.1), the encoded signal associated with the transition labeled $x_j(k)$ will be equal to $h(x_j(k)+\sum_{i=1}^{N} f_i\hat{x}(k-i))$, where $j = 0, \ldots, M-1$ and h is the normalizing gain given by (2.2). Correspondingly, there will be M branch metrics which based on the Euclidean distance measure can be expressed as

$$B_j(k) = (y(k) - h(x_j(k) + \sum_{i=1}^{N} f_i\hat{x}(k-i)))^2 \qquad ,j = 0, \ldots, M-1 \qquad (6.1)$$

where $y(k)$ is the sampled received signal and is contaminated by white Gaussian noise. Now, the VA can be invoked to update the hyper-state metric and the survivor sequence. From (3.1) the update mechanism results

$$M_0(k) = \min_j \{M_0(k-1) + B_j(k)\} \qquad ,j = 0, \ldots, M-1 \tag{6.2}$$

The above minimization problem can be equivalently solved if the terms which are independent of j are dropped from the equation. As a result, and after some simplifications, (6.2), in conjunction with (6.1), reduces to

$$M_0(k) = \min_j \left\{x_j(k)\left(\frac{1}{2}x_j(k) + \sum_{i=1}^{N} f_i\hat{x}(k-i) - \frac{1}{h}y(k)\right)\right\} \qquad ,j = 0, \ldots, M-1 \tag{6.3}$$

In an M-ary signaling scheme with input symbols equally spaced in the interval $[-1, 1]$, the symbols are given by $x_j(k) = -1 + j\Delta$, where $\Delta = 2/(M-1)$ is the separation between two adjacent symbol levels. For this system, one can easily show that if the metric value for $j = r$ is smaller than those obtained for $j = r \pm 1$, that is if it is a local minimum, it will be the global minimum in (6.3). This, implies that finding the global minimum is equivalent to finding j such that

$$-1 + (j - \frac{1}{2})\Delta < \frac{1}{h}y(k) - \sum_{i=1}^{N} f_i\hat{x}(k-i) < -1 + (j + \frac{1}{2})\Delta \qquad ,j = 0, \ldots, M-1 \tag{6.4}$$

Expression (6.4) can be realized by a slicer with M equally-spaced levels. Note that for $j = 0$ the left-side inequality should not be checked, as all of the values below $-1 + \Delta/2$ are detected as a "-1". A similar situation holds for $j = M-1$, where the right-side inequality should be discarded, as all of the values above $1 - \Delta/2$ should lead to a "1".

Comparing the above expression with what the DFE implements (figure 2.9), reveals that the one-state sequence detector indeed functions as a DFE in recovering the data from the PR signal. Also, since the value of the state metric will not be propagated during the iterations, one does not need to calculate it at all[1]. The results of the comparisons expressed by (6.4) are only required.

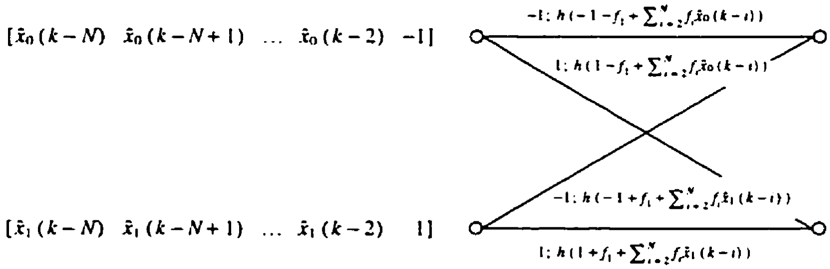
The survivor extension simply takes place by adding the input symbol corresponding to the branch which minimizes (6.2) to the previous survivor sequence. Since there is no other competing sequence, the updated survivor gives the final decision made by the RSSD. Also, this decision will serve as the new estimate of the combined states once it is fed back.

[1] Recall from chapter 3 that the number of propagating quantities can be as low as the number of states minus one.

6.2. Two-State Decoding of Binary PRS

Considering RSSD as a detection technique between MLSD and DFE, the next question arises is how one can take advantage of this fact to reduce the complexity of the detector. Since DFE is located at the lower-complexity end, a general answer to the above question is to exploit the feedback mechanism of the RSSD. To illustrate this issue, two-state decoding of binary-PRS, based on a trellis diagram obtained by grouping even and odd states into two separate hyper-states, is considered in this section. Regardless of any applications, which may exist or not, the example seems to be very useful in illustrating the concept.

For a PRS system characterized by (2.1), with normalized signals, the two-state trellis is shown in figure 6.2



$[\hat{x}_0(k-N)\ \ \hat{x}_0(k-N+1)\ \ \ldots\ \ \hat{x}_0(k-2)\ \ -1]$
$-1;\ h(-1-f_1+\sum_{i=2}^{N} f_i\hat{x}_0(k-i))$
$1;\ h(1-f_1+\sum_{i=2}^{N} f_i\hat{x}_0(k-i))$
$-1;\ h(-1+f_1+\sum_{i=2}^{N} f_i\hat{x}_1(k-i))$
$[\hat{x}_1(k-N)\ \ \hat{x}_1(k-N+1)\ \ \ldots\ \ \hat{x}_1(k-2)\ \ 1]$
$1;\ h(1+f_1+\sum_{i=2}^{N} f_i\hat{x}_1(k-i))$

Figure 6.2: Two-state trellis of a binary-PRS system resulted from grouping even and odd states. The branch labels show the pairs of normalized *uncoded; encoded* signals.

In the above figure, the last $N-1$ decoded outputs for hyper-states **0** and **1** are denoted by $[\hat{x}_0(k-N)\ \ \hat{x}_0(k-N+1)\ \ \ldots\ \ \hat{x}_0(k-2)]$ and $[\hat{x}_1(k-N)\ \ \hat{x}_1(k-N+1)\ \ \ldots\ \ \hat{x}_1(k-2)]$ respectively, and h is given by (2.2).

From the encoded signals shown in figure 6.2, the branch metrics can be calculated. Applying these metrics to the difference-metric algorithm results in the update mechanism given by (3.4) with

$$b_{00}(k)-b_{01}(k) = 2h^2\left(\frac{1}{h}y(k)+f_1-\sum_{i=2}^{N} f_i\hat{x}_0(k-i)\right) \tag{6.5}$$

$$b_{10}(k)-b_{11}(k) = 2h^2\left(\frac{1}{h}y(k)-f_1-\sum_{i=2}^{N} f_i\hat{x}_1(k-i)\right) \tag{6.6}$$

$$b_{00}(k)-b_{11}(k) = 2h^2\left(\frac{1}{h}y(k)-\frac{1}{2}\sum_{i=2}^{N} f_i(\hat{x}_0(k-i)+\hat{x}_1(k-i))\right)\left(1+f_1-\frac{1}{2}\sum_{i=2}^{N} f_i(\hat{x}_0(k-i)-\hat{x}_1(k-i))\right) \tag{6.7}$$

$$b_{10}(k)-b_{01}(k) = 2h^2\left(\frac{1}{h}y(k)-\frac{1}{2}\sum_{i=2}^{N} f_i(\hat{x}_0(k-i)+\hat{x}_1(k-i))\right)\left(1-f_1+\frac{1}{2}\sum_{i=2}^{N} f_i(\hat{x}_0(k-i)-\hat{x}_1(k-i))\right) \tag{6.8}$$

and the decision regions separated by threshold levels

$$b_{10}(k) - b_{00}(k) = 2h^2\left(1 + \frac{1}{h}y(k) - \frac{1}{2}\sum_{i=2}^{N} f_i(\hat{x}_0(k-i) + \hat{x}_1(k-i))\right)\left(-f_1 + \frac{1}{2}\sum_{i=2}^{N} f_i(\hat{x}_0(k-i) - \hat{x}_1(k-i))\right) = T - \Delta T \quad (6.9)$$

$$b_{11}(k) - b_{01}(k) = 2h^2\left(-1 + \frac{1}{h}y(k) - \frac{1}{2}\sum_{i=2}^{N} f_i(\hat{x}_0(k-i) + \hat{x}_1(k-i))\right)\left(-f_1 + \frac{1}{2}\sum_{i=2}^{N} f_i(\hat{x}_0(k-i) - \hat{x}_1(k-i))\right) = T + \Delta T \quad (6.10)$$

where

$$\Delta T = 2h^2\left(f_1 - \frac{1}{2}\sum_{i=2}^{N} f_i(\hat{x}_0(k-i) - \hat{x}_1(k-i))\right) \quad (6.11)$$

Starting from any set of two initial states of $2j$; $2j+1$ for $j = 0, 1, \ldots, N/2-1$ (i.e. $\sum_{i=2}^{N} f_i(\hat{x}_0(k-i) - \hat{x}_1(k-i)) = 0$), (6.9) and (6.10) result in two threshold levels of $T \pm 2f_1h^2$. Depending on the polarity of f_1, one of the middle decision regions in (3.4) will be discarded. Assuming $f_1 > 0$, only one of the three survivor extensions shown below can take place[1]

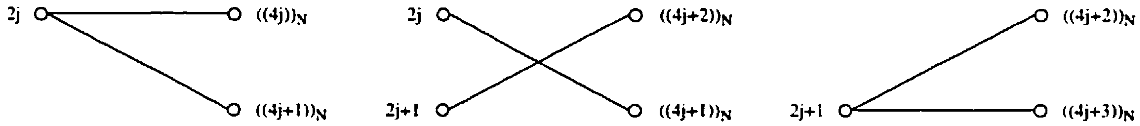


Figure 6.3: Survivor extensions in the binary PRS two-state RSSD, starting from states $2j$ and $2j+1$, and assuming $f_1 > 0$.

where $((u))_v$ denotes the residue of u divided by v.

From the three new sets of states, two are of the form of the starting set. The other set can be categorized as $4j+2$; $4j+1$ for $j = 0, 1, \ldots, N/4-1$. This category for which $\sum_{i=2}^{N} f_i(\hat{x}_0(k-i) - \hat{x}_1(k-i)) = 2f_2$ is, in turn, bootstrapped to serve as the starting states in the new iteration with the threshold levels of $T \pm 4h^2(f_1 - f_2)$, obtained from (6.9), (6.10), and (6.11). Similarly, the new survivor extensions depend on the polarity of $f_1 - f_2$. As an example, for $f_1 - f_2 > 0$ these extensions are shown in figure 6.4.

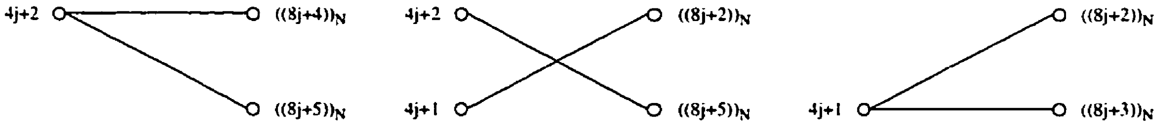


Figure 6.4: Survivor extensions in the binary PRS two-state RSSD, starting from states $4j+2$ and $4j+1$, and assuming $f_1 - f_2 > 0$.

From the three estimates of the new states, two belong to the starting category. The other estimate described by $8j+2$; $8j+5$ for $j = 0, 1, \ldots, N/8-1$, leads to a new category with threshold levels equal to $T \pm 4h^2(f_1 - f_2 + f_3)$. Again, based on the polarity of $f_1 - f_2 + f_3$ survivor exten-

[1] For $f_1 < 0$ a similar approach can be taken.

sions and consequently the new state estimates are determined. This process is repeated N times, where at the end none of the estimated sets is new. This completes all of the pairs of the estimated states through which the RSSD algorithm ever passes. It can be shown that all the remaining sets (out of $2^{2(N-1)}$) will not occur, since, these states eventually lead to the categories described above, where a reverse transition does not exist. Note that all of the update values required in the update mechanism can be calculated from (6.5) for each one of the estimated sets.

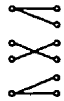
The above discussion shows that the two-state RSSD can be viewed as a difference-metric algorithm with threshold and update values determined from the previous state estimates. This tightly relates the decision feedback part of the RSSD with its sequence detection nature from an implementation point-of-view.

6.2.1. Binary EPR4

EPR4, characterized by $F(D) = 1 + D - D^2 - D^3$, is chosen as an example to illustrate what the RSSD architecture, described above, looks like in a typical realization. Following the general procedure, the first step initiating from states 0;1, 2;3, 4;5, and 6;7 either keeps the RSSD decoder in one of these states or leads it to one of two pairs of 2;1 and 6;5. These new states, in turn, result in either an estimate from the starting category, or the pair 2;5, through the second step. Finally, from this new estimate, the decoder is either returned back to the starting category, or remains in the current states. This last step wraps up the procedure, resulting in only 7 possible estimated pairs of states from which the RSSD decoder passes through during the iterations. All other 9 remaining sets will bring the RSSD decoder to one of the above 7 sets, where a reverse transition does not exist.

Furthermore, the update values and the threshold levels required for each one of the estimated pair of states can be calculated from (6.5), (6.9), (6.10), and (6.11). After some manipulations, the results can be summarized in the following update equation

$$\Delta m(k) = \alpha T - \begin{cases} T - \Delta T & , \quad \Delta m(k-1) < T - \Delta T \\ \Delta m(k-1) & , T - \Delta T < \Delta m(k-1) < T + \Delta T \\ T + \Delta T & , T + \Delta T < \Delta m(k-1) \end{cases} \tag{6.12}$$



where T, ΔT, α, and the new state estimates are given in table 6.2 for all of the above 7 state pairs.

Note that this significant reduction in complexity has been achieved at the expense of a penalty in the noise performance of the decoder. By combining 8 states of the MLSD decoder into only 2 states, the minimum distance of the error events will drop from 1 to $1/\sqrt{2}$, corresponding to $3dB$

Table 6.1: RSSD of the binary-EPR4 based on the difference-metric algorithm applied to the two-state trellis.

state estimates	0;1	2;3	4;5	6;7	2;1	6;5	2;5
T	$-\frac{1}{2}y(k)+\frac{1}{4}$	$-\frac{1}{2}y(k)$	$-\frac{1}{2}y(k)$	$-\frac{1}{2}y(k)-\frac{1}{4}$	$-y(k)+\frac{1}{4}$	$-y(k)-\frac{1}{4}$	$-\frac{1}{2}y(k)$
ΔT	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{8}$
α	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0
new estimates	0;1, 2;3, 2;1	4;5, 6;7, 6;5	0;1, 2;3, 2;1	4;5, 6;7, 6;5	2;3, 4;5, 2;5	2;3, 4;5, 2;5	2;3, 4;5, 2;5

loss from the $6dB$ coding gain. Figure 6.5 illustrates the BER performance of the MLSD and RSSD decoders, obtained from simulations. That of the DFE is also shown.

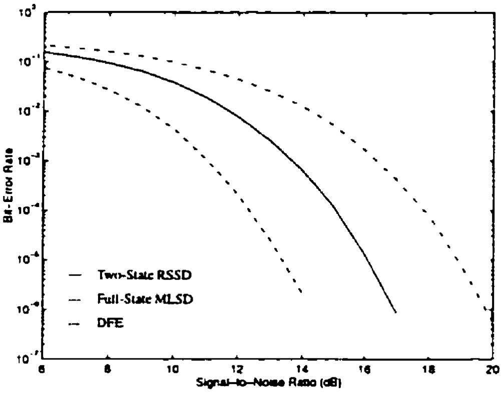


Figure 6.5: Noise performances of the MLSD and RSSD applied to the binary-EPR4 scheme.

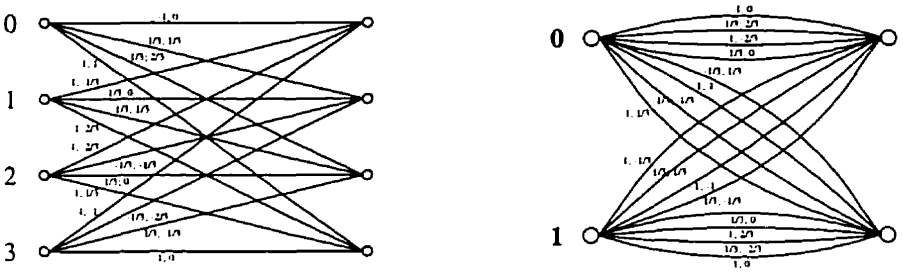
6.3. Two-State Decoding of Quaternary Class-IV

It was mentioned in chapter 2 that a QPRIV system has been recently proposed for high-rate data transmission over unshielded twisted-pair cables. For the specific application considered here, extremely low-cost transceivers are desired [117]. Therefore, MLSD, realized by the VA, is favored only if a detector solution of low-enough complexity exists. Although it was shown that for a binary class-IV system such a solution is known, a sub-optimum decoder seems to be more promising in the case were multi-level signals are employed. Towards this goal, reduced-state sequence detection of multi-level PRS has been studied [115], resulting in a difference-metric algorithm for multi-level class-IV signals [118]. Furthermore, a VLSI implementation for a QPRIV transceiver employing a reduced 2-state Viterbi decoder is recently proposed [42].

The purpose of the work here is to develop the Viterbi algorithm for decoding a quaternary class-IV signal from an analog implementation point-of-view. As it was demonstrated in chapter 4, for the binary scheme analog implementations offer viable alternatives to their digital counterparts and result in high-speed decoders with significant savings in the size and power consumption. Here, the goal is to extend the binary-decoder structure to the quaternary system by first reducing the number of states of the trellis to two and then applying the difference-metric algorithm, or more appropriately the input-interleaved algorithm, to the two-state trellis. Also, it is shown that the derived algorithm here is a simplified version of the difference-metric algorithm described in [118] and can be alternatively used in a digital realization. We shall refer to this algorithm as the "*reduced-state simplified difference-metric algorithm*" in this thesis.

6.3.1. Reducing the Number of States

A quaternary dicode signaling scheme[1] leads to a trellis diagram shown in figure 6.6.a. It has been shown that if the four states of this trellis are reduced to two, by combining the even states into one hyper-state and the odd states into another hyper-state, a significant reduction in the complexity of the sequence detector results with a negligible penalty in the performance [115]. However, the reduced-state trellis will contain some parallel branches, as shown in figure 6.6.b.



a. Full-state trellis

b. Two-state trellis

Figure 6.6: Full-state and two-state trellis diagrams of the quaternary dicode scheme. Branch labels represent the pairs of the normalized *uncoded; encoded* signals.

Similar to the approach presented in chapter 3, one can easily show that the branch metrics of the full-state decoder are given by

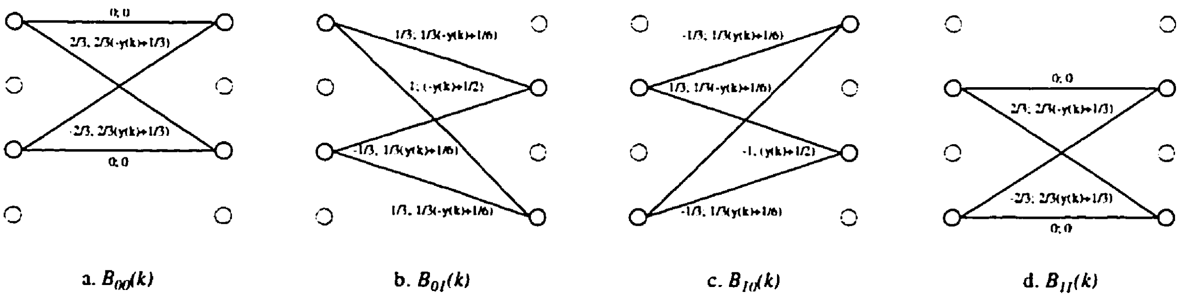
$$b_{ji}(k) = \frac{j-i}{3}\left(y(k) + \frac{j-i}{6}\right) \qquad , \begin{matrix} j = 0, 1, 2, 3 \\ i = 0, 1, 2, 3 \end{matrix} \tag{6.13}$$

However, combining the states leads to ambiguities in deriving branch metrics in the reduced-state decoder, since now there is more than one transition between each two subsequent states. The branch metrics, $B_{JI}(k)$, can take any of the values

[1] Recall that a class-IV system is usually realized by time-interleaving two independent dicodes.

$$B_{JI}(k) = b_{ji}(k) \qquad \begin{matrix} J = 0, 1 \\ I = 0, 1 \end{matrix}, \qquad \begin{matrix} j = J, J+2 \\ i = I, I+2 \end{matrix}, \tag{6.14}$$

The above ambiguities can be only avoided if the algorithm is provided with enough information to resolve four parallel transitions between each two subsequent hyper-states. These parallel transitions are further illustrated in figure 6.7.



a. $B_{00}(k)$ b. $B_{01}(k)$ c. $B_{10}(k)$ d. $B_{11}(k)$

Figure 6.7: Parallel transitions resulted from state grouping. The pairs of *normalized-encoded signal; branch metric* are also shown.

Using estimates of the combined states for each hyper-state at time step $k-1$, eliminates two of each four parallel transitions. Two remaining transitions can be resolved by applying a threshold decision on the received signal. Starting from known initial states, a decision-feedback mechanism completes the loop by boot-strapping the new state estimates to be used in the next iteration.

6.3.2. Reduced-State Simplified Difference-Metric Algorithm

The first step mentioned above in resolving the ambiguities in the branch metrics of the reduced-state decoder, namely estimating the combined states, leads to four different cases (four different combinations of the estimated states) in which the Viterbi algorithm can potentially proceed. This reduces each set of four parallel transitions to a set of two. Further resolution is possible through the second step which gives the most-likely transitions in a sub-optimum manner by slicing the input signal[1]. The threshold levels of the slicer can be determined from the encoded signals shown in figure 6.7. Having resolved the parallel transitions, the difference-metric algorithm given by (3.4) (with b_{ji} and Δm replaced by B_{ji} and ΔM, respectively) can be invoked. Tables 6.2 to 6.5 summarize the results. Each table corresponds to one of the four different combinations of the estimated states with entries representing the resolved branch metrics. Survivor extensions are also shown. In addition to updating the path memory, estimated states required for the next iteration of the algorithm, should be also determined from these extensions.

[1] Note that the slicing described here is equivalent to resolving the first three bits in the binary representation of the signal in [118].

Table 6.2: Resolving parallel transitions in the RSSD of the quaternary dicode PR signal. Estimates of hyper-states **0** and **1** = states 0 and 1.

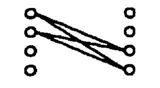
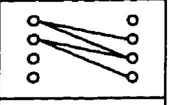
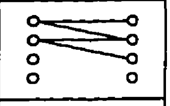
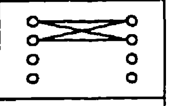
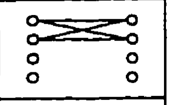
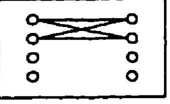
y(k)	$B_{00}(k)$	$B_{01}(k)$	$B_{10}(k)$	$B_{11}(k)$	Survivor Extension
↕ — 2/3	$\frac{2}{3}(-y(k)+\frac{1}{3})$	$-y(k)+\frac{1}{2}$	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{2}{3}(-y(k)+\frac{1}{3})$	
↕ — 1/3	$\frac{2}{3}(-y(k)+\frac{1}{3})$	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{2}{3}(-y(k)+\frac{1}{3})$	
↕ — 0	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(-y(k)+\frac{1}{6})$	0	
↕ — -1/3	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕ — -2/3	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	

Table 6.3: Resolving parallel transitions in the RSSD of the quaternary dicode PR signal. Estimates of hyper-states **0** and **1** = states 0 and 3.

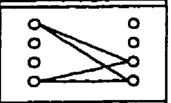
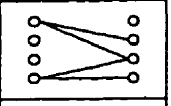
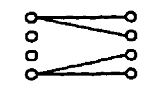
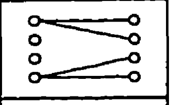
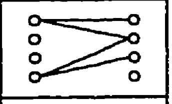
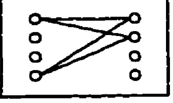
y(k)	$B_{00}(k)$	$B_{01}(k)$	$B_{10}(k)$	$B_{11}(k)$	Survivor Extension
↕ — 2/3	$\frac{2}{3}(-y(k)+\frac{1}{3})$	$-y(k)+\frac{1}{2}$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕ — 1/3	$\frac{2}{3}(-y(k)+\frac{1}{3})$	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕ — 0	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕ — -1/3	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕ — -2/3	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	$\frac{2}{3}(y(k)+\frac{1}{3})$	
↕	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$y(k)+\frac{1}{2}$	$\frac{2}{3}(y(k)+\frac{1}{3})$	

Table 6.4: Resolving parallel transitions in the RSSD of the quaternary dicode PR signal. Estimates of hyper-states **0** and **1** = states 2 and 1.

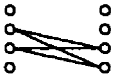
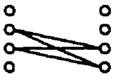
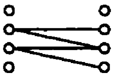
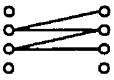
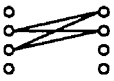
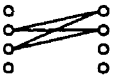
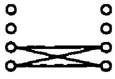
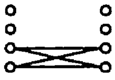
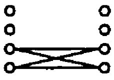
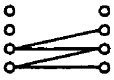
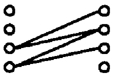
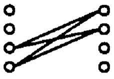
y(k)	$B_{00}(k)$	$B_{01}(k)$	$B_{10}(k)$	$B_{11}(k)$	Survivor Extension
↕ 2/3	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{2}{3}(-y(k)+\frac{1}{3})$	
↕ 1/3	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{2}{3}(-y(k)+\frac{1}{3})$	
↕ 0	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(-y(k)+\frac{1}{6})$	0	
↕ -1/3	0	$\frac{1}{3}(y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕ -2/3	$\frac{2}{3}(y(k)+\frac{1}{3})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕	$\frac{2}{3}(y(k)+\frac{1}{3})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	

Table 6.5: Resolving parallel transitions in the RSSD of the quaternary dicode PR signal. Estimates of hyper-states **0** and **1** = states 2 and 3.

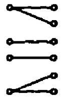
y(k)	$B_{00}(k)$	$B_{01}(k)$	$B_{10}(k)$	$B_{11}(k)$	Survivor Extension
↕ 2/3	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕ 1/3	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕ 0	0	$\frac{1}{3}(-y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕ -1/3	0	$\frac{1}{3}(y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	0	
↕ -2/3	$\frac{2}{3}(y(k)+\frac{1}{3})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	$\frac{2}{3}(y(k)+\frac{1}{3})$	
↕	$\frac{2}{3}(y(k)+\frac{1}{3})$	$\frac{1}{3}(y(k)+\frac{1}{6})$	$y(k)+\frac{1}{2}$	$\frac{2}{3}(y(k)+\frac{1}{3})$	

To further explain how these tables are obtained, consider, as an example, state 0 as the previous estimate of the hyper-state **0**. This reduces the four candidates for the transition from hyper-state **0** to the same hyper-state, shown in figure 6.7.a, to only those two which initiate from state 0. Considering the fact that 0 and $2/3$ are two encoded signals associated with these transitions, applying a threshold level of $1/3$ to the received signal, $y(k)$, uniquely specifies the final transition and consequently the corresponding branch metric the reduced-state decoder should use. As a result, the entries of the first column of table 6.2 are determined. The ending state resulting from this particular transition is also resolved. This fact is being used in deriving the last column which illustrates the survivor extensions. All the other entries of the above tables can be obtained in a similar manner.

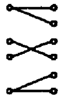
It is interesting to note that the case described by table 6.3 will never occur simply because there exist transitions from this case to other cases (tables 6.2, 6.4, and 6.5), but, not vice versa. This is in agreement with the adjacency relation in [118] which states that two most-likely states should remain adjacent during the iterations. Also, note that some of the adjacent decision regions given in these tables need not to be resolved at all. The last three rows of table 6.2 show an example. Similar situations can be found in other tables. Extending the decision feedback to update the threshold levels of the threshold device, effectively reduces the number of comparators required in the analog implementation.

The rows of tables 6.2, 6.4, and 6.5 can be divided into two categories. *Category-I*, for which the new estimated states are independent of the computations and only depend on the decisions made by the threshold device and *category-II*, for which these estimations depend on the results of the algorithm, as well. As an example, the first and three last rows of table 6.2 belong to category-I, whereas two other rows fall in category-II. The difference-metric algorithm described by (3.4) applied to the members of category-I reduces to very simple update mechanisms each with only three survivor extensions. It is not difficult to see that these update mechanisms and their corresponding survivor extensions can be expressed by one of the two following expressions

$$\Delta M(k) = \begin{cases} T - 1/9 & , \quad \Delta M(k-1) < T - 1/9 \\ \Delta M(k-1) & , T - 1/9 < \Delta M(k-1) < T \\ T & , \quad T < \Delta M(k-1) \end{cases} \tag{6.15}$$



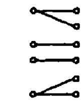
$$\Delta M(k) = \begin{cases} -T + 1/9 & , \quad \Delta M(k-1) < T - 1/9 \\ -\Delta M(k-1) & , T - 1/9 < \Delta M(k-1) < T \\ -T & , \quad T < \Delta M(k-1) \end{cases} \tag{6.16}$$



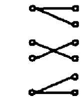
where T is a threshold level derived from (3.4) in conjunction with tables 6.2, 6.4, and 6.5 for each one of the members of this category.

In contrast to the above, applying the difference-metric algorithm directly to category-II does not lead to a similar simplification. However, such a simplification is indeed possible if $y(k)$ is further resolved. The required resolution imposes four additional threshold levels of $\pm 1/6$ and $\pm 1/2$ to the above tables. This extra one-bit resolution is available in a digital realization, since the number of bits in the binary representation of the signal almost always exceeds 4. It is not difficult to see that in each one of the resulted sub-regions, the decision criteria given by (3.4) will be reduced to only three. As well, in each one of the two sub-regions resulted from each member of category-II, one of the two following expressions applies

$$\Delta M(k) = \begin{cases} -T & , \quad \Delta M(k-1) < -T \\ \Delta M(k-1) & , -T < \Delta M(k-1) < T \\ T & , \quad T < \Delta M(k-1) \end{cases} \tag{6.17}$$



$$\Delta M(k) = \begin{cases} T & , \quad \Delta M(k-1) < -T \\ -\Delta M(k-1) & , -T < \Delta M(k-1) < T \\ -T & , \quad T < \Delta M(k-1) \end{cases} \tag{6.18}$$



where T is a positive threshold level, again obtained from (3.4) in conjunction with tables 6.2, 6.4, and 6.5 for each one of the sliced sub-regions.

The results of the above arguments are summarized in table 6.6. Assuming that estimates of the combined states are available from the previous iteration, the appropriate update equation, the threshold level, T, and the new estimates of the hyper-states can be obtained from this table. These information are given for different ranges of the received signal (normalized to noiseless peaks of ± 1) determined by applying it to a slicer with threshold values given in the first column of the table. Note that for the sliced regions which fall in category-II, only one of the state estimates is available before any calculations. The other state can be estimated at the end of the iteration.

As an example, the decision regions, the algorithm output, and the survivor extensions for the first row of the first column, three-first rows of the second column, and five-first rows of the third column of table 6.6 (first, two-first, and three-first rows of tables 6.2, 6.4, and 6.5, respectively), from category-I, and for the second and third rows of the first column of table 6.6 (second row of table 6.2), from category-II, are illustrated in figure 6.8.

6.3.3. The Analog Architecture

The above arguments suggest that the problem of Viterbi detection of a quaternary dicode can be changed to that of a binary one[1]. However, the difference-metric algorithm should be supplied

[1] See section 4.2.

Table 6.6: The reduced-state simplified difference-metric algorithm applied to the two-state trellis of a quaternary dicode PRS scheme.

Estimated States	0, 1			2, 1			2, 3		
y(k)	eqn.	T	New Estimates	eqn.	T	New Estimates	eqn.	T	New Estimates
↕ - 2/3	(6.15)	$\frac{1}{3}(y(k)-\frac{1}{2})$	2, 3	(6.16)	$\frac{1}{3}(-y(k)+\frac{1}{2})$	2, 3	(6.15)	$\frac{1}{3}(y(k)+\frac{1}{6})$	2, 3
↕ - 1/2	(6.17)	$\frac{1}{3}(y(k)-\frac{1}{2})$	2, x						
↕ - 1/3	(6.18)	$\frac{1}{3}(-y(k)+\frac{1}{2})$							
↕ - 1/6		$\frac{1}{3}(y(k)-\frac{1}{6})$	x, 1	(6.18)	$\frac{1}{3}(y(k)-\frac{1}{6})$	2, x			
↕ - 0	(6.17)	$\frac{1}{3}(-y(k)+\frac{1}{6})$		(6.17)	$\frac{1}{3}(-y(k)+\frac{1}{6})$				
↕ - -1/6	(6.15)	$\frac{1}{3}(y(k)+\frac{1}{6})$	0, 1		$\frac{1}{3}(y(k)+\frac{1}{6})$	x, 1	(6.17)	$\frac{1}{3}(y(k)+\frac{1}{6})$	2, x
↕ - -1/3				(6.18)	$\frac{1}{3}(-y(k)-\frac{1}{6})$		(6.18)	$\frac{1}{3}(-y(k)-\frac{1}{6})$	
↕ - -1/2				(6.16)	$\frac{1}{3}(-y(k)-\frac{1}{6})$	0, 1		$\frac{1}{3}(y(k)+\frac{1}{2})$	x, 1
↕ - -2/3							(6.17)	$\frac{1}{3}(-y(k)-\frac{1}{2})$	
↕							(6.15)	$\frac{1}{3}(y(k)+\frac{5}{6})$	0, 1

with threshold levels derived from the previous state estimates and the results of slicing the input signal. These threshold levels are shifted versions of the sampled input signal (possibly with a sign change) and are easy to calculate. The first step is to slice the input signal, $y(k)$, and generate six digital signals corresponding to six rows of table 6.6. Five comparators, which compare $y(k)$ with five equally-spaced threshold levels, followed by four AND gates provide six mutually exclusive outputs. The slicing boundaries should all be biased with a DC value depending upon the previous estimates. Figure 6.9 depicts the idea. In this figure, the pairs of estimated states 0;1, 2;1, and 2;3 are indicated by I, III, and V respectively.

A simple look at the threshold values, T, given in table 6.6 shows that the appropriate polarity of the input signal and the required DC shifts to construct two upper and lower threshold levels can be ruled by the slicer outputs and the previous state estimates. The signals for controlling the polarity of $y(k)$, P_u and P_l, and selecting the DC shifts, $L_u^{-1/2}$, $L_u^{-1/6}$, $L_u^{1/6}$, $L_u^{1/2}$, $L_u^{5/6}$, $L_l^{-5/6}$, $L_l^{-1/2}$, $L_l^{-1/6}$, $L_l^{1/6}$, and $L_l^{1/2}$, can be generated by the typical combinational circuits shown in figures 6.10 and 6.11. Here, subscripts are used to distinguish between the upper and lower threshold

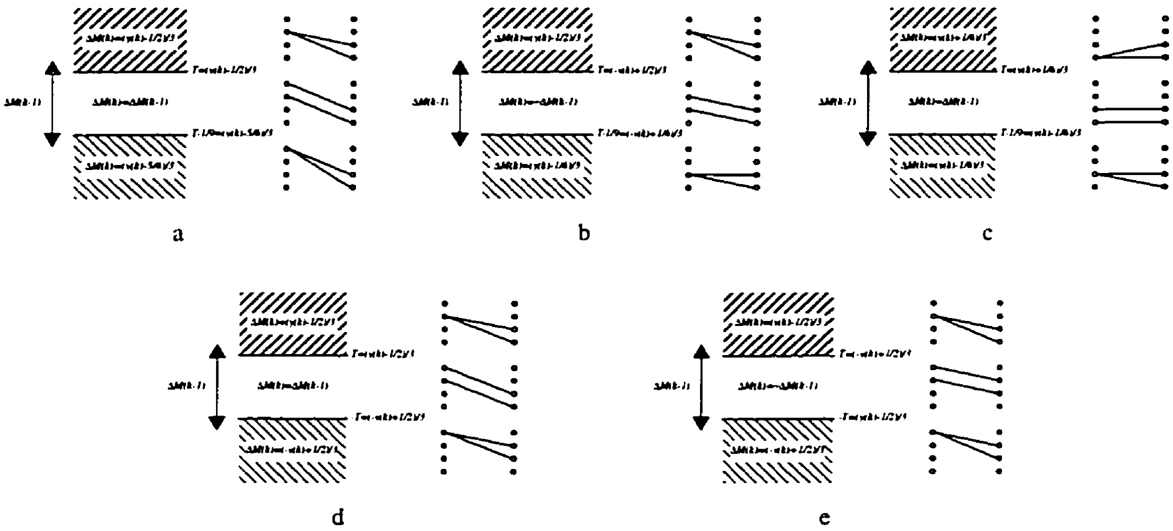


Figure 6.8: Difference-metric algorithm applied to the: a. first row of the first column (first row of table 6.2), b. three-first rows of the second column (two-first rows of table 6.4), c. five-first rows of the third column (three-first rows of table 6.5), d. second row of the first column (second row of table 6.2 for $1/2 < y(k) < 2/3$), and e. third row of the first column (second row of table 6.2 for $1/3 < y(k) < 1/2$) of table 6.6.

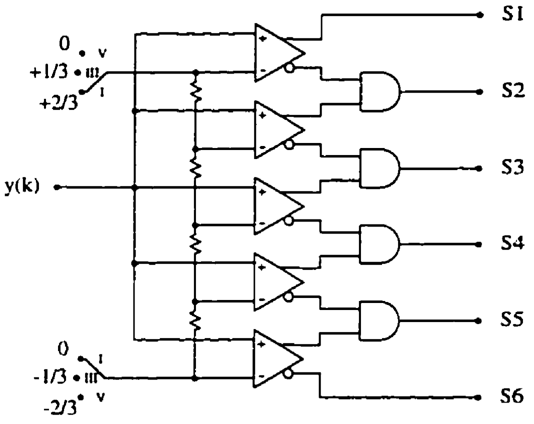


Figure 6.9: Slicing the input signal with five comparators and by making use of the previous state estimates.

levels, whereas superscripts denote the values of the required DC shifts.

Limiting mechanisms expressed by (6.15) to (6.18) are used to update the difference signal. In updating this signal, the polarity should be positive if either (6.15) or (6.17) applies, negative if either (6.16) or (6.18) applies and the difference signal does undergo an update, and reversed if either (6.16) or (6.18) applies but the difference signal does not undergo the update. As a result, the polarity of the updated difference signal can be set by a control signal generated by the circuit

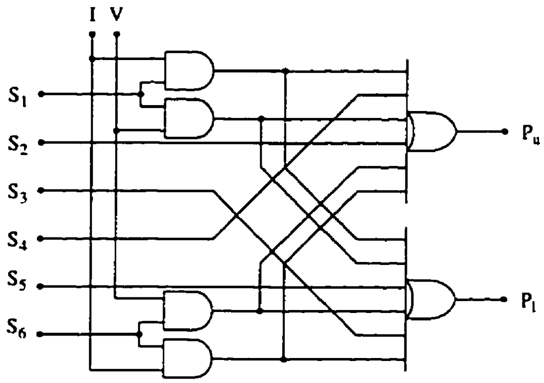


Figure 6.10: A typical combinational circuit for generating signals which set the polarity of the input signal in constructing the upper and lower threshold levels.

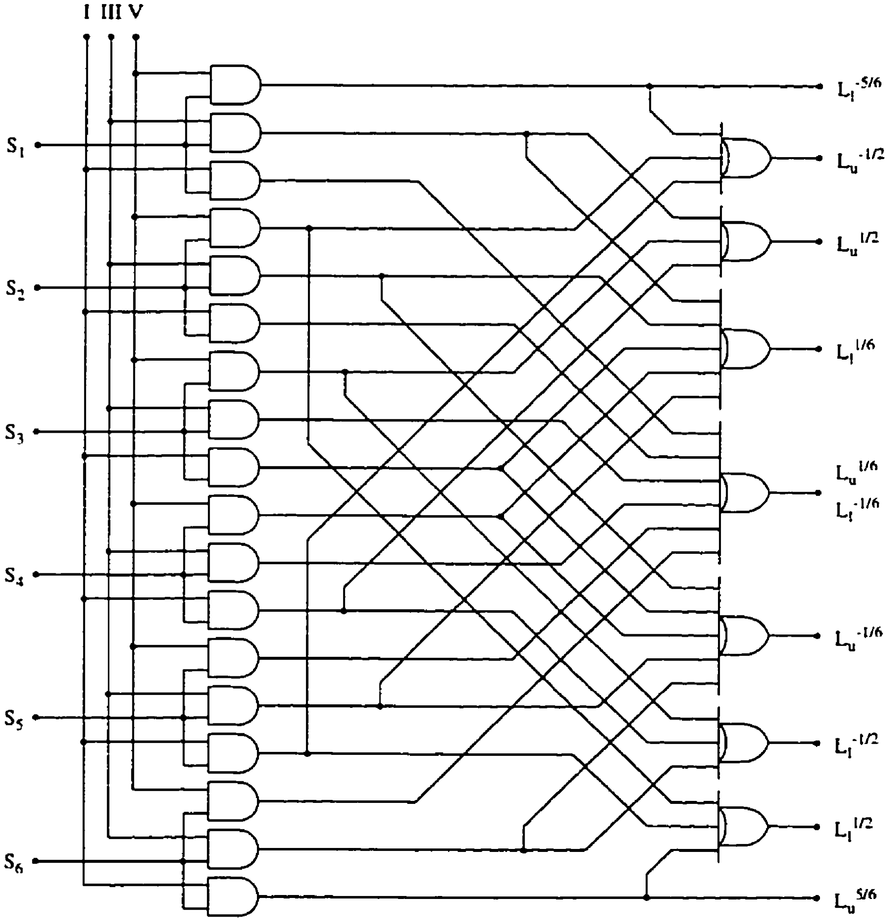


Figure 6.11: A typical combinational circuit for generating signals which set the DC values in constructing the upper and lower threshold levels.

shown in figure 6.12. In this figure, E_a is a signal which indicates when either (6.15) or (6.17) is effective, and Q_u and Q_l correspond to the latched outputs of the comparators in the adaptive threshold detector (figure 4.3).

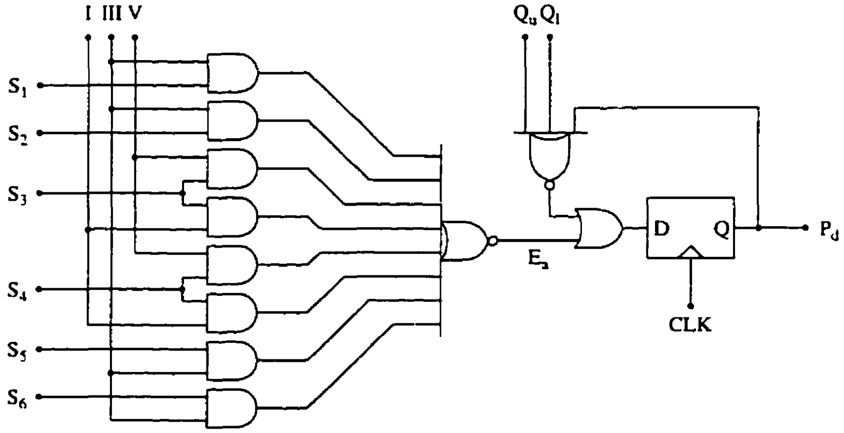


Figure 6.12: Generating the control signal which sets the polarity of the difference-metric signal. A "1" at the output corresponds to a positive polarity and a "0" to a negative polarity. AND gates are redrawn from figure 6.11.

At the end of each iteration, the new estimates of the combined states should be bootstrapped to be used in the next iteration. Again, table 6.6 can be employed to generate these estimates. While for the first and last rows of this table (category I) estimates are available soon after the start of the iteration, for the middle rows (category II) they depend on the calculations. However, to prevent any destructive feedback effect, the results should be latched and only fed back at the start of the next iteration. It is straight-forward to see that the circuit in figure 6.13 is capable of generating the new estimates based on the information given in the above table.

Finally, path memory of the decoder should be updated at the end of each iteration. With quaternary signals, this memory should be constructed from two-bit storage elements interconnected similar to what was shown in figure 4.4. In fact, path memory is a two-bit, but deeper, version of this circuit. Figure 6.14 illustrates the path memory structure. Update rules come from the memory extensions shown by (6.15) to (6.18) and the new inserted words for each hyper-state are the estimates generated by the circuit of figure 6.13. In the usual cases where the quaternary scheme has been chosen to reduce the baud rate of the overall binary communication system, the inserted bits can be selected such that the detected two-bit words are directly mapped to the original binary data by converting them into a serial bit stream. Note that unlike the binary decoder, the comparator outputs do not directly control the registers, but through the simple logic circuit depicted in figure 6.15. This circuit adds simultaneous parallel loading of both shift registers, required in the quaternary decoder, to other loading features of the binary path memory.

It should be mentioned that the circuits shown here are typical and the control signals can be

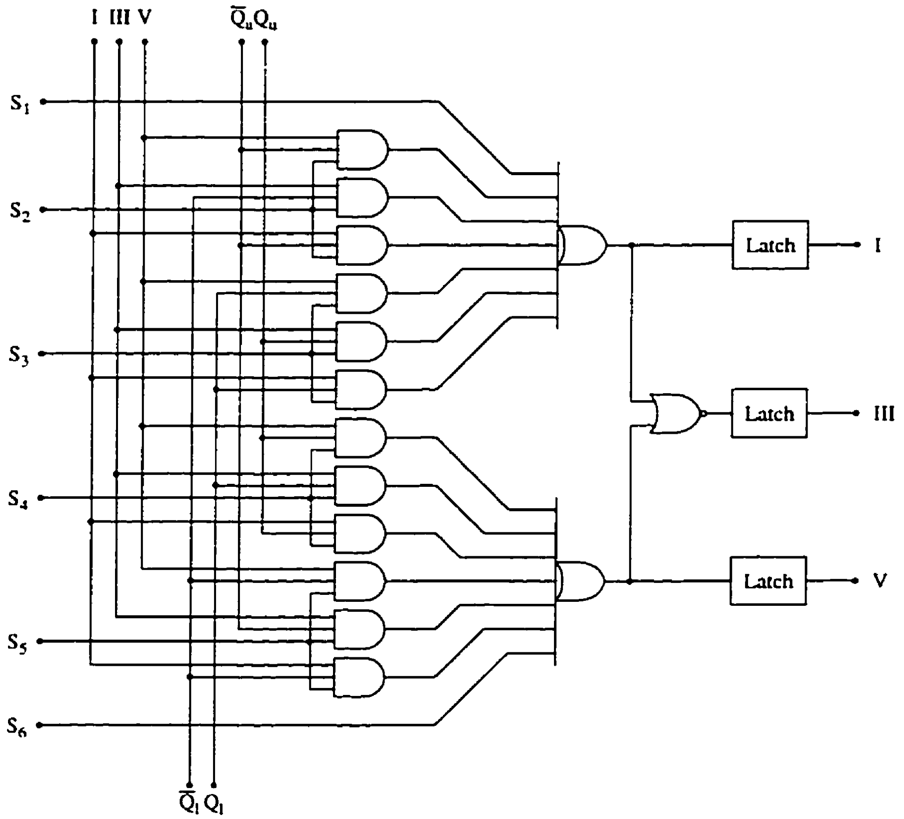


Figure 6.13: A typical circuit for generating new estimates of the combined states.

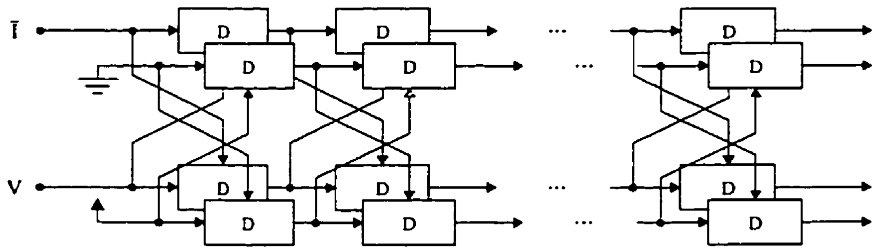


Figure 6.14: Path memory of the reduced-state quaternary decoder.

generated by implementing the logic functions extracted from table 6.6 in different ways. A circuit-level block diagram of the quaternary dicode decoder is further illustrated in figure 6.16. The QPRIV decoder consists of two independent time-interleaved dicode decoders.

6.3.4. Performance Evaluation

When resolving the parallel transitions by the slicer, the decisions made by these devices are reliable since the distance between two branches initiating from each state is larger than the minimum distance of the error events of the sequence detector. Consequently, the performance degra-

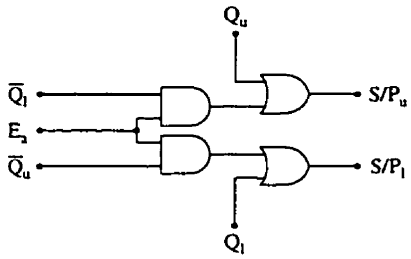


Figure 6.15: Generating signals for updating the path memory. Each signal controls the serial/parallel loading of its corresponding shift register. A "1" corresponds to a serial shift and a "0" to a parallel load.

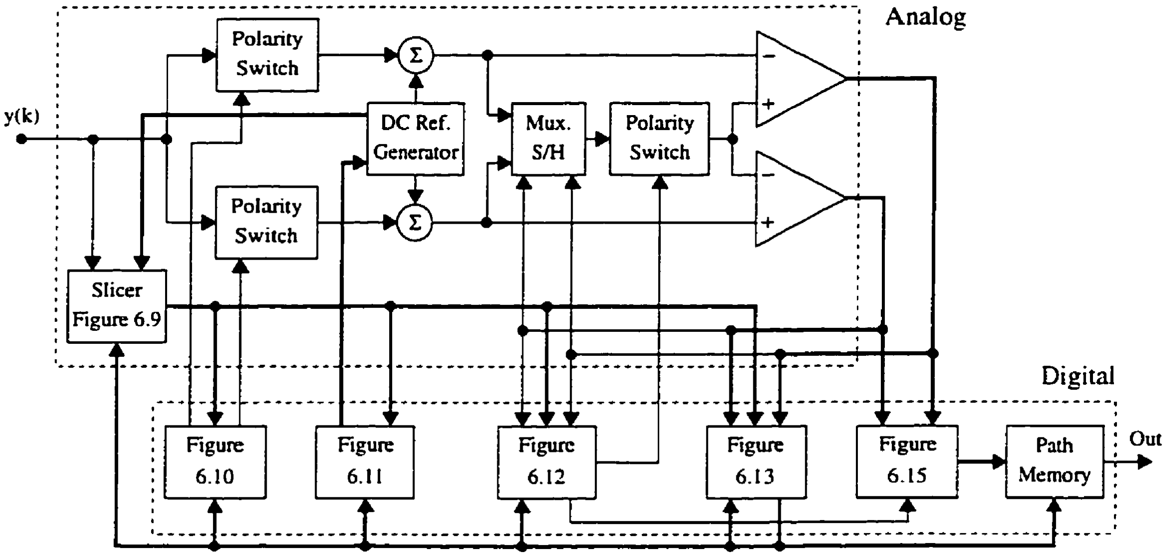


Figure 6.16: Circuit-level block diagram of the analog quaternary dicode decoder. Bold lines indicate multi-signal connections.

dation of the decoder and the error propagation effect are expected to be negligible. This, can be confirmed by comparing the minimum distance of the error events in the two-state trellis diagram with that in the four-state diagram. Figure 6.17.a shows typical minimum-distance events (starting from state 0), which give rise to errors in both of the full-state and reduced-state decoders. However, due to combining state 2 with state 0 in the reduced-state decoder, there still exist other error events with the same distance. These additional error events are shown in figure 6.17.b.

Following the error analysis in appendix B, the probabilities of events shown in figures 6.17.a and b can be calculated as $(3/4)^n Q(1/(3\sqrt{2}\sigma))$ and $(1/2)\,(3/4)^n Q(1/(3\sqrt{2}\sigma))$, respectively. Here, similar to the appendix, $n = N_e - N$ represents the length of the error sequence corresponding to the error event with the length of N_e. SER of the full-state and reduced-state decoders can be calculated by employing (B.2) and considering the fact that each minimum-distance error

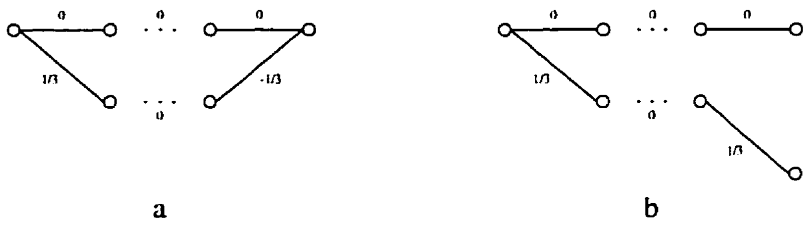


Figure 6.17: Minimum-distance error events in the full-state (a) and reduced-state (a and b) quaternary-dicode decoders.

event entails n symbol errors. This yields

$$SER(MLSD) = 24Q\left(\frac{1}{\sqrt{5}}10^{\frac{SNR}{20}}\right) \tag{6.19}$$

$$SER(RSSD) = 39Q\left(\frac{1}{\sqrt{5}}10^{\frac{SNR}{20}}\right) \tag{6.20}$$

in which (2.10) has been employed to express the results in terms of SNR.

Pre-coding, used in systems with DFE detection approach, is useful in reducing the SER even with MLSD when multi-level signals are employed [12]. This technique applied to the quaternary scheme, results in error events with one incorrect symbol at the beginning and another incorrect symbol at the end of the event [118]. Consequently, (6.19.a and b) in the presence of pre-coding reduce to

$$SER(MLSD) = 12Q\left(\frac{1}{\sqrt{5}}10^{\frac{SNR}{20}}\right) \tag{6.21}$$

$$SER(RSSD) = 18Q\left(\frac{1}{\sqrt{5}}10^{\frac{SNR}{20}}\right) \tag{6.22}$$

The above equations clearly show that the SER performance of the RSSD decoder is only 62.5% worse than that of the MLSD decoder. Furthermore, in the presence of pre-coding this degradation drops to 50% (a minor decrease). More sensitivity of the reduced-state decoder to error propagation due to the decision-feedback mechanism accounts for this decrease. However, this error-propagation effect is negligible, as expected.

Figure 6.18 illustrates the noise performances of the full-state and reduced-state decoders, obtained from simulations, as well as theory. In simulating the RSSD, the architecture shown in figure 6.16 was used. The results are shown both in the presence and absence of pre-coding. From this figure, it can be seen that the reduced-state decoder presented here follows the theoretical predictions, causes a negligible degradation in the coding gain compared to the full-state decoder, and has a performance equivalent to that of the algorithm described in [118].

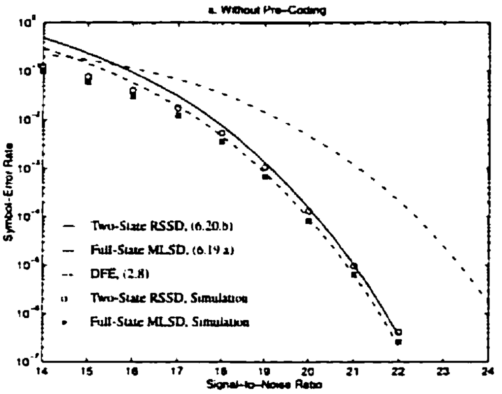


Figure 6.18: Noise performances of the quaternary dicode RSSD and MLSD decoders, obtained from theory and simulations. That of the DFE ia also shown. The performances are plotted in the absence (a) and presence (b) of pre-coding. Note that (6.20) and (6.21) are not accurate at low SNR.

6.4. Summary

Reduced-state sequence detection is a sub-optimum detection technique which reduces the implementation complexity of the maximum-likelihood sequence detector at the expense of some penalty in its noise performance. The reduction in complexity is achieved by decreasing the number of states of the trellis diagram. A decision-feedback mechanism is then incorporated to prevent total loss of information. This mechanism causes the decoder to perform as a hybrid between a full-state sequence detector (corresponding to the full-state trellis) and a decision-feedback equalizer (corresponding to the single-state trellis). From an implementation stand-point, the realization problem of the reduced-state sequence detector can be answered by demonstrating its connection with the decision feedback equalizer, from one side, and the maximum-likelihood sequence detector, from another side.

In this chapter, after investigating the aforementioned connections, it was shown that by properly exploiting the decision-feedback mechanism and employing it in the Viterbi algorithm, the goal of implementing the decoder at the desired reduced complexity can be achieved.

Being a suitable scheme for high-rate data transmission over band-limited channels, quaternary class-IV drew most of our attentions in this chapter. Since this scheme results in a relatively complex trellis diagram, sub-optimum decoders are particularly useful in this case. Also, to obtain a high-speed low-complexity decoder, analog implementations are strong alternatives. As a result, the Viterbi algorithm for decoding a quaternary class-IV signal with a reduced-state trellis was developed from an analog implementation point-of-view and an analog architecture for realizing the reduced-state algorithm was proposed. The goal was to extend the existing binary-decoder structure to the quaternary system by applying the difference-metric algorithm, or more appropriately the input-interleaved algorithm described in chapter 4, to the reduced two-state trellis of the quaternary system. Although the preliminary motivation has been an analog implementation, the results can alternatively be used in digital realizations as well.

Chapter 7

Conclusions and Future Directions

Partial-response signaling, first proposed for data transmission, has now found more applications in other areas such as magnetic recording. Consequently, sequence-detection techniques, and in particular the Viterbi algorithm, for detecting a partial-response signal in a noisy environment has drawn a lot of attention. The challenge is to realize the Viterbi algorithm such that the compatibility with other parts of the system is maintained. These usually include small size, low power, and high speed in today's most, if not all, applications. Magnetic recording and miniature data transceivers are two important examples in which the aforementioned requirements are essential.

Toward realizing the Viterbi algorithm in an efficient way, analog implementations have shown to be viable alternatives to their traditional digital counterparts. The key idea is to eliminate the power-hungry analog-to-digital converter at the front-end of the receiver. However, other savings may be accompanied if the algorithm is carefully examined from an analog-implementation perspective.

Described in this thesis was another attempt for realizing the Viterbi algorithm in the analog domain. Partial-response sequence detectors were of special interest, however, other applications were addressed as well. Beside chapter 1, which was an introduction to the work, a brief summary of what was covered in this thesis is presented in the next section. Some potential areas of research and future directions are then suggested.

7.1. Conclusions

In chapter 2, the partial-response concept, modeling, and applications were explained. The symbol-by-symbol detection technique based on decision feedback equalization for detecting a noisy partial-response signal was reviewed and an approximate approach for the error analysis, which takes into account the effects of error propagation, was introduced. The approach applied to binary and quaternary class-IV partial-response schemes was verified by simulations.

Maximum-likelihood sequence detection of partial-response signals was considered in chapter 3. In particular, a general difference-metric Viterbi algorithm applied to a two-state trellis diagram was developed. From this algorithm, a new derivation of the difference-metric algorithm for decoding a dicode signal was proposed. The proposed algorithm was named the input-interleaved algorithm, since it was shown in chapter 4 to result in a very fast and efficient analog realization with an interleaved input structure. Also introduced in chapter 3, was the maximal-distance approach to the partial-response codes. It was shown that if the coding polynomial of the system satisfies some certain properties, the loss in the signal-to-noise ratio can be asymptotically recovered by the sequence detector. This loss is due to the increased number of levels and can not be combatted by the symbol-by-symbol detector. Implementation issues such as path-memory truncation, signal limiting, and computational accuracy were discussed at the end of the chapter.

Chapter 4 described one of the most important contributions of the current thesis. Two novel analog architectures for realizing a class-IV partial-response Viterbi decoder were introduced. These were the adaptive-threshold architecture, based on the adaptive-threshold interpretation of the decoding algorithm, and the input-interleaved architecture, based on the new input-interleaved algorithm. The robustness of these structures to analog imperfections such as offsets and mismatches was investigated. Two actual realizations of these decoders were reported. A discrete prototype, based on the first architecture, and an integrated decoder, based on the latter architecture, were constructed and tested. The experimental results confirmed the validity of the approaches and their feasibility of implementation. The integrated decoder was fabricated in a $0.8\mu m$ BiCMOS process in a $0.5mm^2$ silicon area and was shown to be able to decode a $200MSymbols/s$ class-IV signal with a total power consumption of $30mW$ drawn from a $3.3V$ single power supply.

To extend the idea of analog realizations to the Viterbi algorithms for which simplifications of the sort of the difference-metric algorithm are not applicable, a novel implementation technique was introduced in chapter 5. The technique was shown to be quite general, covering a variety of other applications such as convolutional coding and M-ary digital communication. Two partial-response decoders were designed and fabricated in a $0.8\mu m$ BiCMOS process. A prove-of-concept dicode decoder was chosen to illustrate the feasibility of the approach. Also, an extended partial-response scheme was fabricated to demonstrate the extendibility of the approach to relatively complicated sequence decoders. The latter scheme was chosen as it was predicted to find its first application in the new generation of computer hard disks. The experimental results of the dicode decoder were presented, confirming the feasibility of the approach. The second decoder has not been tested yet due to test-equipment limitations. It should be mentioned that digital path memories were not included on the chip for these decoders to save design time. As a result, speeds up to $80Mb/s$ were achieved. Simulations, however, show that much higher speeds are attainable if the path memory is included on the chip.

Since the work presented in chapter 4 could also be applied to some reduced-state sequence detectors, chapter 6 was devoted to the implementation issues of these sub-optimum detectors. The complexity reduction compared to the maximum-likelihood sequence detector was investigated by fully exploiting the decision feedback mechanism in the decoder. This was illustrated through examples, where it was shown how the analog sub-optimum decoders benefit these exploitations. Finally, a quaternary class-IV partial-response scheme and its reduced-state decoder (which in fact had partly motivated the work) were explained as the major part of this chapter. A new algorithm, shown to be suitable for an analog implementation, was developed. Furthermore, the algorithm was shown to be equivalent to an existing one, but, less complex. It should be pointed out here that the full-state decoder for this signaling scheme could be alternatively implemented following the general approach introduced in chapter 5.

In conclusion, it was shown through system-level and circuit-level analysis and simulations as well as experiments that analog approaches are not only feasible in implementing digital sequence detectors, but they outperform their digital counterparts in a variety of applications. Although the primary motivations were applications which do not demand a sophisticated signal processing prior to detection, the simplicity of the analog decoder may compensate for the possible increase in the complexity of the preceding circuitry if the idea of analog decoders is extended beyond these applications.

7.2. Future Directions

In addition to searching for new applications where analog sequence detectors are beneficial, more tightly related subjects can be proposed as extensions to the work presented in this thesis. These cover a variety of system-level and circuit-level aspects of which some are briefly addressed below.

CMOS implementations of the implemented decoders (chapters 4 and 5) are desired by many industrial companies. Lower cost and more compatibility with the existence of digital circuits and processes are the most important advantages over BiCMOS implementations. Nevertheless, the leading company in this area employs an advanced BiCMOS processes for state-of-the-art production.

In addition to being motivated by the existence of a real application, a complete circuit realization for the reduced-state quaternary class-IV decoder discussed in chapter 6 seems to be very helpful in demonstrating the extendibility of the analog realization concept, and its advantages, to cases where only digital realizations have been considered for so far. This can be considered as a major step toward exploring more applications for analog sequence detectors.

Applying the automatic-layout generation tools to the implementation approach presented in chapter 5 facilitates the generation of the layout and post-simulation of the analog decoders and biases the industry more toward using the analog-realization approaches in their future products.

Soft-output Viterbi decoders are being actively pursued in many applications. However, due to implementation-complexity issues, these decoders have not become very popular so far. Generating soft decisions which contain more information than an ordinary Viterbi decoder at its output, a soft-output Viterbi decoder is perhaps an even more likely candidate for an analog implementation. Preliminary investigations by the author show the feasibility of the idea, although no published work was found in this open area.

A real maximum-likelihood detector results only if the received signal undergoes the sequence detection which is matched to the encoder. Although an intermediate equalization is essential in compensating any mismatches, it enhances the noise at the same time. By properly absorbing the equalizer (even partly) into the sequence detector, the amount of the enhanced noise can be minimized (This can be achieved by adjusting the polynomial coefficients in a partial-response scheme, for example.). The implementation technique of chapter 5 seems to be flexible enough to let the designer combine a part of the equalizer with the sequence detector. Apparently, using a decoder with a larger number of states improves this flexibility as more degrees of freedom will be available in programming the sequence detector.

Another step toward combining the equalizer and sequence detector is to exploit the flexibility of adjusting the sequence detector to realize an adaptive Viterbi decoder. This can be accomplished by employing the aforementioned programmable sequence detector in a feedback loop. Magnetic read channels are good potential applications for adaptive Viterbi decoders. However, many other applications may be found as well.

Appendix

Error Analysis of a Partial-Response System

A. Symbol-by-Symbol Detection

A PRS system with a DFE-based symbol-by-symbol detector is shown in figure 2.9. In this figure, if the N-previous samples of the input signal and their detected values are denoted by $x(k-1), \ldots, x(k-N)$ and $\hat{x}(k-1), \ldots, \hat{x}(k-N)$ respectively, one can easily show that

$$z(k) = x(k) + \frac{1}{h}n(k) + e(k) \tag{A.1}$$

where

$$e(k) = \sum_{i=1}^{N} f_i(x(k-i) - \hat{x}(k-i)) \tag{A.2}$$

reflects part of the noise contributed by error propagation.

Assuming equi-probable equally-spaced input symbols, straight-forward analysis yields

$$SER = 2\frac{M-1}{M}P\left(\frac{1}{h}n(k) + e(k) > \frac{\Delta}{2}\right) \tag{A.3}$$

for the symbol-error rate of the detector. Here,

$$\Delta = \frac{2}{M-1} \tag{A.4}$$

is the interval between two adjacent levels at the output of the slicer.

A lower bound on the symbol-error rate can be found by ignoring the error propagation (i.e. setting $e(k) = 0$). This bound can be expressed by

$$SER = 2\frac{M-1}{M}Q\left(\frac{h\Delta}{2\sigma}\right) \tag{A.5}$$

where $Q(\ldots)$ is the cumulative Gaussian distribution function defined by [9]

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-x^2/2}dx \tag{A.6}$$

and σ^2 is the variance of the channel noise.

B. Sequence Detection

In the detection of a sequence, errors always occur in groups. The optimum path diverges from the actual path upon the occurrence of the first error of the group. Depending on the trellis diagram, the detected path can merge with the actual path only after a minimum number of transitions between states. In an N'th-order PRS system, if the number of transitions during each error event is N_e $(N_e > N)$, then the actual length of the error sequence is $n = N_e - N$, since the last N detected transitions have to be correct. Associated with each error event, three sub-events can be identified. The probability that a particular error event occurs, can then be described in terms of the probability of each one of these sub-events.

The first sub-event is that the optimum path is equal to the actual path at the time the first error of the error event occurs. The probability of this sub-event is not easy to calculate, however, since the probability that this sub-event is not true is of the order of the probability of error, it can be approximated by unity.

The second sub-event is that if the error sequence is added to the input sequence, the result should be an allowable sequence. In an M-ary PRS scheme with independent equi-probable equally-spaced (with a separation of Δ) inputs, if the i'th element of the error sequence, ε_i, is equal to $\pm i\Delta$, only $M - |i|$ values of the corresponding input are permissible. As a result, the probability of the second sub-event is equal to $\prod_{N_e - N} (M - |i|)/M$, where the multiplication spans over the entire error sequence.

The third sub-event occurs if the noise terms added to the signal, over the length of the error event, are strong enough to make the wrong estimate more likely than the actual sequence. With independent and identical Gaussian noise components of variance σ^2, the probability of this sub-event is equal to $Q(d/(2\sigma))$, where d is the Euclidean distance between the transmitted signal and its estimate and $Q(\ldots)$ is given by (A.6).

The probability of a particular error event can be computed by combining the above arguments. The result is

$$P_{ee} = \left(\prod_n \frac{M - |i|}{M}\right) Q\left(\frac{d}{2\sigma}\right) \tag{B.1}$$

To calculate the symbol-error probability, one should note that among the different error events those which provide the minimum Euclidean distance, d_{min}, are the most-likely ones and usually dominate the error-rate expression at moderate-to-high SNR. This expression can be derived by discarding all the other events and weighting each minimum-distance error event, ε_{min}, by the number of symbol errors it entails, $w(\varepsilon_{min})$. The result is the following upper bound which, incidentally, is rather tight at high SNR

$$SER = \sum_{\varepsilon_{min}} (w(\varepsilon_{min}) \prod_n \frac{M - |i|}{M}) Q\left(\frac{d_{min}}{2\sigma}\right) \tag{B.2}$$

Here, the summation takes place over all error events with Euclidean distances equal to d_{min}.

References

[1] P. Kabal and S. Pasupathy, "Partial-Response Signaling," *IEEE Trans. on Commun.*, Vol. 23, No. 9, pp. 921-934, Sep. 1975.

[2] A. Lender, "The Duobinary Technique for High-Speed Data Transmission," *IEEE Trans. on Commun. and Electron.*, Vol. 82, pp. 214-218, May 1963.

[3] E. R. Kretzmer, "Generalization of a Technique for Binary Data Communication," *IEEE Trans. on Commun. Technol.*, Vol. 14, No. 1, pp. 67-68, Feb. 1966.

[4] H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *AIEE Trans.*, Vol. 47, No. 2, pp. 617-644, Apr. 1928.

[5] J. G. Proakis and M. Salehi, "Communication Systems Engineering," *Prentice Hall*, 1994.

[6] W. R. Bennett, "Introduction to Signal Transmission," *McGraw-Hill*, 1970.

[7] P. P. Vaidyanathan, " Multirate Systems and Filter Banks," *Prentice Hall*, 1993.

[8] E. A. Lee and D. G. Messerschmitt, "Digital Communication," *Kluwer Academic Publishers*, 1994.

[9] P. Z. Peebles, Jr., "Probability, Random Variables, and Random Signal Principles," *McGraw-Hill*, 1980.

[10] R. W. Lucky, J. Salz, and E. J. Weldon, Jr., "Pronciples of Data Communication," *McGraw-Hill*, 1968.

[11] H. Kobayashi, "Correlative Level Coding and Maximum-Likelihood Decoding," *IEEE Trans. on Inform. Theory*, Vol. 17, No. 5, pp. 586-594, Sep. 1971.

[12] G. D. Forney, Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. on Inform. Theory*, Vol. 18, No. 3, pp. 363-378, May 1972.

[13] R. Wood, "Magnetic and Optical Storage Systems: Opportunities for Communications Technology," *Proc. of the IEEE Int'l Conf. on Commun.*, Vol. 3, pp. 1605-1612, 1989.

[14] P. H. Siegel and J. K. Wolf, "Modulation and Coding for Information Storage," *IEEE Commun. Mag.*, Vol. 29, No. 12, pp. 68-86, Dec. 1991.

[15] A. A. Abidi, "Integrated Circuits in Magnetic Disk Drives," *Proc. of the 20'th Eroupean Solid-State Circuits Conf.*, pp. 48-57, Sep. 1994.

[16] R. G. Yamasaki, "Hard Disk Drive Data Signal Processing," *Silicon Systems Inc.*, 1995.

[17] R. A. Baugh, R. F. Chiu, and D. L. Hart, "The Magnetic Recording Channel: Noise, Distortion and Overwrite," *Proc. of the IEEE Int'l Conf. on Commun.*, Vol. 3, pp. 1613-1617, 1989.

[18] W. L. Abbott, J. M. Cioffi, and H. K. Thapar, "Performance of Digital Magnetic Recording with Equalization and Offtrack Interference," *IEEE Trans. on Magn.*, Vol. 27, No. 1, pp. 705-716, Jan. 1991.

[19] C. A. Laber and P. R. Gray, "A 20-MHz Sixth-Order BiCMOS Parasitic-Insensitive Continuous-Time Filter and Second-Order Equalizer Optimized for Disk-Drive Read Channels," *IEEE J. of Solid-State Circ.*, Vol. 28, No. 4, pp. 462-470, Apr. 1993.

[20] J. W. M. Bergmans, "Density Improvements in Digital Magnetic Recording by Decision Feedback Equalization," *IEEE Trans on Magn.*, Vol. 22, No. 3, pp. 157-162, May 1986.

[21] H. Kobayashi and D. T. Tang, "Application of Partial-Response Channel Coding to Magnetic Recording Systems," *IBM J. of Res. and Develop.*, Vol. 14, pp. 368-375, Jul. 1970.

[22] R. Price, et. al., " An Experimental, Multilevel, High Density Disk Recording System," *IEEE Trans. on Magn.*, Vol. 14, No. 5, pp. 315-317, Sep. 1978.

[23] R. W. Wood and D. A. Petersen, "Viterbi Detection of Class IV Partial Response on a Magnetic Recording Channel," *IEEE Trans. on Commun.*, Vol. 34, No. 5, pp. 454-461, May 1986.

[24] F. B. Dolivo, G. Ungerboeck, and T. D. Howell, "Decoding the Output Signal of a Partial-Response Class-IV Communication or Recording Device Channel," *U. S. Pat.*, No. 4,644,564, Feb. 17, 1987.

[25] A. J. Armstrong and J. K. Wolf, " Coded Partial Response Signaling with Peak Detection," *Proc. of the IEEE Globecom Conf.*, Vol. 3, pp. 1782-1785, 1990.

[26] D. G. Messerschmitt, L. C. Barbosa, and T. D. Howell, "A Study of Sampling Detectors for Magnetic Recording," *Proc. of the IEEE Asilomar Conf. on sig., Sys., and Comp.*, pp. 450-455, 1989.

[27] J. Moon and L. R. Carley, "Performance Comparison of Detection Methods in Magnetic Recording," *IEEE Trans. on Magn.*, Vol. 26, No. 6, pp. 3155-3172, Nov. 1990.

[28] H. Shafiee, M. Melas, and P. Sutardja, "Performance Comparison of EPRML and Peak Detection in High Density Digital Magnetic Recording," *IEEE Trans. on Magn.*, Vol. 29, No. 6, pp. 4015-4017, Nov. 1993.

[29] K. Han and R. Spencer, "Comparison of Different Detection Techniques for Digital Magnetic Recording Channels," *IEEE Trans. on Magn.*, Vol. 31, No. 2, pp. 1128-1133, Mar. 1995.

[30] T. D. Howell, et. al., "Error Rate Performance of Experimental Gigabit per Square Inch Recording Components," *IEEE Trans. on Magn.*, Vol. 26, No. 5, pp. 2298-2302, Sep. 1990.

[31] C. Tsang, M. M. Chen, and T. Yogi, "Gigabit-Density Magnetic Recording," *Proc. of the IEEE*, Vol. 81, No. 9, pp. 1344-1359, Sep. 1993.

[32] H. K. Thapar and A. M. Patel, "A Class of Partial Response Systems for Increasing Storage Density in Magnetic Recording," *IEEE Trans. on Magn.*, Vol. 23, No. 5, pp. 3666-3668, Sep. 1987.

[33] W. Schott, "Equalization for Partial-Response Systems," *Proc. of the IEEE Asilomar Conf. on sig., Sys., and Comp.*, pp. 461-465, 1989.

[34] W. L. Abbott and J. M. Cioffi, "Combined Equalization and Coding on the Binary Lorentzian Storage Channel," *Proc. of the IEEE Asilomar Conf. on sig., Sys., and Comp.*, pp. 471-475, 1989.

[35] J. M. Cioffi, et. al., "Adaptive Equalization in Magnetic-Disk Storage Channels," *IEEE Commun. Mag.*, Vol. 28, No. 2, pp. 14-28, Feb. 1990.

[36] Private Communications, *AT&T, Western Digital, and Silicon Systems.*

[37] G. Cherubini, S. Olcer, and G. Ungerboeck, "A Quaternary Partial-Response Class-IV System for 125 Mbit/s Data Transmission over Unshielded Twisted-Pair Cables," *Proc. of the IEEE Int'l Conf. on Commun.*, Vol. 3, pp. 1814-1819, 1993.

[38] "Fiber Distributed Data Interface.", *ANSI X3T9.5/ISO 9314.*

[39] J. J. Horenkamp and J. S. Eudy, "Structured Cabling Systems for the Global Premises Distribution Market," *AT&T Tech. J.*, Vol. 72, No. 2, pp. 41-49, Mar./Apr. 1993.

[40] G. H. Im and J. J. Werner, "Bandwidth-Efficient Digital Transmission up to 155 Mb/s over Unshielded Twisted Pair Wiring," *Proc. of the IEEE Int'l Conf. on Commun.*, Vol. 3, pp. 1797-1803, 1993.

[41] G. H. Im and J. J. Werner, "Bandwidth-Efficient Digital Transmission over Unshielded Twisted-Pair Wiring," *IEEE J. on Selec. Areas in Commun.*, Vol. 13, No. 9, pp. 1643-1655, Dec. 1995.

[42] G. Cherubini, S. Olcer, and G. Ungerboeck, "A Quaternary Partial-Response Class-IV Transceiver for 125 Mbit/s Data Transmission over Unshielded Twisted-Pair Cables: Principles of Operation and VLSI Realization," *IEEE J. on Selec. Areas in Commun.*, Vol. 13, No. 9, pp. 1656-1669, Dec. 1995.

[43] A. N. Coles, D. G. Cunninghum, and S. G. Methley, "100 Mb/s Data Transmission on UTP and STP Cabling for Demand Priority Networks," *IEEE J. on Selec. Areas in Commun.*, Vol. 13, No. 9, pp. 1684-1691, Dec. 1995.

[44] D. A. Johns and D. Essig, "Integrated Circuits for Data Transmission over Twisted-Pair Channels," *Proc. of the IEEE Cust. Integ.-Circ. Conf.*, pp. 5-12, 1996.

[45] Code of U. S. Federal Regulations 47, *FCC*, Part 15-J.

[46] R. W. Chang and J. C. Hancock, " On Receiver Structures for Channels Having Memory," *IEEE Trans. on Inform. Theory*, Vol. 12, No. 4, pp. 463-468, Oct. 1966.

[47] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. on Inform. Theory*, Vol. 13, No. 2, pp. 260-269, Apr. 1967.

[48] G. D. Forney, Jr., "The Viterbi Algorithm," *Proc. of the IEEE*, Vol. 61, No. 3, pp. 268-278, Mar. 1973.

[49] R. E. Bellman, "Dynamic Programming," *Princeton Univ. Press*, 1957.

[50] J. K. Omura, "On the Viterbi Decoding Algorithm," *IEEE Trans. on Inform. Theory*, Vol. 15, No. 1, pp. 177-179, Jan. 1969.

[51] J. F. Hayes, "The Viterbi Algorithm Applied to Digital Data Transmission," *IEEE Commun. Mag.*, Vol. 13, No. 2, pp. 15-20, Mar. 1975.

[52] C. M. Rader, "Memory Management in a Viterbi Algorithm," *IEEE Trans. on Commun.*, Vol. 29, No. 9, pp. 1399- 1401, Sep. 1981.

[53] M. J. Ferguson, "Optimal Reception for Binary Partial Response Channels," *Bell Sys. Tech. J.*, Vol. 51, No. 2, pp. 493-505, Feb. 1972.

[54] P. H. Siegel, et. al., "Exact Bounds for Viterbi Detector Path Metric Differences," *Proc. of the IEEE Int'l on Acous., Speech, and sig. Process.*, Vol. 2, pp. 1093-1096, 1991.

[55] M. H. Shakiba, D. A. Johns, and K. W. Martin, "Analog Implementation of Class-IV Partial-Response Viterbi Detector," *Proc. of the IEEE Int'l Symp. on Circ. and Sys.*, Vol. 4, pp. 91-94, 1994.

[56] M. H. Shakiba, D. A. Johns, and K. W. Martin, "A 200 MHz 3.3 V BiCMOS Class-IV Partial-Response Analog Viterbi Decoder," *Proc. of the IEEE Cust. Integ.-Circ. Conf.*, pp. 567-570, 1995.

[57] J. Fitzpatrick and X. Che, "An Evaluation of Partial Response Polynomials for Magnetic Recording Systems," *IEEE Trans. on Magn.*, Vol. 31, No. 2, pp. 1095-1102, Mar. 1995.

[58] G. D. Forney, Jr., "Convolutional Codes II: Maximum Likelihood Decoding," *Inform. and Cont.*, Vol. 25, No. 3, pp. 222-266, July 1974.

[59] F. Hemmati and D. J. Costello, Jr., " Truncation Error Probability in Viterbi Decoding," *IEEE Trans. on Commun.*, Vol. 25, No. 5, pp. 530-532, May 1977.

[60] D.A. Johns and K. W. Martin, "Analog Integrated Circuit Design," *John Wiley*, 1997.

[61] A. Acampora, "Decoder for Implementing an Approximation of the Viterbi Algorithm Using Analog Processing Techniques," *U. S. Pat.*, No. 4,087,787, May 2, 1978.

[62] A. S. Acampora and R. P. Gilmore, "Analog Viterbi Decoding for High Speed Digital Satellite Channels," *IEEE Trans. on Commun.*, Vol. 26, No. 10, pp. 1463-1470, Oct. 1978.

[63] R. C. Davis, "Diode-Configured Viterbi Algorithm Error Correcting Decoder for Convolutional Codes," *U. S. Pat.*, No. 4,545,054, Oct. 1, 1985.

[64] R. R. Spencer, "A Parallel Architecture for High-Speed Analog Viterbi Detectors," *Proc. of the 33rd Midwest Symp. on Circ. and Sys.*, Vol. 2, pp. 1030-1033, 1990.

[65] T. Suzuki and Y. Saitoh, "A 100Mbps Optical Transmission Experiment Employing a Viterbi Decoder Composed of Analog Circuits," *Proc. of URSI Int'l Symp. on Sig., Sys., and Electron.*, 1989.

[66] R. R. Spencer and P. J. Hurst, "Analog Implementations of Sampling Detectors," *IEEE Trans. on Magn.*, Vol. 27, No. 6, pp. 4516-4521, Nov. 1991.

[67] R. R. Spencer, "Simulated Performance of Analog Viterbi Detectors," *IEEE J. on Selec. Areas in Commun.*, Vol. 10, No. 1, pp. 277-288, Jan. 1992.

[68] T. W. Matthews and R. R. Spencer, "An Analog CMOS Viterbi Detector for Digital Magnetic Recording," *Proc. of the IEEE Int'l Solid-State Circ. Conf.*, pp. 214-215, 1993.

[69] T. W. Matthews and R. R. Spencer, "An Integrated Analog CMOS Viterbi Detector for Digital Magnetic Recording," *IEEE J. of Solid-State Circ.*, Vol. 28, No. 12, pp. 1294-1302, Dec. 1993.

[70] R. G. Yamasaki, et. al., "A 72Mb/S PRML Disk-Drive Channel Chip with an Analog Sampled-Data Signal Processor," *Proc. of the IEEE Int'l Solid-State Circ. Conf.*, pp. 278-279, 1994.

[71] M. H. Shakiba, D. A. Johns, and K. W. Martin, "General Approach to Implementing Analogue Viterbi Decoders," *IEE Electron. Lett.*, Vol. 30, No. 22, pp. 1823-1824, Oct. 27, 1994.

[72] Private Communications, *Silicon Systems*.

[73] T. J. Schmerbeck, et. al., "A 27MHz Mixed Analog/Digital Magnetic Recording Channel DSP Using Partial Response Signaling with Maximum Likelihood Detection," *Proc. of the IEEE Int'l Solid-State Circ. Conf.*, pp. 136-137, 1991.

[74] A. M. Patel, "A New Digital Signal Processing Channel for Data Storage Products," *IEEE Trans. on Magn.*, Vol. 27, No. 6, pp. 4579-4584, Nov. 1991.

[75] C. B. Shung, et. al., "A 30-MHz Trellis Codec Chip for Partial-Response Channels," *IEEE J. of Solid-State Circ.*, Vol. 26, No. 12, pp. 1981-1987, Dec. 1991.

[76] R. D. Cideciyan, et. al., "A PRML System for Digital Magnetic Recording," *IEEE J. on Selec. Areas in Commun.*, Vol. 10, No. 1, pp. 38-56, Jan. 1992.

[77] R. T. Behrens and A. J. Armstrong, "An Advanced Read/Write Channel for Magnetic Disk Storage," *Proc. of the IEEE Asilomar Conf. on sig., Sys., and Comp.*, Vol. 2, pp. 956-960, 1992.

[78] P. A. Ziperovich and J. K. Wolf, "CMOS Implementation of a Viterbi Detector for Hard Disk Drives," *Proc. of the IEEE Cust. Integ.-Circ. Conf.*, pp. 10.3.1-10.3.4, 1993.

[79] R. Philpott, et. al., " A 7 Mbyte/s (65 MHz), Mixed-Signal, Magnetic Recording Channel DSP Using Partial-Response Signalling with Maximum Likelihood Detection," *Proc. of the IEEE Cust. Integ.-Circ. Conf.*, pp. 10.4.1-10.4.4, 1993.

[80] G. J. Kerwin, R. L. Galbraith, and J. D. Coker, " Performance Evaluation of the Disk Drive Industry's Second Generation Partial Response Maximum Likelihood Data Channel," *IEEE Trans. on Magn.*, Vol. 29, No. 6, pp. 4005-4008, Nov. 1993.

[81] T. Sugawara, et. al., "Viterbi Detector Including PRML and EPRML," *IEEE Trans. on Magn.*, Vol. 29, No. 6, pp. 4021-4023, Nov. 1993.

[82] D. R. Welland, et. al., "A Digital Read/Write Channel with EEPR4 Detection," *Proc. of the IEEE Int'l Solid-State Circ. Conf.*, pp. 276-277, 1994.

[83] D. Choi, et. al., "An Analog Front-End Signal Processor for a 64Mb/s PRML Hard-Disk Drive Channel," *Proc. of the IEEE Int'l Solid-State Circ. Conf.*, pp. 282-283, 1994.

[84] W. L. Abbott, et. al., "A Digital Chip with Adaptive Equalizer for PRML Detection in Hard-Disk Drives," *Proc. of the IEEE Int'l Solid-State Circ. Conf.*, pp. 284-285, 1994.

[85] G. T. Uehara, et. al., "A 50 MHz 70mW 8-Tap Adaptive Equalizer/Viterbi Sequence Detector in 1.2μm CMOS," *Proc. of the IEEE Cust. Integ.-Circ. Conf.*, pp. 51-54, 1994.

[86] R. A. Richetta, et. al., " A 16Mbyte/s PRML Read/Write Data Channel," *Proc. of the IEEE Int'l Solid-State Circ. Conf.*, pp. 78-79. 1995.

[87] J. D. Welland, et. al., "Implementation of a Digital Read/Write Channel with EEPR4 Detection," *IEEE Trans. on Magn.*, Vol. 31, No. 2, pp. 1180-1185, Mar. 1995.

[88] Sontag, et. al., " A High Speed, Low power PRML Read Channel Device," *IEEE Trans. on Magn.*, Vol. 31, No. 2, pp. 1186-1195, Mar. 1995.

[89] P. K. D. Pai, et. al., "Analog Front-End Architectures for High-Speed PRML Magnetic Read Channels," *IEEE Trans. on Magn.*, Vol. 31, No. 2, pp. 1103-1108, Mar. 1995.

[90] Private Communications, *Silicon Systems, TI.*

[91] R. G. Yamasaki, "Maximum Likelihood Sequence Metric Calculator," *U. S. Pat.*, No. 5,384,560, Jan. 24, 1995.

[92] B. Razavi and B. A. Wooley, "Design Techniques for High-Speed, High-Resolution Comparators," *IEEE J. of Solid-State Circ.*, Vol. 27, No. 12, pp. 1916-1926, Dec. 1992.

[93] L. Singer, "Analog Circuit Techniques Utilizing BiCMOS Technology," *Proc. of the IEEE Int'l Symp. on Circ. and Sys.*, Vol. 3, pp. 1979-1982, 1990.

[94] P. J. Lim and B. A. Wooley, "An 8-bit 200-MHz BiCMOS Comparator," *IEEE J. of Solid-State Circ.*, Vol. 25, No. 1, pp. 192-199, Feb. 1990.

[95] D. A. Hodges and H. G. Jackson, "Analysis and Design of Digital Integrated Circuits," *McGraw Hill*, 1988.

[96] T. J. Schmerbeck, "Minimizing Mixed-Signal Coupling and Interaction," *Proc. of the 20'th Eroupean Solid-State Circuits Conf.*, pp. 28-37, Sep. 1994.

[97] R. Wood, "Turbo-PRML: A Compromise EPRML Detector," *IEEE Trans. on Magn.*, Vol. 29, No. 6, pp. 4018-4020, Nov. 1993.

[98] R. C. Davis and H. A. Loeliger, "A Nonalgorithmic Maximum Likelihood Decoder for Trellis Codes," *Unpublished.*

[99] B. R. Owen, et. al., "BALLISTIC: An Analog Layout Language," *Proc. of the IEEE Cust. Integ.-Circ. Conf.*, pp. 41-44, 1995.

[100] C. Shung, et. al., "VLSI Architectures for Metric Normalization in The Viterbi Algorithm," *Proc. of the IEEE Int'l Conf. on Commun.*, Vol. 4, pp. 1723-1728, 1990.

[101] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Application," *Proc. of the IEEE Globecom Conf.*, Vol. 3, pp. 1680-1686, 1989.

[102] C. Berrou, et. al., "A Low Complexity Soft-Output Viterbi Decoder Architecture," *Proc. of the IEEE Int'l Conf. on Commun.*, Vol. 2, pp. 737-740, 1993.

[103] O. J. Joeressen, M. Vaupel, and H. Meyr, "Soft-Output Viterbi Decoding: VLSI Implementation Issues," *Proc. of the IEEE Veh. Tech. Conf.*, pp. 941-944, 1993.

[104] O. J. Joeressen and H. Meyr, "A 40Mb/s Soft-Output Viterbi Decoder," *IEEE J. of Solid-State Circ.*, Vol. 30, No. 7, pp. 812-818, Jul. 1995.

[105] M. V. Eyuboglu and S. U. Qureshi, "Reduced-State Sequence Estimation with Set Partitioning and Decision Feedback," *IEEE Trans. on Commun.*, Vol. 36, No. 1, pp. 13-20, Jan. 1988.

[106] S. U. Qureshi and E. E. Newhall, "An Adaptive receiver for Data Transmission over Time-Dispersive Channels," *IEEE Trans. on Inform. Theory*, Vol. 19, No. 4, pp. 448-457, Jul. 1973.

[107] D. D. Falconer and F. R. Magee, Jr., "Adaptive Channel Memory Truncation for Maximum-Likelihood Sequence Estimation," *Bell Sys. Tech. J.*, Vol. 52, No. 9, pp. 1541-1562, Nov. 1973.

[108] F. L. Vermuelen and M. E. Hellman, "Reduced-State Viterbi Decoding for Channels with Intersymbol Interference," *Proc. of the IEEE Int'l Conf. on Commun*, pp. 37.B.1-37.B.4, 1974.

[109] W. U. Lee and F. S. Hill, "A Maximum-Likelihood Sequence Estimator with Decision-Feedback Equilization," *IEEE Trans. on Commun.*, Vol. 25, No. 9, pp. 971-979, Sep. 1977.

[110] G. J. Foschini, "A Reduced-State Variant of Maximum-Likelihood Sequence Detection Attaining Optimum Performance for High Signal-to-Noise Ratio Performance," *IEEE Trans. on Inform. Theory*, Vol. 23, No. 5, pp. 605-609, Sep. 1977.

[111] C. T. Beare, "The Choice of the Desired Impulse Response in Combined Linear-Viterbi Algorithm Equalizers," *IEEE Trans. on Commu*n., Vol. 26, No. 8, pp. 1301-1307, Aug. 1978.

[112] A. P. Clark and M. Clayden, "Pseudobinary Viterbi Detector," *Proc. of IEE*, Vol. 131, Part F, pp. 208-218, Apr. 1984.

[113] A. P. Clark, et. al., "Pseudobinary and pseudoquaternary Detection Processes for Linearly Distorted Multilevel QAM Signals," *IEEE Trans. on Commun.*, Vol. 33, No. 7, pp. 639-645, Jul. 1985.

[114] M. V. Eyuboglu and S. U. Qureshi, "Reduced-State Sequence Estimation with Set Partitioning and Decision Feedback," *Proc. of the IEEE Globecom Conf.*, Vol. 2, pp. 1023-1028, 1986.

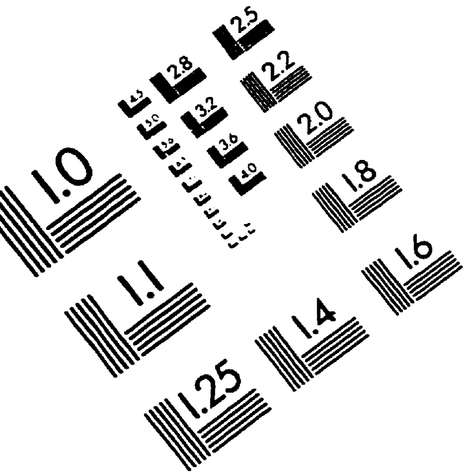
[115] S. Olcer, "Reduced-State Sequence Detection of Multilevel Partial-Response Signals," *IEEE Trans. on Commun.*, Vol. 40, No. 1, pp. 3-6, Jan. 1992.

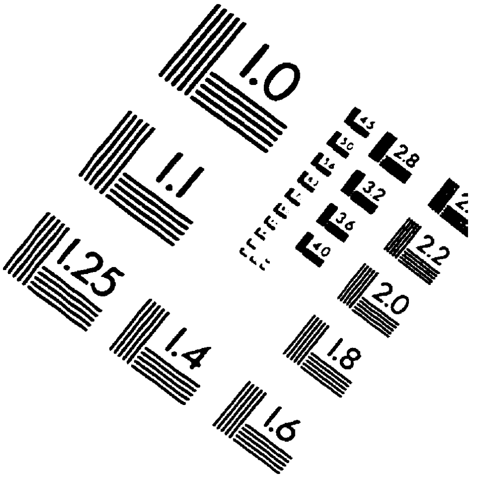
[116] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. on Inform. Theory*, Vol. 28, No. 1, pp. 55-67, Jan. 1982.

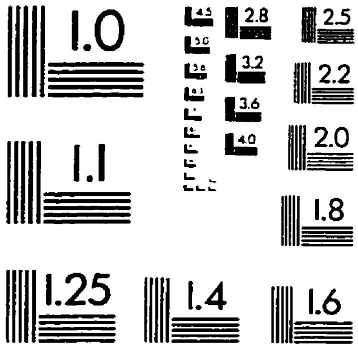
[117] Private Communications, *IBM*.

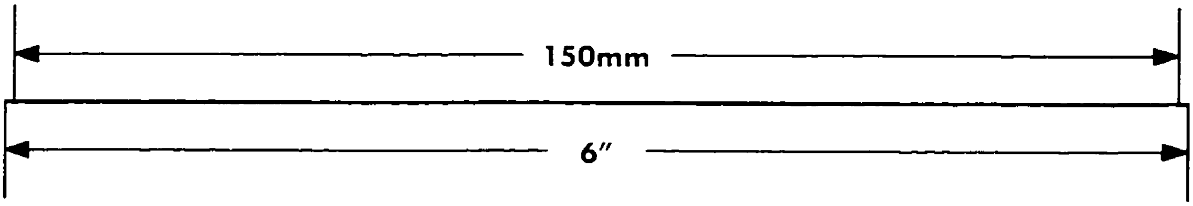
[118] S. Olcer and G. Ungerboeck, "Difference-Metric Viterbi Decoding of Multilevel Class-IV Partial-Response Signals," *IEEE Trans. on Commun.*, Vol. 42, No. 2/ 3/ 4, pp. 1558-1570, Feb./ Mar./ Apr. 1994.

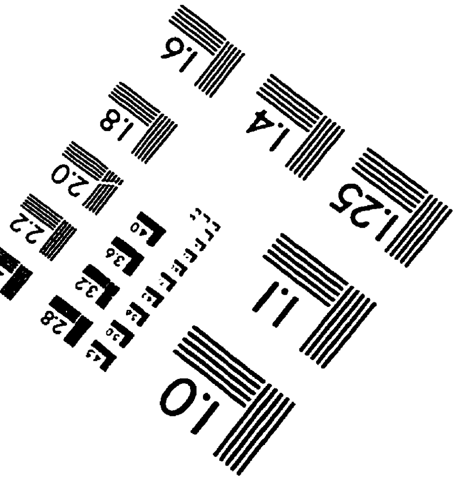
IMAGE EVALUATION TEST TARGET (QA-3)

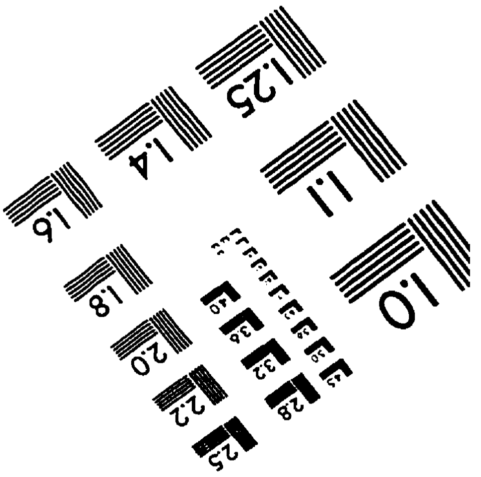












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989