

UNIVERSITY OF CALGARY

**Differential Threshold Reliant Detectors for
Ternary Partial Response Channels**

by

Omole, Ibiyemi Akintomide

A DISSERTATION

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

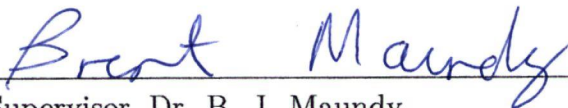
CALGARY, ALBERTA

NOVEMBER, 2003

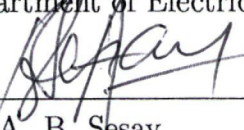
© Omole, Ibiyemi Akintomide 2003

UNIVERSITY OF CALGARY
FACULTY OF GRADUATE STUDIES

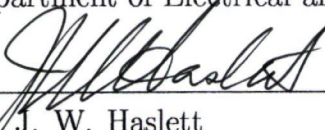
The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a dissertation entitled "Differential Threshold Reliant Detectors for Ternary Partial Response Channels" submitted by Omole, Ibiyemi Akintomide in partial fulfillment of the requirements for the degree of Doctor of Philosophy.



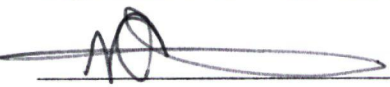
Supervisor, Dr. B. J. Maundy
Department of Electrical and Computer Engineering



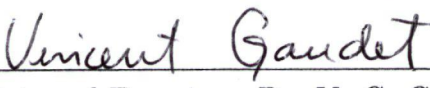
Dr. A. B. Sesay
Department of Electrical and Computer Engineering



Dr. J. W. Haslett
Department of Electrical and Computer Engineering



Dr. N. El-Sheimy
Department of Geomatics Engineering



External Examiner, Dr. V. C. Gaudet
Department of Electrical and Computer Engineering
University of Alberta

19/10/03
Date

Abstract

An in-depth look at the detection problem of both fundamental Partial Response Signaling (PRS) channels; duobinary ($1 + D$) and dicode ($1 - D$) used in a ternary mode is addressed in this dissertation from the Maximum-Likelihood Sequence Detection (MLSD) stand-point. Literature surveys reveal that the ternary partial response signaling channel detector(s) would be very helpful in simplifying detectors for higher order partial response signaling channels.

Literature surveys further show that the detection problem of ternary duobinary and dicode channels have not been efficiently dealt with in either the analog or digital domain and only a handful of research publications exist on the issue.

Mathematical derivation from first principles for a new detection algorithm for both channels from the difference metrics perspective is presented. Analysis, interpretation and characterization of the algorithms in the analog domain were done to provide a better insight to their functionality. The algorithms sensitivity and robustness to undesirable analog effects such as saturation effects was investigated.

The algorithm translation for hardware implementation was carried out and implementation of an architecture was investigated. For proof of concept and also as the first to be reported for the ternary PRS channel detector implementation, a mixed-signal detector circuitry was designed using digital CMOS $0.18\mu m$ technology with a single 3.3V supply and simulation showed a maximum working speed of 100MHz for the design.

The design was fabricated and experimental tests were conducted. Experimental analysis and problems encountered were discussed and the dissertation concludes with future work.

Acknowledgement

My sincere and very profound gratitude goes to my supervisor Dr. Brent Maundy for providing me with the opportunity to be able to pursue my research interest, for always finding time to guide me through the course of the research, and for his patience and support during the time I was still groping for the way forward.

I would also like to thank Dr. Abu Sesay for his immense contributions, suggestions and positive critics to this research. Thanks for believing that a worthwhile research could stem from this topic. You are appreciated.

Also, Dr. Bob Davies (TRLabs) is well deserving of my thanks for helping to provide access to very vital equipments for the testing of my chip. To Simon Arse-neault (TRLabs), I appreciate your help during the course of the experiment. John-Peter Vanzelm, thank you for all your assistance in securing the test space. Balvinder Vardee (Agilent Technologies), your provision of the Infiniium oscilloscope was timely and it is very well appreciated. John Shelley and Angela Morton you are both awesome; thanks for your technical assistance.

My parents, brother and sisters, I love you all for your continual support and prayers. My wife, thanks for being the light in my dark hours. You were always there for me when I was discouraged. Moreover, thank you for being selfless and understanding while my interest got in the way of our life together.

Lastly but not the least, I would like to show my appreciation to Mr. A. A. Akinola, Mr. Tony and Mrs. Idowu Osibodu, Dr. J. B. Olomo (Physics Department, O. A. U. Nigeria), Dr. G. O. Ajayi (Electronic and Electrical Engineering Department, O. A. U. Nigeria), and Dr. C. Papavassiliou (Imperial College, London) for your fatherly advice and timely support.

Dedication

*To my Parents, my wife Olanike
and
my precious daughter Eyilayomi*

Contents

Approval Page	ii
Abstract	iii
Acknowledgement	iv
Dedication	v
Contents	ix
List of Tables	x
List of Figures	xv
List of Symbols	xvi
List of Acronyms	xix
1 Introduction	1
1.1 Background	1
1.2 Trend and Motivation	3
1.3 Research Objectives	4
1.4 Overview	6
2 Literature Review	8
2.1 Partial Response Signaling	8
2.2 The Viterbi Algorithm	10
2.2.1 Difference Metric Approach	12

2.3	EPR4 Channel Detection	13
2.3.1	Wood's Detection Scheme	15
2.3.2	Friedmann's Detection Scheme	15
2.3.3	Knudson's Detection Scheme	17
3	A New Detection Algorithm: The Derivation	18
3.1	A Preview	18
3.2	Ternary Dicode Channel	19
3.2.1	States Survivor Derivation	20
3.2.2	Merger Observations	24
3.2.3	The Updates	26
3.3	Ternary Duobinary Channel	31
3.3.1	States Survivor Derivation	32
3.3.2	Merger Observations	34
3.4	Preliminary Evaluation	36
4	Interpretation and Characterization	42
4.1	An Interpretation	42
4.1.1	Ternary Dicode: Threshold Δm_c	43
4.1.2	Ternary Dicode: Threshold Δm_d	46
4.1.3	Ternary Duobinary: Threshold Δm_a	47
4.1.4	Ternary Duobinary: Threshold Δm_b	48
4.2	Saturation Effect	50
4.2.1	Ternary Dicode: Input Limitation	52
4.2.2	Ternary Dicode: Signal Swing Limitation	55
4.2.3	Ternary Duobinary: Input Limitation	56

4.3	Error Rate Performance	57
4.3.1	Ternary Channel Detection vs. Binary Channel Detection . .	58
4.3.2	New technique vs. Friedmann's technique	61
4.4	Algorithm translation for hardware implementation	64
5	Detector Design and Implementation	68
5.1	Architectures	68
5.2	Detector Design	71
5.2.1	Input Track and Hold circuit	71
5.2.2	Buffer circuitry	74
5.2.3	Level-shifting circuitry	77
5.2.4	The clocked comparator	79
5.2.5	The Cross-over Multiplexing T/H	83
5.2.6	Control Signal Generator	85
5.2.7	Clock Generator	85
5.2.8	Bias Circuitry	86
5.3	Detector simulation results	87
5.3.1	Detector decisions	90
5.4	Survivor Memory Management	94
5.5	Experimental Results	97
5.5.1	Test Signal Generation	97
5.5.2	Chip Test Bench	100
5.5.3	Experimental Data Analysis	102
6	Related Research Work	112
6.1	Loser-Take-All Circuits	112

6.1.1	A New Proposition	113
6.1.2	First Circuit Implementation	114
6.1.3	Second Circuit Implementation	117
6.1.4	Simulation Results	119
6.2	Novel Differential Logic	121
6.2.1	Differential AND/NAND Logic	123
6.2.2	Differential OR/NOR Logic	126
7	Contributions, Future Work and Conclusion	131
7.1	Contributions	131
7.2	Future Work	132
7.3	Conclusion	134
A	Duobinary channel detection	145
A.1	Derivation of updates	145
B	Limitation Effect: Ternary 1-D channel	150
B.1	Positive plane limitation	151
B.2	Negative plane limitation	152
C	Detector Output Validation	154
C.1	Output Logic Combinations	154
C.2	Survivor Memory Management Validation	156
D	Detection Scheme Modifications	159

List of Tables

3.1	Updates for ternary duobinary channel detector	38
4.1	Comparison of ternary dicode channel detection techniques	62
5.1	Logic operations for control signal generator	86
5.2	Comparison of MATLAB and Cadence simulation result	92
5.3	Survivor memory path control logic	96
6.1	Transistors' aspect ratio	117
6.2	Differential AND/NAND logic truth table	124
6.3	Differential OR/NOR logic truth table	126
C.1	Group A: Comparators 1 to 3	155
C.2	Group B: Comparators 4 to 6	155
C.3	Detector valid 6-bit outputs	156
D.1	A sample of region occurrence probability	160
D.2	Logic operations for updated control signal generator	160

List of Figures

2.1	Arbitrary two state trellis	13
2.2	EPR4 channel trellis	14
2.3	Friedmann's form of EPR4 channel coding	16
2.4	Friedmann's EPR4 channel detection technique	16
3.1	Dicode channel FIR representation	19
3.2	Dicode channel trellis	20
3.3	Transitions to state +1 at time time $k+1$ for the dicode channel trellis	21
3.4	Transitions to state 0 at time time $k+1$ for the dicode channel trellis	21
3.5	Transitions to state -1 at time time $k+1$ for the dicode channel trellis	22
3.6	Region-dependency of the thresholds for the dicode channel	27
3.7	Duobinary channel FIR representation	32
3.8	Duobinary channel trellis	32
3.9	Region-dependency of the thresholds for the duobinary channel . . .	37
3.10	Typical decoding example: Ternary dicode channel (<i>Dark thick lines show the trace back path</i>)	39
3.11	Typical decoding example: Ternary duobinary channel (<i>Dark thick lines show the trace back path</i>)	40
3.12	Threshold adaptation: Dicode Channel	40
3.13	Threshold adaptation: Duobinary Channel	41

4.1	Flow-chart: Ternary Dicode Channel Detection	44
4.2	Flow-chart: Ternary Duobinary Channel Detection	45
4.3	Threshold Δm_c adaptation type 1	46
4.4	Threshold Δm_c adaptation type 2	47
4.5	Threshold Δm_d adaptation type 1	48
4.6	Threshold Δm_d adaptation type 2	49
4.7	Threshold Δm_a adaptation type 1	50
4.8	Threshold Δm_a adaptation type 2	51
4.9	Threshold Δm_b adaptation type 1	52
4.10	Threshold Δm_b adaptation type 2	53
4.11	Input limitation effect on ternary dicode channel	54
4.12	Limitation effect: Binary Channel(B) vs. Ternary Channel(T)	55
4.13	Signal swing limitation effect on ternary dicode Channel	56
4.14	Input limitation effect on ternary duobinary channel	57
4.15	Error rate performance: Ternary dicode channel	58
4.16	Error rate performance: Ternary duobinary channel	59
4.17	Error-rate comparison	61
4.18	Threshold adaption comparison	63
4.19	Bit error rate vs. Signal-to-noise ratio	64
4.20	Reference communication system with detector	65
4.21	Linear Input Conditioner	66
4.22	Limitation effect on conditioned input $y(t)'$	67
5.1	Ternary dicode detector architecture	72
5.2	Input Track and Hold Circuitry	73

5.3	Buffer block diagram	74
5.4	Buffer small-signal diagram	74
5.5	Actual Buffer Circuit	75
5.6	Simulated buffer dc response	76
5.7	Simulated buffer transient response @ $V_{in} = 600mV$, frequency 100MHz	77
5.8	Simulated buffer transient response @ $V_{in} = 400mV$, frequency 100MHz	77
5.9	Simulated buffer transient response @ $V_{in} = 200mV$, frequency 100MHz	78
5.10	Level Shifting Circuit	78
5.11	A typical differential pair	80
5.12	Clocked comparator circuit	80
5.13	Comparator small-signal representation	81
5.14	Comparator decision for $0.8V < V_{in} < 3.0V$	83
5.15	Cross-over multiplexing T/H mechanism	84
5.16	Control signal generator	87
5.17	On-Chip clock generator	88
5.18	Bias Circuitry	88
5.19	Simulated signal shift	89
5.20	Simulated threshold movement	90
5.21	Captured comparator outputs	91
5.22	Circuit mapped error rate computation	94
5.23	Input limiting effect within 1.2V and 2.1V	95
5.24	Survivor memory path construction	97
5.25	Chip layout diagram	98
5.26	State diagram for ternary sequence generator	99
5.27	A typical example of ternary sequence generation	99

5.28	The chip test bench	101
5.29	The off-chip level shifter	101
5.30	A sample of acquired signals @ 0dB SNR	103
5.31	Type 1: Initialization imbalance (Threshold Movement)	104
5.32	Type 1: Initialization imbalance(Error Rate)	105
5.33	Type 2: Initialization imbalance (Threshold Movement)	106
5.34	Type 2: Initialization imbalance(Error Rate)	107
5.35	Acquired signal sample 1: y_t and off-chip clock	108
5.36	Acquired signal sample 1: corresponding DUT output (D0-D5), off-chip (AWG) clock (D6) and DUT clock (D7)	109
5.37	Acquired signal sample 2: y_t and off-chip clock	110
5.38	Acquired signal sample 2: corresponding DUT output (D0-D5), off-chip (AWG) clock (D6) and DUT clock (D7)	111
6.1	Block diagram representation of the LTA	115
6.2	First type of CMOS implementation of the LTA	115
6.3	Second type of CMOS implementation of the LTA	118
6.4	Simulated current sweep result of the LTA	120
6.5	Simulation result: first circuit @ 10MHz	121
6.6	Simulation result: first circuit @ 15MHz	122
6.7	Simulation result: second circuit @ 35MHz	123
6.8	Novel AND/NAND differential logic	124
6.9	Simulation result of AND/NAND logic with 0.02pF load	126
6.10	Simulation result of AND/NAND logic with 0.3pF load	127
6.11	Simulation result of AND/NAND logic with 0.5pF load	128
6.12	Novel OR/NOR differential logic	129

6.13	Simulation result of OR/NOR logic with 0.02pF load	129
6.14	Simulation result of OR/NOR logic with 0.3pF load	130
6.15	Simulation result of OR/NOR logic with 0.5pF load	130
D.1	Updated Region-dependency of thresholds for the dicode channel . . .	159
D.2	Updated Region-dependency of thresholds for the duobinary channel	161
D.3	Updated Flow-chart for the dicode channel detection	162
D.4	Updated Flow-chart for the duobinary channel detection	163
D.5	Updated ternary dicode detector architecture	164
D.6	Updated cross-over multiplexing T/H	165
D.7	Typical timing diagram for feed-forward and cross-over transmission gates	165
D.8	Logic diagram for feed-forward and cross-over control signals	166

List of Symbols

L	Number of partial-response signaling channel output level
D	Unit time delay
Δm	Metric difference
m_x	State metric
k	Time instance
T_n	Threshold
S_m	Shifts
γ	A constant
z^{-1}	Unit time delay in the z-domain
w_n	Tap weight
s_k	Uncoded source symbol
x_k	Coded transmitted symbol
n_k	Noise sample
y_k	Unbounded received noisy symbol sample
$y(t)$	Unbounded received noisy symbol
$y(t)'$	Conditioned received noisy symbol
S_{+1}	Trellis State +1
S_0	Trellis State 0
S_{-1}	Trellis State -1
S_t	Number of states in a channel trellis
Δm_a	Difference threshold A
Δm_b	Difference threshold B
Δm_c	Difference threshold C
Δm_d	Difference threshold D

L_y	Input limitation level
d_{min}	Minimum distance
σ_n	Noise variance
$Q(.)$	Cumulative Gaussian distribution
ξ	Friedmann's thresholds separation
α	Additive constant
β	Normalizing constant
C_{ox}	Transistor's gate oxide capacitance
W/L	Transistor's aspect ratio
g_m	Transconductance
g_o	Conductance
τ	Time constant
ϕ_1, ϕ_2	Clock phase
z	Truncation length
$\mu_{n/p}$	Electron/Hole mobility
C_{1-6}	Comparator output 1 to 6
S_{1-7}	Select signal 1 to 7
A_x	Region in ternary dicode detector
B_x	Region in ternary duobinary detector
$1 - D$	Dicode channel
$1 + D$	Duobinary channel
$1 - D^2$	Class-IV partial-response channel
V_{max}	Maximum signal swing level
V_{min}	Minimum signal swing level
V_{ref}	Reference voltage
V_{cm}	Common-mode level

\bar{P}_{symbol}	Symbol error probability
V_{th}	Threshold voltage
V_{dd}	Positive power supply
V_{in}	Input voltage
V_{out}	Output voltage
V_{ind}	Differential input voltage
C_{hold}	Holding capacitance
A_{gain}	DC gain
μm	micrometer
μF	microfarad
μA	microampere
mV	millivolts
$K\Omega$	kiloohms

List of Acronyms

PRS	Partial Response Signaling
ISI	Inter-symbol Intereference
MLSD	Maximum-Likelihood Sequence Detection
VA	Viterbi Algorithm
VD	Viterbi Detector
PR4	Class-IV Partial-Response Signaling
EPR4	Extended Class-IV Partial-Response Signaling
E^2 PR4	Double Extended Class-IV Partial-Response Signaling
PRML	Partial Response Maximum Likelihood
CMOS	Complementary Metal Oxide Semiconductor
BiCMOS	Bipolar-Complementary Metal Oxide Semiconductor
MAP	Maximum a posteriori probability
ACS	Add-Compare-Select
FIR	Finite Impulse response
SNR	Signal to Noise Ratio
BER	Bit Error Rate
dB	Decibel
A/D	Analog to Digital Converter
CMC	The Canadian Microelectronics Corporation
Mb/s	Megabit per second
T/H	Track and Hold
CLK	Clock
nMOS	n-type Complementary Metal Oxide Semiconductor
pMOS	p-type Complementary Metal Oxide Semiconductor

SMM	Survivor Memory Management
TSMC	Taiwan Semiconductor Manufacturing Company
I/O	Input/Output
AWGN	Additive White Gaussian Noise
AWG	Arbitrary Waveform Generator
DUT	Device Under Test
MUX	Multiplexer
OPA	Operational Amplifier
MSO	Mixed Signal Oscilloscope
DSO	Digital Storage Oscilloscope
PC	Personal Computer
CSV	Comma Separated Variables
LTA	Loser-Take-All
WTA	Winner-Take-All
RSSD	Reduced State Sequence Detection
SOVA	Soft Output Viterbi Algorithm

Chapter 1

Introduction

1.1 Background

A remarkable growth was triggered by a widely known detrimental, yet “obscurely advantageous” phenomenon, when it was proposed as being applicable to a type of digital communication system; specifically the magnetic recording system. Unapparent was the immense benefit that would be derived from this proposition. Virtually unknown as well to groping researchers was the reconciliation of rather under utilized ideas of previous decades with a relatively new but creative work of the time in history, which the proposition would eventually bring in years to come. Prior to the year 1970, it would have sounded more of an in-ordinate curiosity than an intellectual, methodical, and logical line of thought; awaiting time to unravel the imperative reality of the tremendous wave of change that the work of Kobayashi and Viterbi brought to the world of digital magnetic recording communication systems.

The previous paragraph is somewhat of a descriptive situation of the state of the magnetic recording community over 30 years ago. Primitive was the state of, and

stunted was the progress made on, modelling magnetic recording channel(s) accurately and developing suitable detector(s) (high error correcting device(s)) for such a system that would only tolerate a low error rate level in real applications.

In the process of developing efficient communication systems, one of the major or notable problems that could limit the functionality of the system is the overlapping of bits or smearing of pulses in a corruptive channel. This detrimental phenomenon is usually referred to as Inter-symbol interference (ISI). Inter-symbol interference has been studied over the years and researchers have proposed ways and means of eliminating or minimizing ISI and/or its effects. It is known that ISI is a direct consequence of the corruptive nature of communication channels, since it occurs due to the amplitude and phase distortion of transmitted symbols in the channel. Also known is the fact that it limits the transmission rate in high-data-rate communication systems because it becomes pronounced with increasing distance as well as frequency. Since it seems unavoidable due to the non-existence of perfect communication channels, measures to restrain ISI were studied several decades ago. Most notable of the research work done on this issue is that of Nyquist in 1928 [1]; on how to restrict the bandwidth requirement of a communication channel while keeping ISI at bay.

Although not exactly the topic of discussion in this dissertation, it is pertinent to state that Nyquist's methods of controlling ISI served as a precedence for other techniques used to combat ISI in a communication system. Specifically, the focus of this dissertation will be on the Partial-Response Signaling (PRS) technique otherwise referred to as Correlative Level Coding, which is the technique that was revolutionary in the art of data storage in magnetic recording systems; and in present times is finding application in other types of data communication systems.

1.2 Trend and Motivation

The era of Elias [2], Lender [3, 4], Kretzmer [5], and Gerrish [6], will remain indelible in the world of data communication for introducing the concept of signal coding, which utilizes controlled ISI in a way that the receiver could eventually cancel. This attractive concept is called Partial-Response Signaling (PRS).

By using the PRS technique, the resulting coded output signal is spectrally shaped allowing a direct concentration of signal energy at specific frequencies of interest [7]. In this way, some level of immunity to noise and distortion is provided [8]. Although not immediately obvious at the time of inventing the technique, partial-response signaling was noted several years later, as being applicable to magnetic recording channels to achieve higher storage density [9]. Simply put, this is made possible because the technique inherently allows a controlled ISI and thus, it is possible to stuff bits closely to each other in a defined manner instead of having bit spacing that minimizes bit interference.

Since the inception of the PRS technique, there have been several classes of PRS polynomials of varying attributes [6, 8]. However, there are only two fundamental PRS polynomials namely; *duobinary* or class I, and *dicode* polynomial. All other PRS polynomials can be seen as a higher order combination of these polynomials. A common “denominator” to all the PRS polynomials is that they all yield a multi-level coded output signal with level L , which depends on the m -ary input. The various classes of PRS polynomial provide a varying degree of approximation to the modeling of a physical magnetic recording channel as well as a varying achievable storage capacity. Usually, the higher the order, the better the approximation, with resulting higher storage capacity, at the expense of more complex detection [10].

Due to the effect of channel non-idealities such as noise, recovery of the coded signals in PRS channels could be cumbersome. A sub-optimal technique used in the data recovery process from these channels is the bit-by-bit threshold detector, which invariably results in an abysmally low error detection. The increasing quest to improve on the detection technique has brought about the application of the Maximum-Likelihood Sequence Detection (MLSD) technique to the partial-response channel detection problem [7, 11]. In 1971 and 1972, it was shown that MLSD would provide optimum decoding for correlative coded signals [7, 12].

In 1967, Viterbi proposed a revolutionary algorithm, which has now become widely accepted as the most practical means of implementing the MLSD concept [13, 14]. Since then, interest has been brewing in devising means of interpreting and implementing the Viterbi algorithm (VA). Shortly afterwards, Forney and Ferguson [12, 15] independently provided two different interpretations of the classical Viterbi algorithm. Retrospectively, the latter could be perceived as somewhat addressing the shortcoming(s) of the former [12, 15].

Ever since, there has been tremendous application of the Viterbi algorithm to various PRS channel detection problems. Consequently, the race to develop efficient, fast and cost effective Viterbi detectors was started.

1.3 Research Objectives

Despite the applicability of several PRS polynomials as viable models for a physical magnetic recording channel, the class-IV PRS polynomial (PR4) seems to be the dominant choice; partly due to the relative simplicity with which the detector could be implemented for moderate density requirements. Another reason for this as well,

is the fact that the four-state class-IV (PR4) model could be achieved by directly time-interleaving two, two-state, binary duobinary channels. However, for very high-density requirements, the class-IV model is generally insufficient. In this case, a more appropriate solution can be found using the extended class-IV (EPR4) polynomial or the double extended class-IV (E^2PR4) polynomial; which are respectively an order, and two higher than the PR4 model. Nevertheless, the detection of the E^2PR4 model is very complex in comparison to the EPR4 model, since it has a higher number of states; hence, the preferred choice is the EPR4 model for very high-density magnetic recording.

Although it is preferred for very high-density recording, the number of states in the EPR4 channel model is double that which is found in the PR4 channel model. Hence, its detection is more involved than that of the PR4 model. Moreover, in contrast to the four-state PR4 channel that can be directly time-interleaved, the eight-state EPR4 channel cannot be directly time-interleaved; invariably asking for some sort of simplification that could be carried out on the EPR4 model to ensure the feasibility of implementation using the maximum-likelihood sequence detection principle efficiently.

The simplification requirement for the EPR4 channel has drawn the attention of a wide variety of researchers. However, only Wood and Friedmann's proposition on the simplification of the EPR4 channel detection seems to have been widely embraced [16, 17]. The two proposed techniques theoretically attain EPR4 detection simplification using either of the two fundamental partial-response signaling polynomials (*duobinary* and *dicode*), in a binary or ternary mode. Using either of the duobinary or dicode channels in a ternary mode would warrant an extra effort in detection compared to the popular two-state binary duobinary or dicode channel detection. Thus, only an efficient detection scheme for these ternary partial-response signaling channels

would be acceptable; of course based on the maximum-likelihood sequence detection principle.

A review of the literature on this topic has revealed the need for an effective ternary partial-response channel detection scheme, while there are a whole lot of propositions and implementations of binary partial-response channel detectors [15, 18, 19, 20, 21, 22, 23, 24, 25, 26]. Moreover, the only existing piece of work on the detection of the ternary dicode channel [27] seems to be predominantly theoretical, with major emphasis on the relevant signal processing.

Thus, it is the major objective of this thesis research to delve more into the detection of both the ternary dicode and duobinary channel with the ultimate goal of presenting an efficient detection scheme based on the maximum-likelihood sequence detection principle.

The major objective would be dealt with by:

- applying the difference metric approach to the development of an efficient detection algorithm for ternary PRS channels.
- proposing and implementing possible design architecture(s).
- validating concepts presented via physical testing.

1.4 Overview

This section elucidates the arrangement of the remaining contents of this dissertation. Chapter two provides some insight into the partial-response signaling technique as well as the maximum-likelihood sequence detection in the form of the Viterbi algorithm.

It further explains the extended class-IV PRS channel and the proposed detection simplification schemes.

Chapter three focusses on the two fundamental PRS polynomials; duobinary and diode polynomial. A complete derivation of a new detection scheme is presented for these two channels, from first principles.

Chapter four shifts the focus from the signal processing involved in the new proposed detection scheme and investigates some relevant issues about the proposed scheme itself. Characterization of the ternary PRS channels in terms of error-event occurrence is discussed. It also investigates the feasibility of the hardware implementation of the new detector. The results of MATLAB simulations of the new detection scheme are presented.

Chapter five discusses possible architectures for the detector implementation and emphasizes the design of the detector in CMOS $0.18\mu m$ technology. The schematic designs are discussed and simulation results are presented. Experimental results on the fabricated detector are also presented along with a discussion on the generation of the test vector and the test bench.

Other related research works carried out are presented and discussed with the obtained results in chapter six and the dissertation is finally concluded in chapter seven with a brief summary of the contributions made, followed by an insight into possible future work, and the relevant conclusion is drawn.

Chapter 2

Literature Review

2.1 Partial Response Signaling

It is generally known that communication channels are imperfect, usually exhibiting some amplitude and phase distortion, which causes the smearing of transmitted bits. The phenomenon of bit smearing or pulse spreading resulting in un-warranted symbol overlap is widely known as Inter-symbol interference (ISI). ISI generally limits the transmission rate in a communication system.

In an effort to combat ISI while efficiently utilizing the bandwidth in a communication channel, Nyquist in his work in 1928, proposed a zero-memory communication model, that could achieve high data rate [1]. In his work, Nyquist assumed that the transmitted symbols are independent and the noise at the sampler (receiver end) is uncorrelated, such that each symbol can be recovered without the knowledge of the signal history.

However, there are several drawbacks to the Nyquist's approach of combating ISI

using zero-memory systems, namely:

- it has a high degree of perturbation intolerance.
- zero ISI condition is too restrictive.
- independency requirement on transmitted symbol is stringent.
- theoretical pulse shaping filter is impractical, while practical pulse-shaping requires some excess bandwidth.

These drawbacks invariably led to the discovery of the partial response signaling (PRS) technique.

Invented in 1963 by Lender [4], the PRS technique was developed to accommodate pulse amplitudes selected dependently by relaxing the zero ISI criterion of the Nyquist system. It actually utilizes the destructive ISI in a channel positively to efficiently use the available channel bandwidth. It also achieves the theoretical Nyquist maximum symbol rate by using practicable and perturbation-tolerant filters [28].

Generally, the PRS technique provides a multi-level coded output using a controlled amount of inter-symbol interference such that the receiver logic can cancel out the inter-symbol interference [29]. There are several classes of PRS polynomials, each uniquely combining the two fundamental coding polynomials, dicode $(1 - D)$ and duobinary $(1 + D)$ to attain varying spectral properties. Available classes of useful PRS polynomials can be found in [8]. Mathematically, these PRS polynomials are generally given by:

$$P(D) = (1 - D)^w(1 + D)^x(1 + (1 - D))^y((1 - D) + (1 + D)^2 - D)^z \quad (2.1)$$

where w, x, y, z are restricted to non-negative integer values not greater than two, and D is the unit time delay z^{-1} in the z -domain.

Due to the fact that the PRS technique accommodates a controlled ISI, it is easily possible to place more bits closer to each other; hence the reason for its proposition by Kobayashi and Tang for modelling magnetic recording channels to achieve high storage density [9]. Ever, since this proposition, there have been various classes proposed for achieving high storage density in magnetic recording systems; some of which are class-IV PRS (PR4) for moderately high density, and the extended class-IV (EPR4) for very high density requirements [30].

However, in order to realize the full potential of the PRS technique, in data communication applications such as the magnetic recording system, a technique more sophisticated than the conventional bit-by-bit threshold detector has to be used. This prompted the work of Kobayashi to propose the use of the maximum-likelihood sequence detection (MLSD) technique for implementing PRS channel detectors [7, 11]. By using the MLSD technique, it was discovered that even though the PRS technique uses multi-level transmission (thought to be more susceptible to noise effects; and demanding complex detection procedure), lower error rates can be achieved compared to the conventional bit-by-bit threshold detector.

2.2 The Viterbi Algorithm

Proposed in 1967 as a means of implementing the idea of Maximum-Likelihood Sequence Detection (MLSD) for convolutional codes [13], the Viterbi Algorithm (VA) has extended its tentacles beyond this and found application in PRS channel detection.

The Viterbi algorithm is a very efficient algorithm that can be seen as a forward dynamic programming application [31] for figuratively finding the shortest route to a

destination from one or more likely starting positions based on a probabilistic criterion. Moreover, it is also a suitable solution to the problem of maximum *a posteriori* probability (MAP) estimation of the state sequence in a finite-state, discrete-time Markov chain process, assuming a memory-less output noise [32, 33].

Hypothetically, assuming a finite-state process involving more than one state, according to the VA, each state is estimated as having an initial, equal weighted relative likelihood, referred to as the *state metric*. The states' transition paths are known as *branches* and their data-dependent, weighted value is known as the *branch metric*. Each available path is also characterized with a measure of the acquired Euclidean distance (which is a negative log-likelihood estimate [34, 35]), referred to as the *path metric* [36].

Estimation of the best possible path within the cluster of available paths emanating from various states then proceeds by computing the respective branch metrics (Euclidean distance between time-dependent received noisy symbol and idea branch symbol) for all the transitions from all states that evolves a path to each observed state. These are then added to each state metric that is responsible for that transition to obtain the path metric. The surviving path (*survivor*) to each state in the next time step is then determined by comparing the path metrics for all transitions entering that particular state and choosing the minimum/maximum (depending on the branch metric computation) of all the path metrics.

The above described approach for implementing the classical Viterbi algorithm basically involves adding, comparing and then selecting; thus the tag ACS approach and the use of ACS units in Viterbi decoders in both digital and analog domains. These could be found in many hardware implementations of Viterbi decoders/detectors in voltage-mode [37, 38], current-mode [39, 40, 41, 42] and neural networks [43].

Although the traditional approach works well, the unbounded growth of the path metrics is a common concern that had to be dealt with in all of these implementations. In [44], this problem was combated by the use of common-mode feedback circuitry to monitor the path metrics common-mode signals and keep their value equal to a pre-determined reference voltage; thus effectively minimizing the dynamic range of the circuits in the decoder. Also, in [42], a path metric normalization circuit had to be used to restrain the path metric within specified bounds.

Another drawback of the traditional approach is the speed bottleneck usually created by the ACS units since this is where the longest delay can occur before the survivor and path metric storage [45] and a proposition to enhance the speed can be found in [45, 46]. However, a more promising approach to solving the problem of un-bounded path metric growth and avoiding the computation of absolute state metrics is to compute the state metric difference instead; which is actually bounded [32, 15].

2.2.1 Difference Metric Approach

The difference metric algorithm, which is a variant of the classical Viterbi algorithm was first proposed by Ferguson in 1972 [15]. Ferguson's work not only revealed that absolute state metric computation is un-warranted, it also solves the problem of un-bounded metric growth. The use of this algorithm in hardware implementations of Viterbi decoders [24, 19, 23, 21] has proven how versatile the approach could be and has also shown how it could significantly reduce the complexity of the needed hardware; at least for a two-state PRS channel trellis.

Conceptually, consider an arbitrary two-state trellis involving states a and b as shown in Figure 2.1. The true difference metric algorithm involves the computation

of a non-stationary, time-variant term Δm , which can be given by:

$$\Delta m = m_a - m_b \quad (2.2)$$

where m_a and m_b represent the state metric for states a and b respectively.

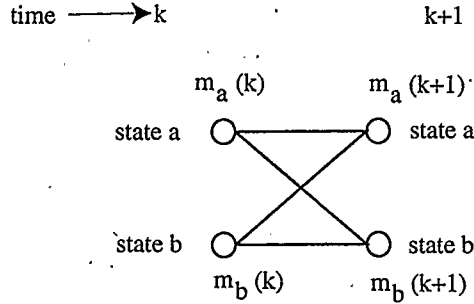


Figure 2.1: Arbitrary two state trellis

The use of such a defined term (time-variant differential threshold) in hardware implementations of Viterbi decoders has somewhat been restricted to decoders for binary decode PRS channels; partly because a very useful class of PRS polynomial (class IV) $1 - D^2$ could be formed by time-interleaving two decode channels. However, since the fundamental and higher-order PRS coding polynomials were not proposed exclusively for binary type systems only, there is the need to investigate the use of the true difference metric algorithm concept for m -ary (where $m > 2$) systems; if it could be applied at all for PRS channel trellises with more than two states.

2.3 EPR4 Channel Detection

For very high density requirements in magnetic recording systems, the extended class-IV PRS channel is generally preferred [10]. This channel is an eight-state, three memory element FIR-type channel (trellis as shown in Figure 2.2) with a polynomial

model:

$$\begin{aligned} P(D) &= (1 - D)(1 + D)^2 \\ &= 1 + D - D^2 - D^3 \end{aligned} \quad (2.3)$$

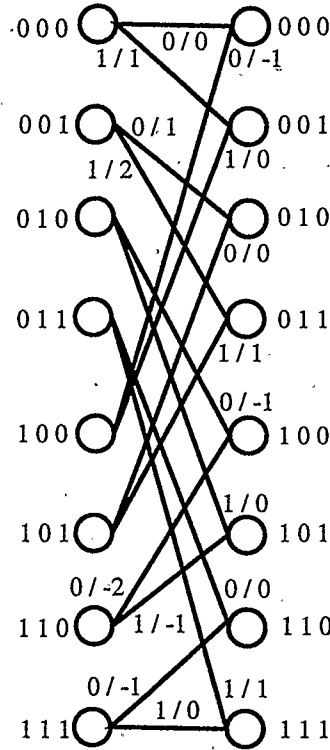


Figure 2.2: EPR4 channel trellis

Thus, with a binary input (0,1), this channel gives a five level output signal ($\pm 2, \pm 1, 0$). In contrast to the four-state, two memory element, three level output ($\pm 1, 0$) class-IV (PR4) channel, which can be time-interleaved using two binary dicode channels, the EPR4 cannot be directly time-interleaved. Hence, for this reason and its high number of states, it is generally more complex to implement the detector.

Several attempts had been made by researchers to simplify EPR4 detection based on the maximum-likelihood sequence detection concept. Worthy of note are the re-

ported works of Wood [16], Friedmann [17], and Knudson [47].

2.3.1 Wood's Detection Scheme

Wood's approach [16] referred to as "Turbo-PRML" is a compromise EPR4 detection approach which is based on factorization of the EPR4 polynomial as:

$$P(D) = (1 - D^2)(1 + D). \quad (2.4)$$

Thus, by realizing that the PR4 channel is represented by $1 - D^2$, the received signal can be equalized to the PR4 target, enabling the use of a PR4 detector; the output error sample of which is then made to be processed by the duobinary channel.

The major advantage of this technique is the ease of using existing PR4 detectors with some additional low-complexity post processors. However, this approach has two notable drawbacks which are:

- more aggressive equalization is required than for an actual EPR4 detector [16].
- loss in signal to noise ratio (SNR) may result at high densities mainly because of equalization to PR4 instead of the desired EPR4 target [17].

2.3.2 Friedmann's Detection Scheme

In an effort to make the inherent benefit of time-interleaving (used in PR4 detectors) available for EPR4 detection (i.e. speed doubling), Friedmann proposed another simplification method for EPR4 detection based on equation(2.4), re-written as:

$$P(D) = (1 + D)(1 - D^2). \quad (2.5)$$

The philosophy is that if a binary sequence is first coded by the duobinary channel, a ternary sequence results. The EPR4 channel coding can then be completed by allowing the ternary sequence to pass through two time-interleaved ternary decode channels, which accounts for the remaining $1 - D^2$ factor in the polynomial as shown in Figure 2.3. Therefore, a backward organization of this EPR4 coding technique should result in an EPR4 channel detector; this is depicted in Figure 2.4.

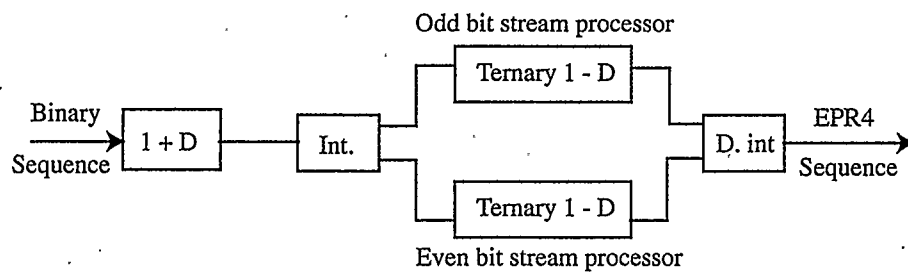


Figure 2.3: Friedmann's form of EPR4 channel coding

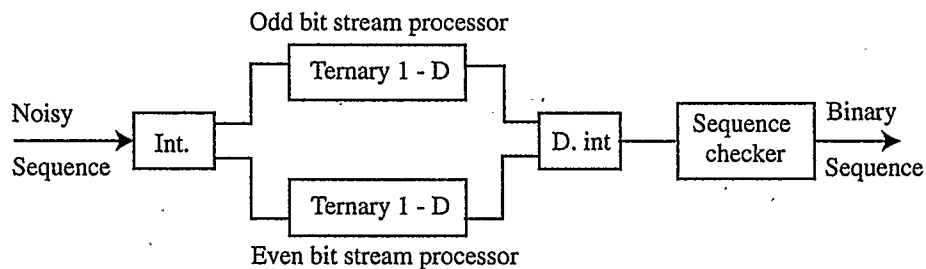


Figure 2.4: Friedmann's EPR4 channel detection technique

Obviously, the advantage of this method is the possibility of time-interleaving two decode channels, which forms the core of the detection process. A post-processor is only needed occasionally to help correct some error when discovered in the output of the two interleaved ternary decode detectors. This method thus makes it possible to design an EPR4 detector which works at double the speed at which each ternary decode detector can be made to operate.

The only disadvantage for this method is the requirement to use ternary dicode detectors, which is expected to be more susceptible to error-event occurrence than a binary dicode channel. Nonetheless, Friedmann's approach overcomes one of the drawbacks of Wood's approach since the actual EPR4 equalization target remains applicable. The problem then becomes that of developing a fast and efficient ternary dicode detector.

2.3.3 Knudson's Detection Scheme

In this scheme, Knudson tried solving the EPR4 detection problem by venturing into a difference metric approach. In his technique, four threshold definitions and three quantities called "shifts", of similar definition as the thresholds are required. Each of this quantities are of the general form:

$$T_n/S_m = 0.5\Delta m + \gamma \quad (2.6)$$

where T_n implies threshold, S_m is shift, Δm is state metric difference, and γ is such that for T_n , $\gamma \in \{\pm 0.5, 1.5\}$ and for S_m , $\gamma \in \{0.5, 1\}$.

It is worth noting that this technique does not really result in a simplification of the EPR4 detection except for the fact that metrics overflow is prevented as a result of using threshold/shift definition based on metrics difference (even though the true difference metric approach is not used in the strict sense).

Amongst all these detection simplification techniques, Friedmann's approach seems to stand out and could be the best means of ultimately simplifying EPR4 detection, if only efficient ternary fundamental channel detectors can be developed and implemented. This is the motivation for pursuing the research work in this dissertation.

Chapter 3

A New Detection Algorithm: The Derivation

3.1 A Preview

As previously pointed out in the preceding chapter, a simplified detection scheme for the extended class-IV partial-response signaling channel (EPR4) may require the use of ternary duobinary or dicode channels. Also stated is the fact that, to achieve a high-level of error detection, the maximum-likelihood sequence detection concept is sufficient for the PRS channel. Therefore, the derivations in this chapter for what will be referred to as the “differential threshold reliant detectors” will be commensurate using the Viterbi algorithm.

From the first principle, the true difference metric variation of the Viterbi algorithm as originally proposed by Ferguson [15] will be employed since it has been shown to circumvent the shortcoming(s) of the rather naive interpretation (add-compare-select approach) of the Viterbi algorithm. The goal is to attain a new and very

efficient algorithm for both the ternary dicode and duobinary PRS channel that will make the simplification of the EPR4 channel detection possible.

3.2 Ternary Dicode Channel

Typically, a dicode partial-response polynomial is a first order polynomial given by:

$$P(D) = 1 - D \quad (3.1)$$

where D is the unit time delay z^{-1} in the z -domain. In terms of a finite impulse response (FIR) representation, the dicode channel is a 2-tap FIR structure with one channel memory element D and weights w_n ; where n goes from 0 to 1, and $w_n \in \{1, -1\}$ is as shown in Figure 3.1 (without the bandlimiting filter).

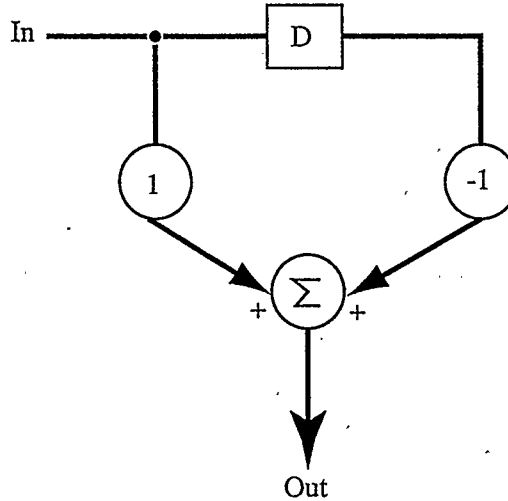


Figure 3.1: Dicode channel FIR representation

Therefore, with a tri-state channel memory element and a ternary input sequence s_k , where $s_k \in \{\pm 1, 0\}$, the coded channel output x_k will be a 5-level signal such that, $x_k \in \{\pm 2, \pm 1, 0\}$. Thus the trellis for this channel is depicted by Figure 3.2.

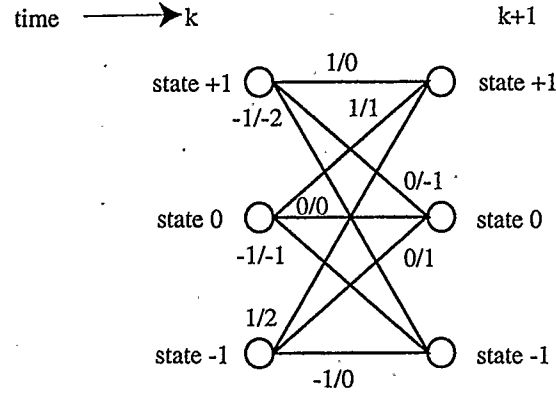


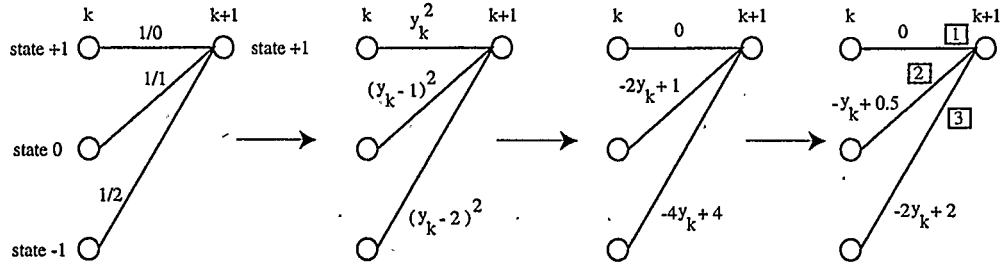
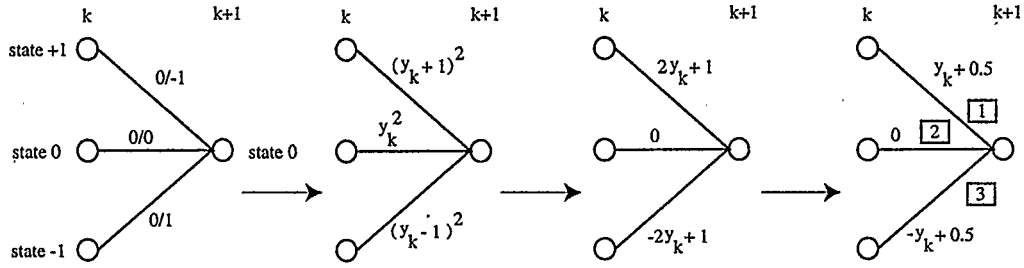
Figure 3.2: Dicode channel trellis

The problem then becomes that of finding the most appropriate path (survivor) in the trellis which results in the noise-less symbol closest in distance to the transmitted symbol that was corrupted by the channel at any given instance of time.

3.2.1 States Survivor Derivation

In the following derivations, let k represent an index of time. The three states in the trellis i.e. +1, 0, and -1 are represented by S_{+1} , S_0 , S_{-1} , respectively, and their respective state metrics are given by m_{+1} , m_0 , and m_{-1} . Let y_k be the received noisy sample at the instance of time k . Between time instances k and $k+1$, there are three distinct transitions emanating from all the three states and terminating on each of the states S_{+1} , S_0 , and S_{-1} as illustrated in Figure 3.3-3.5 with their respective branch metrics¹.

¹All the branch metrics have been derived by first subtracting a common term y_k^2 from the original branch metric calculation for all transitions. Then, the remaining terms for all transitions were uniformly scaled by a positive factor of 2. This is done in order to avoid the need for a squaring operation. This approach for simplifying branch metric computation was first proposed and vindicated in [7].

Figure 3.3: Transitions to state +1 at time time $k+1$ for the dicode channel trellisFigure 3.4: Transitions to state 0 at time time $k+1$ for the dicode channel trellis

Therefore by applying the Viterbi algorithm, the metrics for the three states at time $k+1$ can be expressed as:

$$m_{+1}(k+1) = \max[m_{+1}(k), m_0(k) - (-y_k + 0.5), m_{-1}(k) - (-2y_k + 2)] \quad (3.2)$$

$$m_0(k+1) = \max[m_{+1}(k) - (y_k + 0.5), m_0(k), m_{-1}(k) - (-y_k + 0.5)] \quad (3.3)$$

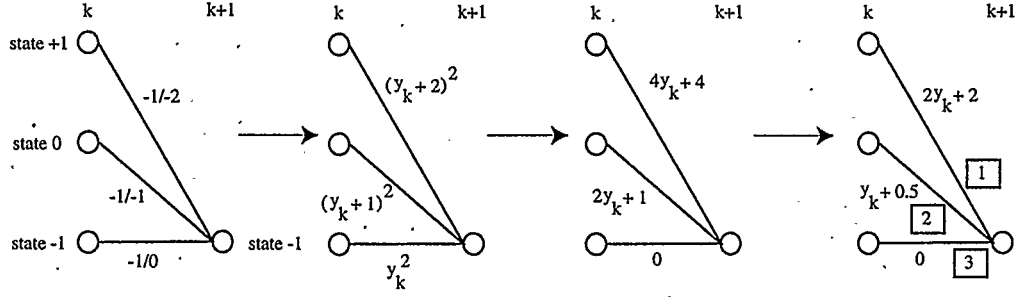
$$m_{-1}(k+1) = \max[m_{+1}(k) - (2y_k + 2), m_0(k) - (y_k + 0.5), m_{-1}(k)] \quad (3.4)$$

Considering equation(3.2), if path 1 should be the survivor, then the corresponding path metric must be greater than that for paths 2 and 3; implying that

$$y_k < m_{+1}(k) - m_0(k) + 0.5 \quad \text{and} \quad y_k < \left(\frac{m_{+1}(k) - m_{-1}(k)}{2} \right) + 1. \quad (3.5)$$

In a similar manner, for path 2 to be the survivor, then

$$y_k > m_{+1}(k) - m_0(k) + 0.5 \quad \text{and} \quad y_k < m_0(k) - m_{-1}(k) + 1.5. \quad (3.6)$$

Figure 3.5: Transitions to state -1 at time time $k+1$ for the dicode channel trellis

Using the same kind of argument, path 3 will be the survivor if

$$y_k > \left(\frac{m_{+1}(k) - m_{-1}(k)}{2} \right) + 1 \quad \text{and} \quad y_k > m_0(k) - m_{-1}(k) + 1.5. \quad (3.7)$$

Following a similar argument and using equation(3.3), for state S_0 , path 1, 2 and 3 will, respectively be the survivor at time $k+1$ if

$$y_k < m_{+1}(k) - m_0(k) - 0.5 \quad \text{and} \quad y_k < \left(\frac{m_{+1}(k) - m_{-1}(k)}{2} \right). \quad (3.8)$$

$$y_k > m_{+1}(k) - m_0(k) - 0.5 \quad \text{and} \quad y_k < m_0(k) - m_{-1}(k) + 0.5. \quad (3.9)$$

$$y_k > \left(\frac{m_{+1}(k) - m_{-1}(k)}{2} \right) \quad \text{and} \quad y_k > m_0(k) - m_{-1}(k) + 0.5. \quad (3.10)$$

Also, for S_{-1} , at time $k+1$, path 1, 2 and 3 will be the respective surviving path if

$$y_k < m_{+1}(k) - m_0(k) - 1.5 \quad \text{and} \quad y_k < \left(\frac{m_{+1}(k) - m_{-1}(k)}{2} \right) - 1. \quad (3.11)$$

$$y_k > m_{+1}(k) - m_0(k) - 1.5 \quad \text{and} \quad y_k < m_0(k) - m_{-1}(k) - 0.5. \quad (3.12)$$

$$y_k > \left(\frac{m_{+1}(k) - m_{-1}(k)}{2} \right) - 1 \quad \text{and} \quad y_k > m_0(k) - m_{-1}(k) - 0.5. \quad (3.13)$$

Now, let two terms (differential thresholds) Δm_c and Δm_d that are both functions of time be defined according to the true difference metric concept as

$$\Delta m_c = m_{+1} - m_0 \quad (3.14)$$

$$\Delta m_d = m_0 - m_{-1} \quad (3.15)$$

Based on these newly defined terms and the observation that all inequalities involving both m_{+1} and m_{-1} together could be neglected² (this will enable the ternary dicode trellis to be treated as two binary trellises with an added constraint of a feasible transition from S_{+1} to S_{-1} or vice versa); then the conditions and corresponding survivor paths for all states in the ternary dicode channel trellis can be deduced as follows:

$$\text{If } y_k - 0.5 < \Delta m_c(k) \text{ then } S_{+1}(k) \Rightarrow S_{+1}(k+1) \quad (3.16)$$

$$\begin{aligned} \text{If } y_k - 0.5 > \Delta m_c(k) \text{ and } y_k - 1.5 < \Delta m_d(k) \\ \text{then } S_0(k) \Rightarrow S_{+1}(k+1) \end{aligned} \quad (3.17)$$

$$\text{If } y_k - 1.5 > \Delta m_d(k) \text{ then } S_{-1}(k) \Rightarrow S_{+1}(k+1) \quad (3.18)$$

$$\text{If } y_k + 0.5 < \Delta m_c(k) \text{ then } S_{+1}(k) \Rightarrow S_0(k+1) \quad (3.19)$$

$$\begin{aligned} \text{If } y_k + 0.5 > \Delta m_c(k) \text{ and } y_k - 0.5 < \Delta m_d(k) \\ \text{then } S_0(k) \Rightarrow S_0(k+1) \end{aligned} \quad (3.20)$$

$$\text{If } y_k - 0.5 > \Delta m_d(k) \text{ then } S_{-1}(k) \Rightarrow S_0(k+1) \quad (3.21)$$

$$\text{If } y_k + 1.5 < \Delta m_c(k) \text{ then } S_{+1}(k) \Rightarrow S_{-1}(k+1) \quad (3.22)$$

$$\begin{aligned} \text{If } y_k + 1.5 > \Delta m_c(k) \text{ and } y_k + 0.5 < \Delta m_d(k) \\ \text{then } S_0(k) \Rightarrow S_{-1}(k+1) \end{aligned} \quad (3.23)$$

$$\text{If } y_k + 0.5 > \Delta m_d(k) \text{ then } S_{-1}(k) \Rightarrow S_{-1}(k+1). \quad (3.24)$$

²For instance, at the start of decoding, all state metrics assume equal value of zero. Therefore equation(3.5) will imply that $y_k < +0.5$ and $y_k < +1$. However, as long as $y_k < +0.5$, the second condition will be automatically satisfied. Also applying similar reasoning to equation(3.7) yields $y_k > +1$ and $y_k > +1.5$ but as long as $y_k > +1.5$, the remaining condition is invariably satisfied.

3.2.2 Merger Observations

It would be desirable to establish conditions by which the detector will be able to simultaneously determine all the three survivor paths during each decoding time frame. Such feasible conditions will be advantageous and enhance having an efficient implementation of the detector.

A meticulous observation of all the derived mathematical relations in section 3.2.1 could lead to a variety of interesting mergers of state transitions for the derived conditions for state survivor determination in the ternary dicode channel trellis. Such a feasible merger could imply that any single valid condition would lead to one or more state survivor paths being determined during each decoding time frame. However, it is worth noting that while some mergers will conspicuously follow from the stated conditions in section 3.2.1, some other mergers may be non-trivial as they require some indirect mathematical link between the inequalities based on the fact that the differential thresholds Δm_c and Δm_d are unequal (except at the start of decoding when all state metrics are the same and assume value equal to zero).

Recalling equations(3.16)-(3.24), four sets of possible state survivor mergers could directly follow from the relations and these are:

$$\begin{aligned} \text{If } y_k - 1.5 &> \Delta m_d(k) \text{ then} \\ y_k - 0.5 &> \Delta m_d(k) \text{ and } y_k + 0.5 > \Delta m_d(k) \end{aligned} \quad (3.25)$$

$$\begin{aligned} \text{If } y_k + 1.5 &< \Delta m_c(k) \text{ then} \\ y_k + 0.5 &< \Delta m_c(k) \text{ and } y_k - 0.5 < \Delta m_c(k) \end{aligned} \quad (3.26)$$

$$\text{If } y_k - 0.5 > \Delta m_d(k) \text{ then } y_k + 0.5 > \Delta m_d(k) \quad (3.27)$$

$$\text{If } y_k + 0.5 < \Delta m_c(k) \text{ then } y_k - 0.5 < \Delta m_c(k) \quad (3.28)$$

Assuming these relations are always true (which they are); and then by making the assumption that the pair of non-stationary differential thresholds will not be equal (in this case $\Delta m_c(k) < \Delta m_d(k)$)³, the conditions for states survivor path determination during each decoding time frame then finally become simplified to:

$$\begin{aligned} &\text{If } y_k + 1.5 < \Delta m_c(k) \text{ then} \\ &S_{+1}(k) \Rightarrow S_{+1}(k+1), S_0(k+1), S_{-1}(k+1) \end{aligned} \quad (3.29)$$

$$\begin{aligned} &\text{If } y_k + 0.5 < \Delta m_c(k) < y_k + 1.5 \text{ then} \\ &S_{+1}(k) \Rightarrow S_{+1}(k+1), S_0(k+1), \text{ and } S_0(k) \Rightarrow S_{-1}(k+1) \end{aligned} \quad (3.30)$$

$$\begin{aligned} &\text{If } y_k - 0.5 < \Delta m_c(k) < y_k + 0.5 \text{ and } \Delta m_d(k) > y_k + 0.5 \text{ then} \\ &S_0(k) \Rightarrow S_0(k+1), S_{-1}(k+1) \text{ and } S_{+1}(k) \Rightarrow S_{+1}(k+1) \end{aligned} \quad (3.31)$$

$$\begin{aligned} &\text{If } y_k + 0.5 > \Delta m_d(k) \text{ and } y_k - 0.5 < \Delta m_c(k) \text{ then} \\ &S_{+1}(k) \Rightarrow S_{+1}(k+1), S_0(k) \Rightarrow S_0(k+1), S_{-1}(k) \Rightarrow S_{-1}(k+1) \end{aligned} \quad (3.32)$$

$$\begin{aligned} &\text{If } y_k - 0.5 < \Delta m_d(k) < y_k + 0.5 \text{ and } \Delta m_c(k) < y_k - 0.5 \text{ then} \\ &S_0(k) \Rightarrow S_{+1}(k+1), S_0(k+1) \text{ and } S_{-1}(k) \Rightarrow S_{-1}(k+1) \end{aligned} \quad (3.33)$$

$$\begin{aligned} &\text{If } y_k - 1.5 < \Delta m_d(k) < y_k - 0.5 \text{ then} \\ &S_{-1}(k) \Rightarrow S_0(k+1), S_{-1}(k+1), \text{ and } S_0(k) \Rightarrow S_{+1}(k+1) \end{aligned} \quad (3.34)$$

³Considering equations (3.6), (3.9) and (3.12) will reveal that Δm_d is involved in the relations that set the upper condition for y_k while Δm_c is involved in that for the lower condition in these equations regardless of whether $m_{+1}(k)$ and $m_{-1}(k)$ are both set to zero with respect to $m_0(k)$ or Δm_c and Δm_d are set to zero at the start of decoding.

$$\begin{aligned} &\text{If } y_k - 1.5 > \Delta m_d(k) \text{ then} \\ &S_{-1}(k) \Rightarrow S_{+1}(k+1), S_0(k+1), S_{-1}(k+1) \end{aligned} \quad (3.35)$$

All the above conditions translate to five simple, data-dependent, well defined, bounded regions over which the differential thresholds operate to correctly detect the transmitted ternary sequence in a noisy dicode PRS channel. This is depicted in Figure 3.6⁴. It is worth noting that the most critical region out of all the five regions is region A_3 since there exist multiple sets of different combinations of survivor paths in this region. However, it is only one set of survivor path combinations that will always be valid.

3.2.3 The Updates

With such clearly defined regions and boundaries for the movement of the two differential thresholds already defined in the previous sections, and with the conditions for all three survivor paths to each of the states S_{+1} , S_0 , and S_{-1} during any time step established, the next appropriate step will be to determine how the differential thresholds will be algorithmically guided to help decode transmitted sequences through the entire decoding time frame.

Recalling equations(3.29)-(3.35) and Figure 3.4, there exist five clearly defined regions for which the update has to be determined for both time-variant terms Δm_c and Δm_d simultaneously. The following updates will be derived in conjunction with equations(3.2)-(3.4); which express the three state metrics for all distinct paths entering states S_{+1} , S_0 , and S_{-1} at time $k+1$.

⁴At this point, it should be noted that the algorithm omits the case $A_3(iv)$ (condition of occurrence is $\Delta m_d(k) > y_k + 0.5$ and $\Delta m_c(k) < y_k - 0.5$) which can actually occur as discovered in our experimental results (see Appendix D for details).

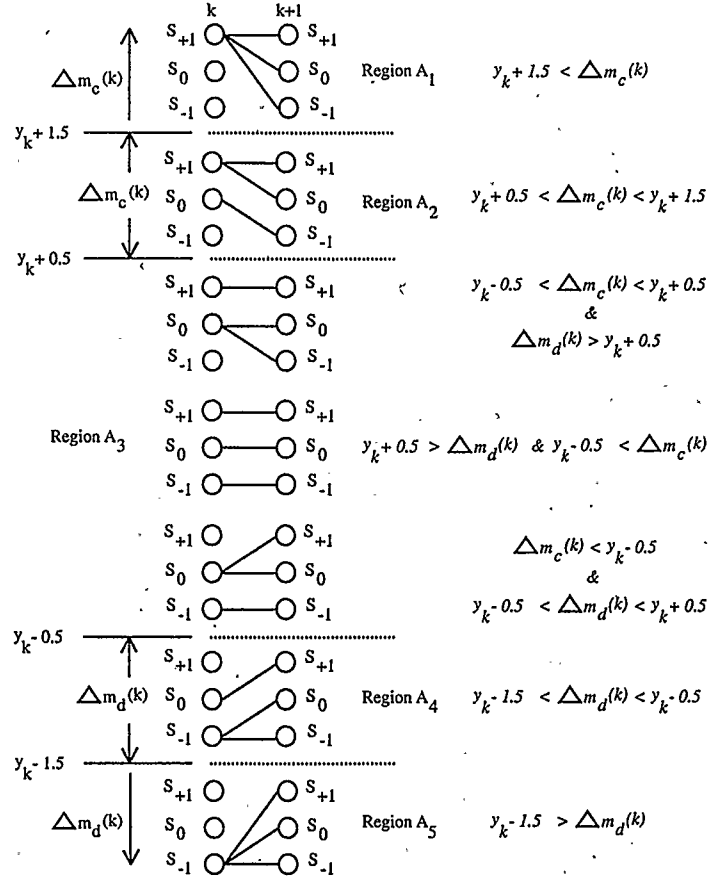


Figure 3.6: Region-dependency of the thresholds for the dicode channel

Region A_1 :

At time $k+1$,

$$\begin{aligned}
 m_{+1}(k+1) &= m_{+1}(k) \\
 m_0(k+1) &= m_{+1}(k) - y_k - 0.5 \\
 m_{-1}(k+1) &= m_{+1}(k) - 2y_k - 2
 \end{aligned} \tag{3.36}$$

Therefore based on equation(3.36),

$$\begin{aligned}
 \Delta m_c(k+1) &= m_{+1}(k+1) - m_0(k+1) \\
 &= m_{+1}(k) - [m_{+1}(k) - y_k - 0.5]
 \end{aligned}$$

$$= y_k + 0.5 \quad (3.37)$$

and

$$\begin{aligned} \Delta m_d(k+1) &= m_0(k+1) - m_{-1}(k+1) \\ &= m_{+1}(k) - y_k - 0.5 - [m_{+1}(k) - 2y_k - 2] \\ &= y_k + 0.5 \end{aligned} \quad (3.38)$$

Region A_2 :

At time $k+1$,

$$\begin{aligned} m_{+1}(k+1) &= m_{+1}(k) \\ m_0(k+1) &= m_{+1}(k) - y_k - 0.5 \\ m_{-1}(k+1) &= m_0(k) - y_k - 0.5 \end{aligned} \quad (3.39)$$

Therefore based on equation(3.39),

$$\begin{aligned} \Delta m_c(k+1) &= m_{+1}(k) - [m_{+1}(k) - y_k - 0.5] \\ &= y_k + 0.5 \end{aligned} \quad (3.40)$$

and

$$\begin{aligned} \Delta m_d(k+1) &= m_{+1}(k) - y_k - 0.5 - [m_0(k) - y_k - 0.5] \\ &= \Delta m_c(k) \end{aligned} \quad (3.41)$$

Region $A_3(i)$:

At time $k+1$,

$$\begin{aligned} m_{+1}(k+1) &= m_{+1}(k) \\ m_0(k+1) &= m_0(k) \\ m_{-1}(k+1) &= m_0(k) - y_k - 0.5 \end{aligned} \quad (3.42)$$

Therefore based on equation(3.42),

$$\begin{aligned}\Delta m_c(k+1) &= m_{+1}(k) - m_0(k) \\ &= \Delta m_c(k)\end{aligned}\tag{3.43}$$

and

$$\begin{aligned}\Delta m_d(k+1) &= m_0(k) - [m_0(k) - y_k - 0.5] \\ &= y_k + 0.5\end{aligned}\tag{3.44}$$

Region $A_3(ii)$:

At time $k+1$,

$$\begin{aligned}m_{+1}(k+1) &= m_{+1}(k) \\ m_0(k+1) &= m_0(k) \\ m_{-1}(k+1) &= m_{-1}(k)\end{aligned}\tag{3.45}$$

Therefore based on equation(3.45),

$$\begin{aligned}\Delta m_c(k+1) &= m_{+1}(k) - m_0(k) \\ &= \Delta m_c(k)\end{aligned}\tag{3.46}$$

and

$$\begin{aligned}\Delta m_d(k+1) &= m_0(k) - m_{-1}(k) \\ &= \Delta m_d(k)\end{aligned}\tag{3.47}$$

Region $A_3(iii)$:

At time $k+1$,

$$\begin{aligned}m_{+1}(k+1) &= m_0(k) + y_k - 0.5 \\ m_0(k+1) &= m_0(k) \\ m_{-1}(k+1) &= m_{-1}(k)\end{aligned}\tag{3.48}$$

Therefore based on equation(3.48);

$$\begin{aligned}\Delta m_c(k+1) &= m_0(k) + y_k - 0.5 - m_0(k) \\ &= y_k - 0.5\end{aligned}\tag{3.49}$$

and

$$\begin{aligned}\Delta m_d(k+1) &= m_0(k) - m_{-1}(k) \\ &= \Delta m_d(k)\end{aligned}\tag{3.50}$$

Region A_4 :

At time $k+1$,

$$\begin{aligned}m_{+1}(k+1) &= m_0(k) + y_k - 0.5 \\ m_0(k+1) &= m_{-1}(k) + y_k - 0.5 \\ m_{-1}(k+1) &= m_{-1}(k)\end{aligned}\tag{3.51}$$

Therefore based on equation(3.51),

$$\begin{aligned}\Delta m_c(k+1) &= m_0(k) + y_k - 0.5 - [m_{-1}(k) + y_k - 0.5] \\ &= \Delta m_d(k)\end{aligned}\tag{3.52}$$

and

$$\begin{aligned}\Delta m_d(k+1) &= m_{-1}(k) + y_k - 0.5 - m_{-1}(k) \\ &= y_k - 0.5\end{aligned}\tag{3.53}$$

Region A_5 :

At time $k+1$,

$$\begin{aligned}m_{+1}(k+1) &= m_{-1}(k) + 2y_k - 2 \\ m_0(k+1) &= m_{-1}(k) + y_k - 0.5 \\ m_{-1}(k+1) &= m_{-1}(k)\end{aligned}\tag{3.54}$$

Therefore based on equation(3.54),

$$\begin{aligned}\Delta m_c(k+1) &= m_{-1}(k) + 2y_k - 2 - [m_{-1}(k) + y_k - 0.5] \\ &= y_k - 1.5\end{aligned}\tag{3.55}$$

and

$$\begin{aligned}\Delta m_d(k+1) &= m_{-1}(k) + y_k - 0.5 - m_{-1}(k) \\ &= y_k - 0.5\end{aligned}\tag{3.56}$$

These updates are relatively simple and might render themselves easily to hardware implementation.

3.3 Ternary Duobinary Channel

Generally, the class-I or the duobinary PRS channel is modelled by the following first order polynomial:

$$P(D) = 1 + D\tag{3.57}$$

where D has its usual meaning as already explained in the previous section. Thus, the duobinary PRS channel is equivalently a 2-tap FIR structure shown in Figure 3.7 (without the bandlimiting filter). In a similar way to the dicode channel, this channel also gives a five level coded output with a ternary input sequence, assuming a tri-state channel memory. Its time-unwrapped state diagram is as shown in Figure 3.8. Therefore, the derivation of the optimal detection algorithm for this channel using the MLSD concept will be very similar to that of the ternary dicode channel.

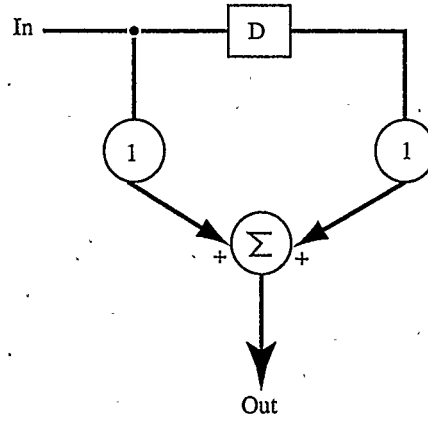


Figure 3.7: Duobinary channel FIR representation

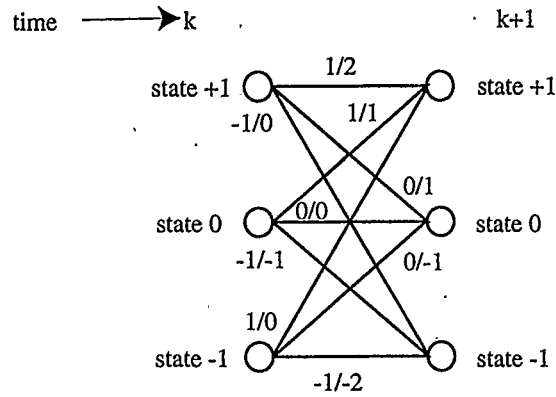


Figure 3.8: Duobinary channel trellis

3.3.1 States Survivor Derivation

Following the terminologies and terms used in the previous section, there are also three distinct transitions emerging from all three states and ending on each of the states S_{+1} , S_0 , and S_{-1} . Therefore, at time $k+1$, the state metric for all three states can be expressed as:

$$m_{+1}(k+1) = \max[m_{+1}(k) - (-2y_k + 2), m_0(k) - (-y_k + 0.5), m_{-1}(k)] \quad (3.58)$$

$$m_0(k+1) = \max[m_{+1}(k) - (-y_k + 0.5), m_0(k), m_{-1}(k) - (y_k + 0.5)] \quad (3.59)$$

$$m_{-1}(k+1) = \max[m_{+1}(k), m_0(k) - (y_k + 0.5), m_{-1}(k) - (2y_k + 2)] \quad (3.60)$$

Thus, for S_{+1} , the survivor will be path 1, path 2 or path 3 respectively if the following conditions are satisfied.

$$y_k > m_0(k) - m_{+1}(k) + 1.5 \quad \text{and} \quad y_k > \left(\frac{m_{-1}(k) - m_{+1}(k)}{2} \right) + 1 \quad (3.61)$$

$$y_k < m_0(k) - m_{+1}(k) + 1.5 \quad \text{and} \quad y_k > m_{-1}(k) - m_0(k) + 0.5 \quad (3.62)$$

$$y_k < \left(\frac{m_{-1}(k) - m_{+1}(k)}{2} \right) + 1 \quad \text{and} \quad y_k < m_{-1}(k) - m_0(k) + 0.5 \quad (3.63)$$

The necessary conditions for the survivor determination for the two remaining states, S_0 and S_{-1} can be derived as well.

The two time-variant, differential thresholds for this channel can be defined according to the true difference metric algorithm as:

$$\Delta m_a = m_0 - m_{+1} \quad (3.64)$$

$$\Delta m_b = m_{-1} - m_0 \quad (3.65)$$

If all the inequalities involving both m_{-1} and m_{+1} together are neglected⁵ the conditions and corresponding survivor paths for all states in the ternary duobinary channel trellis can be derived and expressed as:

$$\text{If } y_k - 1.5 > \Delta m_a(k) \quad \text{then} \quad S_{+1}(k) \Rightarrow S_{+1}(k+1) \quad (3.66)$$

$$\begin{aligned} \text{If } y_k - 0.5 > \Delta m_b(k) \quad \text{and} \quad y_k - 1.5 < \Delta m_a(k) \\ \text{then} \quad S_0(k) \Rightarrow S_{+1}(k+1) \end{aligned} \quad (3.67)$$

⁵For instance, at the beginning of decoding, all states metrics are equal to zero. Therefore equation(3.61) will imply that $y_k > 1.5$ and $y_k > 1$. As long as the former is true, the latter will always be satisfied. Also, equation(3.63) states that $y_k < 1$ and $y_k < 0.5$. Thus, the latter being true will invariably make the former satisfied

$$\text{If } y_k - 0.5 < \Delta m_b(k) \text{ then } S_{-1}(k) \Rightarrow S_{+1}(k+1) \quad (3.68)$$

$$\text{If } y_k - 0.5 > \Delta m_a(k) \text{ then } S_{+1}(k) \Rightarrow S_0(k+1) \quad (3.69)$$

$$\begin{aligned} \text{If } y_k + 0.5 > \Delta m_b(k) \text{ and } y_k - 0.5 < \Delta m_a(k) \\ \text{then } S_0(k) \Rightarrow S_0(k+1) \end{aligned} \quad (3.70)$$

$$\text{If } y_k + 0.5 < \Delta m_b(k) \text{ then } S_{-1}(k) \Rightarrow S_0(k+1) \quad (3.71)$$

$$\text{If } y_k + 0.5 > \Delta m_a(k) \text{ then } S_{+1}(k) \Rightarrow S_{-1}(k+1) \quad (3.72)$$

$$\begin{aligned} \text{If } y_k + 1.5 > \Delta m_b(k) \text{ and } y_k + 0.5 < \Delta m_a(k) \\ \text{then } S_0(k) \Rightarrow S_{-1}(k+1) \end{aligned} \quad (3.73)$$

$$\text{If } y_k + 1.5 < \Delta m_b(k) \text{ then } S_{-1}(k) \Rightarrow S_{-1}(k+1). \quad (3.74)$$

3.3.2 Merger Observations

In a similar way to the merger observations for the dicode channel discussed in section 3.2.2, it will be assumed that the pair of non-stationary, differential thresholds Δm_a and Δm_b are unequal (in this case $\Delta m_a > \Delta m_b$)⁶. Also, by recalling equations(3.65)-(3.73), it can be noted that the four sets of possible state survivor mergers that could directly follow from these relations are:

$$\begin{aligned} \text{If } y_k - 1.5 > \Delta m_a(k) \text{ then} \\ y_k - 0.5 > \Delta m_a(k) \text{ and } y_k + 0.5 > \Delta m_a(k) \end{aligned} \quad (3.75)$$

⁶Considering equation(3.62) will reveal that Δm_a is involved in the relations that set the upper condition for y_k while Δm_b is involved in that for the lower condition in this equation regardless of whether $m_{+1}(k)$ and $m_{-1}(k)$ are both set to zero with respect to $m_0(k)$ or Δm_a and Δm_b are set to zero at the start of decoding.

$$\begin{aligned} \text{If } y_k + 1.5 < \Delta m_b(k) \text{ then} \\ y_k + 0.5 < \Delta m_b(k) \text{ and } y_k - 0.5 < \Delta m_b(k) \end{aligned} \quad (3.76)$$

$$\text{If } y_k - 0.5 > \Delta m_a(k) \text{ then } y_k + 0.5 > \Delta m_a(k) \quad (3.77)$$

$$\text{If } y_k + 0.5 < \Delta m_b(k) \text{ then } y_k - 0.5 < \Delta m_b(k) \quad (3.78)$$

Therefore, based on these merger relations and the assumption that $\Delta m_a > \Delta m_b$, the region-dependent conditions for the state survivor determination in the ternary duobinary channel can be stated as:

Region B_1 :

$$\begin{aligned} \text{If } y_k + 1.5 < \Delta m_b(k) \text{ then} \\ S_{-1}(k) \Rightarrow S_{+1}(k+1), S_0(k+1), S_{-1}(k+1) \end{aligned} \quad (3.79)$$

Region B_2 :

$$\begin{aligned} \text{If } y_k + 0.5 < \Delta m_b(k) < y_k + 1.5 \text{ then} \\ S_{-1}(k) \Rightarrow S_{+1}(k+1), S_0(k+1), \text{ and } S_0(k) \Rightarrow S_{-1}(k+1) \end{aligned} \quad (3.80)$$

Region $B_3(i)$:

$$\begin{aligned} \text{If } y_k - 0.5 < \Delta m_b(k) < y_k + 0.5 \text{ and } \Delta m_a(k) > y_k + 0.5 \text{ then} \\ S_0(k) \Rightarrow S_0(k+1), S_{-1}(k+1) \text{ and } S_{-1}(k) \Rightarrow S_{+1}(k+1) \end{aligned} \quad (3.81)$$

Region $B_3(ii)$:

$$\begin{aligned} \text{If } y_k + 0.5 > \Delta m_a(k) \text{ and } y_k - 0.5 < \Delta m_b(k) \text{ then} \\ S_{+1}(k) \Rightarrow S_{-1}(k+1), S_0(k) \Rightarrow S_0(k+1), S_{-1}(k) \Rightarrow S_{+1}(k+1) \end{aligned} \quad (3.82)$$

Region $B_3(iii)$:

$$\begin{aligned} \text{If } y_k - 0.5 < \Delta m_a(k) < y_k + 0.5 \text{ and } \Delta m_b(k) < y_k - 0.5 \text{ then} \\ S_0(k) \Rightarrow S_{+1}(k+1), S_0(k+1) \text{ and } S_{+1}(k) \Rightarrow S_{-1}(k+1) \end{aligned} \quad (3.83)$$

Region B_4 :

$$\begin{aligned} &\text{If } y_k - 1.5 < \Delta m_a(k) < y_k - 0.5 \text{ then} \\ &S_{+1}(k) \Rightarrow S_0(k+1), S_{-1}(k+1), \quad \text{and} \quad S_0(k) \Rightarrow S_{+1}(k+1) \end{aligned} \quad (3.84)$$

Region B_5 :

$$\begin{aligned} &\text{If } y_k - 1.5 > \Delta m_a(k) \text{ then} \\ &S_{+1}(k) \Rightarrow S_{+1}(k+1), S_0(k+1), S_{-1}(k+1) \end{aligned} \quad (3.85)$$

All these relations also translate to five data-dependent, well defined regions over which the two differential thresholds Δm_a and Δm_b operate to decode any transmitted sequence over the ternary duobinary channel. This is shown in Figure 3.9⁷.

The necessary updates for the differential thresholds in this type channel detector is summarized in Table 3.1 (*see Appendix A for derivations*). These updates are also of equivalent complexity as those for the ternary dicode channel that was previously discussed. Thus, the ternary duobinary channel detector should be implementable using simple hardware.

3.4 Preliminary Evaluation

A cogent issue that needs to be addressed after deriving the two algorithms in the previous sections, will be to verify their validity before any further evaluation. Of course, in order to do this, the algorithms may be tested with a small sample of random ternary symbols while assuming a relatively high signal-to-noise ratio (SNR).

⁷At this point, it should be noted that the algorithm omits the case $B_3(iv)$ (condition of occurrence is $\Delta m_a(k) > y_k + 0.5$ and $\Delta m_b(k) < y_k - 0.5$) which can actually occur (*see Appendix D for details*).

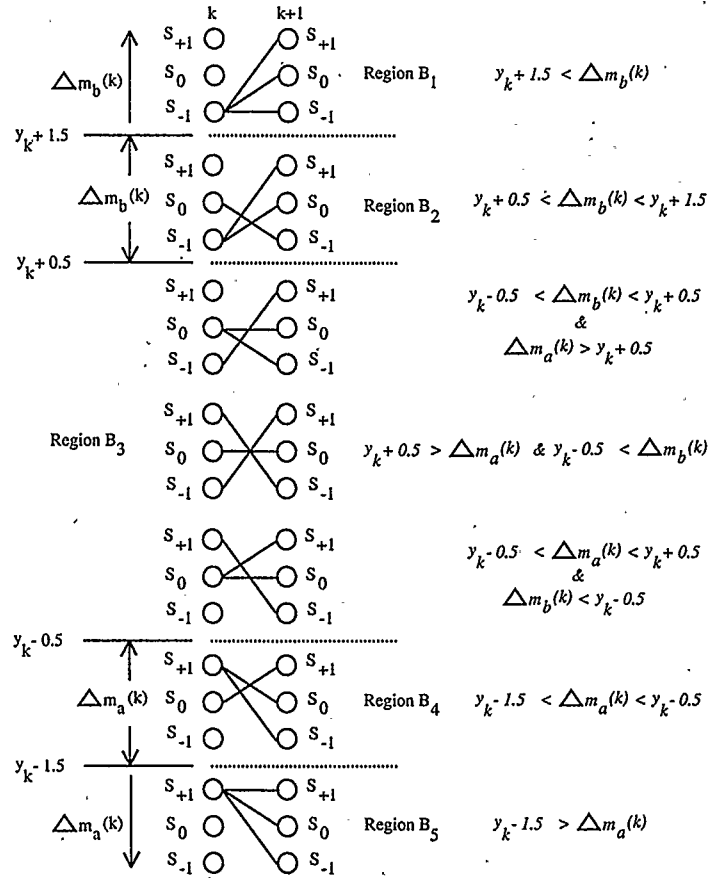


Figure 3.9: Region-dependency of the thresholds for the duobinary channel

High SNR has to be used since it is common knowledge that at low SNR, detectors (either bit-by-bit threshold detectors or maximum-likelihood detectors) are prone to high error-rate; although maximum-likelihood detectors will still be expected to perform better than the bit-by-bit threshold detector. Thus assuming a low SNR for verification could be misleading.

Therefore, in this evaluation, it will be assumed that the newly derived detectors are operating in channels where the signal-to-noise ratio is as high as 20dB. At this SNR, it will be expected that the signal power is high enough to lead to virtually zero error in estimating the actual transmitted symbols.

Table 3.1: Updates for ternary duobinary channel detector

Region	$\Delta m_a(k+1)$	$\Delta m_b(k+1)$
B_1	$-y_k - 0.5$	$-y_k - 1.5$
B_2	$-y_k - 0.5$	$-\Delta m_b(k)$
$B_3(i)$	$-\Delta m_b(k)$	$-y_k - 0.5$
$B_3(ii)$	$-\Delta m_b(k)$	$-\Delta m_a(k)$
$B_3(iii)$	$-y_k + 0.5$	$-\Delta m_a(k)$
B_4	$-\Delta m_a(k)$	$-y_k + 0.5$
B_5	$-y_k + 1.5$	$-y_k + 0.5$

In carrying out the preliminary evaluation of the two differential threshold reliant detectors for both the ternary dicode and duobinary channels, an arbitrary ternary sequence was generated via MATLAB.

This sequence was coded for the dicode and duobinary channels separately in each case and channel non-ideality was taken into consideration by adding a random noise sequence to the coded sequence such that a signal-to-noise ratio of 20dB is attained.

The corrupted symbols were then fed into the detection algorithms (as provided in the previous sections) and the eventual surviving path as predicted by the algorithms was obtained using the trace-back survivor memory management technique [48].

For both channels, a small sample size of 10 (arbitrarily chosen) ternary symbols was used. In the case of the dicode channel, the uncoded sequence is [1 1 0 -1 1 0 1 -1 0 0]; while for the duobinary channel, the arbitrary uncoded sequence is [-1 -1 0 1 0 -1 1 0 1 1]. Figures 3.10 and Figure 3.11 show the simulation results obtained. In both cases, the algorithm was able to estimate all of the transmitted symbols correctly

without a single error.

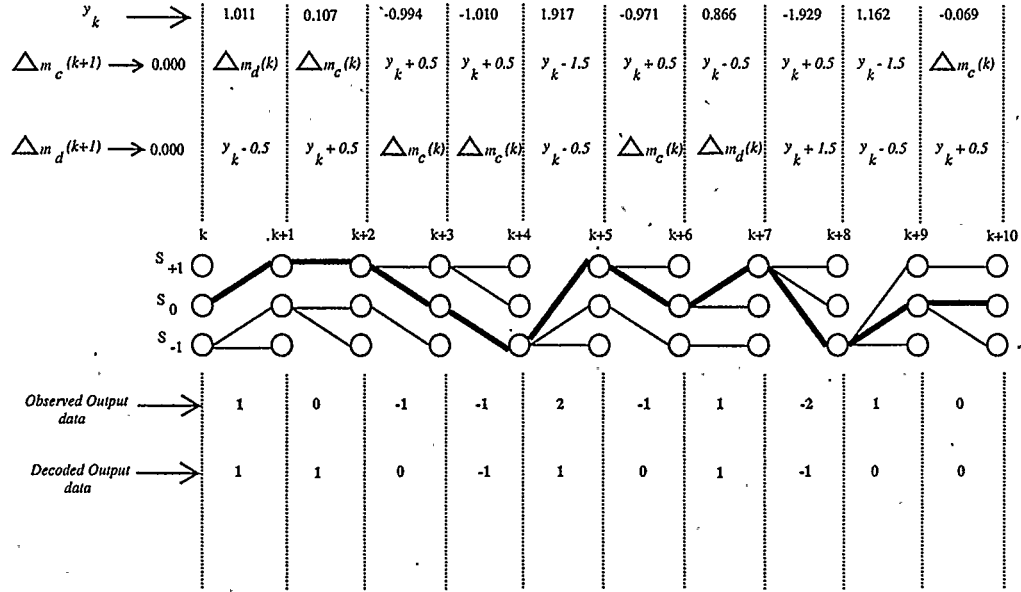


Figure 3.10: Typical decoding example: Ternary dicode channel (*Dark thick lines show the trace back path*)

Furthermore, the two differential thresholds in each case need to be monitored, in order to verify that the assumptions made about them in the process of deriving the algorithms is actually correct. Re-stated again, it is expected that $\Delta m_a > \Delta m_b$ in the case of duobinary channel while $\Delta m_d > \Delta m_c$ for the the dicode channel.

The simulated threshold movement pattern in the above decoding examples was captured and shown in Figure 3.12 and Figure 3.13. These simulation results show that the assumptions are true; at least for the small data size used. Further simulations would be needed to fully authenticate all the derivations provided in the previous sections of this chapter.

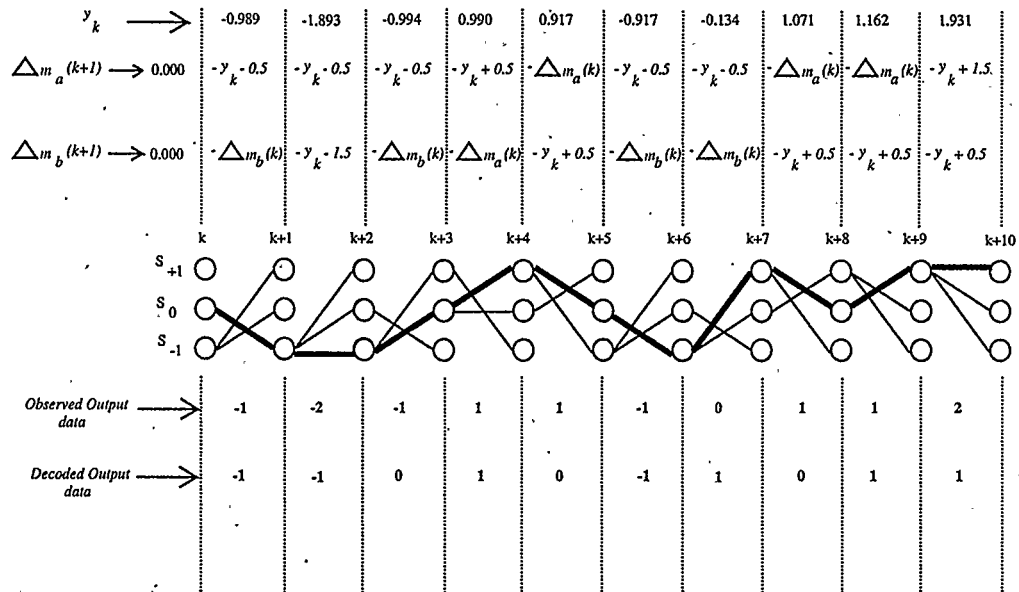


Figure 3.11: Typical decoding example: Ternary duobinary channel (*Dark thick lines show the trace back path*)

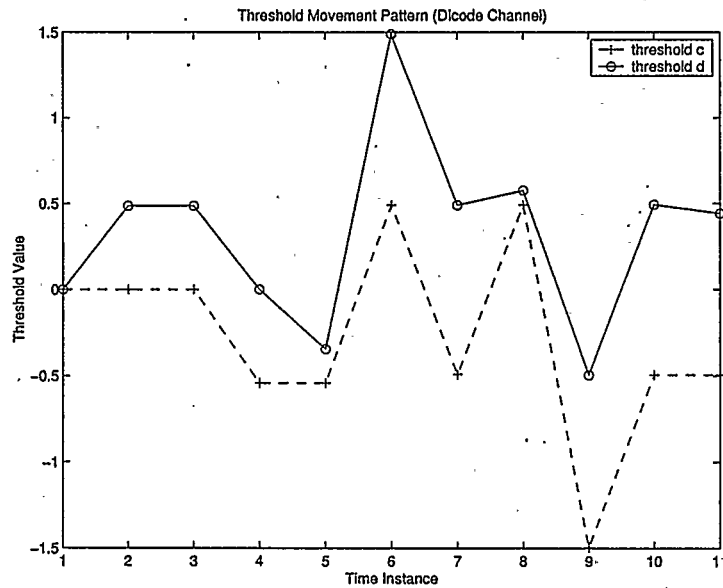


Figure 3.12: Threshold adaptation: Dicode Channel

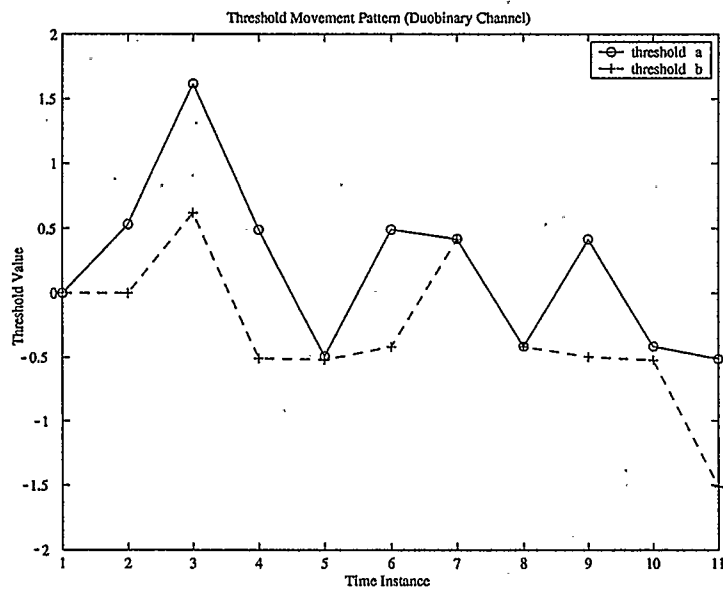


Figure 3.13: Threshold adaptation: Duobinary Channel

Chapter 4

Interpretation and Characterization

4.1 An Interpretation

In accordance with the outcome of the preliminary investigation of the differential threshold reliant detection scheme presented in the preceding chapter for both the ternary dicode and duobinary PRS channels, the algorithm derivation seem to be correct. However, the complete characterization of the algorithms cannot be inferred from the simple test that was carried out. A further analysis is needed to be able to understand the behavior of the propagating signals in the detector. A careful study of the algorithms presented in the preceding chapter would reveal that these differential thresholds (in either case) are dynamic in nature. Therefore, to further explain their characteristic behavior, the underlining computations in the detection scheme can be

explained as depicted in the flow-charts of Figure 4.1¹ and Figure 4.2².

Meticulous observation of the charts would show that the two differential thresholds required for the detection in each ternary PRS channel can only assume five possible values (although the five values vary with k); which depend on the region of operation of the thresholds. This is actually very similar in nature to that of the binary PRS channel, where only one threshold that could assume three possible values is required [15, 18, 25, 22].

In general, the number of possible values that the differential threshold could assume in either of the binary or ternary PRS channels, is related to the number of levels L that their coded output could attain (recall that $L = 2m - 1$).

For the ternary dicode channel, the two differential thresholds (Δm_c and Δm_d) required in this channel detection could take on any of five quantities at the instance of time $k + 1$, such that:

$$\Delta m_c(k + 1) \in \{y_k + 0.5, y_k - 0.5, y_k - 1.5, \Delta m_c(k), \Delta m_d(k)\} \quad (4.1)$$

$$\Delta m_d(k + 1) \in \{y_k + 1.5, y_k + 0.5, y_k - 0.5, \Delta m_c(k), \Delta m_d(k)\} \quad (4.2)$$

However, the quantity chosen by each threshold depends on the region that is detected and thus, leading to an interesting behavior of the thresholds in each possible region of operation.

4.1.1 Ternary Dicode: Threshold Δm_c

The characteristic behavior of the threshold Δm_c can be shown by both Figure 4.3 and Figure 4.4. In these figures, when the region of operation is between $y_k + 0.5$

¹see Appendix D for the updated version with region $A_3(iv)$ included.

²see Appendix D for the updated version with region $B_3(iv)$ included.

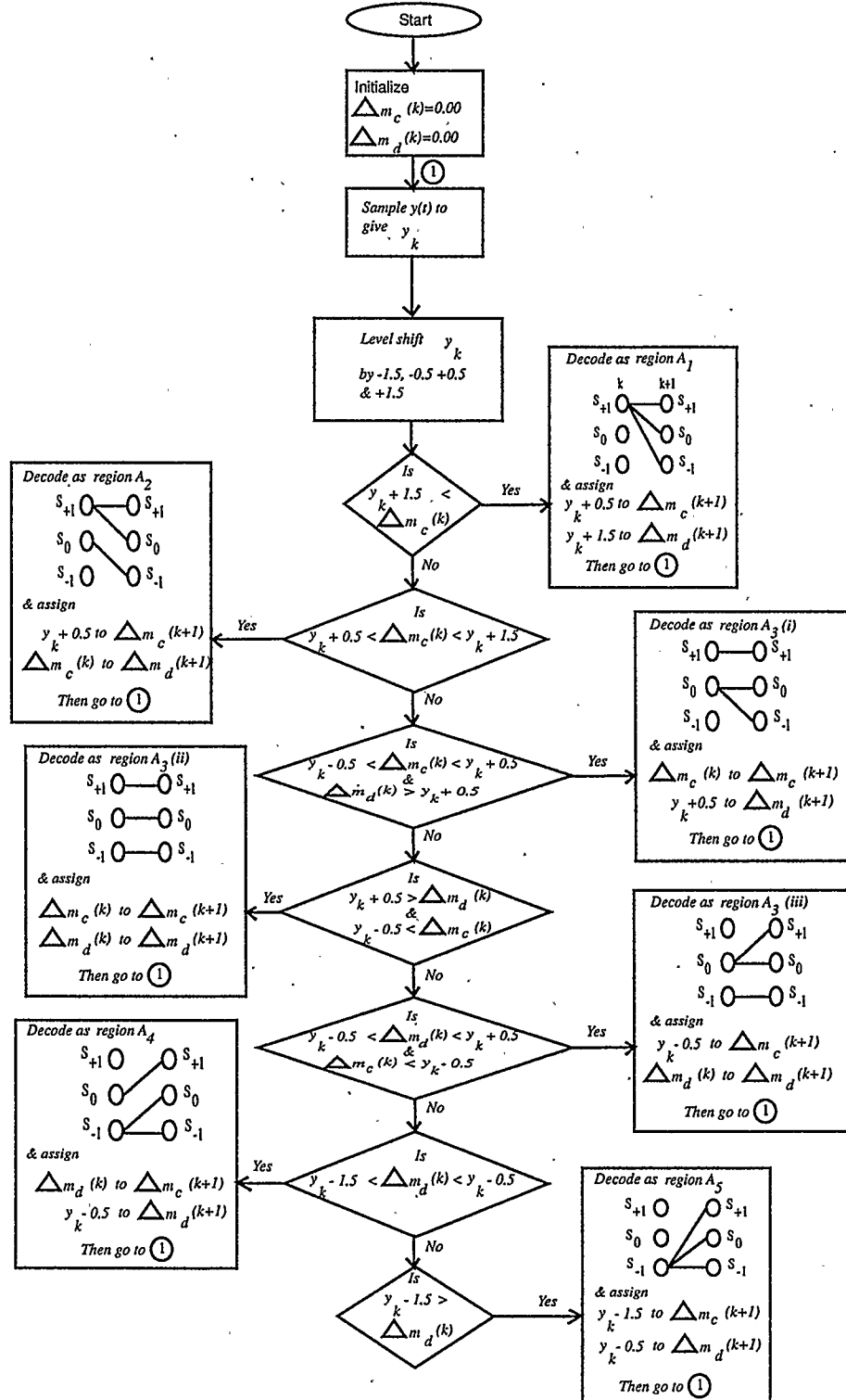


Figure 4.1: Flow-chart: Ternary Dicode Channel Detection

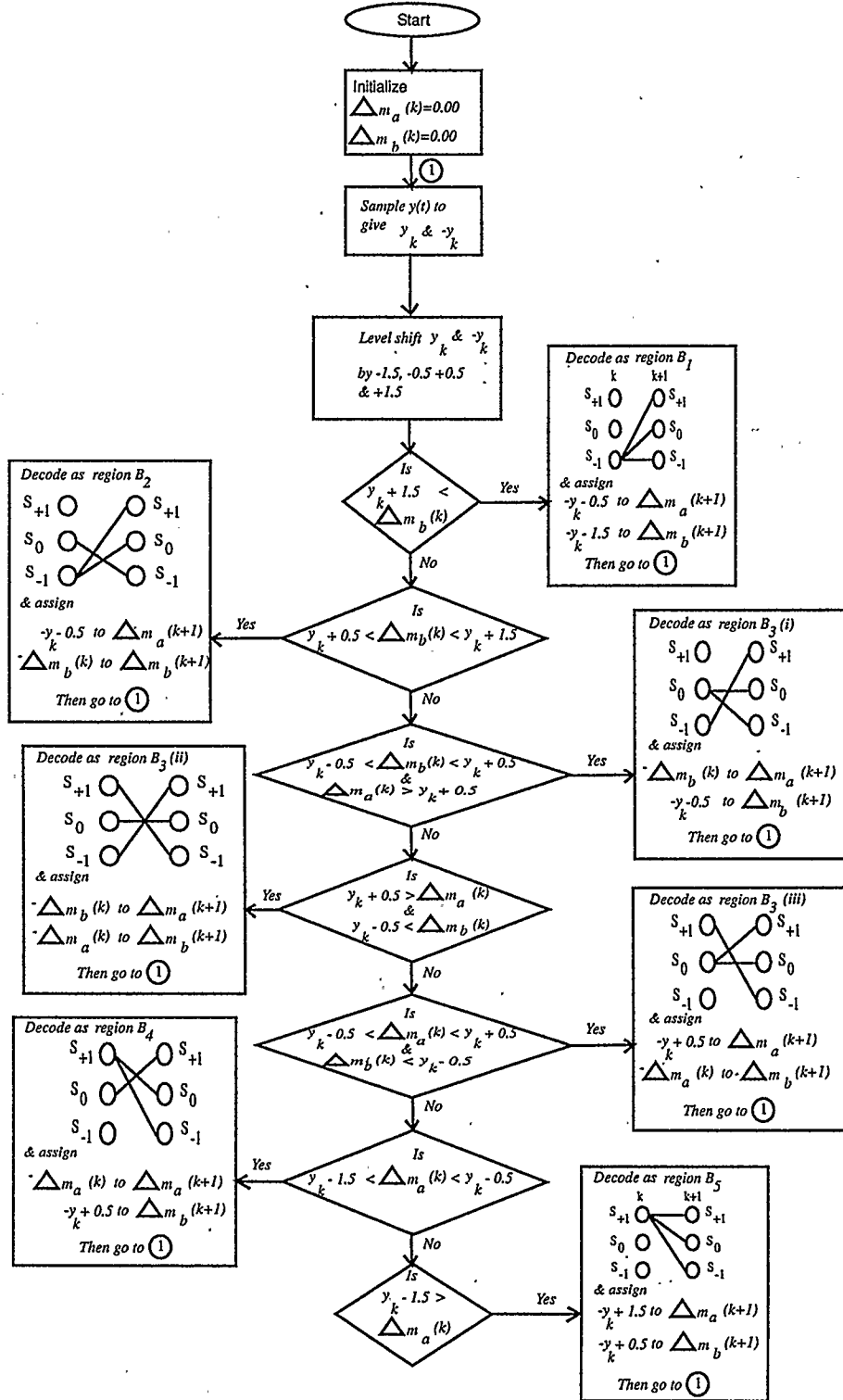


Figure 4.2: Flow-chart: Ternary Duobinary Channel Detection

and $y_k + 1.5$ or beyond $y_k + 1.5$, Δm_c would exhibit a zero gradient characteristic at a hard-limited value of $y_k + 0.5$. Beyond $y_k - 1.5$, Δm_c stays saturated at $y_k - 1.5$, thereby exhibiting a zero gradient characteristic, while between $y_k + 0.5$ and $y_k - 0.5$, it will have a steady, positive, unity gradient by assuming its previous value. However, between $y_k - 0.5$ and $y_k - 1.5$, there are two possibilities. Δm_c could either be saturated to a zero gradient threshold at the value of $y_k - 0.5$ or its gradient could be completely dependent on the history of the other propagating threshold (Δm_d).

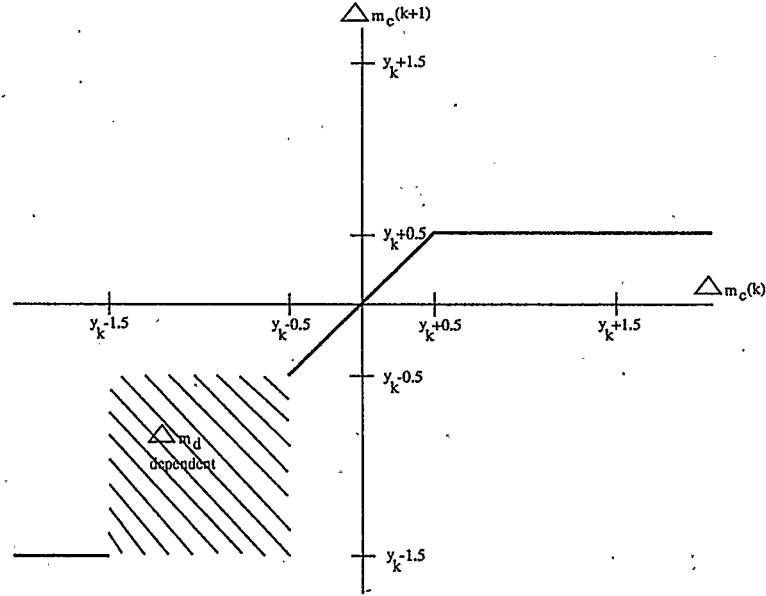
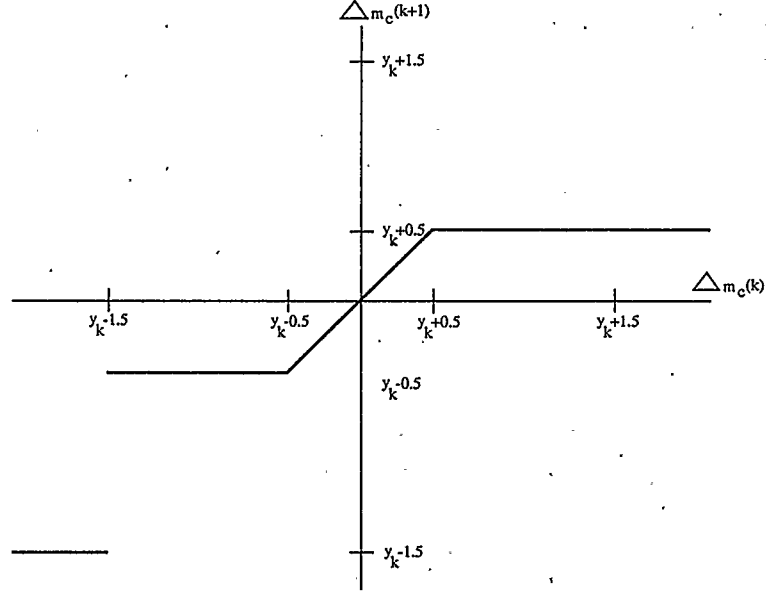


Figure 4.3: Threshold Δm_c adaptation type 1

4.1.2 Ternary Dicode: Threshold Δm_d

The behavior of threshold Δm_d is very similar to that of Δm_c , as might be expected. In this case, in the region beyond $y_k + 1.5$, Δm_d will have a zero gradient characteristic at a saturation value of $y_k + 1.5$. However, in contrast to the behavior of Δm_c in the region between $y_k + 0.5$ and $y_k + 1.5$, the gradient of Δm_d is either dependent on the

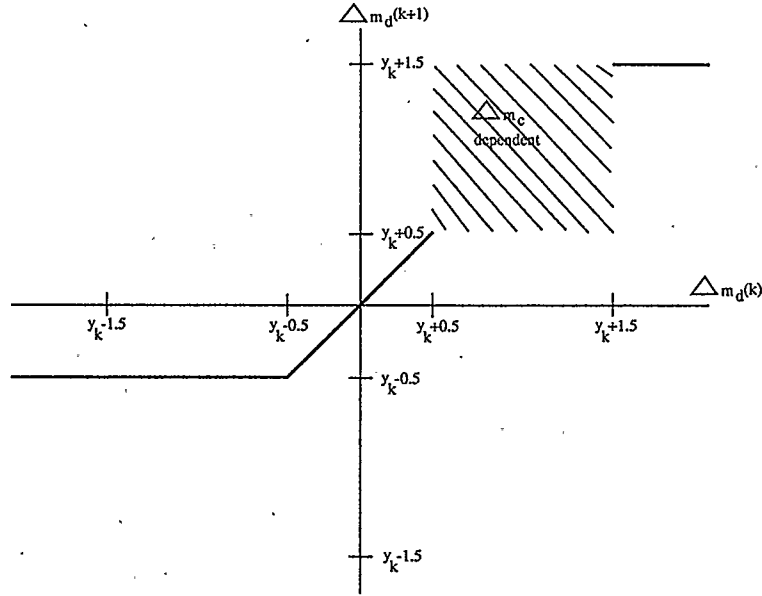
Figure 4.4: Threshold Δm_c adaptation type 2

history of the threshold Δm_c or limited to $y_k + 0.5$. Nevertheless, between $y_k + 0.5$ and $y_k - 0.5$, Δm_d also exhibits characteristics very similar to that of Δm_c ; since it would have a positive, unity gradient by retaining its own value from the previous instance of time. Furthermore, beyond $y_k - 1.5$, and between $y_k - 1.5$ and $y_k - 0.5$, the value of Δm_d is purely saturated to a value of $y_k - 0.5$. The above explanation is depicted in Figures 4.5 and Figure 4.6.

4.1.3 Ternary Duobinary: Threshold Δm_a

For the ternary duobinary channel, the regions of operation for the differential thresholds are the same as exists in the case of the ternary dicode channel. Within these regions, the value of threshold Δm_a is taken from the set:

$$\Delta m_a(k+1) \in \{-y_k - 0.5, -y_k + 0.5, -y_k + 1.5, \Delta m_a(k), \Delta m_b(k)\} \quad (4.3)$$

Figure 4.5: Threshold Δm_d adaptation type 1

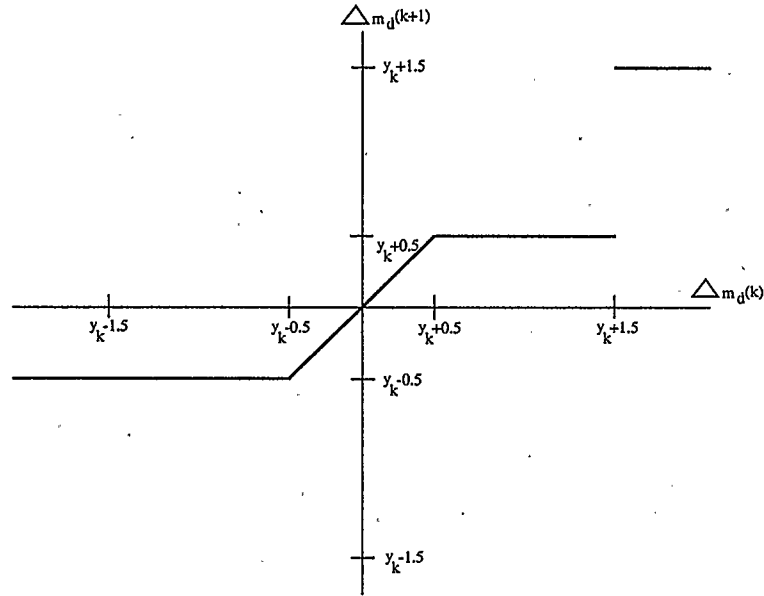
Therefore Δm_a exhibits characteristics that are shown in Figure 4.7 and Figure 4.8, and which can be explained as follows:

- Beyond $y_k + 1.5$, it is saturated at a zero gradient to $-y_k - 0.5$.
- It could have a negative gradient dependent on the history of Δm_b , or it could be limited to $-y_k + 0.5$ in the region from $y_k - 0.5$ to $y_k + 1.5$.
- Δm_a has steady, negative, unity gradient between $y_k - 0.5$ and $y_k - 1.5$.
- Beyond $y_k - 1.5$, Δm_a is hard-limited to a value of $-y_k + 1.5$.

4.1.4 Ternary Duobinary: Threshold Δm_b

At time $k + 1$, the value of Δm_b is chosen from the set:

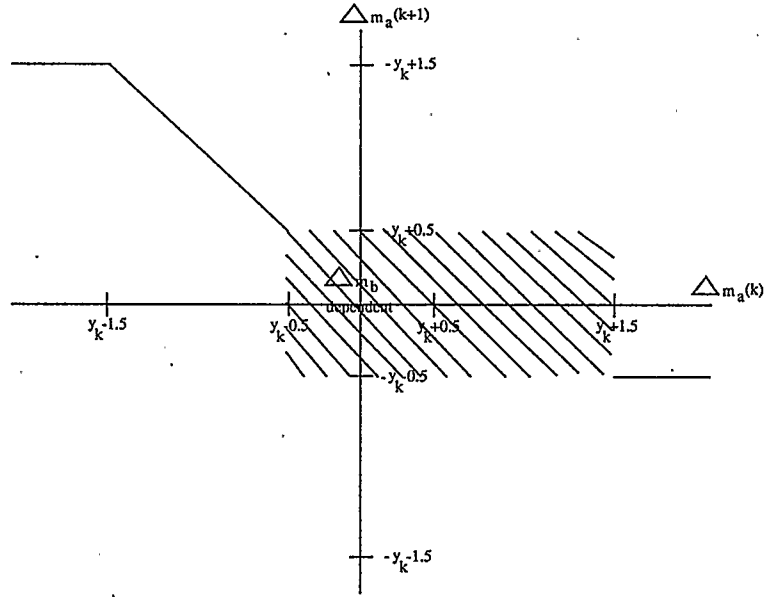
$$\Delta m_b(k + 1) \in \{-y_k + 0.5, -y_k - 0.5, -y_k - 1.5, \Delta m_a(k), \Delta m_b(k)\} \quad (4.4)$$

Figure 4.6: Threshold Δm_d adaptation type 2

Depending on the detected region, threshold Δm_b exhibits the following characteristics:

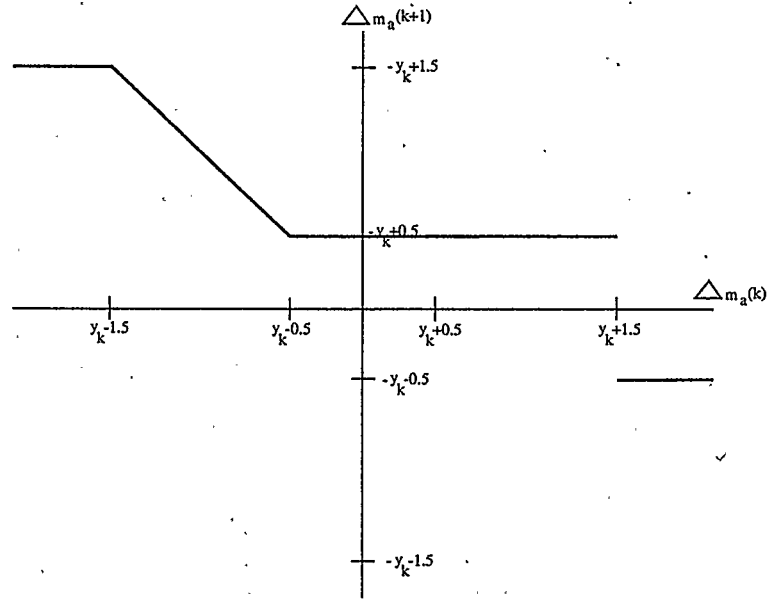
- Beyond $y_k - 1.5$, Δm_b is hard-limited to a zero gradient with value of $-y_k + 0.5$.
- It could be limited to $-y_k - 0.5$ in the region from $y_k - 1.5$ to $y_k + 0.5$, or it could have a negative gradient dependent on the history of Δm_a .
- Between $y_k + 0.5$ and $y_k + 1.5$, Δm_b has steady, negative, unity gradient as a result of negating its retained previous value.
- Beyond $y_k + 1.5$, Δm_b is saturated to a value of $-y_k - 1.5$.

The above characteristic behavior of Δm_b is depicted in Figures 4.9 and Figure 4.10.

Figure 4.7: Threshold Δm_a adaptation type 1

4.2 Saturation Effect

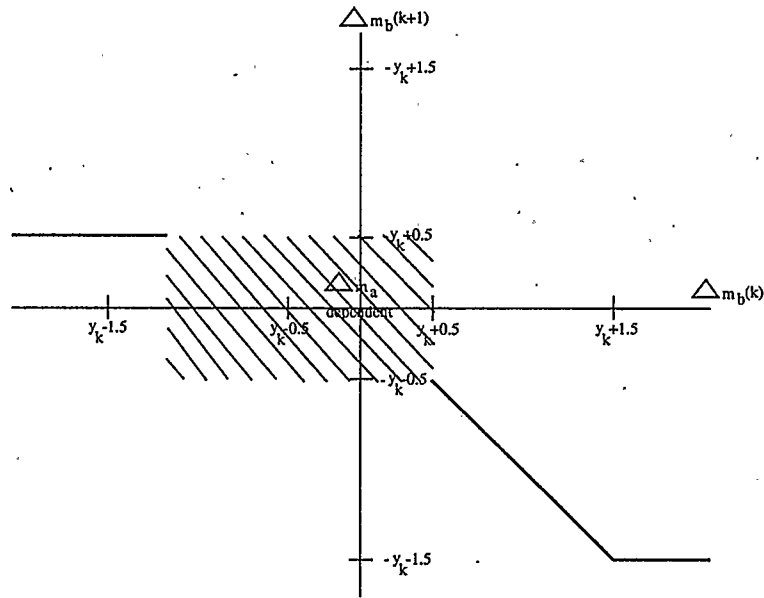
Usually, in any digital communication system, the transmitted digital input sequence s_k is corrupted such that the eventual output of the channel $y(t)$ can be regarded as being analog in nature [49]. In the magnetic recording channel, which serves as a case study in this dissertation, the source of corruption is assumed to be additive white Gaussian noise $n(t)$. From a digital point of view, $y(t)$ has to be quantized back to a digital sample using a fast analog-to-digital (A/D) converter before detection can be done by a digital Viterbi decoder. However, from an analog point of view, since the channel output is already analog in nature, the A/D converter could be eliminated and the noisy channel output $y(t)$ can be directly sampled to y_k , which can then be used as input to an analog Viterbi detector. It has already been studied and established that only a moderate resolution of 6-bits is required for analog circuitry in an analog Viterbi detector [24]. Thus, only simple analog circuitry need be used.

Figure 4.8: Threshold Δm_a adaptation type 2

By eliminating the A/D converter, analog implementations have seemingly gained three advantages over their digital counterpart; namely the absence of quantization noise, the removal of the high power consumption of the fast A/D converter, and the elimination of the silicon area of the A/D converter in an integrated implementation. However, the analog implementation itself suffers from effects such as offset, saturation, and mismatch. This section delves into the effect of saturation for the ternary dicode and duobinary channel detector.

The seemingly analog outputs of these channels are not unbounded or rather, the analog input samples for the detector cannot be unbounded in value since all electronic hardware is confined to operate within specific power supplies. Moreover, not all the electronic hardware possesses the ability to function from rail-to-rail. This invariably raises two questions:

- what effect does a limited input signal have on the performance of the detection

Figure 4.9: Threshold Δm_b adaptation type 1

algorithms?.

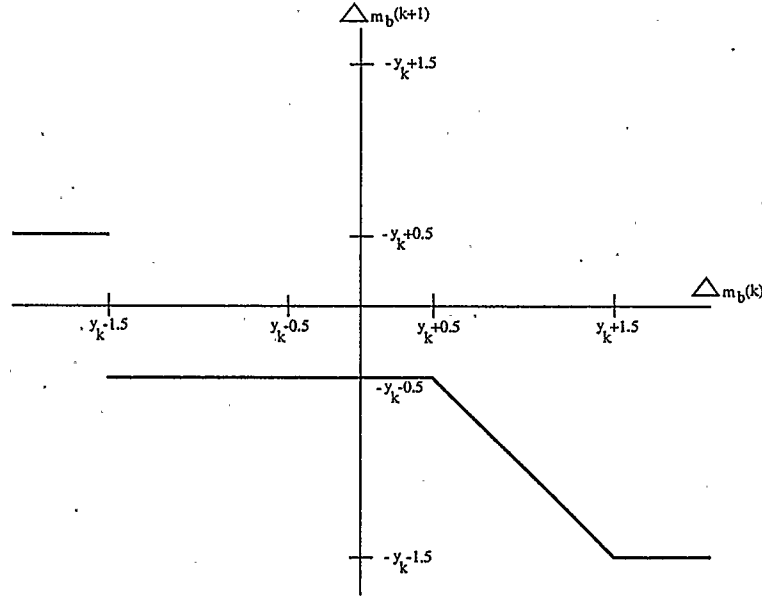
- how sensitive are the algorithms to signal swing limitation?

To answer these questions, the algorithms were tested and evaluated for the effect of input limitation and varying dynamic range.

4.2.1 Ternary Dicode: Input Limitation

In order to study the effect of input limitation on the differential threshold reliant ternary dicode detector, the noisy sample y_k that was fed into the algorithm was clipped to various values at constant signal-to-noise ratio. The simulation result as obtained from MATLAB is given in Figure 4.11.

It can be observed that when the input y_k is below a limiting value 0.5, no update

Figure 4.10: Threshold Δm_b adaptation type 2

of the algorithm took place. However, between the limiting value of 0.5 and 1.5, the algorithm update begins and the error rate starts to decrease even though a steady error rate value is not reached. Eventually, as the limiting value of y_k become greater than 1.5, a rapid update occurs and the algorithm quickly reaches the steady error rate value for the pertinent signal to noise ratio (SNR).

A meticulous observation of the derived detection algorithm clearly provides a concise explanation of this simulation result. Assuming the input limitation level of y_k is given by L_y , and by considering region $A_3(ii)$ in the algorithm, it would be expected that no update should occur if $L_y - 0.5 < 0^3$. However, when $L_y > 1.5$, which corresponds to region A_5 , the algorithm updates all the thresholds to non-zero value of $L_y + 0.5$ and $L_y + 1.5$ (see detailed explanation in Appendix B).

³At start of detection, Δm_c and Δm_d are both zero. By considering only the situation where $L_y > 0$, then in this region, when $L_y - 0.5 < 0$, the pertinent threshold(s) involved, is constantly held at zero.

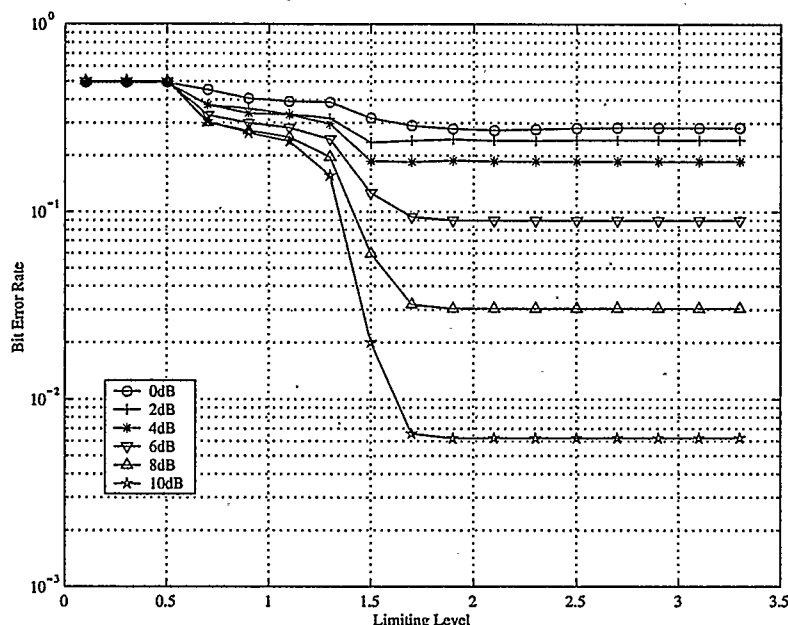


Figure 4.11: Input limitation effect on ternary dicode channel

This result compares favorably with the observation presented for the binary dicode channel [21]. Figure 4.12 shows the simulation result of a comparative study that was done for both the binary and ternary dicode channel detection algorithms. As can be seen in the binary channel, the algorithm quickly converges to its steady minimum error rate value once L_y exceeds 0.5 but L_y has to be greater than three times this value before convergence can occur for the ternary channel.

It can also be observed that once the steady minimum error rate value is reached, the ternary channel performance is worse than that for the binary channel, and this can be directly attributed to the difference in error-event probability in the two channels, and not the detection algorithm itself. This observation is an expected result and will be further explained later. Meanwhile, suffice to say that these simulation results have revealed the appropriate input limitation necessary to fully utilize the capability of the detection algorithm while avoiding unwarranted performance degradation.

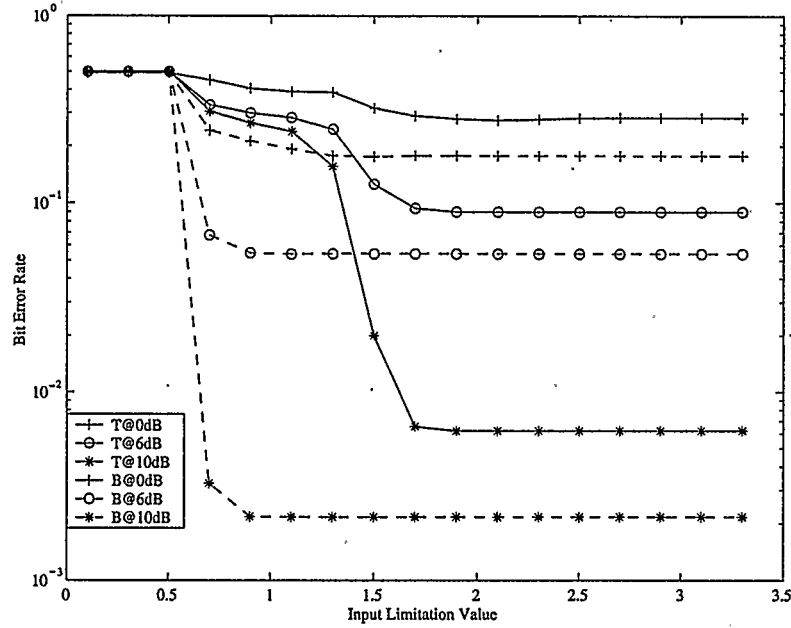


Figure 4.12: Limitation effect: Binary Channel(B) vs. Ternary Channel(T)

4.2.2 Ternary Dicode: Signal Swing Limitation

The second stage in determining the sensitivity of the proposed detection algorithms to saturation effects in an analog implementation, was done by testing for the effect of varying signal swing at constant signal-to-noise ratio. In this case, the algorithm was mapped to a conceptual circuit implementation with limited signal swing of $V_{max} - V_{min}$, (algorithm to circuit mapping will be discussed later). V_{max} and V_{min} would imply the power supply for a rail-to-rail operation or otherwise, the circuit limit of operation.

Figure 4.13 shows the obtained simulation result from MATLAB. From this result, it can be observed that at constant signal-to-noise ratio, regardless of the dynamic range changes, the error-rate remains the same. This invariably suggests a low sensitivity of the algorithm to dynamic range limitation. Furthermore, at each

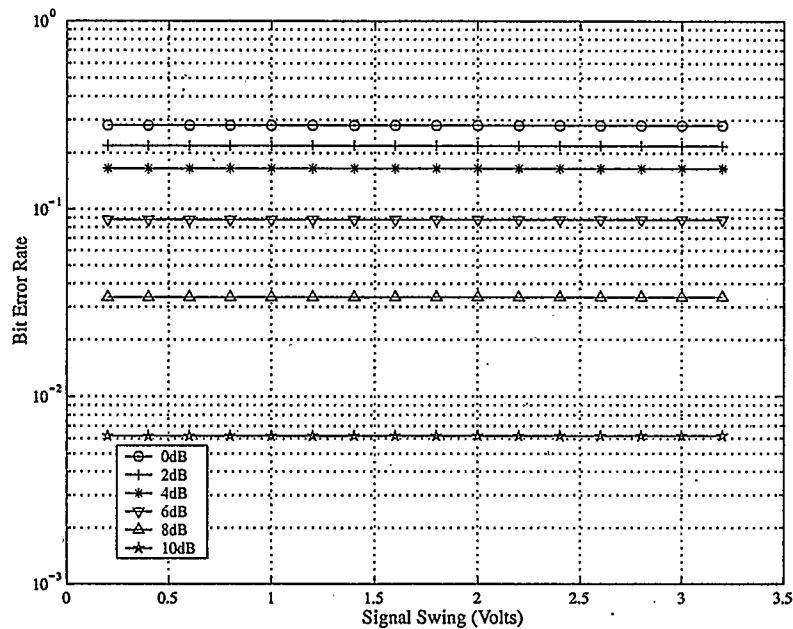


Figure 4.13: Signal swing limitation effect on ternary dicode Channel

signal-to-noise ratio, the algorithm yields the same error-rate that was obtained at the steady state when the input was limited. This reinforces the robustness of the detection algorithm to saturation effects.

4.2.3 Ternary Duobinary: Input Limitation

From the ternary dicode channel detection point of view, the proposed algorithm can effectively attain a steady minimum error-rate value, if a proper input limitation level is chosen to avoid unnecessary degradation in its implementation. Nonetheless, it may still be worthwhile to investigate the ternary duobinary channel detection algorithm also, so as to establish its degree of sensitivity to saturation effect. Therefore, a similar test was carried out and the obtained simulation result is shown in Figure 4.14.

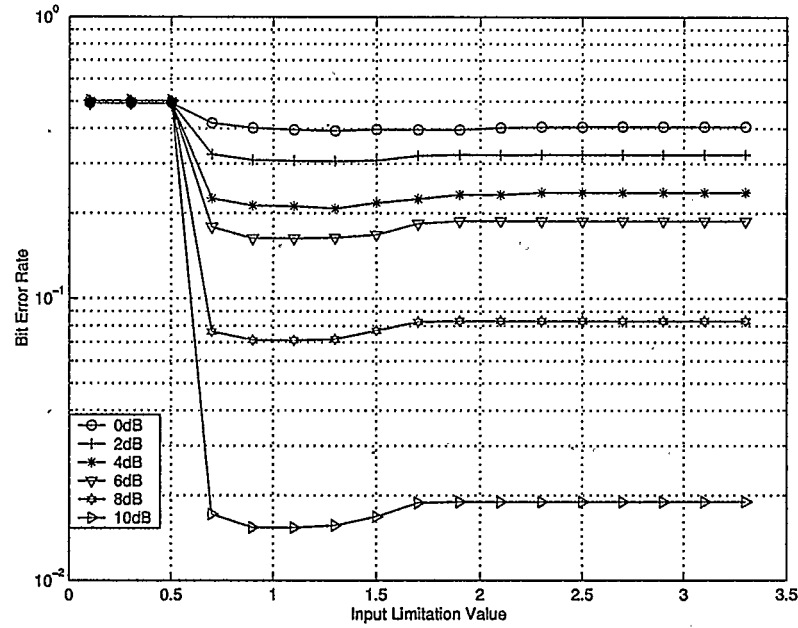


Figure 4.14: Input limitation effect on ternary duobinary channel

As might be expected, the algorithm also settles with a rapid convergence to its steady error rate value only after the input limitation level L_y , exceeded 1.5. The closeness in behavior of the duobinary detection algorithm to that of its counterpart (dicode) can be explained by noting that the two algorithms are indeed very similar, with the same boundaries except for the difference in valid survivor in each allowable region of operation. Thus it might be safe to conclude that this algorithm is also robust to saturation effects.

4.3 Error Rate Performance

The evaluation of the detection algorithms is not complete without investigating error-rate performance; since this usually serves as a benchmark for algorithm testing and verification. For this test on both ternary channels, the respective algorithm was

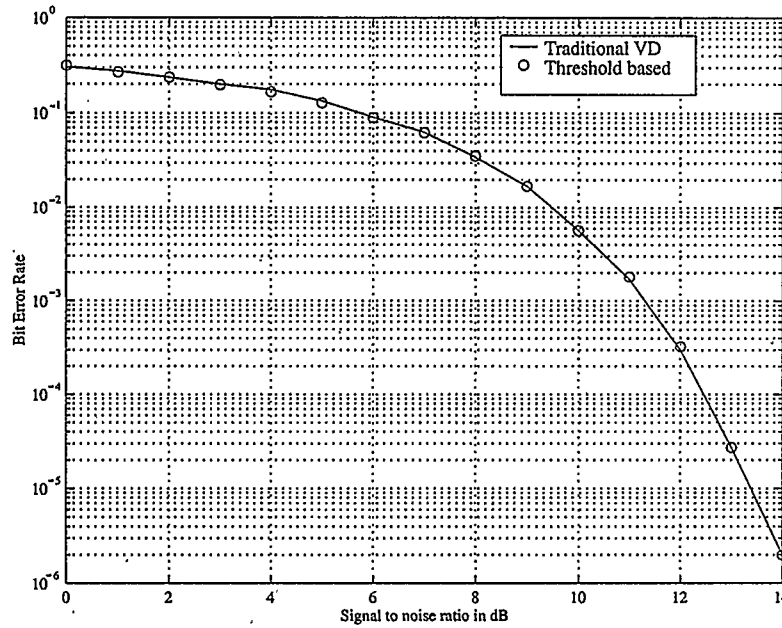


Figure 4.15: Error rate performance: Ternary dicode channel

simulated in MATLAB. Also, the traditional Viterbi detector without path metrics simplification or normalization was simulated at several signal-to-noise ratios and the obtained results are shown in Figures 4.15 and Figure 4.16.

As can be seen that the results exhibit a very high degree of closeness between the derived algorithms and the typical Viterbi detector for both channels, this further authenticates the correctness of the derived algorithms.

4.3.1 Ternary Channel Detection vs. Binary Channel Detection

The fundamental partial-response signaling channels are subject to error-event occurrence, however, the degree/type of error-event occurrence varies from one type of

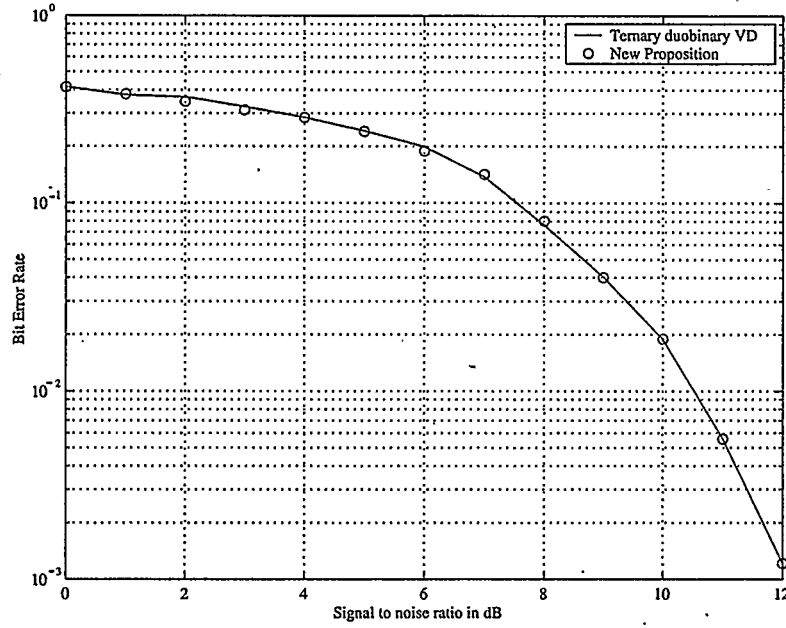


Figure 4.16: Error rate performance: Ternary duobinary channel

channel to the other.

The transmission mode in the channel could also lead to varying degrees of error-event occurrence. This section focusses on the error-event occurrence in both the binary and the ternary dicode channel (just to serve as an illustration).

Generally, error-event occurrence in PRS channels has been studied [50]. The error rate performance at high signal-to-noise ratio for the dicode channel can be given as:

$$P_{\text{error-event/symbol}} = K_{\text{error-event/symbol}} Q\left(\frac{d_{\min}}{2\sigma_N}\right). \quad (4.5)$$

where $P_{\text{error-event/symbol}}$ is the probability of error-event/symbol error occurrence, d_{\min} is the minimum distance, σ_N is the noise variance, $K_{\text{error-event/symbol}}$ is a minimum distance dependent constant, and $Q(\cdot)$ is the Q-function or the cumulative Gaussian

distribution, which is defined as:

$$Q(n) = \int_n^\infty \left(\frac{1}{\sqrt{2\Pi}} \right) e^{-\frac{n^2}{2}} dn. \quad (4.6)$$

For the symbol error rate probability P_{symbol} , it can be calculated from three factors [25], which are:

- probability of minimum distance event occurrence.
- probability of the minimum distance events being supported by the data sequence.
- number of errors in the event.

However, for the error-event probability $P_{error-event}$, only the first two factors are sufficient. In terms of the number of states present in a channel, P_{symbol} can be expressed as:

$$P_{symbol} = 2S_t(S_t - 1)Q\left(\frac{d_{min}}{2\sigma_N}\right) \quad (4.7)$$

where S_t is the number of states in the channel [7] and the noise variance (σ_N) is

$$\sigma_N = \left(\frac{10^{-0.05SNR_{dB}}}{\sqrt{2}} \right). \quad (4.8)$$

Thus for the binary and ternary dicode channel, K_{symbol} is 4 and 12, respectively. Also, for a normalized input $s_k \in \{\pm 1, 0\}$ as assumed in the derivation of the algorithms, the minimum distance of an error event in both the ternary and binary dicode channels results in $d_{min} = \sqrt{2}$.

Therefore, for the same signal-to-noise ratio, it is expected that the performance of the ternary dicode channel will be worse than that of the binary channel. This is so because the ternary channel has more states that could support an error event

occurrence than the binary channel and thus results in the increased probability of minimum distance error event occurrence from 0.5 to approximately 0.67 [17].

Figure 4.17 shows the asymptotic error rate curve in comparison with the error rate curves for both the binary and ternary decode Viterbi detector. This result shows the degree of accuracy of the new algorithm and it also confirms the expected degradation in performance of the ternary channel in comparison to the binary channel.

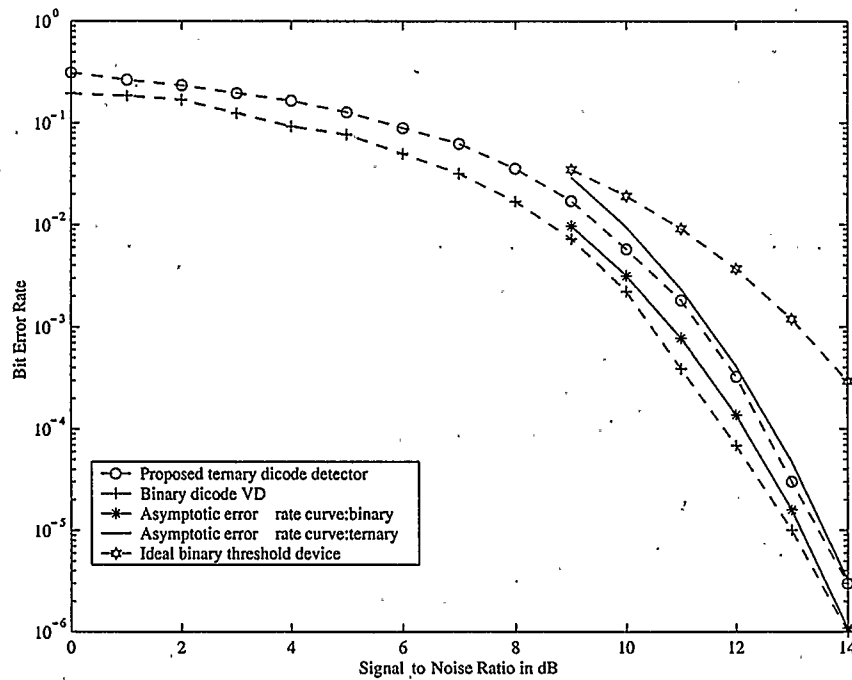


Figure 4.17: Error-rate comparison

4.3.2 New technique vs. Friedmann's technique

The proposed algorithm for the detection of a ternary decode channel was compared with the technique that previously existed in the literature (Friedmann et. al) [27]. The benefits of the new proposition are summarized in Table 4.1.

Table 4.1: Comparison of ternary dicode channel detection techniques

Criteria	Friedmann et al. [27]	New Proposition
Number of thresholds	2	2
Threshold definition	$T_1 = 0.5((S_{-1} - S_0) + 1)$	$\Delta m_d = m_0 - m_{-1}$
	$T_3 = 0.5(S_0 - 1)$	$\Delta m_c = m_{+1} - m_0$
Hardware Implementation	6 comparators needed	6 comparators needed
	11 quantities computed at every time step	7 quantities computed (of comparable complexity)
Starting condition	$ T_1 = T_3 = 0.5$ (non-zero) $sign(T_1) = -sign(T_3)$	$\Delta m_d = \Delta m_c = \text{zero}$ since all state metric assume equal value
Number of regions	7	5

The algorithms were simulated in MATLAB with the necessary comparisons and updates done as appropriate in each case, in order to compare the two techniques. Figure 4.18 shows the simulation result obtained from the two techniques using the same input data (small sample size of 25) while Figure 4.19 shows the bit error rate curve for the two techniques.

From the simulation results obtained, it could be inferred that the new proposition achieves the same error rate performance as obtainable from the technique of Friedmann. It does so with a lower signal swing for the propagating differential signals

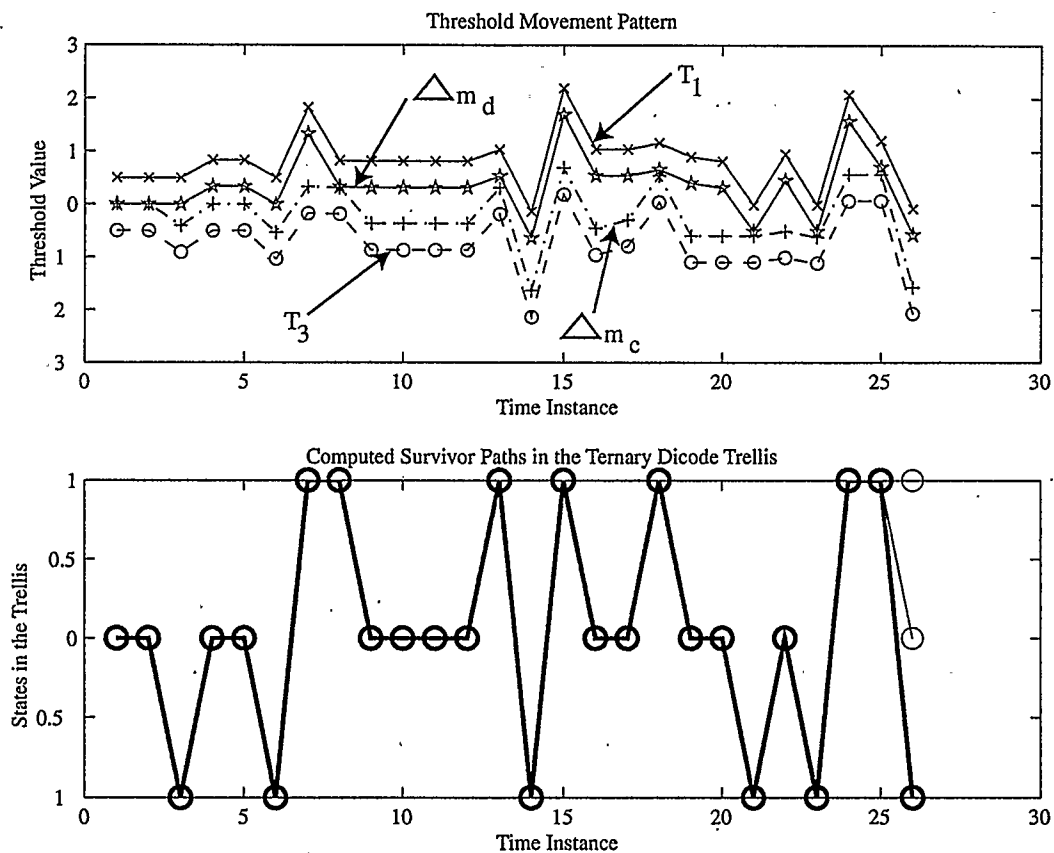


Figure 4.18: Threshold adaption comparison

and it will also required less hardware in its implementation.

Moreover, it circumvents the need to have the two propagating signals (thresholds) at a separation of $\xi = 1^4$ and at opposite signs at the start of detection. The last stated advantage could be seen as the direct consequence of using a more computationally efficient threshold definition based on the true difference metric of the states in the trellis.

⁴Actually, $\xi = V_{ref}$, where in this case, V_{ref} of unity value is assumed. The choice of V_{ref} is dictated by the implementation. At the start of detection, the two thresholds, as defined in [27], take on the following values: $T_1 = 0.5V_{ref}$ and $T_3 = -0.5V_{ref}$.

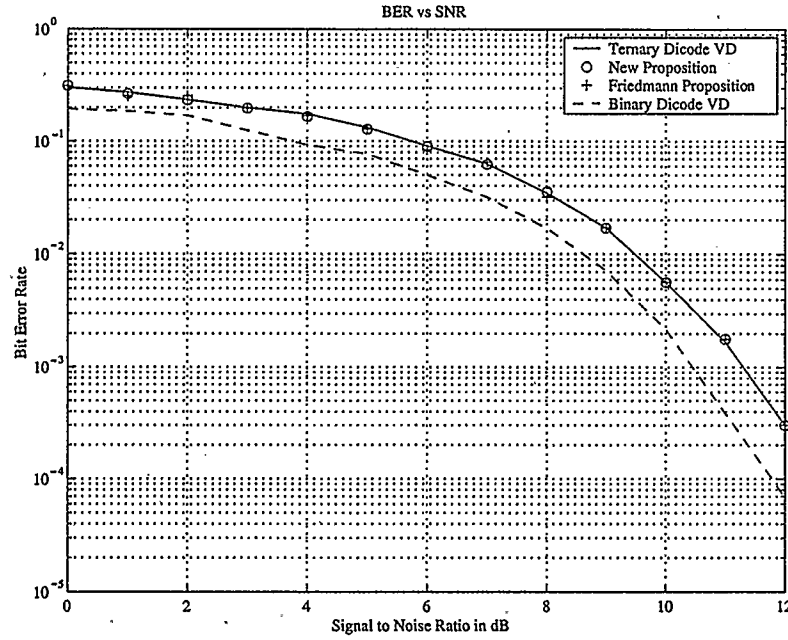


Figure 4.19: Bit error rate vs. Signal-to-noise ratio

4.4 Algorithm translation for hardware implementation

In order to develop the required hardware implementation of the algorithm, there is the need to interpret the algorithm, taking into consideration the limits and electrical parameters with which the implementation would operate.

So far, the algorithm has been interpreted using normalized transmitted symbols $s_k \in \{\pm 1, 0\}$. However, in reality, the hardware will operate on voltages and currents and not symbols. Therefore, an arbitrary reference electrical parameter (voltage) V_{ref} will henceforth be used, such that $s_k \in \{\pm V_{ref}, 0\}$.

Consider, the pertinent communication system as shown in Figure 4.20. The noisy received sample y_k , could be either negative or positive, and it could shoot above or

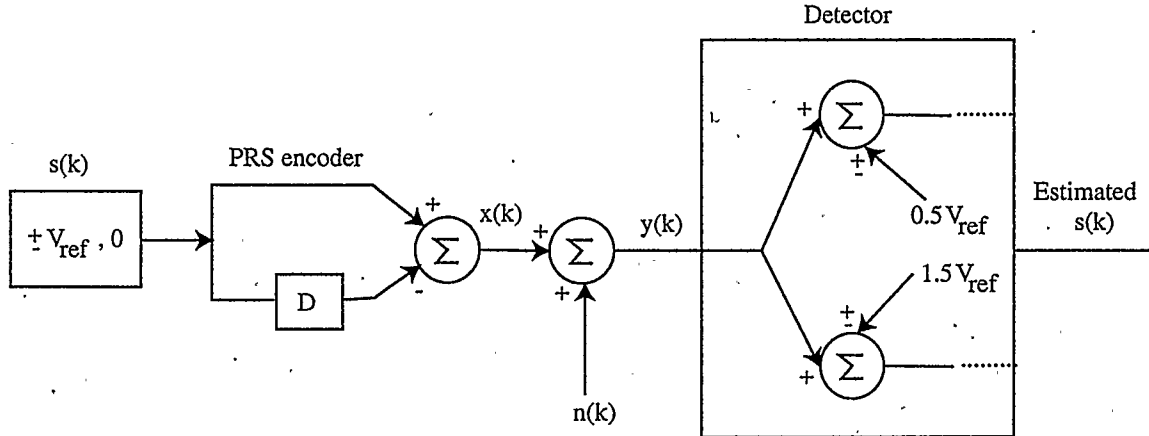


Figure 4.20: Reference communication system with detector

below the power supplies of the detector. Therefore, the received noisy signal $y(t)$ has to be mapped to the appropriate and desired hardware implementation before being fed into the detector. i.e. $y(t)$ has to be conditioned.

In order to condition $y(t)$ appropriately, a worst case scenario of $SNR = 0dB$ is considered (at this SNR, the noise and signal power are equal). Recall that;

$$y_k = x_k + n_k \quad (4.9)$$

where n_k is the noise sample at the instance of time k . Thus, the extremes of y_k are such that the upper and lower limits are $+4V_{ref}$ and $-4V_{ref}$ respectively (assuming that n_k which could range from $-\infty$ to ∞ has extreme values equal to that of the noiseless coded symbol). By taking into consideration the extremes of the shifting operation that is required in the detection algorithm (i.e. $\pm 1.5V_{ref}$), then, the possible limits of the signal that the detector would be expected to handle are $\pm 5.5V_{ref}$. This is the signal that should be mapped to the smallest signal swing ($V_{max} - V_{min}$) of the hardware.

Since the system under consideration is a linear system, the required input condi-

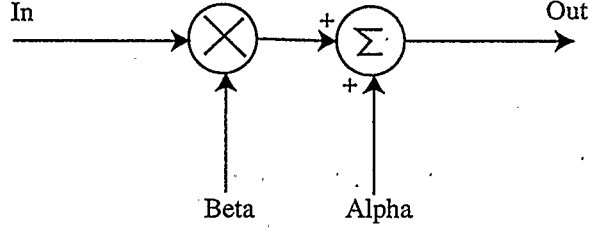


Figure 4.21: Linear Input Conditioner

tioner is a linear type as shown in Figure 4.21. This leads to the following equations:

$$\begin{aligned}\beta[y(t)_{upper} + 1.5V_{ref}] + \alpha &= V_{max} \\ \beta[y(t)_{lower} - 1.5V_{ref}] + \alpha &= V_{min}\end{aligned}\quad (4.10)$$

where β is a normalizing constant and α is an additive constant.

Thus,

$$\alpha = \left(\frac{V_{max} + V_{min}}{2} \right) \quad (4.11)$$

$$\beta = \left(\frac{V_{max} - V_{min}}{11V_{ref}} \right) \quad (4.12)$$

Rearranging the LHS of equation(4.10) yields

$$\beta[y(t) \pm 1.5V_{ref}] + \alpha = y(t)' \pm 1.5\beta V_{ref} \quad (4.13)$$

where the conditioned input into the detector hardware $y(t)'$ is given by

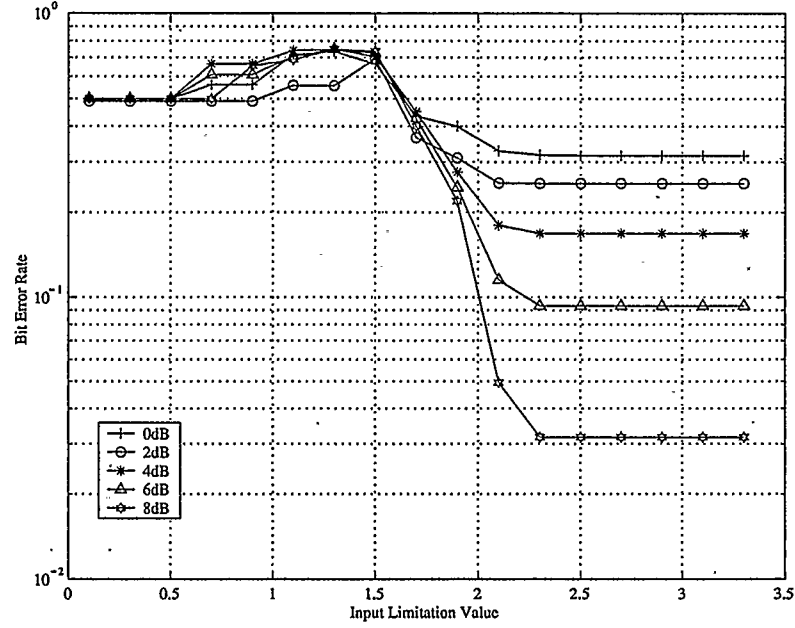
$$y(t)' = \beta y(t) + \alpha \quad (4.14)$$

implying that α is the common-mode level of $y(t)'$, and $\beta y(t)$ is the amplitude.

Hence, the maximum/minimum amplitude of the conditioned input is,

$$y(t)' = \pm \left(\frac{4}{11} \right) (V_{max} - V_{min}) \quad (4.15)$$

and this has a common-mode level of α . It means that the detector hardware would tolerate an input conditioned to about 36% of its smallest signal swing.

Figure 4.22: Limitation effect on conditioned input $y(t)'$

Moreover, the amount of level-shifting required in the detector hardware is given by:

$$\pm 0.5\beta V_{ref} = \pm \left(\frac{V_{max} - V_{min}}{22} \right) \quad (4.16)$$

and

$$\pm 1.5\beta V_{ref} = \pm 3 \left(\frac{V_{max} - V_{min}}{22} \right) \quad (4.17)$$

representing about 4.5% and 13.6% of the smallest signal swing respectively. Thus, the extremes of the shifted conditioned input is guaranteed to stay within 50% of the available signal swing above and below α .

The simulation result obtained by limiting the conditioned input between arbitrary values $V_{max} = 3.3V$ and $V_{min} = 0V$ is shown in Figure 4.22. It can be observed from the simulation result that the steady state error rate value attained without a conditioned input $y(t)$ (Figure 4.11), is the same as that of a conditioned input $y(t)'$ (Figure 4.22).

Chapter 5

Detector Design and Implementation

5.1 Architectures

It was established in chapter 4 that the proposed algorithm for the detection of ternary fundamental channels exhibits the tendency to be robust to some undesirable effects when implemented from an analog perspective. It must be noted however, that the algorithms can also be digitally interpreted and implemented but for the reasons already stated in the preceding chapter, the focus will be on an analog (mixed-signal) implementation.

The choice of an analog/mixed-signal implementation is also supported by reported works in the literature, which have not only proven the feasibility of this type of implementation (at least for a two-state PRS channel) but have also demonstrated the possibility of high-speed operation, at lower power consumption and smaller die-area in comparison to their digital counterpart.

So far, reported works have basically been implemented based on three types of architectures; one of which (the input inter-leaved architecture) is very unique and provides a novel architecture for implementing differential threshold analog detectors (for a two state dicode PRS channel) [22, 23]. However, the investigation of the applicability of the underlying principle behind the input inter-leaved architecture to the proposed ternary channel detector will not be discussed in this chapter.

The second architecture of note is that described in the work of Altarriba et. al [51] and Matthews et. al [19, 20] (referred to here as the analog feed-forward/analog feedback architecture). This architecture employs the use of control logic to sense the digital decision of a clocked analog comparator in order to output a digital signal that will activate the appropriate sampling switch to track and hold the survivor path metric. The determined analog, sampled state metric is then fed back for the computation of the state metric difference Δm , which is used to calculate the necessary path metric to be compared in the next time instance.

Even though this architecture has been reported to have worked at a maximum speed of 50Mb/s for a two-state dicode PRS channel detector using the $2\mu m$ MOSIS CMOS technology with a single 5Volt supply, three drawbacks are noticeable. These are:

- the implementation did not utilize the true difference metric algorithm effectively, since the computation of absolute state metrics was done within the implementation.
- it utilizes master-slave track and hold circuits along with the needed input track and hold circuitry.
- it uses an analog feedback for the computation of the path metrics.

The remaining architecture, referred to here as the analog feed-forward/digital feedback architecture was used by Shakiba [21] also for the implementation and verification of a two-state dicode PRS channel detector. It must be noted that a similar architecture was also proposed by Bergmans et. al [18].

In this architecture, the level-shifted input signals are constantly compared with the lone required differential threshold using fast analog clocked comparators. The comparators' digital decisions are then fed back to activate the appropriate sampling switch to track and hold one of or neither of the two level-shifted input signals which updates the differential threshold for the next detection time instance. Notably, this architecture, addresses all of the aforementioned drawbacks of the analog feed-forward/analog feedback architecture because:

- it uses the true difference metric as originally proposed by Ferguson [15].
- it eliminates the requirement for master-slave track and hold (T/H) circuits, thereby reducing the number of T/H circuits needed.
- the implementation avoids the accumulation of analog error by employing digital feedback to activate the switch that would update the metric appropriately in the feed-forward path.

The discrete implementation of this architecture has been shown to be robust to a variety of undesirable analog errors [21], while an integrated input-interleaved version using a $0.8\mu\text{m}$ BiCMOS technology with a single 3.3Volts supply has been demonstrated to operate at 100Mb/s for a binary dicode detector (twice as fast as the analog feed-forward/analog feedback approach) [22]; corresponding to 200Mb/s channel rate for the PR4 channel detector.

Therefore, based on the above observations, it is the aim of this research to extend

the use of the analog feed-forward/digital feedback architecture to the virtually unexplored territory of ternary maximum-likelihood sequence detector design; solely using a CMOS technology for proof of concept.

5.2 Detector Design

In order to demonstrate the implementation of the proposed detection algorithms, the ternary dicode detector will be implemented to serve as a proof of concept. As already pointed out in table 4.1 of the previous chapter, the implementation will require the use of six comparators and only seven analog quantities need be computed at each detection time instance; which are $y'_k, y'_k \pm 0.5, y'_k \pm 1.5, \Delta m_c$ and Δm_d . Thus, based on the principle of the analog feed-forward/digital feedback architecture, and a meticulous observation of the algorithm, a voltage-mode ternary dicode detector architecture is shown in Figure 5.1¹ without the survivor memory circuitry.

The detector uses a single digital block (control signal generator) and simple analog functional blocks such as a simple track and hold circuits, buffers, summer or level-shifter, a cross-over multiplexed track and hold circuit, and clocked analog comparators.

5.2.1 Input Track and Hold circuit

As already pointed out in the previous chapters, analog Viterbi detectors need only a moderate resolution of 6-bits; thus simple analog building blocks can be used. In this detector implementation, an open-loop track and hold circuit was used to obtain

¹see Appendix D for an updated version

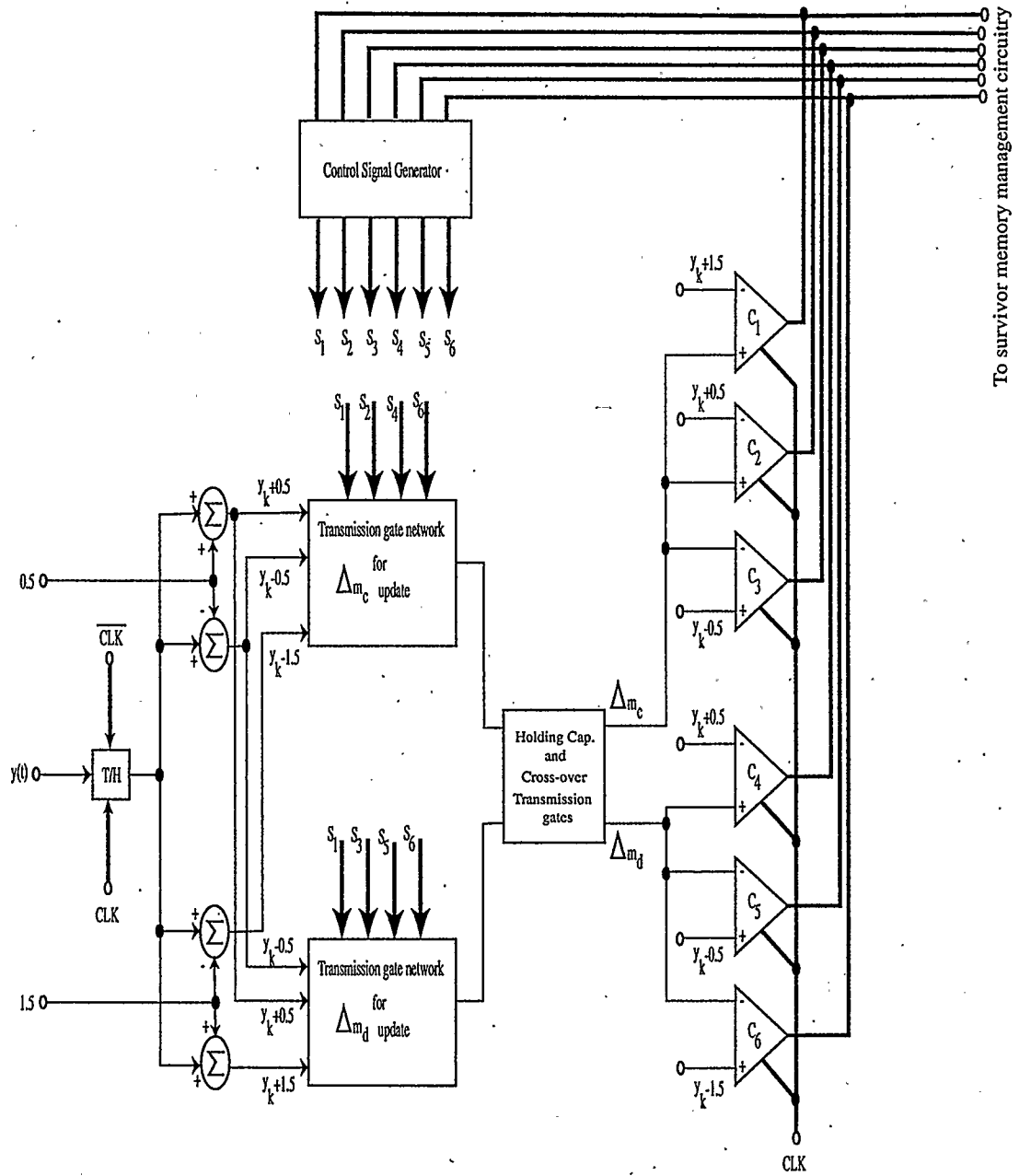


Figure 5.1: Ternary dicode detector architecture

the needed noisy sample y'_k from the conditioned input y'_t . However, it is known that this type of T/H circuit suffers from clock feed-through effects and signal-dependent hold step $V_{hold-step}$ which is given by:

$$|V_{hold-step}| = \left(\frac{C_{ox} W_{in} L_{in} (V_{dd} - V_{th} - y'_t)}{2C_{hold(in)}} \right) \quad (5.1)$$

where C_{ox} is the transistor gate oxide capacitance, W_{in} is the width of the sampling transistor, L_{in} is the length of the sampling transistor, V_{dd} is the power supply, V_{th} is the process dependent sampling transistor threshold, y'_t is the conditioned analog input, and $C_{hold(in)}$ is the input holding capacitance.

Although $V_{hold-step}$ varies linearly with the input y'_t , and it is also inversely proportional to the holding capacitance $C_{hold(in)}$, the sampling transistor aspect ratio can be chosen to try and keep the hold step small even with the use of a small holding capacitance. A small holding capacitance will help to ensure that the encountered time constant in the sampling mechanism is small enough to aid a high speed operation. Also, in order to minimize charge injection, a dummy transistor, half the size of the input sampling transistor was used. It should also be noted that a limited input y'_t , would aid minimizing the hold-step too. Moreover, differential signalling can be very useful in minimizing the undesirable effects of the track and hold mechanism. Figure 5.2 shows the track and hold circuit that was employed.

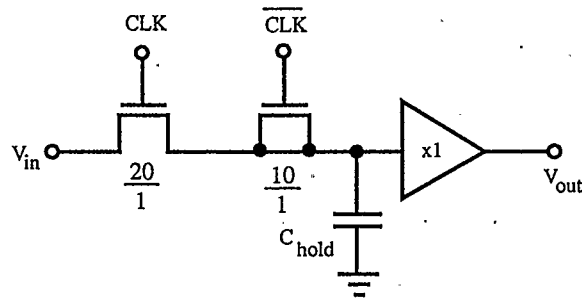


Figure 5.2: Input Track and Hold Circuitry

5.2.2 Buffer circuitry

In order to prevent the other stages in the detector implementation from loading the input T/H circuit, a buffer circuit was used. Since buffering at high-speed is needed, a transconductance circuit can be used to implement the buffer [52]. Figure 5.3 shows the block diagram of the buffer circuitry.

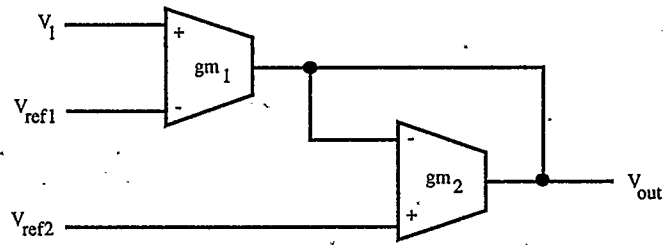


Figure 5.3: Buffer block diagram

A small-signal analysis of this diagram reveals the condition under which it can act as a buffer. The small-signal diagram is presented in Figure 5.4.

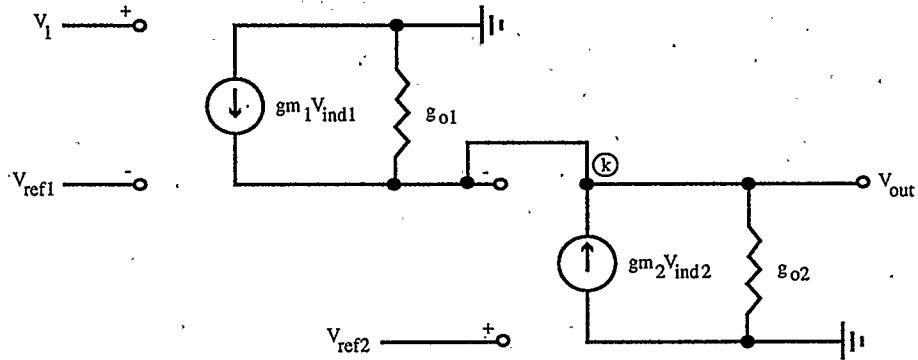


Figure 5.4: Buffer small-signal diagram

Considering node k in the diagram, and applying kirchoff's current law, we have:

$$g_{m1}V_{ind1} + g_{m2}V_{ind2} = g_{o1}V_{out} + g_{o2}V_{out} \quad (5.2)$$

where $V_{ind1} = V_1 - V_{ref1}$ and $V_{ind2} = V_{ref2} - V_{out}$.

This implies that,

$$(g_{o1} + g_{o2} + g_{m2})V_{out} = g_{m1}V_1 + g_{m2}V_{ref2} - g_{m1}V_{ref1}. \quad (5.3)$$

Assuming that $g_{m2} \gg g_{o1} + g_{o2}$, then the output voltage is given by:

$$V_{out} = \left(\frac{g_{m1}}{g_{m2}} \right) V_1 + V_{ref2} - \left(\frac{g_{m1}}{g_{m2}} \right) V_{ref1}. \quad (5.4)$$

Thus, if $g_{m1} = g_{m2}$, then,

$$V_{out} = V_1 + V_{ref2} - V_{ref1}. \quad (5.5)$$

Therefore as long as V_{ref1} and V_{ref2} are equal, the output voltage would follow the input voltage, however with a gain error which may occur if g_{m2} is not greater than the sum of g_{o1} and g_{o2} , and if $g_{m1} \neq g_{m2}$.

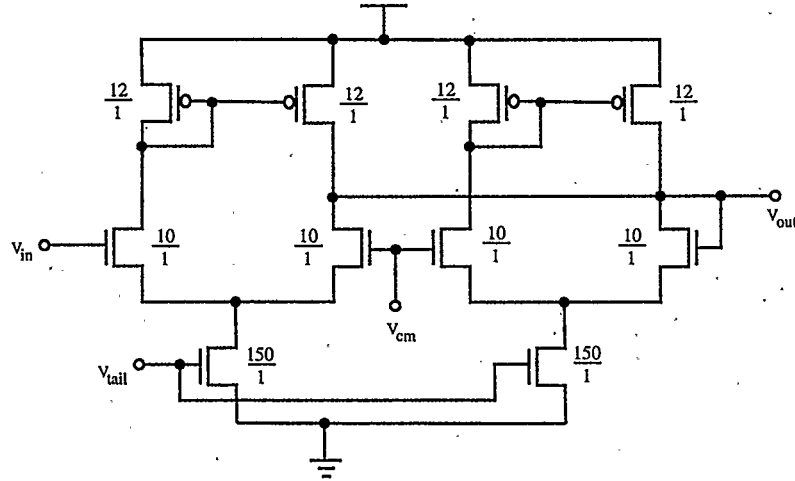


Figure 5.5: Actual Buffer Circuit

The actual circuit implementation uses the simple differential pair to implement the transconductor block and is shown in Figure 5.5. The dc-response of the buffer operating with a single supply voltage of 3.3V, using the 3V 0.18 μ m technology is

shown in Figure 5.6, which exhibits a linear output characteristic between V_{min} and V_{max} of approximately 1.2V and 2.1V respectively. This linear output should be obtained within a maximum input signal swing of 400mV as shown by the simulated dc response of the buffer.

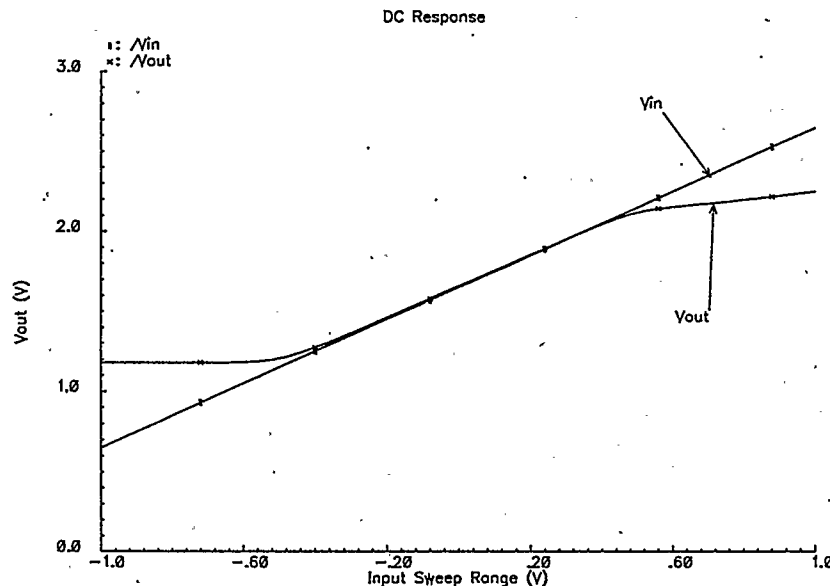


Figure 5.6: Simulated buffer dc response

To further investigate the operation of the circuit as a buffer, several transient simulations were performed with varying input signal swings. Figure 5.7 shows the circuit response to an excitation of 600mV at 100MHz, which is beyond the expected linear range of operation.

As can be observed, the signal distortion is extremely high and the buffer output is clipped. With a reduced input excitation of 400mV, the buffer output tries to follow the input signal, however with some offset and phase shift. This is shown in Figure 5.8. However, with a 200mV input swing, the distortion is much more reduced as shown in Figure 5.9.

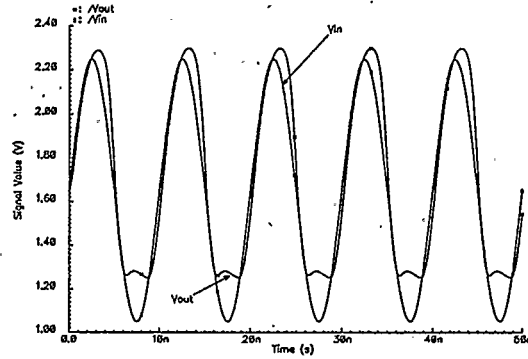


Figure 5.7: Simulated buffer transient response @ $V_{in} = 600mV$, frequency 100MHz

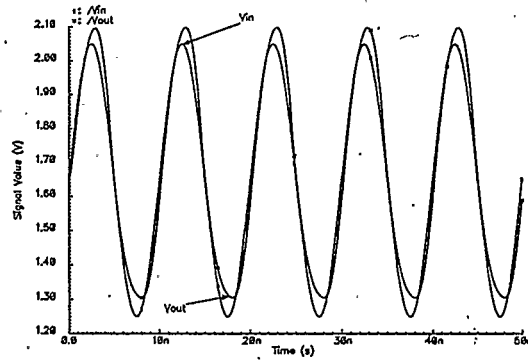


Figure 5.8: Simulated buffer transient response @ $V_{in} = 400mV$, frequency 100MHz

5.2.3 Level-shifting circuitry

Based on the limited range of linearity of the buffer circuitry and the shifting requirement of the algorithm, the level-shifting circuitry is constrained to ensure a steady shifting of the sampled input by $\pm 0.5\beta V_{ref}$ and $\pm 1.5\beta V_{ref}$.

These translate to approximately $\pm 40.9mV$ and $\pm 122.7mV$ respectively, by substituting $V_{min}=1.2V$ and $V_{max}=2.1V$ in Equation (4.16) and (4.17). These voltages are not enough to surmount the transistor threshold in a $0.18\mu m$ CMOS process.

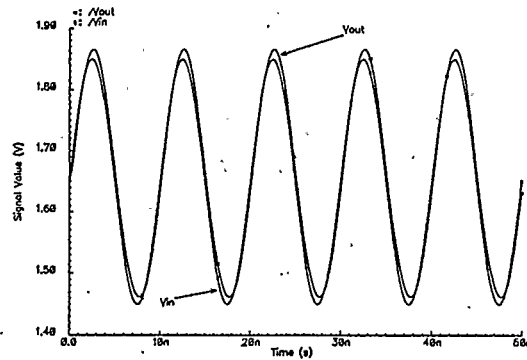


Figure 5.9: Simulated buffer transient response @ $V_{in} = 200mV$, frequency 100MHz

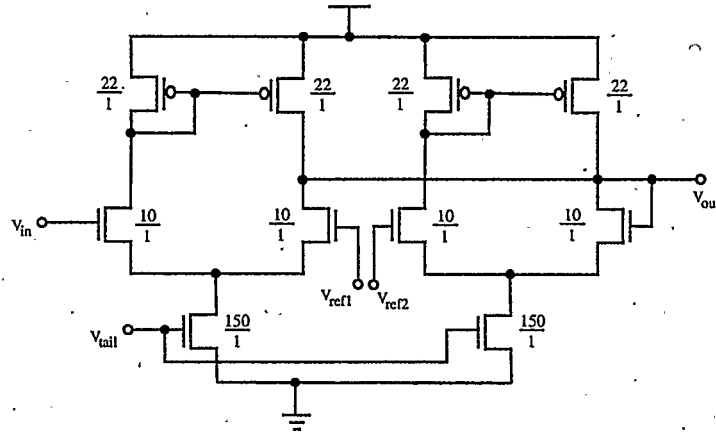


Figure 5.10: Level Shifting Circuit

However, Equation (5.5), will prove invaluable in achieving the required shifting operation since the difference between V_{ref1} and V_{ref2} can be used to provide these voltages for down-shifting or up-shifting the sampled input. Figure 5.10 shows the level-shifting circuit.

5.2.4 The clocked comparator

Analog comparators are used to serve as interface between the analog blocks employed to carry out the necessary analog signal processing in the detector and the digital survivor memory management circuitry.

Functionally, an analog comparator is expected to provide full-scale digital signal for driving digital logic from very small difference in the available input analog signals [53, 54]. For the application at hand, high-speed operation at good resolution is desirable even at extremely small differences in the input signal, in order to ensure that the detector does not make erroneous decisions.

The characteristic behavior of a differential pair in response to a differential input can actually be exploited to implement the comparator. Consider a typical differential pair as shown in Figure 5.11. If the differential input voltage V_{ind} should change polarity, then the single-ended output at node k , will respond accordingly at a gain value determined by the transconductance of the input transistors and the output conductance of the differential pair. Thus, if two of such differential pairs are connected using the same input but at opposite polarity, then two pre-amplified single-ended outputs of opposite sense are obtainable.

The pre-amplified outputs can then be directly used to drive positive feedback circuitry which will be able to provide additional gain for the input signal, to drive a latch that is needed to obtain full-scale digital signal levels. Figure 5.12 depicts the actual comparator circuit with the appropriate reset switches.

In the track mode (i.e. when the comparator clock goes high), the positive feedback circuitry is disabled, and the back-to-back inverters are prevented from competing, thus both outputs of the comparator are reset to logic '0'. In the latch mode

however (i.e. when the comparator clock goes low), the positive feedback circuit as well as the back-to-back inverters are enabled and competition quickly proceeds to ensure a correct comparator decision in response to the direction of change in V_{ind} even at very small differences in input signal values.

A small-signal representation of the designed comparator is shown in Figure 5.13.

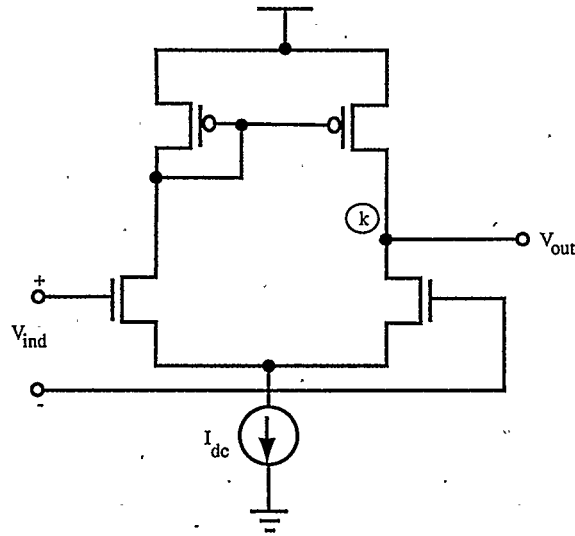


Figure 5.11: A typical differential pair

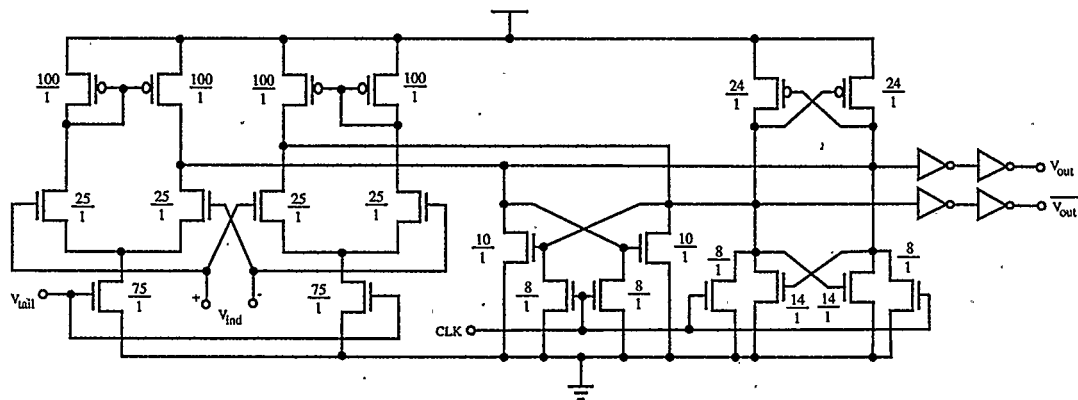


Figure 5.12: Clocked comparator circuit

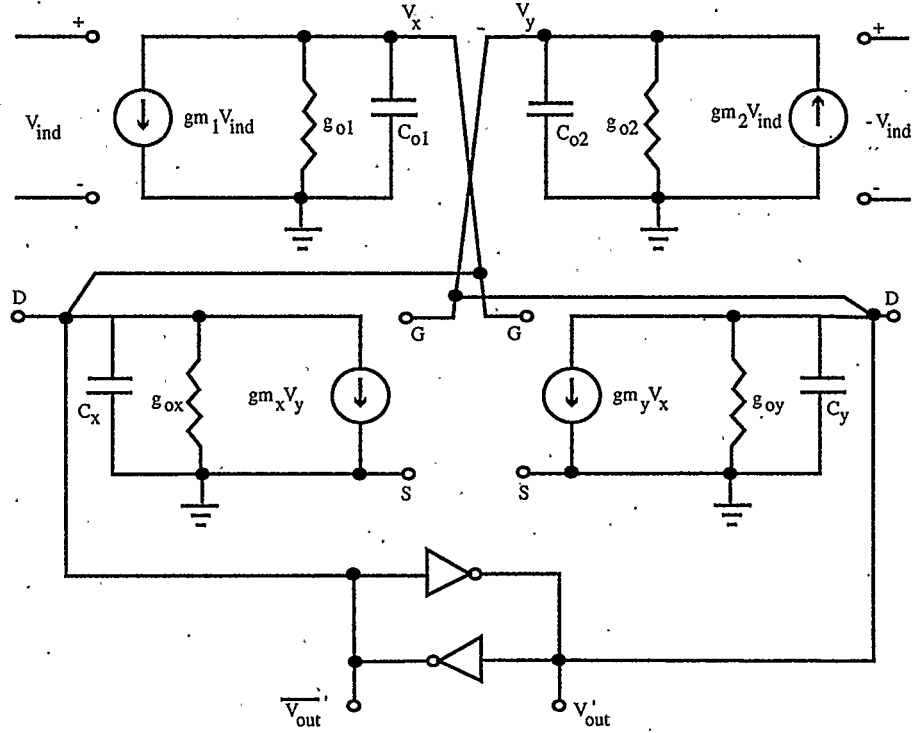


Figure 5.13: Comparator small-signal representation

Applying Kirchoff's current law at node x , we have:

$$\begin{aligned} -g_{m1}V_{ind} &= g_{o1}V_x + sC_{o1}V_x \\ V_x &= -\left(\frac{g_{m1}}{g_{o1} + sC_{o1}}\right)V_{ind}. \end{aligned} \quad (5.6)$$

Also at node y , we have:

$$\begin{aligned} g_{m2}V_{ind} &= g_{o2}V_y + sC_{o2}V_y \\ V_y &= \left(\frac{g_{m2}}{g_{o2} + sC_{o2}}\right)V_{ind}. \end{aligned} \quad (5.7)$$

Moreover,

$$\begin{aligned} -g_{my}V_x &= g_{oy}V'_{out} + sC_yV'_{out} \\ V'_{out} &= -\left(\frac{g_{my}}{g_{oy} + sC_y}\right)V_x \end{aligned}$$

$$= \left(\frac{g_{m1}g_{my}}{g_{o1}g_{oy}} \right) \frac{1}{\left(1 + s\frac{C_{o1}}{g_{o1}}\right) \left(1 + s\frac{C_y}{g_{oy}}\right)} V_{ind}. \quad (5.8)$$

Also,

$$\begin{aligned} -g_{mx}V_y &= g_{ox}\overline{V'_{out}} + sC_x\overline{V'_{out}} \\ \overline{V'_{out}} &= -\left(\frac{g_{mx}}{g_{ox} + sC_x}\right)V_y \\ &= -\left(\frac{g_{m2}g_{mx}}{g_{o2}g_{ox}}\right) \frac{1}{\left(1 + s\frac{C_{o2}}{g_{o2}}\right) \left(1 + s\frac{C_x}{g_{ox}}\right)} V_{ind}. \end{aligned} \quad (5.9)$$

Therefore, if $g_{o1} = g_{o2} = g_{oA}$, $g_{ox} = g_{oy} = g_{oB}$, $C_{o1} = C_{o2} = C_{oA}$, $C_x = C_y = C_{oB}$, $g_{m1} = g_{m2} = g_{md}$ and $g_{mx} = g_{my} = g_{mn}$, then

$$\begin{aligned} V'_{out} &= -\overline{V'_{out}} \\ &= A_{gain} \frac{1}{(1 + s\tau_1)(1 + s\tau_2)} V_{ind}. \end{aligned} \quad (5.10)$$

where

$$A_{gain} = \left(\frac{g_{md}g_{mn}}{g_{oA}g_{oB}} \right) \quad (5.11)$$

$$\tau_1 = \frac{C_{oA}}{g_{oA}} \quad (5.12)$$

$$\tau_2 = \frac{C_{oB}}{g_{oB}} \quad (5.13)$$

The operational input signal range for the comparator can be estimated by:

$$V_{in(min)} \geq \sqrt{\frac{I_{tail}}{2K_n \left(\frac{W_{in}}{L_{in}}\right)}} + \sqrt{\frac{I_{tail}}{K_n \left(\frac{W_{tail}}{L_{tail}}\right)}} + V_{tn(in)} \quad (5.14)$$

$$V_{in(max)} \leq V_{dd} - \sqrt{\frac{I_{tail}}{2K_p \left(\frac{W_{p-load}}{L_{p-load}}\right)}} - |V_{tp}| + V_{tn(in)} \quad (5.15)$$

where $K_{n/p} = \frac{1}{2}\mu_{n/p}C_{ox(n/p)}$.

This implies that the estimated input signal range for the comparator based on the

3V 0.18 μ m technology used is:

$$0.74V \leq V_{in(comp)} \leq 3.2V. \quad (5.16)$$

Figure 5.14 shows the simulated comparator decision for an input difference of 1mV over the estimated range at a clock speed of 100MHz . The actual observable input range for the comparator according to the simulation is $0.8V < V_{in} < 3.0V$. By resolving a 1mV input voltage difference, over this range for frequencies up to 100MHz, the comparator exhibits a 10-bit resolution using a single 3.3V supply; which is in excess of the stipulated requirement for an analog Viterbi detector.

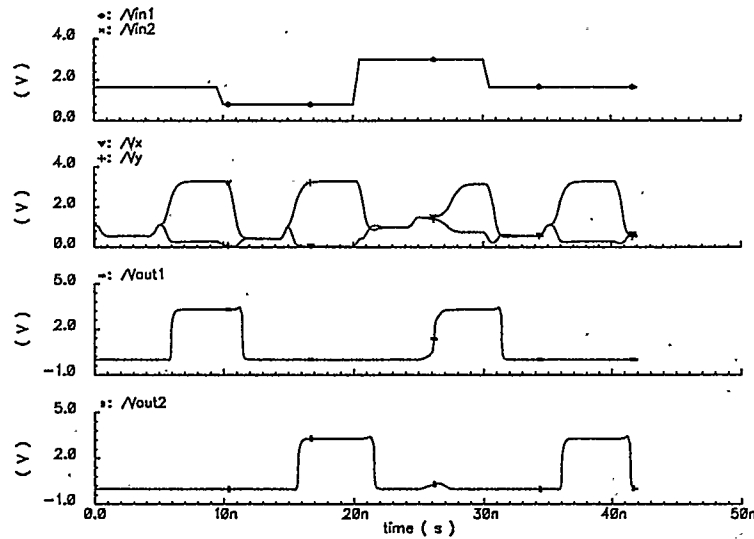


Figure 5.14: Comparator decision for $0.8V < V_{in} < 3.0V$

5.2.5 The Cross-over Multiplexing T/H

The cross-over multiplexing track and hold mechanism interfaces the level-shifters to the six comparators for simultaneous update of the two differential thresholds Δm_c

and Δm_d . It comprises two holding capacitors, two buffers, feed-forward transmission gate networks and two cross-over transmission gates.

Figure 5.15² shows the mechanism with the activating control signals. The activating signals for the transmission gates are obtained from the control signal generator, which output the digital signals based on the decisions of the comparators.

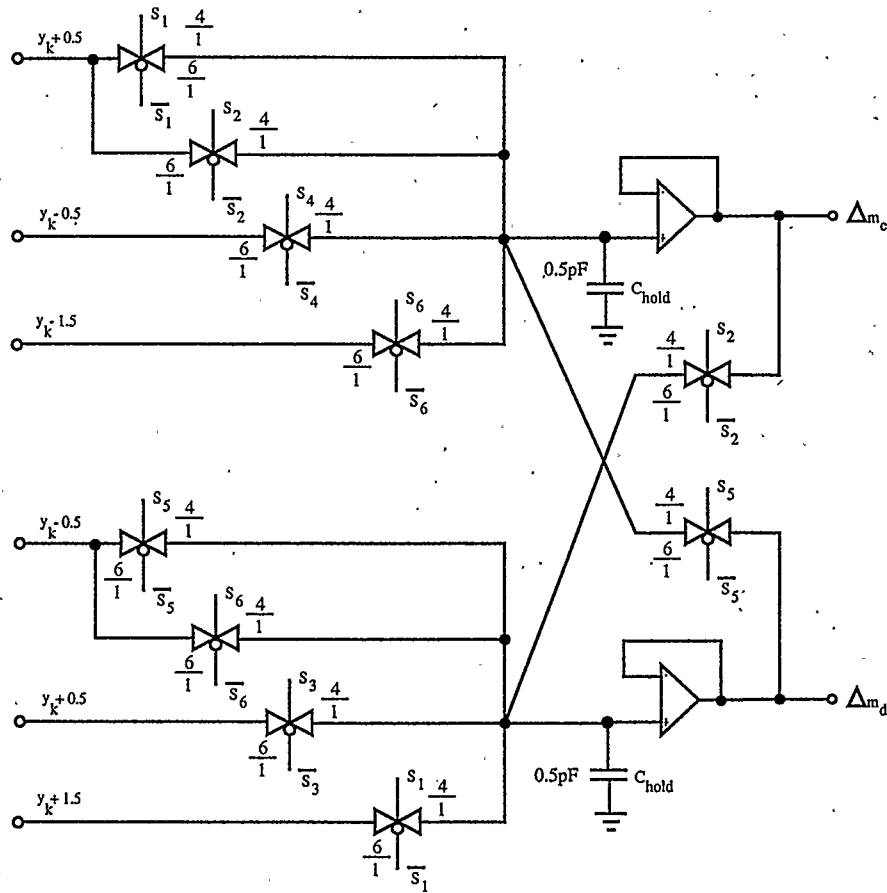


Figure 5.15: Cross-over multiplexing T/H mechanism

The cross-over multiplexing track and hold mechanism ensures that the differential thresholds are updated appropriately for the correct signal processing to take place.

²see Appendix D for an updated version

The mechanism holds the previous differential thresholds' values when the comparators are in their track mode. In this mode, both the feed-forward and cross-over transmission gates are disabled.

However, during the latch mode of the comparators, the appropriate control signals are generated, thereby making the mechanism to track the signals updating the thresholds. It is during this mode, that cross-over transmission gates can be activated (either one or both) or not as appropriate.

5.2.6 Control Signal Generator

The control signal generator is very critical in the detector operation since, an incorrectly generated control signal would send the detector into making wrong decisions; which would result in an extremely high number of errors.

Table 5.1³ summarizes the mapping of the comparator decisions to the region detected in the algorithm. Based on this table, and the requirement that an all zero six bit address should be generated by this block during the track mode of the comparators, the required logic operation for each region can be derived and the control signal generator can then be designed. Figure 5.16⁴ shows the designed control signal generator.

5.2.7 Clock Generator

A clock generator was designed to provide the needed on-chip clock for the detector from an off-chip fifty percent duty cycle clock. Two clock signals are needed for the

³see Appendix D for an updated version

⁴see Appendix D for an updated version

Table 5.1: Logic operations for control signal generator

Region	Comparator Output needed	Required Logic Operation
A_1	$C_1 = 1$	C_1 (appropriately delayed to minimize timing mis-match)
A_2	$C_1 = 0$ and $C_2 = 1$	$\overline{C_1 + C_2}$
$A_3(i)$	$C_2 = 0, C_3 = 0$ and $C_4 = 1$	$\overline{C_2 + C_3 C_4}$
$A_3(ii)$	$C_3 = 0$ and $C_4 = 0$	No additional logic required
$A_3(iii)$	$C_3 = 1, C_4 = 0$ and $C_5 = 0$	$\overline{C_3 C_4} + C_5$
A_4	$C_5 = 1$ and $C_6 = 0$	$\overline{C_5 + C_6}$
A_5	$C_6 = 1$	C_6 (appropriately delayed to minimize timing mis-match)

detector operation, which are the main clock used to activate the input sampling switch as well as the comparators, and an opposite phase clock, used to control the input dummy switch.

The main clock (CLK) ensures that while the input signal is being sampled (i.e. data not ready), the comparators are reset. When data is now ready (i.e. input sample held), the comparators are made to operate in the latch mode. Figure 5.17 shows the on-chip clock generator.

5.2.8 Bias Circuitry

The bias circuitry was designed to work using an off-chip current source providing $50\mu A$ current input. Based on the common-mode level (α), and the specified signal limit ($V_{min} = 1.2V$ and $V_{max} = 2.1V$), the amount of level shifting required in the

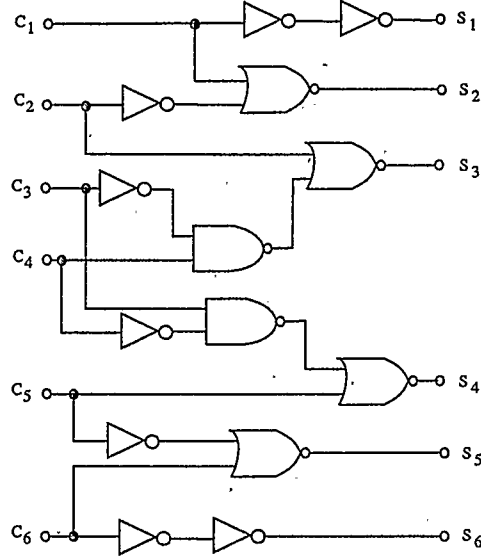


Figure 5.16: Control signal generator

detector, $\pm 0.5\beta V_{ref}$ and $\pm 1.5\beta V_{ref}$ can be calculated.

The bias circuitry generates the needed voltage levels for shifting as $V_{cm} + 0.5\beta V_{ref} \approx 1.691V$, $V_{cm} - 0.5\beta V_{ref} \approx 1.61V$, $V_{cm} + 1.5\beta V_{ref} \approx 1.773V$ and $V_{cm} - 1.5\beta V_{ref} \approx 1.527V$. The differential pairs tail biasing voltages are also appropriately generated. Figure 5.18 shows the bias circuit.

5.3 Detector simulation results

All the above designed building blocks were put together to implement the ternary dicode PRS detector (without the survivor memory management section). Using a build to verify approach, a $55.2K\Omega$ resistor was placed between V_{dd} and the input current mirror of the biasing circuitry to mimic the generation of the $50\mu A$ off-chip current source.

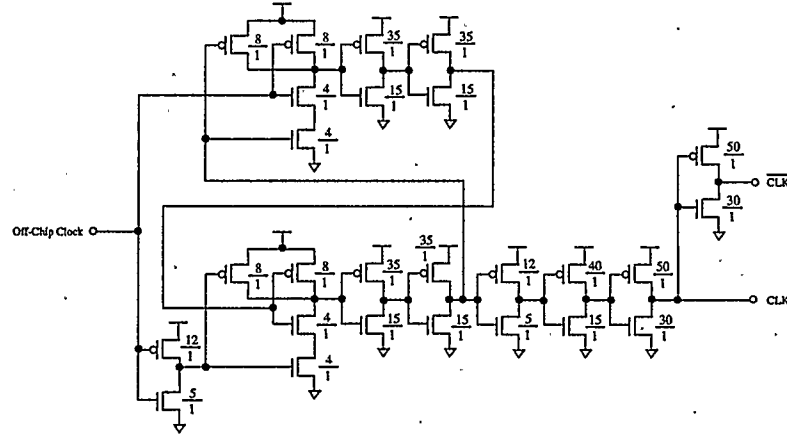


Figure 5.17: On-Chip clock generator

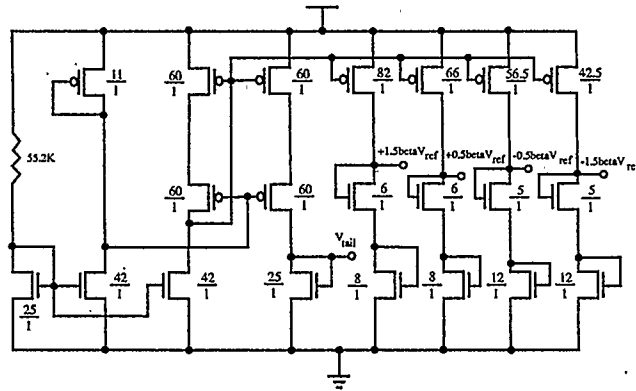


Figure 5.18: Bias Circuitry

In order to establish a means of authenticating the simulation results obtained from Cadence (industry standard CAD design program), we decided to use the signal values generated by MATLAB with the input conditioner already incorporated. By so doing, the simulated detector decisions from Cadence can be compared with the MATLAB prediction, which assumes that all the building blocks functionality are ideal (i.e. no gain error incorporated).

Furthermore, it was decided to use a small sample size of 21 (arbitrarily chosen) input data at a high signal-to-noise ratio. The detector was tested based on a condi-

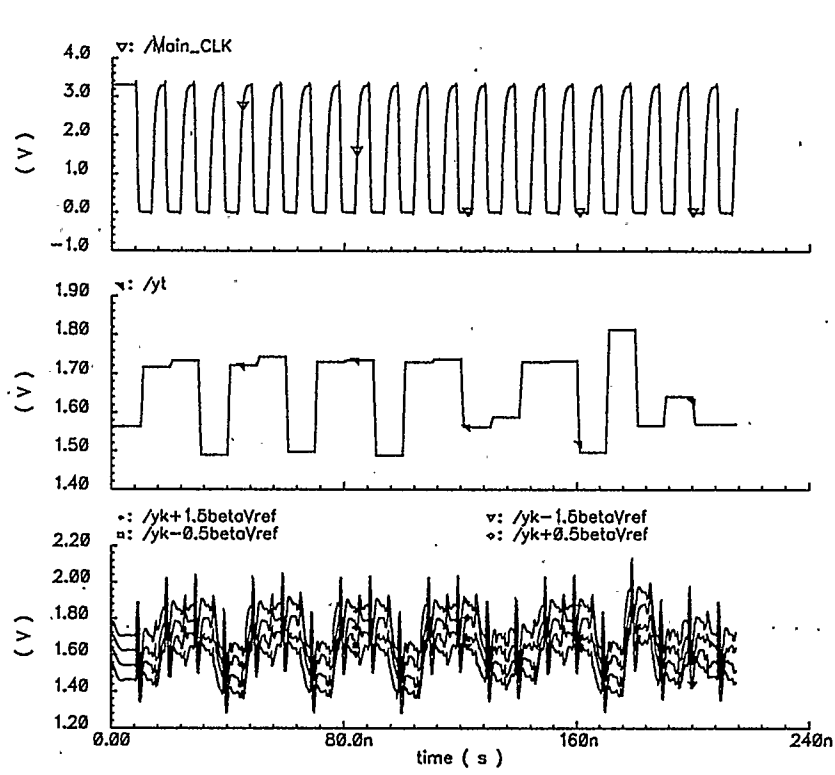


Figure 5.19: Simulated signal shift

tioned input at SNR=16dB, 14dB and 12dB, with the 50 percent duty cycle off-chip clock operating at 100MHz. Figure 5.19 shows the input signal at SNR=16dB, the on-chip main clock, as well as the level-shifted sampled signals.

The two thresholds Δm_c and Δm_d were captured in Cadence and shown in Figure 5.20. It can be observed that this implementation suffers greatly from the effect of clock feed-through and charge injection. However, this is not unexpected since a single-ended design was used for the proof of concept. A fully differential design would ultimately help in reducing these effects. Nonetheless, the detector was still able to uphold the assumption that $\Delta m_c < \Delta m_d$ all through the detection time frame.

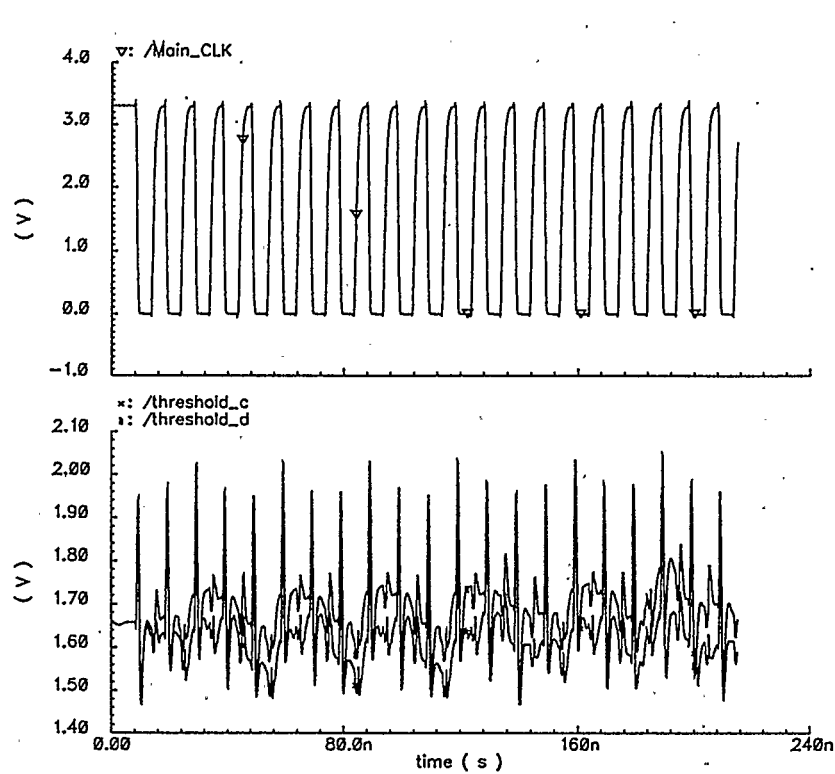


Figure 5.20: Simulated threshold movement

5.3.1 Detector decisions

Since the overall design did not incorporate the survivor memory management, the detector decisions can only be determined to be correct or not by observing the comparators' output. The results (6-bit address in each detection time instance) can then be fed into an algebraic trace-back memory management program written in MATLAB, to obtain the detected ternary sequence. The captured comparator outputs are shown in Figure 5.21.

Table 5.2 summarizes these results along with the MATLAB predictions. The observed variation in the captured 6-bit address from that of MATLAB in some in-

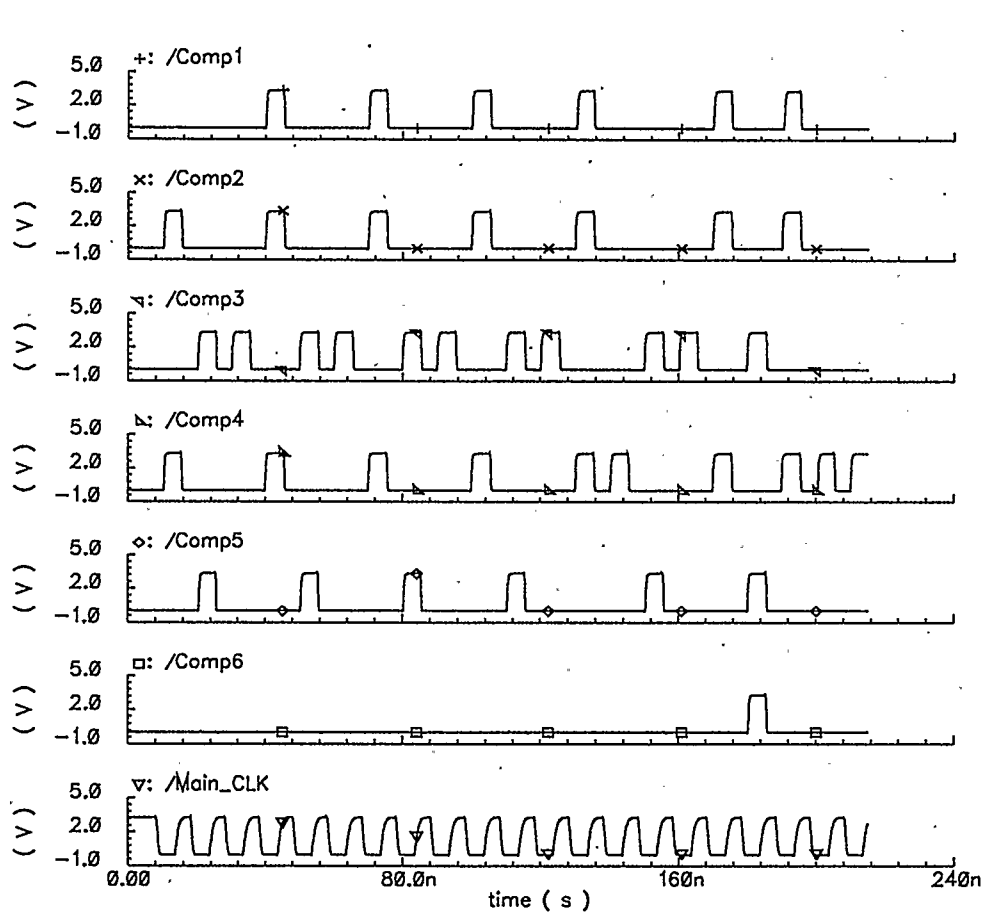


Figure 5.21: Captured comparator outputs

stances could be attributed to the fact that MATLAB predictions did not incorporate any gain error and non-ideal effects, which affects the detector design. Nonetheless, the detected ternary sequence is the same as the uncoded transmitted symbols. This is so because the actual bit responsible for determining the detected region is not affected in these instances.

Furthermore, bit error rate computation was done for the detector within the operational limiting signal range of 1.2V and 2.1V, as was used for the detector design. The obtained result is shown in Figure 5.22 and that for the input limiting effect

Table 5.2: Comparison of MATLAB and Cadence simulation result

MATLAB predictions	Captured results	Detected sequence	Transmitted symbols
010100	010100	-1	-1
001010	001010	0	0
001010	0010 <u>0</u> 0	1	1
110100	110100	-1	-1
001010	001010	0	0
001010	0010 <u>0</u> 0	1	1
110100	110100	-1	-1
001010	001010	0	0
001010	0010 <u>0</u> 0	1	1
110100	110100	-1	-1
001010	001010	0	0
001010	0010 <u>0</u> 0	1	1
110100	110100	0	0
000100	000100	-1	-1
001010	001010	0	0
001010	0010 <u>0</u> 0	1	1
110100	110100	-1	-1
001011	001011	1	1
110100	110100	0	0
000100	000100	0	0
000100	000100	-1	-1

within these range is shown in Figure 5.23. These results validates the correctness of the design.

To obtain these results, simulation was carried out in MATLAB. The unbounded input $y(t)$ was conditioned using the values 81.82×10^{-3} and 1.65 for the normalizing constant (β) and additive constant (α) respectively. These values were calculated from the operational limiting signal range of 1.2V and 2.1V using Equations (4.11)-(4.12).

The obtained conditioned input $y(t)'$ was then fed into the detection algorithm that was modelled to provide the shifting operations such that Equations (4.16)-(4.17) were satisfied. Different lengths of ternary input test vectors were used for simulation at different SNR such that the bit-error-rate (BER) at any SNR was obtained only after at least a hundred errors were counted (e.g. for BER at 10^{-4} , the length of the input test vector is not less than 1×10^6).

It is worth noting that it is also possible to perform the simulation directly in Cadence by making use of the file-based piece-wise linear voltage source (Vpwl) in the "analoglib". The Vpwl voltage source provides the functionality similar to that of an arbitrary waveform generator (AWG), and thus different lengths of test waveforms with varying voltage levels could be generated for testing.

The two column ASCII file describing the waveform in terms of time and voltage level can be easily generated in MATLAB and the file could be placed in a directory such that the path and the filename is fully stated in the Vpwl. Cadence simulation can then be done and the results "marched" so that a program can be written to sort out the obtained simulation results.

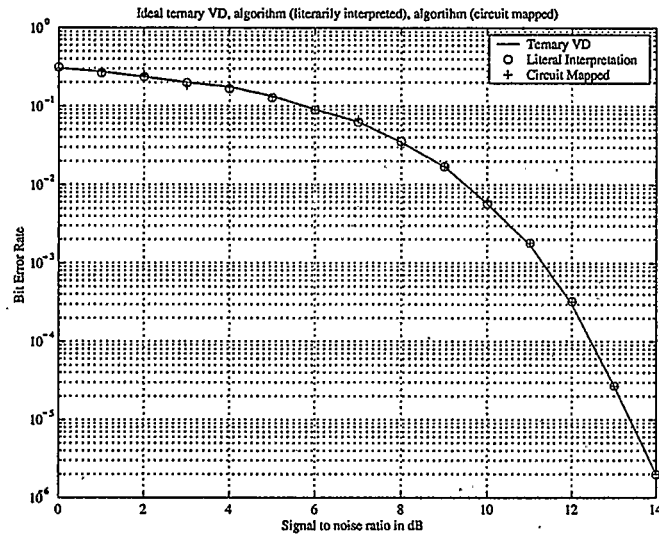


Figure 5.22: Circuit mapped error rate computation

5.4 Survivor Memory Management

The above design did not take into consideration the possible architecture for the implementation of the survivor memory management (SMM) circuitry. It is generally known that there are two primary means of implementing the SMM; namely the register-exchange method and the trace-back method [55, 56, 57, 58, 48].

Considered typically as a naive implementation of the SMM circuitry for Viterbi detectors, the register-exchange method mimics the trellis of the channel for which the detector is designed. Its operation is based on the exchange of information (register content) between the registers involved, in response to the appropriate control signals.

The depth of the memory required is usually carefully chosen to be long enough in order to ensure negligible error detection degradation. At every detection time-step, all registers exchange information thus, data has to be read, and re-written. This method is simple and has a high throughput, but generally requires large area for

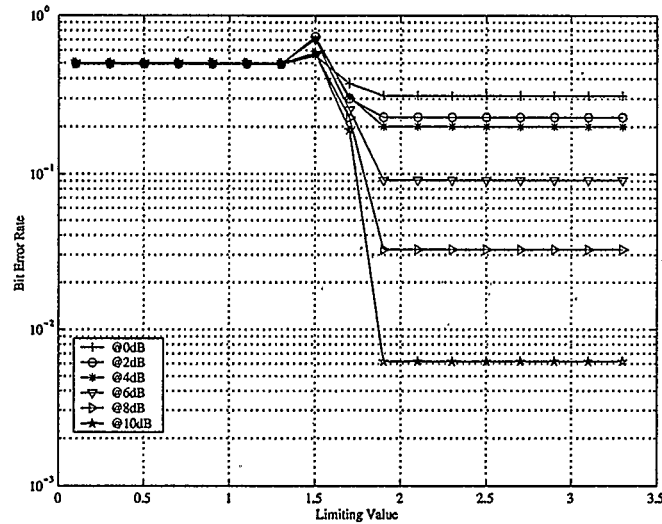


Figure 5.23: Input limiting effect within 1.2V and 2.1V

implementation.

In 1981, Rader published the idea of the trace-back method [55], which avoids the movement of data (usually long bit sequences) at every detection time-step by the use of pointers. The method is based on backward processing of the survivor path update in contrast to the forward processing approach of the register-exchange method. It basically involves three stages:

- traceback read, which when run to a predetermined depth, is used to initiate the decode read operation.
- decode read, the stage where actually decoding is done and bits are sent to a bit-ordering (reversing) circuitry.
- new data write, during which detector decisions are written to locations just made free by the decode read operation.

Although this method is area efficient, it is generally more complex than the

Table 5.3: Survivor memory path control logic

Condition	Comparator Decision	Survivor	Logic Operation
equation (3.16)	$C_3 = 0$	$S_{+1} \rightarrow S_{+1}$	$S_7 = \overline{C_3}$
equation (3.17)	$[C_3 C_6] = [10]$	$S_0 \rightarrow S_{+1}$	$S_8 = \overline{C_3} + C_6$
equation (3.18)	$C_6 = 1$	$S_{-1} \rightarrow S_{+1}$	$S_9 = C_6$
equation (3.19)	$C_2 = 1$	$S_{+1} \rightarrow S_0$	$S_{10} = C_2$
equation (3.20)	$[C_2 C_5] = [00]$	$S_0 \rightarrow S_0$	$S_{11} = \overline{C_2} + \overline{C_5}$
equation (3.21)	$C_5 = 1$	$S_{-1} \rightarrow S_0$	$S_{12} = C_5$
equation (3.22)	$C_1 = 1$	$S_{+1} \rightarrow S_{-1}$	$S_{13} = C_1$
equation (3.23)	$[C_1 C_4] = [01]$	$S_0 \rightarrow S_{-1}$	$S_{14} = \overline{C_1} + \overline{C_4}$
equation (3.24)	$C_4 = 0$	$S_{-1} \rightarrow S_{-1}$	$S_{15} = \overline{C_4}$

register-exchange method.

Therefore for simplicity, the SMM for the ternary dicode detector can be implemented using the register-exchange method. Recalling equations (3.16)-(3.24) and Figure 5.1, the design table for the survivor memory control logic results and is presented in Table 5.3.

Based on this design table, a sample survivor memory path can be developed as shown in Figure 5.24 (*see Appendix C for derivation and validation*). In this case, a tri-state, multi-valued D flip-flop is required, since there exist three stable states in the channel trellis. A possible circuit implementation of the tri-state, multi-valued D flip-flop was reported in [59].

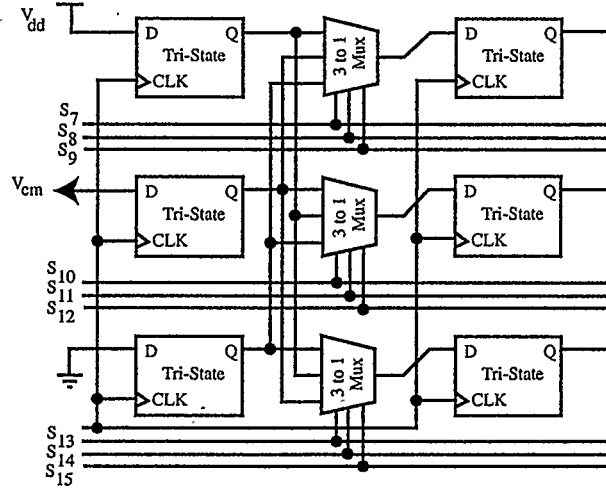


Figure 5.24: Survivor memory path construction

5.5 Experimental Results

The design was fabricated via The Canadian Microelectronics Corporation (CMC) in a 40 pin-DIP package using TSMC CMOS $0.18\mu m$ process. The analog and digital power supply rails (V_{dd} and gnd) were kept separate from each other in the design to minimize switching noise interference. The overall chip size (bond pad inclusive) is $1.44mm \times 1.24mm$ equivalent to an area of $1.78mm^2$; twenty two circuit nodes (I/O and power supplies) were bonded for testing purposes. The chip layout is shown in Figure 5.25.

5.5.1 Test Signal Generation

In order to generate the appropriate test vector (random ternary sequence) for the designed chip, minimum distance error-event occurrence in a typical ternary dicode channel (stated in subsection 4.3.1) was taken into consideration.

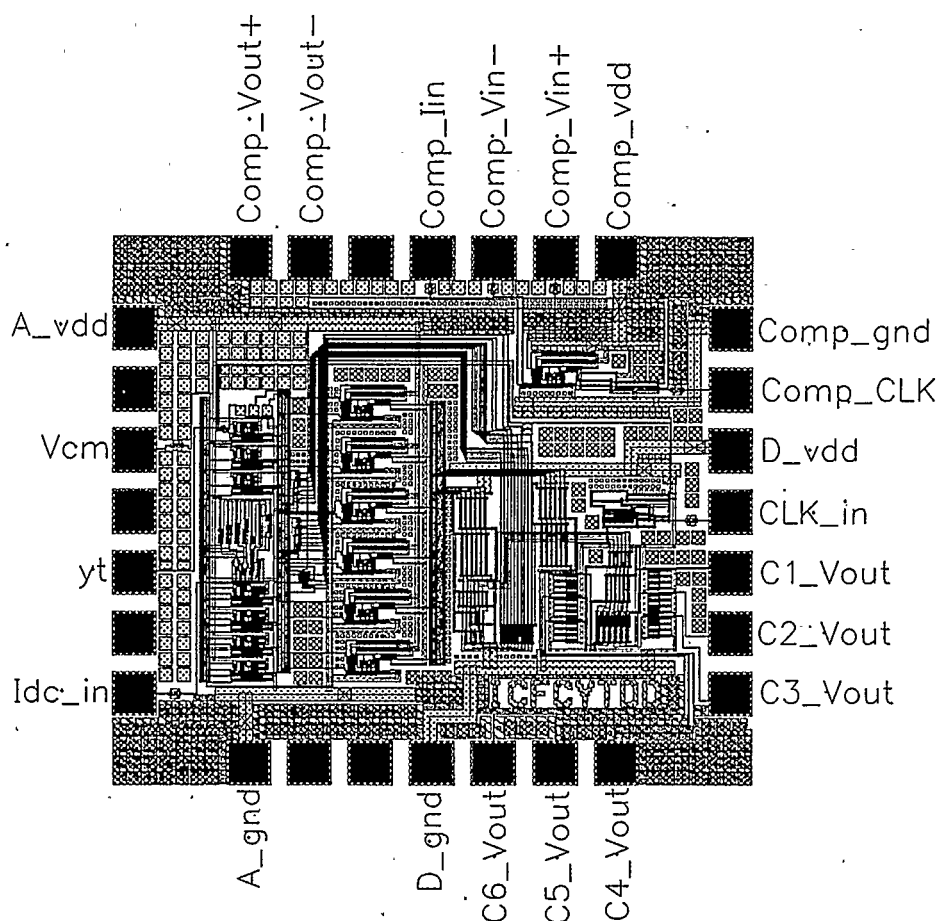


Figure 5.25: Chip layout diagram

As noted in subsection 4.3.1, a typical ternary decode channel has a higher minimum distance error event occurrence probability than a typical binary decode channel. Thus, the input (uncoded) ternary sequence has to be designed to minimize free minimum distance error event propagation by truncating the length of propagation and ensuring frequent alternation of non-zero input symbols [60].

The design of such codes had been studied and the relationship between the capacity and truncation length z is already known [60]. Thus, a simplified version of the code design state diagram found in [60] was used as shown in Figure 5.26.

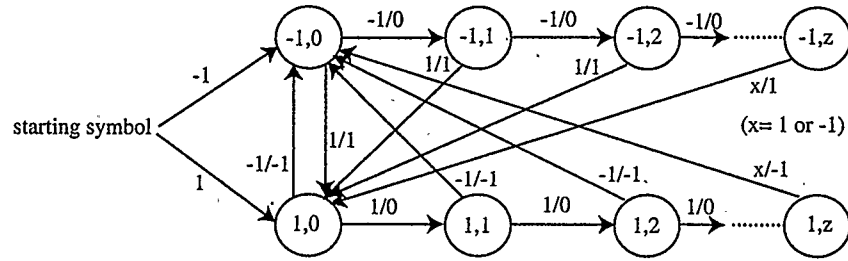


Figure 5.26: State diagram for ternary sequence generator

Basically, this state diagram ensures that zero symbols limited by the specified truncation length are substituted for recurring non-zero symbols, while forcing the next occurring non-zero symbols to be opposite in sign to the last (or previously) occurring non-zero symbol. The implication of this is to frequently force survivor paths from state S_{+1} to S_{-1} and vice-versa preventing minimum distance error events from propagating freely within the channel.

For instance, by using an arbitrary truncation length of 3, the following binary sequence (antipodal) $[1\ 1\ 1\ 1\ -1\ 1\ -1\ -1\ -1\ -1]$ is converted to the ternary sequence $[1\ 0\ 0\ 0\ -1\ 1\ -1\ 0\ 0\ 0]$, and the resulting state diagram will be as shown in Figure 5.27.

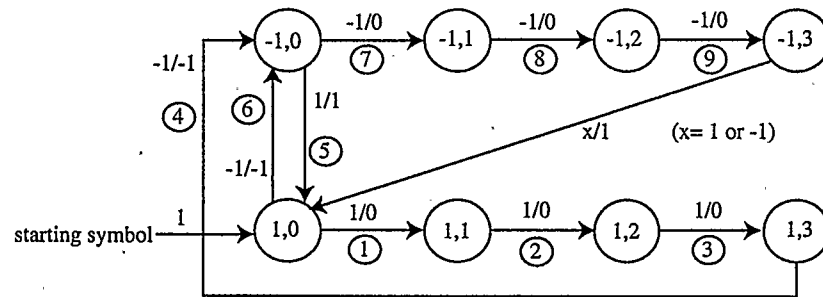


Figure 5.27: A typical example of ternary sequence generation

Thus an arbitrary ternary sequence was generated via the above method from a completely random binary sequence obtained in MATLAB from normally distributed random numbers with zero mean and variance of one. The ternary sequence was then

differentiated and corrupted by additive white gaussian noise (AWGN) of known variance as specified by Equation (4.8). The data was then oversampled by 8 and loaded into an arbitrary waveform generator (Sony/Tektronix AWG520).

Prior to loading the data into the AWG, the symbol length of 512 was generated (since the input data to the AWG must be multiples of 8 in length); representing 4096 sample points in total. Also, the data was constrained by mapping the values within the range 1.2V and 2.1V that the detector was designed for, using Equations (4.15)-(4.17).

Furthermore, the data was then level-shifted down by 1.2V to ensure that the data to be loaded into the AWG satisfies the input condition of ± 1 and an output signal ranging between 0V and 900mV is obtained at the limits. Clock data of equal length as the analog signal was also loaded into the AWG. The clock data was derived using equal length (4 each) of -1 and 1 for the low and high phase of the clock respectively.

5.5.2 Chip Test Bench

The chip test bench is as shown in Figure 5.28 where the chip is depicted by the DUT (Device Under Test) block. Since the output of the AWG has to be shifted up by 1.2V before being fed into the DUT, a level-shifting circuit shown in Figure 5.29 served as the interface between the AWG (channel 1) and the DUT.

A high-speed, wide bandwidth voltage feedback CMOS op-amp (Burr-Brown OPA 355) was used for the level-shifting circuitry and a bypass capacitor of $10\mu F$ was connected between the power rails of the op-amp. Also, the clock data (4096 sample points as well) from the AWG (channel 2) was double inverted using a Fairchild inverter chip 74AC04. This was done to ensure that the off-chip clock going into the

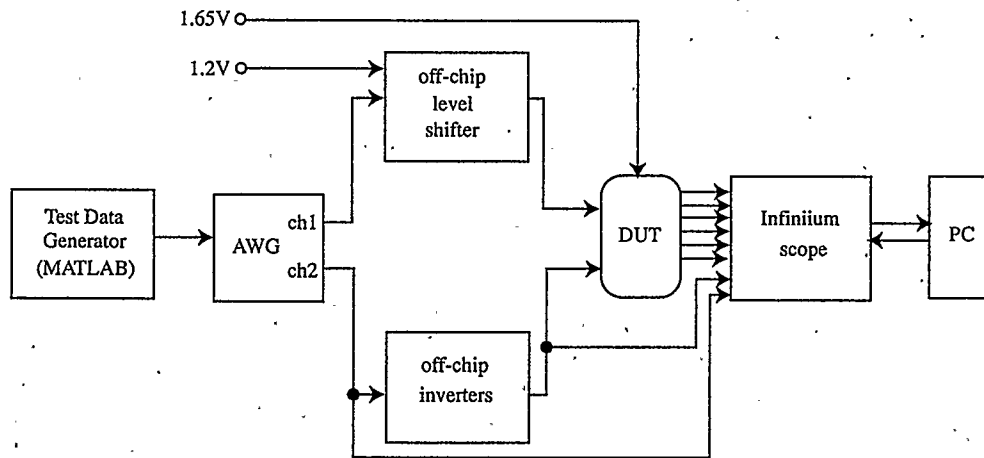


Figure 5.28: The chip test bench

detector will have a logic '1' corresponding to 3.3V since the maximum level of the clock signal from the AWG is 2V.

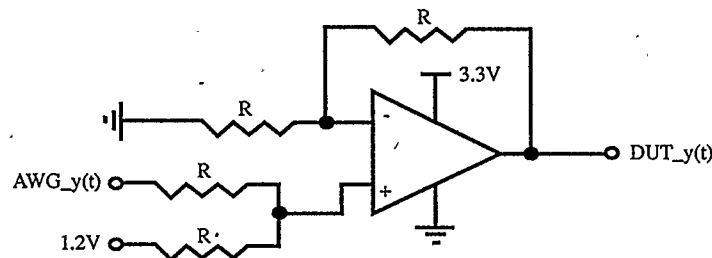


Figure 5.29: The off-chip level shifter

The AWG run mode was set to “Enhanced”, while the trigger was set to external. This was done in order to utilize the functionality of the AWG to generate a specific number of symbols and clock cycles as dictated by the user in its sequence table. By so doing, different lengths of test vectors for the DUT can be generated as appropriate for each desired signal-to-noise ratio, and also the sequence is guaranteed to stop once the specified end of sequence is reached. Also, the clock rate for the AWG was appropriately adjusted bearing in mind that its input data had been oversampled by

8. Thus, for instance, for circuit operation at 1MHz, the AWG clock rate was set to 8MSamples/sec resulting in a bit/clock period of $1\mu s$ for testing at 1Mb/s. The chip test was stopped at the clock rate of 20Mb/s even though simulation predicts operation up to 100Mb/s for the ternary dicode detector (equivalent channel rate of 200Mb/s).

At the time of designing the chip, the survivor memory management part of the detector was not included. Therefore, in order to verify the correct functionality of the fabricated chip, all six digital outputs from the DUT comparators have to be captured in parallel for further analysis. For this purpose, a 16 channel, 4GSamples/sec mixed-signal oscilloscope (Agilent Infiniium MSO 54832D) was employed using its logic analyzer type probe.

The time-base of the MSO was appropriately set to ensure that the entire test sequence is captured in a single-shot, and the trigger was set for 3.3V CMOS operation with a threshold of 1.65V. The MSO was set to trigger off the rising edge of the DUT clock signal, while the digital channels 8 to 15 were turned off since only channels 0 to 7 were used for the test. Since the MSO has deep memory, the acquired six bit address (comparator outputs) at every tested signal-to-noise ratio was stored in the MSO hard disk. After all data acquisition has been completed, the stored data was downloaded to a PC by connecting the MSO to a PC via the available ethernet link. A sample of the acquired signals is shown in Figure 5.30.

5.5.3 Experimental Data Analysis

The acquired data from the chip test was stored in a CSV (Comma Separated Variable) format (a convenient format for the MSO which uses Windows 98 operating system), and had to be converted to a format more compatible for MATLAB anal-

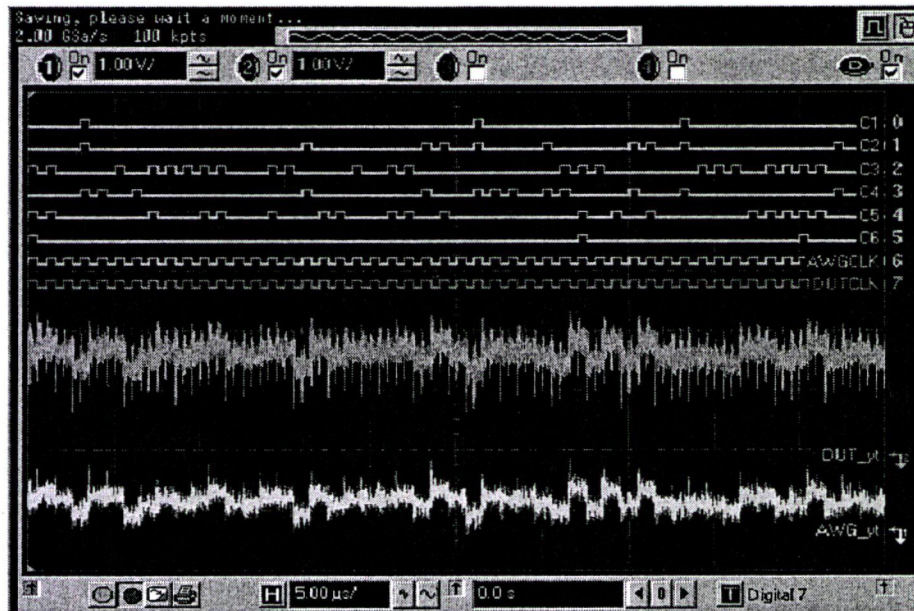


Figure 5.30: A sample of acquired signals @ 0dB SNR

ysis (ASCII space delimited format). After conversion, all pre-trigger information was nullified before loading into MATLAB (easily observable since the MSO trigger position was set at zero second). A MATLAB script was written to extract a single detector decision sample for each clock cycle (out of about 50 samples/bit because of the high sampling rate of the MSO). The extracted decision was then re-arranged in a suitable manner for the remainder of the script to complete the analysis.

During the analysis, it was observed that the acquired chip output was predominantly taken from a set of valid 6-bit combinations (*as shown in Table C.3 in Appendix C*) out of the possible sixty four (2^6) binary combinations. Nonetheless, it was also noticed that within the set of the chip output results, there were instances of invalid 6-bit combinations for the detector. These occurrences were not seen in simulation and a possible cause was investigated.

It was very difficult to ascertain the source of the problem since not all of the

input/output nodes of each block in the design were available for probing. This is a serious oversight during the design of the chip. However, one likely factor thought to be part of the cause of this problem is that of initialization imbalance (unconstrained or un-predetermined threshold(s) starting value(s)) for thresholds Δm_c and Δm_d .

During the design of the detector, the two thresholds were initialized to the same voltage value of $V_{cm} = 1.65V$ (in order to provide equal weight for both thresholds) at the schematic simulation level. However, it was impossible to guarantee equal initialization values for the thresholds during layout since no additional circuitry was introduced in the design to accomplish this. Nonetheless, during the chip layout simulation (schematic versus layout), it was observed that although the two thresholds start at different values, the algorithm settles after the first clock cycle.

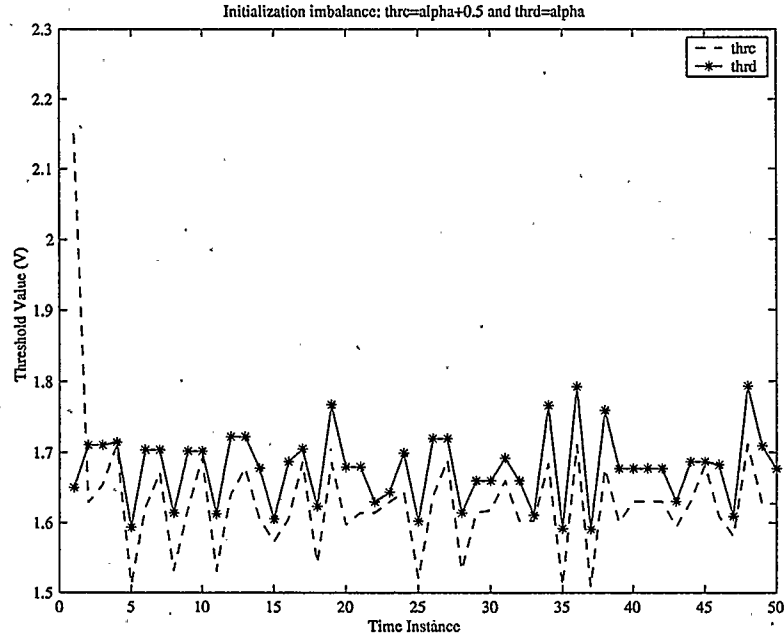


Figure 5.31: Type 1: Initialization imbalance (Threshold Movement)

Due to the invalid combination problem encountered during testing, this observation was further questioned and the effect of initialization imbalance on the algorithm

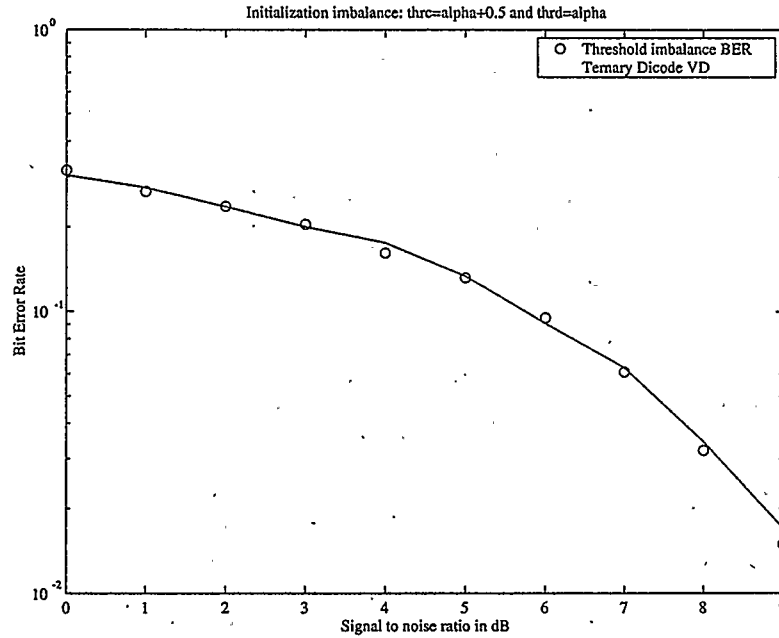


Figure 5.32: Type 1: Initialization imbalance(Error Rate)

under different scenarios was simulated via MATLAB.

Figures 5.31-5.34 shows the obtained results (Type 1 is for the case $\Delta m_c(0) > \Delta m_d(0)$. Type 2 is for the case $\Delta m_c(0) < \Delta m_d(0)$). Based on these results, it was concluded that threshold initialization imbalance might not be such an over-riding factor for the problem at hand.

Another possible cause of the problem was thought to be that of a rather noisier (than expected) input to the DUT as observed in Figure 5.30. The very noisy input going into the DUT was also noticed in the chip test when the signals were checked with another oscilloscope; a 2GSamples/sec Lecroy DSO (Wavepro 960).

The y_t signal input to the DUT was noisier than the output observed from the AWG itself. In order to overcome this problem and its effect on the detector, the test board was examined and in particular the high-speed op-amp used for level-shifting.

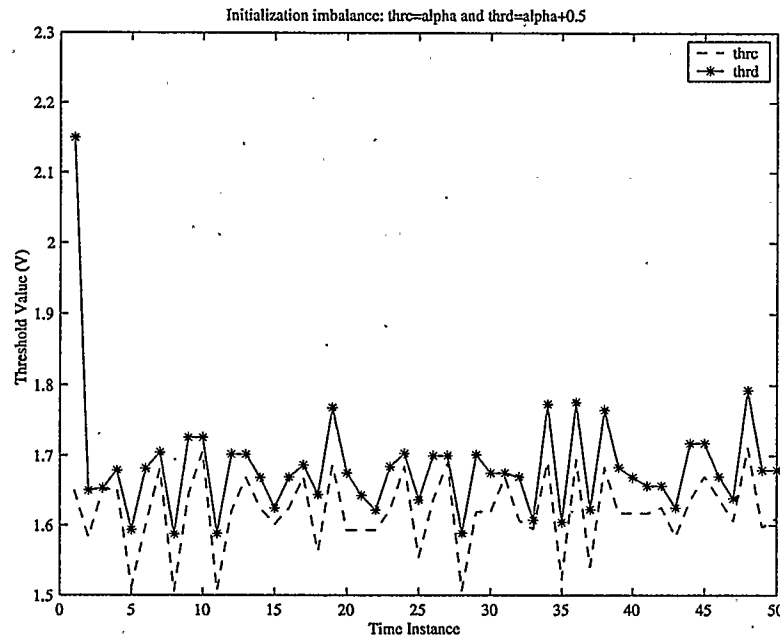


Figure 5.33: Type 2: Initialization imbalance (Threshold Movement)

The op-amp and its passive components were bypassed by shifting down the signal samples (y_t) by 1 in MATLAB and then, using the amplitude and offset parameters of the AWG to re-generate the signal back at the right levels. The signal was then filtered directly from the AWG using built-in filter parameter before being fed into the detector.

With the input y_t “cleaned”, additional chip testing was completed and the output data acquired from four different detector chips and analyzed. Figures 5.35-5.38 show some sample waveforms acquired. During the analysis, it was discovered that there were still some instances of invalid 6-bit combination occurring at the output of the detector.

In finding the source of this problem, a malfunctioning comparator circuitry (assuming every other circuitry in the detector are working appropriately) was investi-

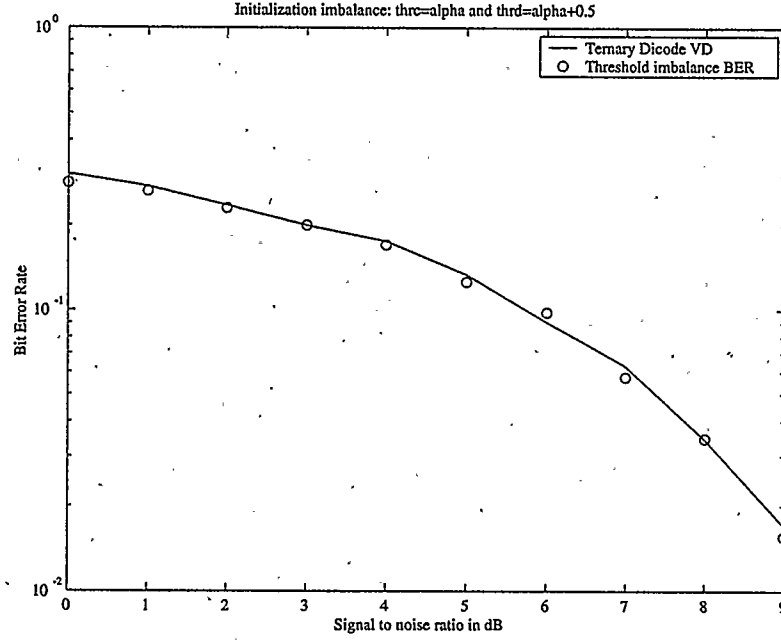


Figure 5.34: Type 2: Initialization imbalance(Error Rate)

gated. Thus the comparator schematic was simulated again to observe its sensitivity to changes in transistor model. Several simulations was carried out on the comparator changing the transistors' model specification from "Typical" to "Fast-Fast", "Slow-Fast", "Fast-Slow" and "Slow-Slow". Despite these changes, the comparator simulation results remain virtually the same.

Two flaws were eventually discovered. In the update of the two differential thresholds, there exists only one instance for any of the thresholds to attain the previous value of the other threshold (Equation (3.41) and (3.52)). Under this circumstance, select signal S_2 or S_5 (in Figure 5.1) would go high. However, based on Figure 5.15, both the feed-forward and cross-over relevant transmission gates will in such instances force $\Delta m_d(k)$ equal to $\Delta m_c(k)$ or vice-versa.

For rectification, select signals S_{2a} , S_{2b} , S_{5a} and S_{5b} are needed such that when

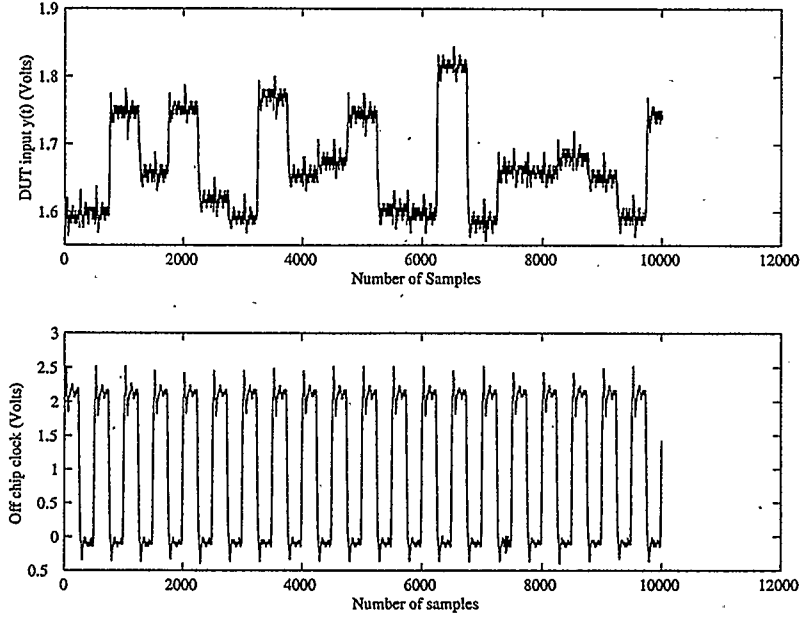


Figure 5.35: Acquired signal sample 1: y_t and off-chip clock

either S_2 or S_5 is high, the pertinent cross-over transmission gate would turn on first, followed sequentially by the feed-forward transmission gate, in a non-overlapping manner within the same clock cycle (see *Figure D6-D8 in Appendix D*).

It was also discovered in conjunction with table C.1 and C.2 (*Appendix C*) that there was an oversight of a rare but possible 6-bit combination “001100” that could occur if the following is satisfied:

$$\text{If } y_k + 0.5 < \Delta m_d(k) \quad \text{and} \quad y_k - 0.5 > \Delta m_c(k). \quad (5.17)$$

Based on this condition, the two differential thresholds must be appropriately updated according to:

$$\begin{aligned} \Delta m_c(k+1) &= m_0(k) + y_k - 0.5 - m_0(k) \\ &= y_k - 0.5 \end{aligned} \quad (5.18)$$

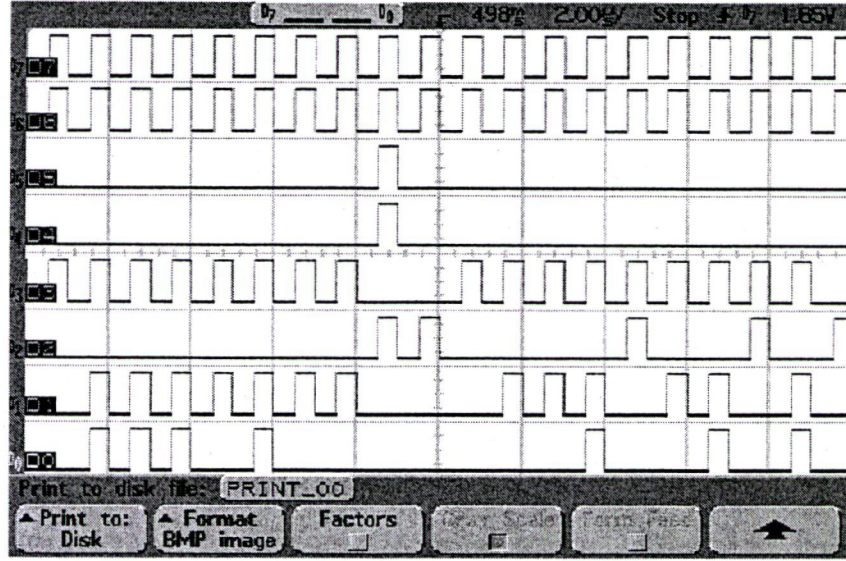


Figure 5.36: Acquired signal sample 1: corresponding DUT output (D0-D5), off-chip (AWG) clock (D6) and DUT clock (D7)

and

$$\begin{aligned}\Delta m_d(k+1) &= m_0(k) - [m_0(k) - y_k - 0.5] \\ &= y_k + 0.5.\end{aligned}\tag{5.19}$$

This was not caught earlier in the derivation and simulation due to its rarity of occurrence. A threshold separation of at least β given in Equation (4.12) is required for this condition to occur while Δm_c is below $y_k - 0.5$ and Δm_d is above $y_k + 0.5$. However, the designed hardware points to the possibility of occurrence of this condition.

These observations could, therefore, explain the difference between the hardware and simulation results since, based on the present design, the two thresholds would be wrongly updated under the two aforementioned circumstances and this would lead the detector to making the wrong decisions once and after the conditions occur.

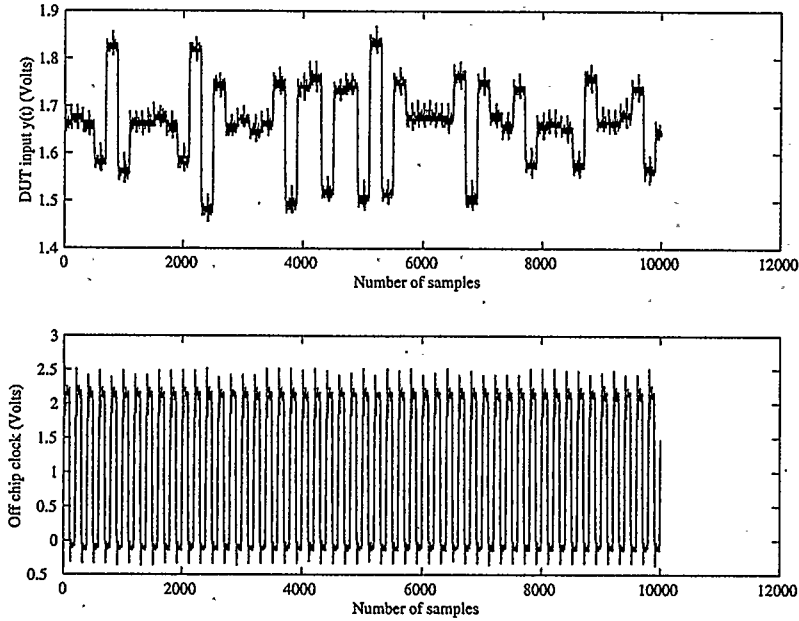


Figure 5.37: Acquired signal sample 2: y_t and off-chip clock

However, only a very minor upgrade would be needed on the current design architecture (for the rectification of the second observation) since only a NOR logic gate is required to process the inverse of comparator outputs C3 and C4. This is in addition to the logic gates shown in Figure 5.16. This gate would generate a select signal to control two additional transmission gates (one for each threshold) to update the thresholds correctly according to Equations (5.18)-(5.19) when the condition of Equation (5.17) occurs. The modified control signal generator logic diagram is shown in Appendix D.

It must be noted that there was an oversight of region $B_3(iv)$ for the ternary duobinary detector as well. This region is governed by the condition:

$$\text{If } y_k + 0.5 < \Delta m_a(k) \text{ and } y_k - 0.5 > \Delta m_b(k). \quad (5.20)$$

Based on this condition, the two differential thresholds must be appropriately updated (*see Appendix A for update*).

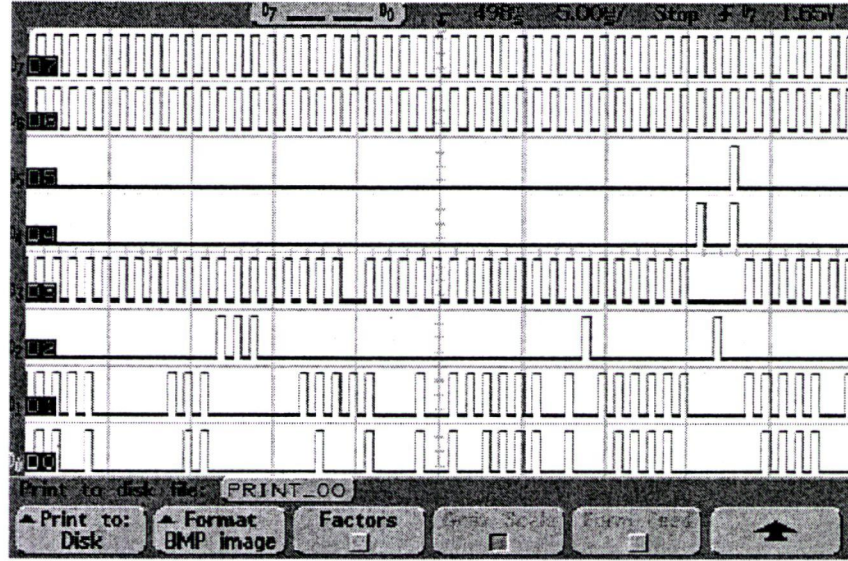


Figure 5.38: Acquired signal sample 2: corresponding DUT output (D0-D5), off-chip (AWG) clock (D6) and DUT clock (D7)

Nonetheless, all the presented simulation results were again verified and are still valid because MATLAB simulations with the newly discovered region ($A_3(iv)$ and $B_3(iv)$) included verified the bit error rate curve to be virtually the same. This could be attributed to rarity of occurrence of the condition in simulation (see *Appendix D for a sample of region occurrence probability*). Modified region dependency diagrams of the thresholds for both ternary dicode and duobinary channels are shown in Appendix D.

However, it should be pointed out that the algorithm originally with the flaw had been published [61] before these discoveries were made.

Chapter 6

Related Research Work

6.1 Loser-Take-All Circuits

During the course of the research work presented in the previous chapters of this dissertation, some useful analog as well as digital building blocks were designed.

One of these circuits is the loser-take-all circuit which is presented in this section. The other circuits are a family of differential logic gates, to be presented in the next section. It is worth noting that the loser-take-all circuit and the differential logic gates could both find application in Viterbi detectors.

The computation of the maximum or minimum of several variables incessantly arises in a wide variety of applications such as Hamming networks, classifiers, vector quantization circuits, and the add-compare-select (ACS) units in Viterbi decoders. Different approaches have been taken and presented in hardware for the realization of the mathematical functions required in computing the maximum or minimum solution [62, 63, 64]. These have led to several circuits of varying sophistication with different

performance limits in terms of speed, resolution, power dissipation, and compactness [65]. Recently, a high resolution, high-speed current-mode maximum seeking circuit was reported [66]. Whenever the maximum is desired, the circuit is traditionally referred to as the winner-take-all (WTA) but when the minimum is desired, a loser-take-all (LTA) is usually adopted.

There have been different techniques proposed for the implementation of a loser-take-all in the literature. In [67], a minimum seeking circuit was obtained by interpreting the minimum function computation as performing the inverse function of the winner-take-all. However, this implementation requires the use of two different power supplies to function properly. Also in [68], a minimum solution was attained by the use of De Morgan's rule on the inputs to a winner-take-all.

All the above stated WTA and LTA circuits were implemented in either the voltage-mode or current-mode. However, current-mode implementations of WTA's or LTA's could be less complex as made evident in [66]. Moreover, current-mode circuits could exhibit better bandwidth as well as dynamic range in comparison to voltage-mode circuits [69].

6.1.1 A New Proposition

A new technique that fully utilizes the inherent ease of adding and/or subtracting current rather than voltages to realize a current-mode LTA will be described. In this technique, the minimum of multiple variables will be obtained using a maximum circuit from a completely different perspective. However, for simplicity, it will be assumed that the number of inputs to the LTA circuit is two.

Conceptually, consider having two time-dependent, signed, unknown variables

$A(t)$ and $B(t)$ of varying magnitude whose minimum is desired. Also, let the minimum of the two time varying variables be represented by $R(t)$. Mathematically, $R(t)$ can be computed as:

$$R(t) = (A(t) + B(t)) - \max(A(t), B(t)) \quad (6.1)$$

Thus, if the sum of the two variables can be found and if the variable with the maximum magnitude can be correctly detected, then the process of computing the minimum is a simple process of subtraction as revealed by Equation(6.1).

Therefore, the overall computation of the minimum involves two basic processes, summation and competitive selection. These two processes could be achieved in either the voltage-mode or current-mode for the hardware implementation of the concept. However, since the overall process involves summation, the ease of adding/subtracting currents could be taken advantage of easily. Thus, it will be assumed that both input variables $A(t)$ and $B(t)$ are currents I_{in1} and I_{in2} respectively. It is worthy of note that a variation of equation (6.1) could also be presented for the computation of the maximum between two variables. Moreover, it should be noted that a similar expression to Equation (6.1) was independently proposed in [43].

6.1.2 First Circuit Implementation

Figure 6.1 shows a block diagram representation of the mathematical model provided in Equation (6.1) [70]. A CMOS implementation is shown in Figure 6.2. The basic processing units are current mirrors, and bulk effect is accounted for in the implementation.

Assuming transistors M1-M4 and M7-M10 are all in saturation, the two current inputs I_{in1} and I_{in2} will be conveyed to the WTA circuit, which comprises transistors M13-M20. Duplication of the inputs to the output stage is achieved via M5, M6, M11

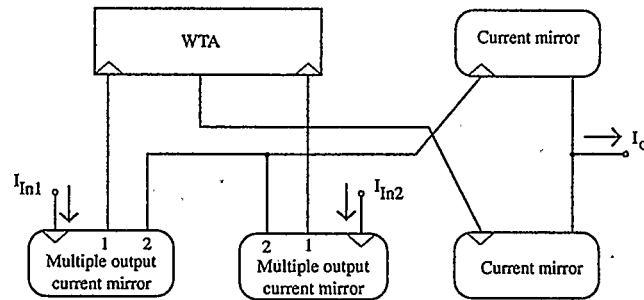


Figure 6.1: Block diagram representation of the LTA

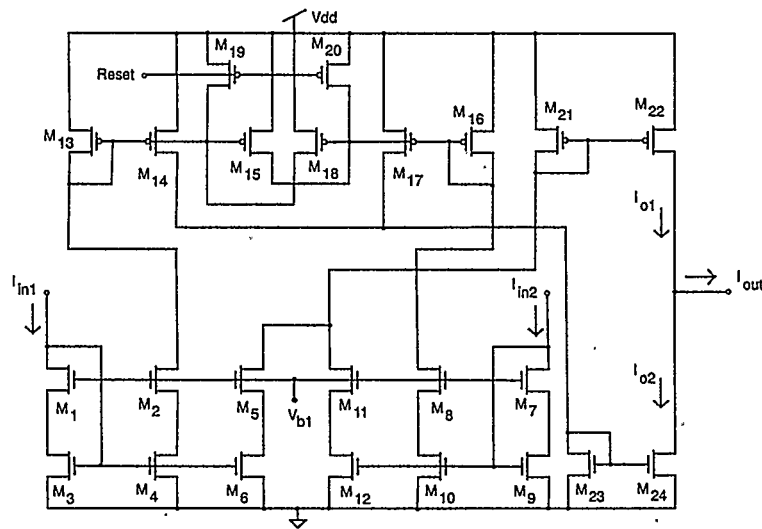


Figure 6.2: First type of CMOS implementation of the LTA

and M_{12} . By tying the drains of M_5 and M_{11} together, the current mirror formed by M_{21} - M_{22} will effectively source the sum of the two input currents towards the output node.

The WTA utilizes two simple, double output current mirrors to competitively determine the winning current between I_{in1} and I_{in2} . By tying the drains of M_{14} and M_{17} (first output transistors of the WTA current mirrors) together and cross-coupling M_{15} and M_{18} (second output transistors of the WTA current mirrors), the winner is determined.

The cross-coupled transistors provide positive feedback within the WTA circuit to ensure that the winning current between I_{in1} and I_{in2} is sunk from the output node through the current mirror formed by M23 and M24. Because positive feedback is used to attain the winner, it would seem appropriate to provide a means of resetting the circuits. This is achieved via M19 and M20. Therefore, the minimum between I_{in1} and I_{in2} will be seen at the output during one of the two phases of the reset clock. For instance, let I_{in1} be assumed to be greater than I_{in2} and let ϕ_1 and ϕ_2 represent the high and low phase of the reset clock, respectively.

During ϕ_1 :

The gates of both M19 and M20 are high and this will result in the two transistors being turned off. Since I_{in1} is assumed to be greater than I_{in2} , then the gate voltage of M14 and M15 should be less than the gate voltage of M17 and M18. This will result in M15 being more likely to be turned on quicker than M18.

With M15 on, it will source more current and tend to raise the gate voltage of M16-M18 towards the positive supply rail. Gradually, M18 will be driven towards cut-off. Eventually, M16-M18 will be turned off leaving M13-M14 to source current I_{in1} towards the output node via M23-M23; thus the maximum of the two currents is correctly determined. Therefore, the drain current of M24 is approximately I_{in1} .

At the same time, the drain current of M22 is approximately the sum of the currents I_{in1} and I_{in2} . Thus the output current I_{out} according to Kirchoff's current law will be given by:

$$I_{out} \cong (I_{in1} + I_{in2}) - \max(I_{in1}, I_{in2}) \quad (6.2)$$

Equation (6.2) is very similar to equation(6.1); therefore I_{out} should be the minimum between I_{in1} and I_{in2} .

During ϕ_2 :

Table 6.1: Transistors' aspect ratio

Transistor(s)	Width(μm)	Length(μm)
M1-M11	20	1
M13-M18	35	1
M19-M20	5	1
M21-M22	70	2
M23-M24	100	2

The gates of M19 and M20 become low, thus turning the two transistors on and the gates of both M15 and M18 will be pulled towards V_{dd} . This effectively results in M13-M18 being turned off and the drain current of M24 becomes virtually zero. However, at this time, the drain current of M22 is still the sum of the two input currents. Therefore, according to KCL, I_{out} will be given by:

$$I_{out} \cong I_{in1} + I_{in2} \quad (6.3)$$

Therefore, this implementation produces the minimum between I_{in1} and I_{in2} when the reset is high but provides the sum when the reset is low. The aspect ratios of the transistors are provided in Table 6.1.

6.1.3 Second Circuit Implementation

The second implementation of Figure 6.1 is presented in Figure 6.3. It is also a CMOS implementation with similar architecture to that of the implementation presented in the preceding subsection. However, the difference between the two implementations is in the implementation of the reset for the WTA circuitry.

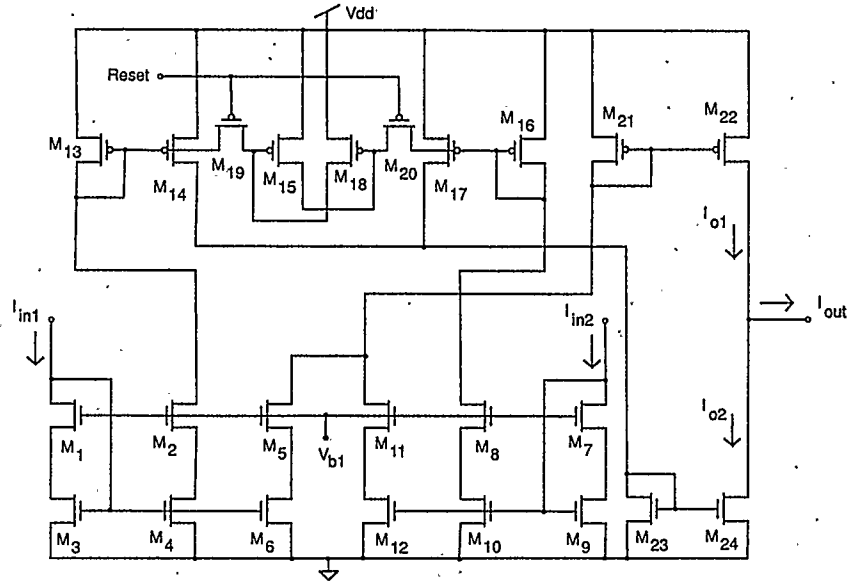


Figure 6.3: Second type of CMOS implementation of the LTA

If I_{in1} is still assumed to be greater than I_{in2} and the phases of the reset clock are still as defined previously, the operation of this circuit implementation will follow.

During ϕ_1 :

The gate voltages of M19 and M20 are both high. Since the two transistors are pMOS and act as switches, they are not turned on due to inadequate gate-to-source voltage. Therefore, the cross-coupled transistors M15 and M18 are prevented from affecting the dynamics of the circuit and M23 will effectively sink the sum of the drain current of M14 and M17. This results in the drain current of M24 being approximately the sum of the two input currents.

Also, the drain current of M21 will be the sum of the drain currents of M5 and M11 and therefore, M22 will source the sum of I_{in1} and I_{in2} towards the output node. Thus, the output current I_{out} will be:

$$I_{out} \cong (I_{in1} + I_{in2}) - (I_{in1} + I_{in2})$$

$$= 0 \quad (6.4)$$

During ϕ_2 :

The gates of M19 and M20 become low and this will result in the gate-to-source voltage of the two transistors being adequate for them to be turned on. With the switches both on, the cross-coupled transistors M15 and M18 now allow competition to begin between the two input currents. Since I_{in1} is still assumed to be greater than I_{in2} , then the gate voltage of M14 and M15 would be less than the gate voltage of M17 and M18. This will result in M15 being ore likely to be turned on quicker than M18. Thus the gate of M18 will be pulled towards V_{dd} and will eventually result in M16-M18 being turned off.

Consequently, the drain current of M23 becomes equal to that of M14, which is approximately equal to I_{in1} . Thus the maximum of the two input currents will be sunk from the output node by M24. Meanwhile, M22 will source the sum of the input currents to the output node, such that the output current I_{out} becomes:

$$I_{out} \cong (I_{in1} + I_{in2}) - \max(I_{in1}, I_{in2}) \quad (6.5)$$

Equation (6.5) is still identical to Equation (6.1) therefore, the output current should be the minimum between the two input currents. Ideally, this implementation should produce zero output current and give the minimum between I_{in1} and I_{in2} whenever the rest is high and low, respectively.

The aspect ratios of all the transistors are the same as provided in Table 6.1.

6.1.4 Simulation Results

In order to verify the authenticity of the new proposition, the two types of circuit implementations were designed and simulated with Cadence using HSPICE. A supply

voltage of 3.3V was used for the $0.35\mu\text{m}$ CMOS process.

The two circuits were tested by having one of the input currents at a constant value while the second input was swept over a range of about $200\mu\text{A}$. During this test, the output of the circuits was observed and the simulation result obtained for the first circuit, shown in Figure 6.4 indicates the ability of the circuit to follow the minimum of the two input currents closely. An identical result was also obtained for the second circuit.

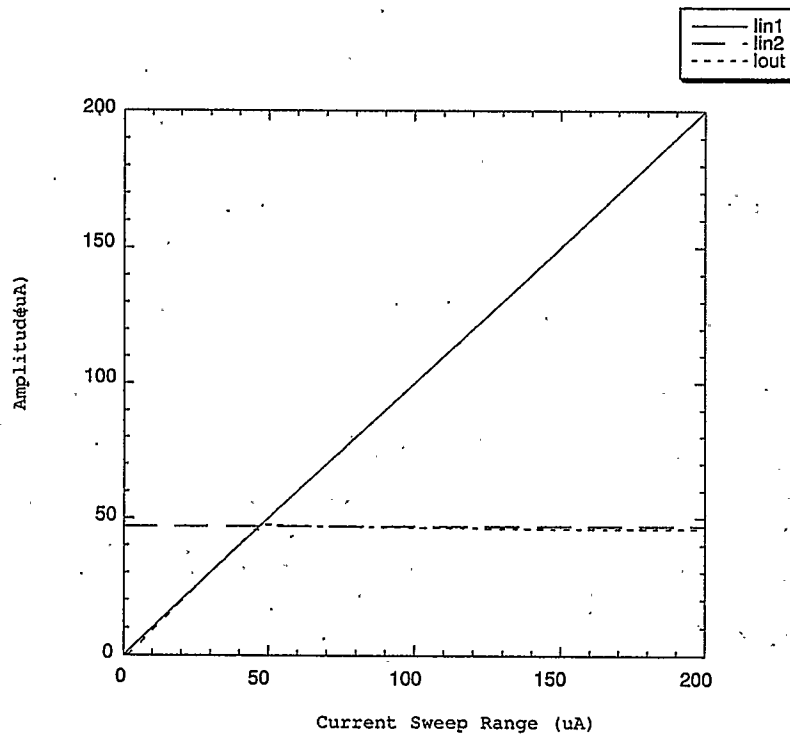


Figure 6.4: Simulated current sweep result of the LTA

Furthermore, the two circuits were tested with two input currents being separated by as little as $1\mu\text{A}$. Over four different time frames, the two input currents are separated by $2\mu\text{A}$, $1\mu\text{A}$, $3\mu\text{A}$ and $1\mu\text{A}$ respectively with either current input alternately becoming the minimum.

Figures 6.5 and 6.6 shows the simulation result for the first circuit at a clock speed of 10MHz and 15MHz respectively; while Figure 6.7 shows the result for the second circuit at a clock speed of 35MHz.

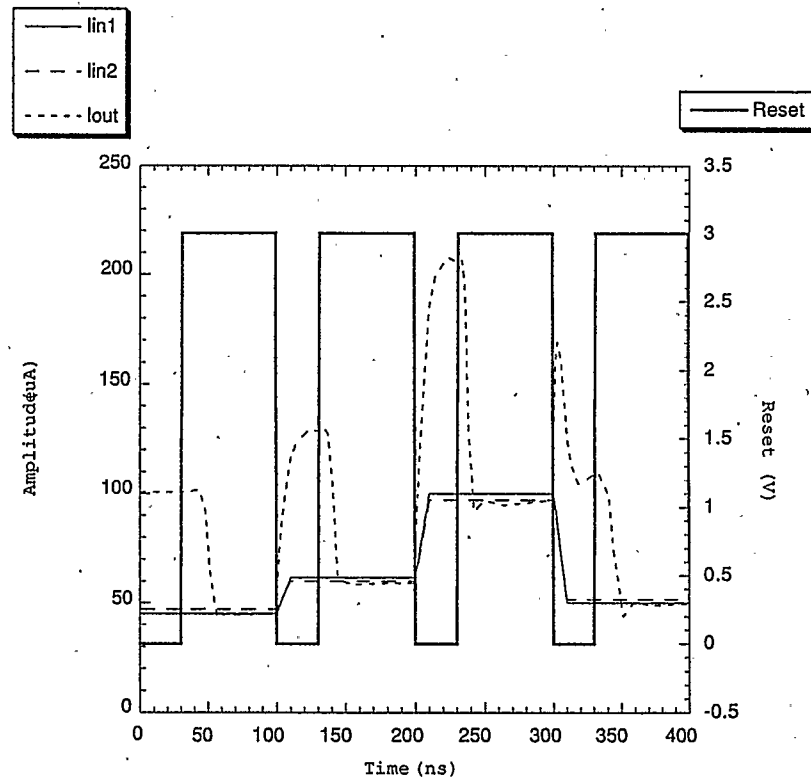


Figure 6.5: Simulation result: first circuit @ 10MHz

6.2 Novel Differential Logic

Incessantly, there arise situations whereby simultaneous use of a logic operator (gate), as well as its complement are required to complete a digital binary operation. A typical example is the computation required in an adder. More often than not, conventional single-ended output logic gates are used in conjunction with the digital inverter

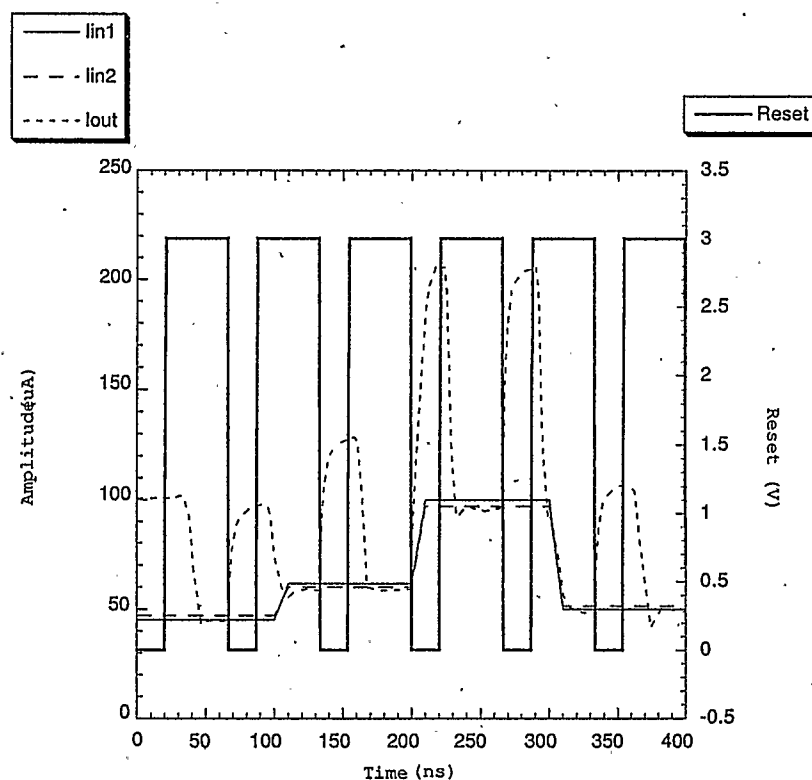


Figure 6.6: Simulation result: first circuit @ 15MHz

or two logic gates are used to obtain a logic value and its complement based on the input to the logic gate(s).

However, there exists a type of digital logic, which can inherently perform differential logic operations for example, OR/NOR and AND/NAND simultaneously using the same number of transistor switches without any further need for an inverter or extra logic gate. Such a logic family is referred to as *Differential Logic* [71].

Differential logic usually minimizes the overall transistor count in a digital design when used as opposed to using a combination of conventional logic gates for the same purpose. A variety of differential logic gates do exist in the literature [72, 73]. However, some of these differential logic gates still use the same number of

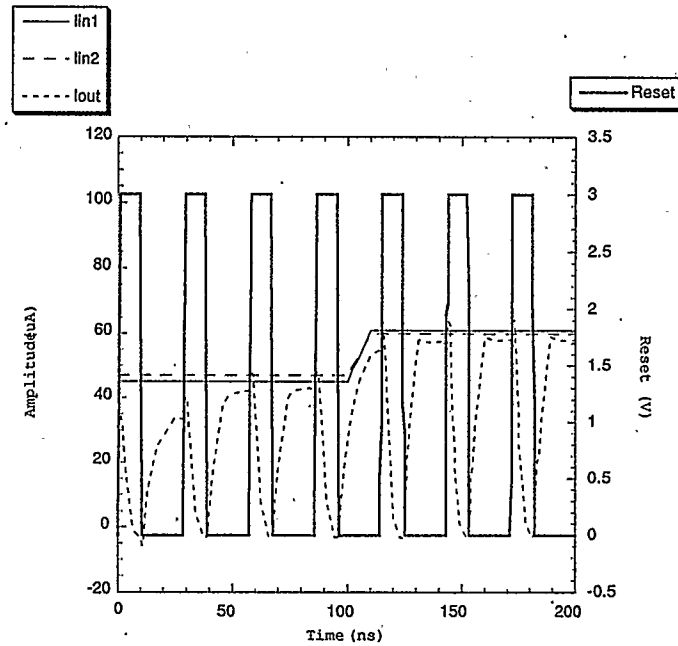


Figure 6.7: Simulation result: second circuit @ 35MHz

transistors between the power rails as can be found in a typical conventional logic gate. Conventional NAND and NOR gates use three transistors between the power rails.

It is therefore the intent and purpose of this research work to propose a family of differential logic gates with a reduced number of transistors stacked between the power rails while still keeping the overall transistor count to a reasonable number, so as not to defeat the set goal.

6.2.1 Differential AND/NAND Logic

The proposed differential AND/NAND logic is shown in Figure 6.8 and the corresponding truth table is provided in Table 6.2.

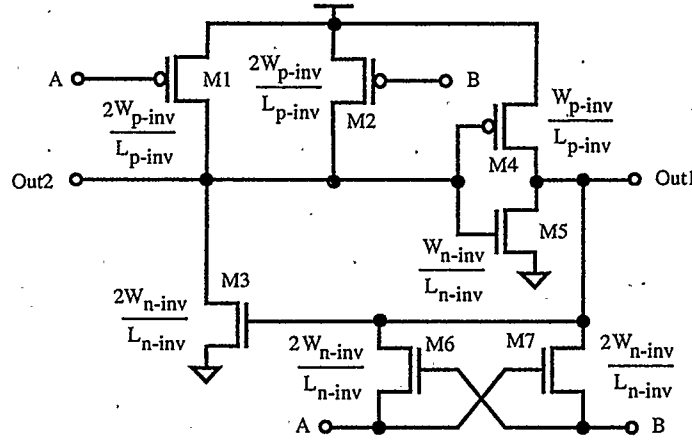


Figure 6.8: Novel AND/NAND differential logic

Table 6.2: Differential AND/NAND logic truth table

A	B	Out1	Out2
0	0	g0	g1
0	1	g0	g1
1	0	g0	g1
1	1	$p1 \rightarrow g1$	g0

The logic circuitry can be thought of as been made up of cross-coupled transistors M6 and M7 (which is very common in reported differential logic gates), two switches M1 and M2 in an OR configuration, a mono-stable circuit comprising nMOS switch M3 and pMOS switch M4, and a level restoration switch M5 which also forms an inverter with M4.

The circuit operation can be explained as follows.

For A='0' and B='0':

Both M1 and M2 are ON and Out2 is pulled up to a good logic '1', however,

M6 and M7 are both OFF. Without the inclusion of M5, the digital level at Out1 is incorrectly determined since M3 and M4 are OFF. Now with the addition of M5, the logic '1' level at Out2 will turn on M5 thereby pulling Out1 low to good logic '0'.

For A='0' and B='1':

M2 is OFF but M1 is ON pulling Out2 up to a good logic '1'. Also, M7 is OFF but M6 is ON and this will pull the gate of M3 low, turning it OFF. However, Out1 is at logic '0' and this logic level is sustained by M5.

For A='1' and B='0':

As long as the input data is readily at these logic levels, the output of the circuit will remain the same as the case for A='0' and B='1', except that it is M2 and M7 that are ON and M1 and M6 are OFF.

For A='1' and B='1':

Both M1 and M2 are OFF while M6 and M7 are ON. However, M6 and M7 will pass a poor logic '1' to the gate of M3 and Out1. Nonetheless, M3 can still be ON thereby pulling down Out2 to a good logic '0'. The logic '0' level at Out2 in turns switches M4 ON to re-instate a good logic '1' at Out1 and thus keep the loop within the mono-stable circuit at a stable state. In this mode, the operation of the differential logic completely depends on how the embedded mono-stable circuit can react to a poor logic '1' level.

Based on the above operation, it is expected that Out1 and Out2 give a AND and NAND output respectively. Simulation results shown in Figures 6.9-6.11 for a load capacitor of 0.02pF, 0.3pF and 0.5pF, respectively authenticate the explained operation of the differential logic.

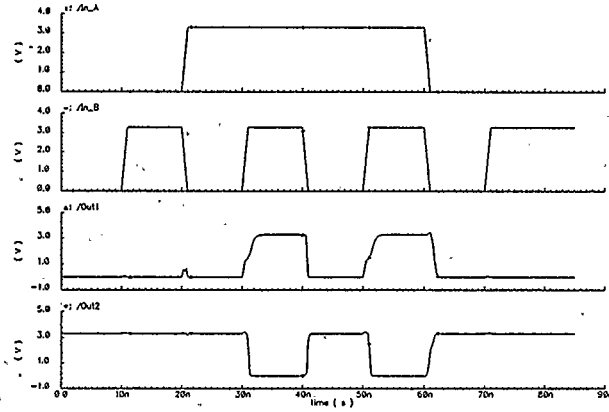


Figure 6.9: Simulation result of AND/NAND logic with 0.02pF load

Table 6.3: Differential OR/NOR logic truth table

A	B	Out1	Out2
0	0	$p0 \rightarrow g0$	g1
0	1	g1	g0
1	0	g1	g0
1	1	g1	g0

6.2.2 Differential OR/NOR Logic

The beauty of the proposed differential logic circuit is revealed in the fact that another type of differential logic (OR/NOR) can be attained by the transposition of the transistors and the power rails in the differential AND/NAND logic circuit just as is done for the conventional logic gates. The resulting circuit is shown in Figure 6.12 and its corresponding truth table is given by Table 6.3.

The circuit operation can be explained as follows.

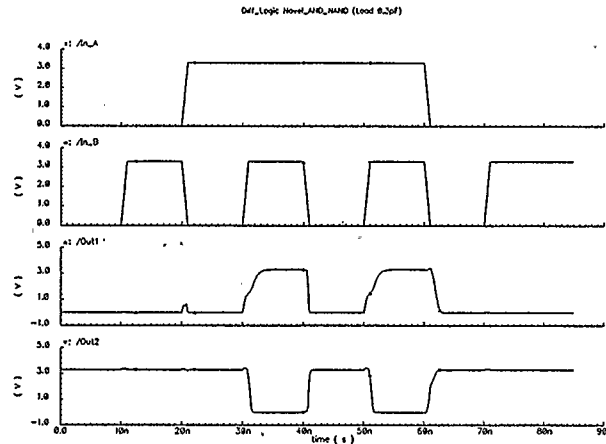


Figure 6.10: Simulation result of AND/NAND logic with 0.3pF load

For A='0' and B='0':

Both M6 and M7 are ON and both switches pass a poor logic '0' to Out1. The gate of M3 senses Out1 to pull Out2 up to a good logic '1'; however, both M1 and M2 are OFF. Therefore because of the good logic '1' at Out2, Out 1 is fully restored to a good logic '0', with this logic level been sustained by M4.

For A='0' and B='1':

M2 is OFF but M1 is ON pulling Out2 up to a good logic '0'. Also, M6 is OFF but M7 is ON and this will pull Out1 up to a good logic '1'.

For A='1' and B='0':

The output of the circuit will remain the same as the case for A='0' and B='1', except that it is M2 and M6 that are ON while M1 and M7 are OFF.

For A='1' and B='1':

Both M6 and M7 are OFF while M1 and M2 are ON, pulling Out2 down to a good logic '0'. Without the restoration switch M5, the logic state of Out1 will not be correctly determined. However, since Out2 is already at logic '0', M5 ensures that

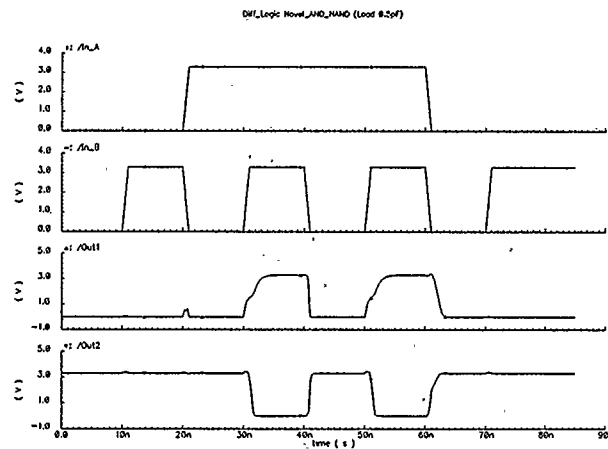


Figure 6.11: Simulation result of AND/NAND logic with 0.5pF load

Out1 stays at a good logic '1'.

Based on the above operation, it is expected that Out1 and Out2 give an OR and NOR output respectively. Figure 6.13-6.15 shows the simulation results obtained for a load capacitances of 0.02pF, 0.3pF and 0.5pF, respectively.

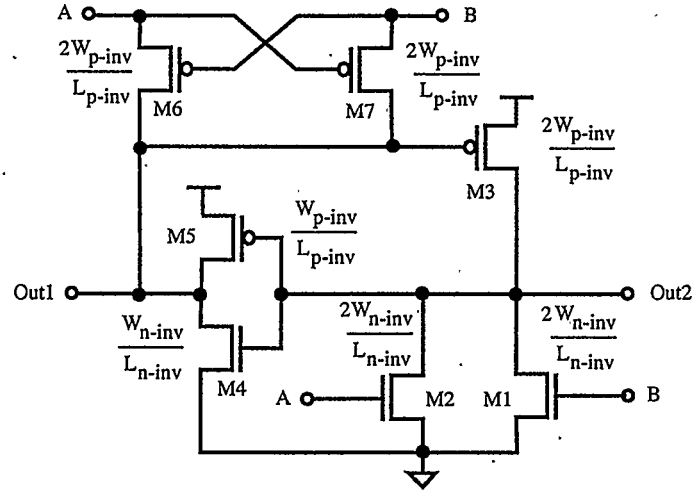


Figure 6.12: Novel OR/NOR differential logic

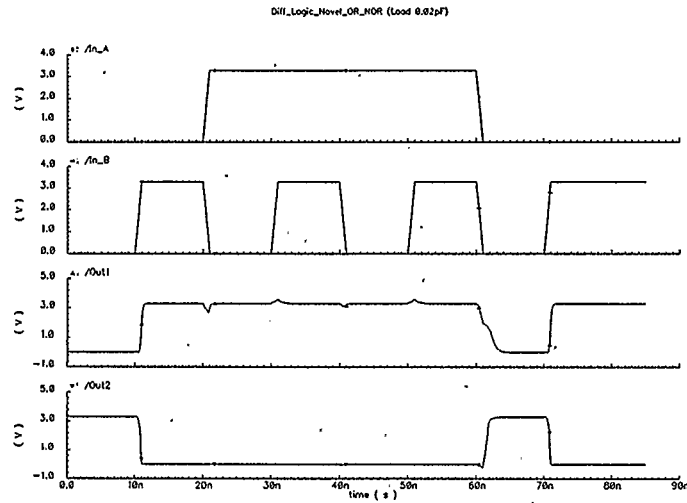


Figure 6.13: Simulation result of OR/NOR logic with 0.02pF load

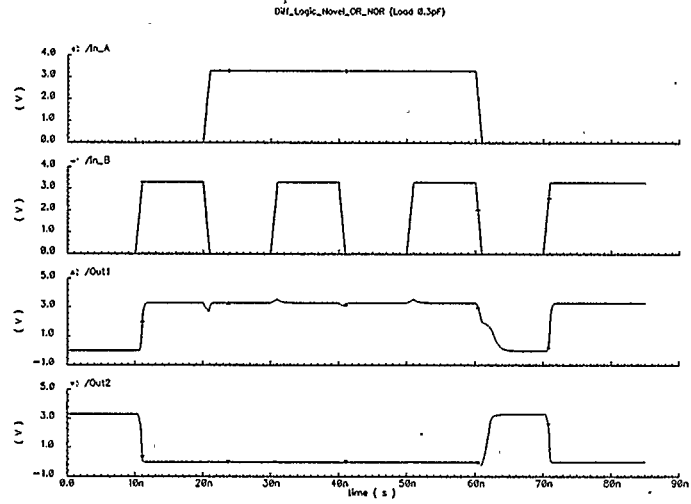


Figure 6.14: Simulation result of OR/NOR logic with 0.3pF load

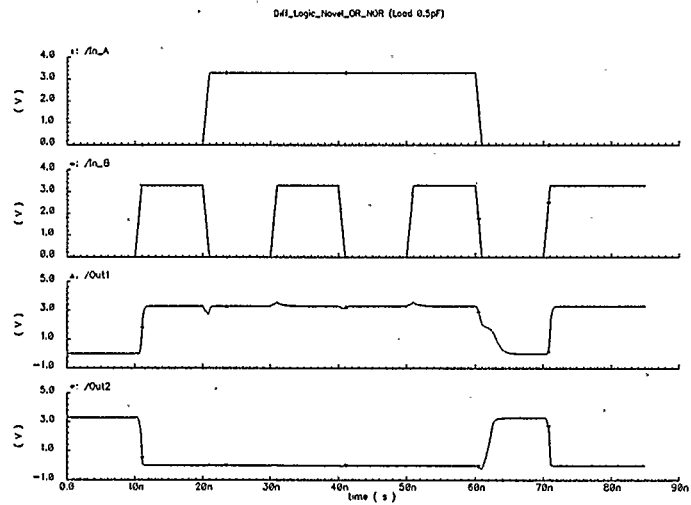


Figure 6.15: Simulation result of OR/NOR logic with 0.5pF load

Chapter 7

Contributions, Future Work and Conclusion

7.1 Contributions

The virtually unexplored MLSD detection for ternary partial-response signaling channel detection has been investigated. A new and efficient differential threshold MLSD approach has been taken (from an analog perspective) and the first reported implementation for the ternary dicode channel has been published [61]. Also the first reported differential threshold MLSD algorithm for the ternary duobinary PRS channel has been presented in this dissertation.

The foundation for ternary PRS channel detection was developed, and a mixed-signal hardware architecture was proposed and implemented. Furthermore, ternary dicode detection was compared with that of the well-known and well-studied binary channel detector for the first time.

The research work presented in this dissertation has resulted in the extension of the sampled-analog signal processing technique beyond the common application in binary type implementations presently available in the literature. This approach is intended to be further targeted at the following likely future work.

7.2 Future Work

The proposed detector implementation in this dissertation was based on a 3.3V single-ended, voltage-mode circuit implementation, just to serve as a proof of concept. It is of further interest to develop, design and fabricate a fully differential mixed-signal circuit implementation (in voltage-mode as well as current-mode) for both the ternary dicode and duobinary channel detectors using reduced power supply voltages (1.8V). The first fully current-mode analog type MLSD detector was recently reported [42] but the implementation was based on the naive add-compare-select (ACS) approach.

Furthermore, it is also of interest to fully investigate and re-interpret the proposed detection algorithms based on the input-interleaved principle, which could further enhance high-speed operation as proven by the reported binary dicode detector implementation [23].

It is also of interest to re-interpret the differential threshold algorithms for both ternary dicode and duobinary channels from a digital implementation perspective; propose architecture(s) and implementation(s), and perform a comparative study with its analog counterpart(s).

Moreover, it is worth pointing out that the survivor memory management circuitry for the detectors was partially investigated. Therefore, it will also be of interest to ensure that a fully implementable register-exchange ternary survivor memory man-

agement circuit is developed and incorporated in subsequent designs either in CMOS or BiCMOS technology.

Another avenue of interest is investigating the possibility of applying reduced-state sequence detection (RSSD) approach to the ternary PRS channels and developing a mixed-signal implementation. It is expected that if applicable, the RSSD approach will further reduce the circuit complexity at the cost of negligible error-rate degradation [74]. A Literature survey revealed that the RSSD approach has not been investigated from an analog perspective until very recently [75], and it would be interesting to know how applicable this approach is to ternary PRS channel detection.

Also, foreseeable future work is the investigation of soft-output Viterbi detection (SOVA) for the two fundamental PRS channels from a differential threshold and analog mixed-signal perspective. Very recently, a difference-metric digital SOVA approach was developed and applied to the binary decode channel for PR4 detection [26]. However, a literature survey reveals the need to have this approach re-interpreted from an analog perspective and applied to the ternary channels investigated in this dissertation. It is expected that by so doing, lower error rates can be accomplished in comparison to that obtainable using the classical Viterbi approach.

Finally, it is of interest to design and implement an actual EPR4 channel detector using the developed ternary PRS channel detectors. By so doing, it would be interesting to examine how closely the simplified EPR4 detection techniques of Friedmann [17] and Wood [16] approximate the actual and conventional EPR4 detection. Also the complexity and performance of the simplified EPR4 detector implementation could be compared to other reported implementations in the literature [76, 77, 78, 79, 80].

It is worth noting that the above mentioned future work is expected to be carried from the conception stage through to design, implementation, and physical testing

and verification.

7.3 Conclusion

Presented in this dissertation, is a new and efficient approach for the maximum-likelihood sequence detection of ternary fundamental ($1-D$ and $1+D$) partial-response channels. Algorithm(s) derivation, interpretation and characterization as well as possible hardware implementation from a mixed-signal perspective were presented in the preceding chapters.

This dissertation represents the first reported research work on the development, and implementation of ternary PRS channel detectors.

Bibliography

- [1] H. Nyquist, "Certain topics on telegraph transmission theory," *Transactions AIEE*, vol. 47, pp. 617-644, April 1928.
- [2] P. Elias, "Coding for noisy channels," *IRE Conv. Rec.*, vol. pt. IV, pp. 37-46, 1955.
- [3] A. Lender, "Correlative digital communication techniques," *IEEE Transactions on Communication Technology*, vol. COM-12, pp. 128-135, 1964.
- [4] —, "The duobinary technique for high speed data transmission," *IEEE Transactions on Communication Electronics*, vol. 82, pp. 214-218, May 1963.
- [5] E. R. Kretzmer, "Generalization of a technique for binary data communication," *IEEE Transactions on Communication Technology (Concise Papers)*, vol. COM-14, pp. 67-68, February 1966.
- [6] A. M. Gerrish and R. D. Howson, "Multilevel Partial-Response Signalling," *IEEE International Conference on Communications Rec.*, p. 186, June 1967.
- [7] H. Kobayashi, "Correlative Level Coding and Maximum-Likelihood Decoding," *IEEE Transactions on Information Theory*, vol. IT-17, no. 5, pp. 586-594, September 1971.

- [8] P. Kabal and S. Pasupathy, "Partial-Response Signaling," *IEEE Transactions on Communications*, vol. COM-23, no. 9, pp. 921-934, September 1975.
- [9] H. Kobayashi and D. T. Tang, "Application of Partial Response Channel Coding to Magnetic Recording Systems," *IBM Journal of Research and Development*, vol. 15, July 1970.
- [10] H. Thapar *et al.*, "Hard Disk Drive Read Channels: Technology and Trends," *Proceedings of the Custom Integrated Circuits Conference*, pp. 309-316, 1998.
- [11] H. Kobayashi, "Application of probability decoding to digital magnetic recording systems," *IBM Journal of Research and Development*, p. 64, January 1971.
- [12] G. D. Forney Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Transactions on Information Theory*, vol. IT-18, no. 3, pp. 363-377, May 1972.
- [13] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, no. 2, pp. 260-269, April 1967.
- [14] ———, "Convolutional codes: The state diagram approach to optimal decoding and performance analysis for memoryless channels," *Jet Propulsion Laboratory, California Institute of Technology, Pasadena, Space Program Summary 37-58*, vol. 3, pp. 50-55, August 1969.
- [15] M. J. Ferguson, "Optimal reception for binary partial response channels," *The Bell Systems Technical Journal*, vol. 51, pp. 493-505, February 1972.
- [16] R. Wood, "Turbo-PRML: A Compromise EPRML Detector," *IEEE Transactions on Magnetics*, vol. 29, no. 6, pp. 4018-4020, November 1993.

- [17] A. Friedmann and J. K. Wolf, "Simplified EPR4 Detection," *IEEE Transactions on Magnetics*, vol. 34, no. 1, pp. 129–134, January 1998.
- [18] J. W. M. Bergmanns, K. D. Fisher, and H. W. Wong-Lam, "Variations on the Ferguson Viterbi Detector," *Philips Journal of Research*, vol. 47, no. 6, pp. 361–386, December 1993.
- [19] T. W. Matthews and R. R. Spencer, "An Integrated Analog CMOS Viterbi Detector for Digital Magnetic Recording," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 1294–1302, 1993.
- [20] —, "An Integrated Analog CMOS Viterbi Detector for Digital Magnetic Recording," *Presented at IEEE International Solid-State Circuits Conference*, pp. 214–216, 1993.
- [21] M. Shakiba, D. A. Johns, and K. Martin, "Analog Implementation of Class-IV Partial-Response Viterbi Detector," *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 91–94, 1995.
- [22] —, "A 200MHz 3.3V BiCMOS Class-IV Partial-Response Analog Viterbi Decoder," *Proceedings of IEEE Custom Integrated Circuits Conference*, pp. 567–570, 1995.
- [23] —, "An Integrated 200MHz 3.3V BiCMOS Class-IV Partial-Response Analog Viterbi Decoder," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 1, pp. 61–75, January 1998.
- [24] R. R. Spencer, "Simulated Performance of Analog Viterbi Detectors," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 1, pp. 277–288, January 1992.

- [25] R. Wood and D. A. Petersen, "Viterbi Detection of Class IV Partial Response on a Magnetic Recording Channel," *IEEE Transactions on Communications*, vol. COM-34, no. 5, pp. 454-461, May 1986.
- [26] W. J. Gross, V. C. Gaudet, and P. G. Gulak, "Difference Metric Soft-Output Detection: Architecture and Implementation," *IEEE Transactions on Circuit and Systems - II*, vol. 48, no. 10, pp. 904-911, October 2001.
- [27] A. Friedmann and J. K. Wolf, "A Sliding Threshold Detector for the Ternary 1-D Channel," *IEEE Transactions on Magnetics*, vol. 32, no. 5, pp. 3959-3961, September 1996.
- [28] S. Pasupathy, "Correlative Coding: A Bandwidth-Efficient Signaling Scheme," *IEEE Communications Society Magazine*, vol. 15, pp. 4-11, 1977.
- [29] L. W. Couch II, *Digital and Analog Communication Systems*. New York: Macmillan Publishing Company, 1987.
- [30] H. Thapar and A. M. Patel, "A Class of Partial Response Systems for Increasing Storage Density in Magnetic Recording," *IEEE Transactions on Magnetics*, vol. MAG-23, no. 5, pp. 3666-3668, September 1987.
- [31] J. K. Omura, "On the Viterbi Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. IT-15, pp. 177-179, 1969.
- [32] A. P. Hekstra, "An Alternative to Metric Rescaling in Viterbi Decoders," *IEEE Transactions on Communications*, vol. 37, no. 11, pp. 1220-1222, November 1989.
- [33] G. D. Forney Jr., "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268-278, March 1973.

- [34] E. A. Lee and D. G. Messerschmitt, *Digital Communication*. Boston: Kluwer Academic Publishers, 1988.
- [35] S. Wicker, *Error Control Systems for Digital Communication and Storage*. Englewood Cliffs, New Jersey: Prentice hall, 1995.
- [36] H. L. Lou, "Implementing the Viterbi Algorithm," *IEEE Signal Processing Magazine*, pp. 42–52, 1995.
- [37] A. S. Acampora and R. P. Gilmore, "Analog Viterbi Decoding for High Speed Digital Satellite Channel," *IEEE Transactions on Communications*, vol. COM-26, pp. 1463–1470, October 1978.
- [38] K. He and G. Cauwenberghs, "An Area-Efficient Analog VLSI Architecture for State-Parallel Viterbi Decoding," *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 432–435, 1999.
- [39] A. Demosthenous and J. Taylor, "BiCMOS Add-Compare-Select Units for Viterbi Decoders," *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 209–212, May 31–Jun 3 1998.
- [40] —, "Low-Power CMOS and BiCMOS Circuits for Analog Convolutional Decoders," *IEEE Transactions on Circuit and Systems II*, vol. 46, no. 8, pp. 1077–1080, August 1999.
- [41] A. Demosthenous, J. Taylor, and C. Verdier, "A New Architecture for Low Power Analogue Convolutional Decoders," *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 37–40, June 9–12 1997.
- [42] A. Demosthenous and J. Taylor, "A 100Mb/s 2.8-V CMOS Current-Mode Analog Viterbi Decoder," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 7, pp. 904–910, July 2002.

- [43] X. A. Wang and S. B. Wicker, "An Artificial Neural Net Viterbi Decoder," *IEEE Transactions on Communications*, vol. 44, no. 2, pp. 165–171, February 1996.
- [44] M. Shakiba, D. A. Johns, and K. Martin, "BiCMOS Circuits for Analog Viterbi Decoders," *IEEE Transactions on Circuits and Systems II*, vol. 45, no. 12, pp. 1527–1537, December 1998.
- [45] G. Fettweis and H. Meyr, "Parallel Viterbi Algorithm Implementation: Breaking the ACS-Bottleneck," *IEEE Transactions on Communications*, vol. 37, no. 8, pp. 785–790, August 1995.
- [46] —, "High-Speed Parallel Viterbi Decoding: Algorithm and VLSI Architecture," *IEEE Communications Magazine*, vol. 29, no. 5, pp. 46–55, May 1991.
- [47] K. J. Knudson, J. K. Wolf, and L. B. Milstein, "Dynamic Threshold Implementation of the Maximum-Likelihood Detector for the EPR4 Channel," *IEEE Global Telecommunications Conference*, vol. 3, pp. 2135–2139, 1991.
- [48] G. Fettweis, "Algebraic Survivor Memory Management Design for Viterbi Detectors," *IEEE Transactions on Communications*, vol. 43, no. 9, pp. 2458–2463, September 1995.
- [49] K. He and G. Cauwenberghs, "Performance of Analog Viterbi Decoding," *Proceedings of the 42nd IEEE Midwest Symposium on Circuits and Systems*, vol. 1, pp. 2–5, August 1999.
- [50] S. A. Altekari, M. Berggren, B. E. Moision, P. H. Siegel, and J. K. Wolf, "Error-Event Characterization on Partial-Response Channels," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 241–247, January 1999.

- [51] M. Altarriba and R. Spencer, "An Architecture for High-Order, Variable Polynomial Analog Viterbi Detectors," *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, vol. 1, pp. 268–271, August 1997.
- [52] T. Deliyannis, J. K. Fidler, and Y. Sun, *Continuous-time active filter design*. Boca Raton, Florida: CRC Press, 1999.
- [53] J. T. Wu and B. A. Wooley, "A 100MHz Pipelined CMOS Comparators," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 6, pp. 1379–1385, December 1988.
- [54] B. Razavi and B. A. Wooley, "Design Techniques for High-Speed, High-Resolution Comparators," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 12, pp. 1916–1926, December 1992.
- [55] C. M. Rader, "Memory Management in a Viterbi Decoder," *IEEE Transactions on Communications*, vol. COM-29, pp. 1399–1401, 1981.
- [56] G. Feygin and P. G. Gulak, "Survivor Sequence Memory Management in Viterbi Decoders," *IEEE International Symposium on Circuits and Systems*, vol. 5, pp. 2967–2970, June 1991.
- [57] P. J. Black and T. H. Meng, "Hybrid Survivor Path Architecture for Viterbi Decoders," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 433–436, April 1993.
- [58] O. Collins and F. Pollara, "Memory Management in Traceback Viterbi Decoders," *The telecommunications and data acquisition progress report, Jet Propulsion Lab, California Institute of Technology*, pp. 98–104, November 1989.
- [59] U. Cilingiroglu and Y. Ozelci, "Multiple-Valued Static CMOS Memory Cell," *IEEE Transactions on Circuit and Systems - II*, vol. 48, no. 3, pp. 282–290, March 2001.

- [60] A. Friedmann, "Measurements, Characterization, and System Design for Digital Storage," *Ph.D. Dissertation, University of California, San Diego*, 1997.
- [61] I. A. Omole, B. J. Maundy, and A. B. Sesay, "Region-Dependent, Algorithmically Directed Thresholds for Maximum-Likelihood Detection of Ternary PRS Channels," *IEEE Transactions on Circuits and Systems II*, vol. 49, no. 12, pp. 775–783, December 2002.
- [62] T. Serrano and B. L. Barranco, "A Modular Current-Mode High-Precision Winner-Take-All Circuit," *IEEE Transactions on Circuits and System II*, vol. 42, pp. 132–134, 1995.
- [63] S. I. Liu, C. Y. Chen, J. G. Hwu, and P. Chen, "Analog Maximum, Median and Minimum Circuit," *Presented at IEEE International Symposium on Circuits and Systems*, 1997.
- [64] D. M. Wilson and S. P. DeWeerth, "Winning isn't Everything," *Presented at IEEE International Symposium on Circuits and Systems*, 1995.
- [65] Z. S. Gunay and E. Sanchez-Sinencio, "CMOS Winner-Take-All Circuits: A Detail Comparison," *IEEE International Symposium on Circuits and Systems*, pp. 41–44, June 1997.
- [66] A. Demosthenous, S. Smedley, and J. Taylor, "A CMOS Analog Winner-Take-All Network for Large-Scale Applications," *IEEE Transaction on Circuit and Systems I*, vol. 45, no. 3, pp. 300–304, March 1998.
- [67] G. Patel and S. P. DeWeerth, "An Analog VLSI Loser-Take-All Circuit," *Presented at IEEE International Symposium on Circuits and Systems*, 1995.

- [68] S. Siskos, S. Vlassis, and I. Pitas, "Analog implementation of fast min/max filtering," *IEEE Transactions on Circuits and System II*, vol. 45, pp. 913–918, 1998.
- [69] C. Toumazou, F. J. Lidgey, and D. G. Haigh, *Analogue IC Design: The Current-Mode Approach*. Stevenage, UK: Peregrinus, 1990.
- [70] I. A. Omole and B. J. Maundy, "Versatile Current-Mode Loser-Take-All Circuit for Analog Decoders," *Proceedings of the 44th IEEE Midwest Symposium on Circuits and Systems*, vol. 2, pp. 748–751, 2001.
- [71] K. Martin, *Digital Integrated Circuit Design*. New York: Oxford University Press, Inc., 2000.
- [72] L. Heller *et al.*, "Cascade Voltage Switch Logic: A Differential CMOS Logic Family," *Proceedings of the IEEE ISSCC Conference*, pp. 16–17, February 1984.
- [73] K. Chu and D. Pulfrey, "Design Procedures for Differential Cascade Logic," *IEEE Journal of Solid-State Circuits*, vol. SC-21, no. 6, pp. 1082–1087, December 1986.
- [74] S. Olcer, "Reduced-state sequence detection of multilevel partial-response signals," *IEEE Transactions on Communication*, vol. 40, pp. 3–6, January 1992.
- [75] B. Zand and D. A. Johns, "High-Speed CMOS Analog Viterbi Detector for 4-PAM Partial-Response Signaling," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 7, pp. 895–903, July 2002.
- [76] M. Demicheli *et al.*, "A 450Mbit/s EPR4 PRML Read/Write Channel," *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 317–320, May 1999.

- [77] T. Conway *et al.*, "A CMOS 260Mbps Read Channel with EPRML performance," *Proceedings of the IEEE Symposium on VLSI Circuits Digest of Technical Papers*, pp. 152–155, June 1998.
- [78] D. C. Wei *et al.*, "An Analog EPR4 Read Channel with an FDTS Detector," *Proceedings of the IEEE International Conference on Communications*, pp. 678–682, June 1998.
- [79] K. Fukahori *et al.*, "An Analog EPR4 Viterbi Detector in Read Channel IC for Magnetic Hard Disks," *IEEE International Solid-State Circuits Conference*, pp. 380–381, February 1998.
- [80] M. Leung *et al.*, "A 300Mb/s BiCMOS EPR4 Read Channel for Magnetic Hard Disks," *Presented at IEEE International Solid-State Circuits Conference*, pp. 378–379, 1998.

Appendix A

Duobinary channel detection

A.1 Derivation of updates

The necessary updates for the propagating differential thresholds in the ternary duobinary channel detector can be derived in a similar manner to those for the ternary decode channel detector in Section 3.2.3. Here, the updates will be derived in conjunction with Equations(3.58)-(3.60).

For Region B_1 at time $k + 1$,

$$\begin{aligned} m_{+1}(k+1) &= m_{-1}(k) \\ m_0(k+1) &= m_{-1}(k) - y_k - 0.5 \\ m_{-1}(k+1) &= m_{-1}(k) - 2y_k - 2 \end{aligned} \tag{A.1}$$

Therefore based on Equation(A.1),

$$\begin{aligned} \Delta m_a(k+1) &= m_0(k+1) - m_{+1}(k+1) \\ &= m_{-1}(k) - y_k - 0.5 - m_{-1}(k) \\ &= -y_k - 0.5 \end{aligned} \tag{A.2}$$

and

$$\begin{aligned}
 \Delta m_b(k+1) &= m_{-1}(k+1) - m_0(k+1) \\
 &= m_{-1}(k) - 2y_k - 2 - [m_{-1}(k) - y_k - 0.5] \\
 &= -y_k - 1.5
 \end{aligned} \tag{A.3}$$

For Region B_2 at time $k+1$,

$$\begin{aligned}
 m_{+1}(k+1) &= m_{-1}(k) \\
 m_0(k+1) &= m_{-1}(k) - y_k - 0.5 \\
 m_{-1}(k+1) &= m_0(k) - y_k - 0.5
 \end{aligned} \tag{A.4}$$

Therefore based on Equation(A.4),

$$\begin{aligned}
 \Delta m_a(k+1) &= m_{-1}(k) - y_k - 0.5 - m_{-1}(k) \\
 &= -y_k - 0.5
 \end{aligned} \tag{A.5}$$

and

$$\begin{aligned}
 \Delta m_b(k+1) &= m_0(k) - y_k - 0.5 - [m_{-1}(k) - y_k - 0.5] \\
 &= -\Delta m_b(k)
 \end{aligned} \tag{A.6}$$

For Region $B_3(i)$ at time $k+1$,

$$\begin{aligned}
 m_{+1}(k+1) &= m_{-1}(k) \\
 m_0(k+1) &= m_0(k) \\
 m_{-1}(k+1) &= m_0(k) - y_k - 0.5
 \end{aligned} \tag{A.7}$$

Therefore based on Equation(A.7),

$$\begin{aligned}
 \Delta m_a(k+1) &= m_0(k) - m_{-1}(k) \\
 &= -\Delta m_b(k)
 \end{aligned} \tag{A.8}$$

and

$$\begin{aligned}\Delta m_b(k+1) &= m_0(k) - y_k - 0.5 - m_0(k) \\ &= -y_k - 0.5\end{aligned}\tag{A.9}$$

For Region $B_3(ii)$ at time $k+1$,

$$\begin{aligned}m_{+1}(k+1) &= m_{-1}(k) \\ m_0(k+1) &= m_0(k) \\ m_{-1}(k+1) &= m_{+1}(k)\end{aligned}\tag{A.10}$$

Therefore based on Equation(A.10),

$$\begin{aligned}\Delta m_a(k+1) &= m_0(k) - m_{-1}(k) \\ &= -\Delta m_b(k)\end{aligned}\tag{A.11}$$

and

$$\begin{aligned}\Delta m_b(k+1) &= m_{+1}(k) - m_0(k) \\ &= -\Delta m_a(k)\end{aligned}\tag{A.12}$$

For Region $B_3(iii)$ at time $k+1$,

$$\begin{aligned}m_{+1}(k+1) &= m_0(k) + y_k - 0.5 \\ m_0(k+1) &= m_0(k) \\ m_{-1}(k+1) &= m_{+1}(k)\end{aligned}\tag{A.13}$$

Therefore based on Equation(A.13),

$$\begin{aligned}\Delta m_a(k+1) &= m_0(k) - [m_0(k) + y_k - 0.5] \\ &= -y_k + 0.5\end{aligned}\tag{A.14}$$

and

$$\begin{aligned}\Delta m_b(k+1) &= m_{+1}(k) - m_0(k) \\ &= -\Delta m_a(k)\end{aligned}\tag{A.15}$$

For Region $B_3(iv)$ at time $k+1$,

$$\begin{aligned}m_{+1}(k+1) &= m_0(k) + y_k - 0.5 \\ m_0(k+1) &= m_0(k) \\ m_{-1}(k+1) &= m_0(k) - y_k - 0.5\end{aligned}\tag{A.16}$$

Therefore based on Equation(A.16),

$$\begin{aligned}\Delta m_a(k+1) &= m_0(k) - [m_0(k) + y_k - 0.5] \\ &= -y_k + 0.5\end{aligned}\tag{A.17}$$

and

$$\begin{aligned}\Delta m_b(k+1) &= m_0(k) - y_k - 0.5 - m_0(k) \\ &= -y_k - 0.5\end{aligned}\tag{A.18}$$

For Region B_4 at time $k+1$,

$$\begin{aligned}m_{+1}(k+1) &= m_0(k) + y_k - 0.5 \\ m_0(k+1) &= m_{+1}(k) + y_k - 0.5 \\ m_{-1}(k+1) &= m_{+1}(k)\end{aligned}\tag{A.19}$$

Therefore based on Equation(A.19),

$$\begin{aligned}\Delta m_a(k+1) &= m_{+1}(k) + y_k - 0.5 - [m_0(k) + y_k - 0.5] \\ &= -\Delta m_a(k)\end{aligned}\tag{A.20}$$

and

$$\begin{aligned}\Delta m_b(k+1) &= m_{+1}(k) - [m_{+1}(k) + y_k - 0.5] \\ &= -y_k + 0.5\end{aligned}\tag{A.21}$$

For Region B_5 at time $k+1$,

$$\begin{aligned}m_{+1}(k+1) &= m_{+1}(k) + 2y_k - 2 \\ m_0(k+1) &= m_{+1}(k) + y_k - 0.5 \\ m_{-1}(k+1) &= m_{+1}(k)\end{aligned}\tag{A.22}$$

Therefore based on Equation(A.22),

$$\begin{aligned}\Delta m_a(k+1) &= m_{+1}(k) + y_k - 0.5 - [m_{+1}(k) + 2y_k - 2] \\ &= -y_k + 1.5\end{aligned}\tag{A.23}$$

and

$$\begin{aligned}\Delta m_b(k+1) &= m_{+1}(k) - [m_{+1}(k) + y_k - 0.5] \\ &= -y_k + 0.5\end{aligned}\tag{A.24}$$

Appendix B

Limitation Effect: Ternary 1-D channel

To understand the convergence characteristics of the detection algorithm in the presence of saturation effect, the relative likelihood of the two propagating differential thresholds Δm_c and Δm_d , will be assumed to be equal. i.e. $\Delta m_c = \Delta m_d = 0$ (or initial condition).

Assuming the initial condition is valid, then in region A_1 :

$$\begin{aligned} L_y + 1.5 &< 0 \\ L_y &< -1.5 \end{aligned} \tag{B.1}$$

Region A_2 :

$$\begin{aligned} L_y + 0.5 &< 0 < L_y + 1.5 \\ -1.5 &< L_y < -0.5 \end{aligned} \tag{B.2}$$

Region $A_3(i)$:

$$L_y - 0.5 < 0 < L_y + 0.5 \text{ and } 0 > L_y + 0.5$$

$$-0.5 < L_y < +0.5 \text{ and } L_y < -0.5 \quad (\text{B.3})$$

Region $A_3(ii)$:

$$\begin{aligned} L_y + 0.5 &> 0 \text{ and } L_y - 0.5 < 0 \\ L_y &> -0.5 \text{ and } L_y < +0.5 \end{aligned} \quad (\text{B.4})$$

Region $A_3(iii)$:

$$\begin{aligned} L_y - 0.5 < 0 < L_y + 0.5 \text{ and } 0 < L_y - 0.5 \\ -0.5 < L_y < +0.5 \text{ and } L_y > +0.5 \end{aligned} \quad (\text{B.5})$$

Region A_4 :

$$\begin{aligned} L_y - 1.5 < 0 < L_y - 0.5 \\ +0.5 < L_y < +1.5 \end{aligned} \quad (\text{B.6})$$

Region A_5 :

$$\begin{aligned} L_y - 1.5 &> 0 \\ L_y &> +1.5 \end{aligned} \quad (\text{B.7})$$

B.1 Positive plane limitation

Limiting Category 1: $0 < L_y < +0.5$.

Under this limiting condition, only Equation (B.4) is satisfied. Hence, the two propagating thresholds are updated such that $\Delta m_c(k+1) = \Delta m_c(k)$ and $\Delta m_d(k+1) = \Delta m_d(k)$. This invariably implies that the two thresholds will remain stuck at the initial condition, and the algorithm is prevented from any convergence.

Limiting Category 2: $+0.5 < L_y < +1.5$.

Based on this limitation on L_y , only Equation (B.6) is satisfied. Accordingly, Δm_c will retain its state of inertia, since $\Delta m_c(k+1) = \Delta m_d(k)$ while Δm_d will ensure a partial adaptation of the algorithm because $\Delta m_d(k+1) = L_y - 0.5$. The change in value for Δm_d would eventually affect the state of Δm_c and this in turn will enhance the adaptation of the algorithm.

Limiting Category 3: $L_y > +1.5$.

Under this condition, only Equation (B.7) is satisfied. Thus, Δm_c and Δm_d will both be prevented from remaining in their state of inertia by assuming the value of $L_y - 1.5$ and $L_y - 0.5$, respectively. These will cause a quick adaptation of the algorithm towards its steady state convergence.

B.2 Negative plane limitation

Limiting Category 1: $-0.5 < L_y < 0$.

Also, under this limiting condition, the algorithm will not experience any adaptation since only Equation (B.4) is satisfied. Thus, Δm_c and Δm_d will remain stuck at their initial condition.

Limiting Category 2: $-1.5 < L_y < -0.5$.

Equation (B.2) is satisfied, and Δm_c and Δm_d are updated such that only Δm_c will not retain its initial condition by taking on the value of $L_y > +0.5$; while $\Delta m_d(k+1) = \Delta m_c(k)$. Under this condition, the algorithm will adapt slowly.

Limiting Category 3: $L_y < -1.5$.

Under this condition, only Equation (B.1) is satisfied. Thus, Δm_c and Δm_d will both simultaneously transit from their state of inertia by assuming the value

of $L_y + 0.5$ and $L_y + 1.5$, respectively. These will cause a quick adaptation of the algorithm towards its steady state convergence.

Appendix C

Detector Output Validation

The detector architecture uses six comparators; all of which need to have their digital outputs read in parallel. Since there are six comparators in all, then $2^6 = 64$ different binary combinations are likely without any restraining condition(s).

C.1 Output Logic Combinations

It can be observed from the detector architecture (Figure 5.1), that a group of three different comparators rely on each of the differential thresholds Δm_c and Δm_d for decision making. Therefore, by grouping into two, it is possible to extract the valid 3-bit combination of binary output from the $2^3 = 8$ likely combinations in each group. The results are shown in table C.1 and C.2.

Tables C.1 and C.2 shows that there exist $2^4 = 16$ valid 6-bit output combinations under no restraining condition(s). However, by recalling the derivations in subsection 3.2.2, the ternary dicode detection algorithm is based on the fact that $\Delta m_d > \Delta m_c$ (justified by Equations (3.6), (3.9) and (3.12)).

Table C.1: Group A: Comparators 1 to 3

$C_1 \longrightarrow C_3$	Condition(s)	Inference
000	$y_k - 0.5 < \Delta m_c < y_k + 0.5$	Valid
001	$\Delta m_c < y_k - 0.5$	Valid
010	$y_k + 0.5 < \Delta m_c < y_k + 1.5$	Valid
011	$y_k + 0.5 < \Delta m_c < y_k + 1.5$ and $y_k - 0.5 > \Delta m_c$	Invalid
100	$\Delta m_c > y_k + 1.5$ and $y_k - 0.5 < \Delta m_c < y_k + 0.5$	Invalid
101	$\Delta m_c > y_k + 1.5$ and $\Delta m_c < y_k - 0.5$	Invalid
110	$\Delta m_c > y_k + 1.5$	Valid
111	$\Delta m_c > y_k + 1.5$ and $\Delta m_c < y_k - 0.5$	Invalid

Table C.2: Group B: Comparators 4 to 6

$C_4 \longrightarrow C_6$	Condition(s)	Inference
000	$y_k - 0.5 < \Delta m_d < y_k + 0.5$	Valid
001	$y_k - 0.5 < \Delta m_d < y_k + 0.5$ and $\Delta m_d < y_k - 1.5$	Invalid
010	$y_k - 1.5 < \Delta m_d < y_k - 0.5$	Valid
011	$\Delta m_d < y_k - 1.5$	Valid
100	$\Delta m_d > y_k + 0.5$	Valid
101	$\Delta m_d > y_k + 0.5$ and $\Delta m_d < y_k - 1.5$	Invalid
110	$\Delta m_d > y_k + 0.5$ and $y_k - 1.5 < \Delta m_d < y_k - 0.5$	Invalid
111	$\Delta m_d > y_k + 0.5$ and $\Delta m_d < y_k - 1.5$	Invalid

Table C.3: Detector valid 6-bit outputs

$C_1 \rightarrow C_6$	Condition	Region Detected
110100	equation (3.29)	A_1
010100	equation (3.30)	A_2
000100	equation (3.31)	$A_3(i)$
000000	equation (3.32)	$A_3(ii)$
001000	equation (3.33)	$A_3(iii)$
001100	equation (5.17)	$A_3(iv)$ (previously un-noticed)
001010	equation (3.34)	A_4
001011	equation (3.35)	A_5

Therefore by restraining the detector architecture with the above condition, there exist only eight 6-bit valid detector comparator outputs as shown in table C.3.

C.2 Survivor Memory Management Validation

Table C.3 provides another means to corroborate the link between Equations (3.29)-(3.35), Equation (5.17) and the survivor memory management (SMM) circuitry shown in Figure 5.24, which is based on table 5.3 (derived from Equations (3.16)-(3.24)).

For the SMM circuitry proposition to be valid, it is expected that adjacent flip-flop in each layer (row) of SMM registers would exchange information chosen from the distinct set $\{V_{dd}, V_{cm}, \text{gnd}\} = \{3.3\text{V}, 1.65\text{V}, 0\text{V}\}$ for this design. This implies that at every detection time-step, one and only one of the select signal in each layer must be “HIGH” and the combination of all “HIGH” select signals (three in total) must

result in one and only one of the detectable regions.

For combination "110100":

$$S_7 = S_{10} = S_{13} = HIGH \quad (C.1)$$

Therefore, region A_1 would be detected.

For combination "010100":

$$S_7 = S_{10} = S_{14} = HIGH \quad (C.2)$$

Therefore, region A_2 would be detected.

For combination "000100":

$$S_7 = S_{11} = S_{14} = HIGH \quad (C.3)$$

Therefore, region $A_3(i)$ would be detected.

For combination "000000":

$$S_7 = S_{11} = S_{15} = HIGH \quad (C.4)$$

Therefore, region $A_3(ii)$ would be detected.

For combination "001000":

$$S_8 = S_{11} = S_{15} = HIGH \quad (C.5)$$

Therefore, region $A_3(iii)$ would be detected.

For combination "001100":

$$S_8 = S_{11} = S_{14} = HIGH \quad (C.6)$$

Therefore, region $A_3(iv)$ would be detected.

For combination "001010":

$$S_8 = S_{12} = S_{15} = HIGH \quad (C.7)$$

Therefore, region A_4 would be detected.

For combination "001011":

$$S_9 = S_{12} = S_{15} = HIGH \quad (C.8)$$

Therefore, region A_5 would be detected.

Appendix D

Detection Scheme Modifications

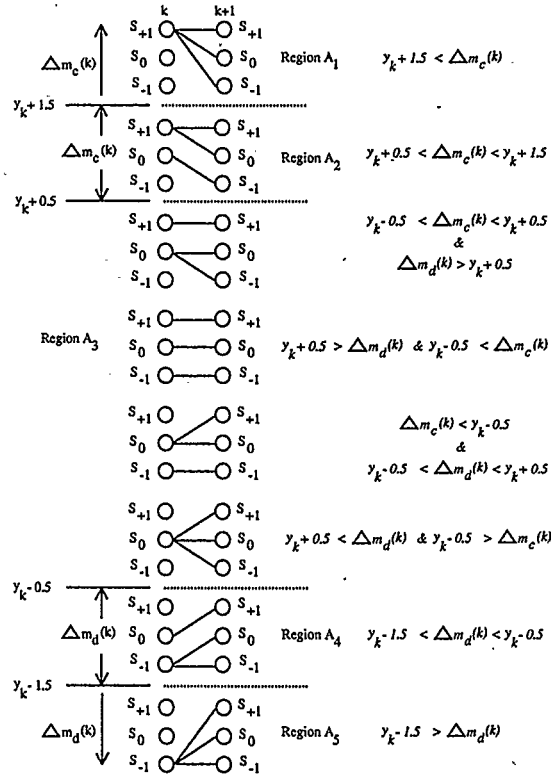


Figure D.1: Updated Region-dependency of thresholds for the decode channel

Table D.1: A sample of region occurrence probability

Region	1024 samples	10240 samples	100352 samples
A_1	0.1748	0.1740	0.1731
A_2	0.1543	0.1602	0.1632
$A_3(i)$	0.1084	0.1081	0.1081
$A_3(ii)$	0.0762	0.0987	0.0997
$A_3(iii)$	0.1406	0.1102	0.1142
$A_3(iv)$	9.766×10^{-5}	9.766×10^{-5}	9.965×10^{-6}
A_4	0.1729	0.1805	0.1725
A_5	0.1719	0.1683	0.1693

Table D.2: Logic operations for updated control signal generator

Region	Comparator Output needed	Required Logic Operation
A_1	$C_1 = 1$	C_1 (appropriately delayed to minimize timing mis-match)
A_2	$C_1 = 0$ and $C_2 = 1$	$\overline{C_1} + \overline{C_2}$
$A_3(i)$	$C_2 = 0$, $C_3 = 0$ and $C_4 = 1$	$C_2 + \overline{C_3}C_4$
$A_3(ii)$	$C_3 = 0$ and $C_4 = 0$	No additional logic required
$A_3(iii)$	$C_3 = 1$, $C_4 = 0$ and $C_5 = 0$	$\overline{C_3}\overline{C_4} + C_5$
$A_3(iv)$	$C_3 = 1$, $C_4 = 1$	$\overline{C_3} + \overline{C_4}$
A_4	$C_5 = 1$ and $C_6 = 0$	$\overline{C_5} + C_6$
A_5	$C_6 = 1$	C_6 (appropriately delayed to minimize timing mis-match)

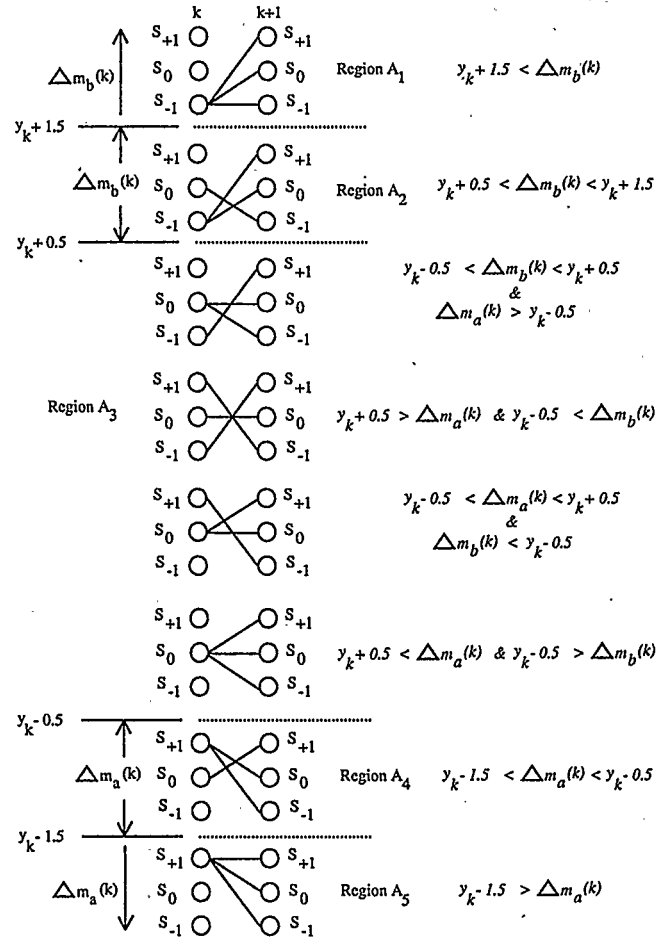


Figure D.2: Updated Region-dependency of thresholds for the duobinary channel

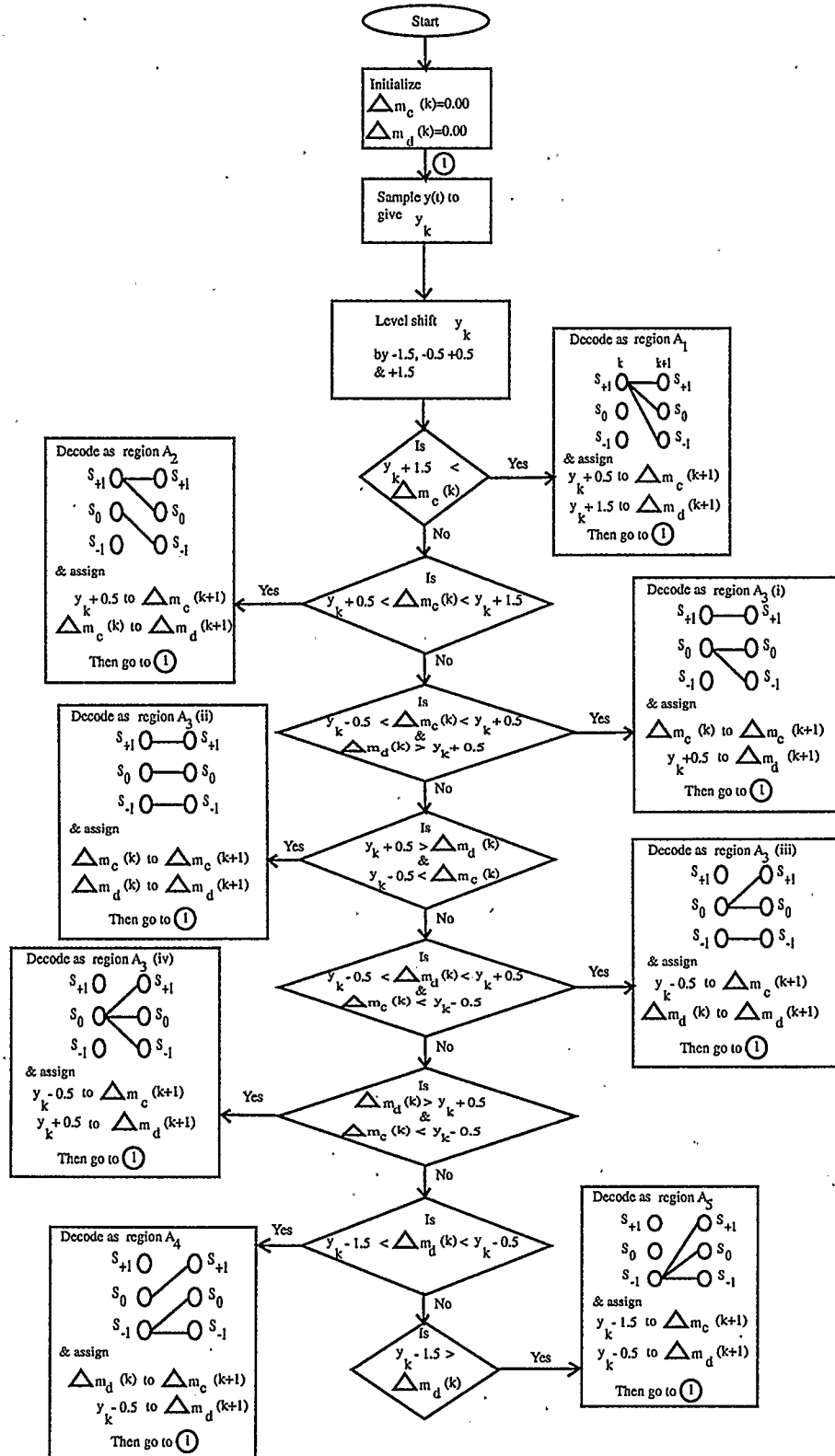


Figure D.3: Updated Flow-chart for the decode channel detection

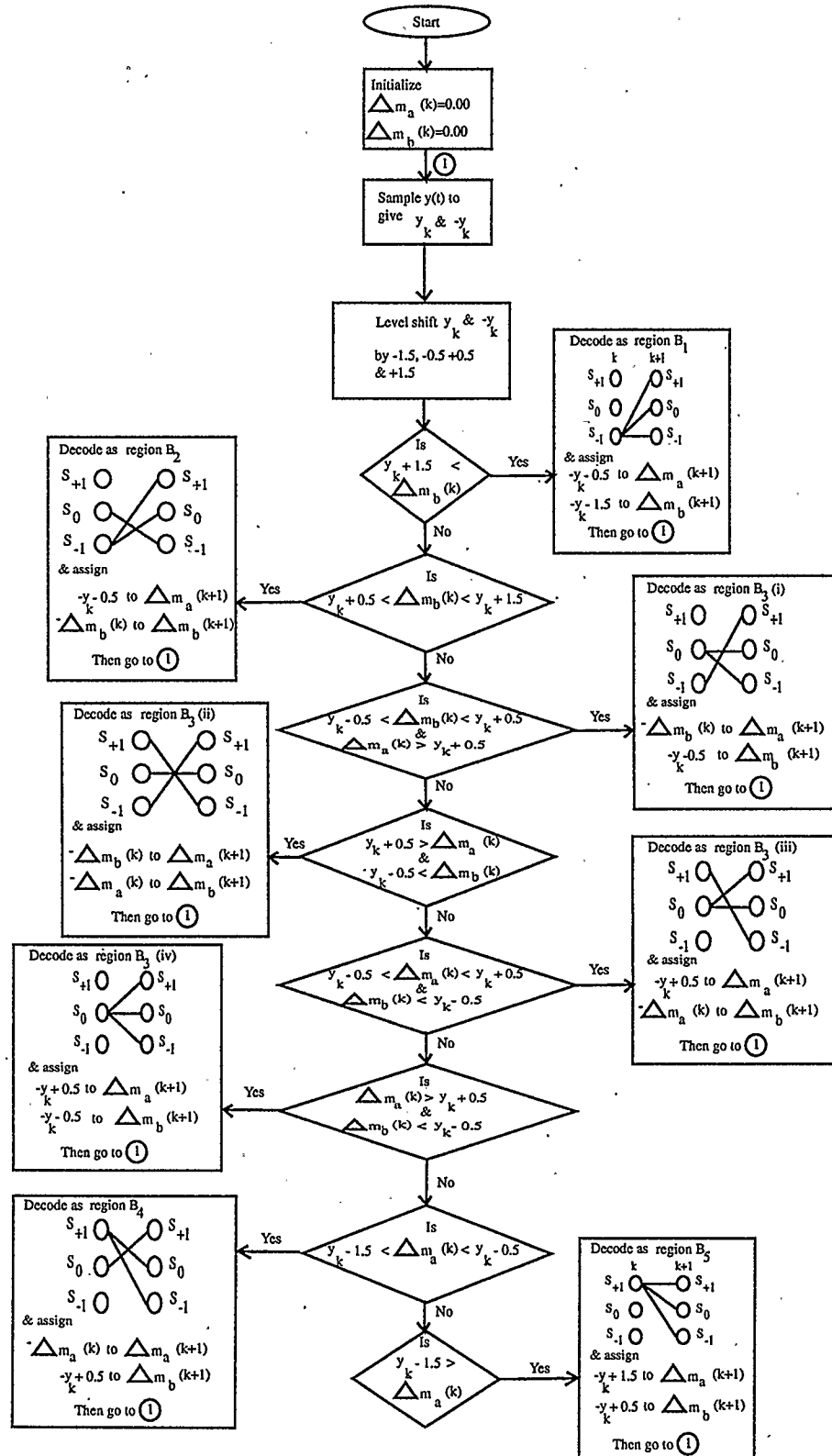


Figure D.4: Updated Flow-chart for the duobinary channel detection

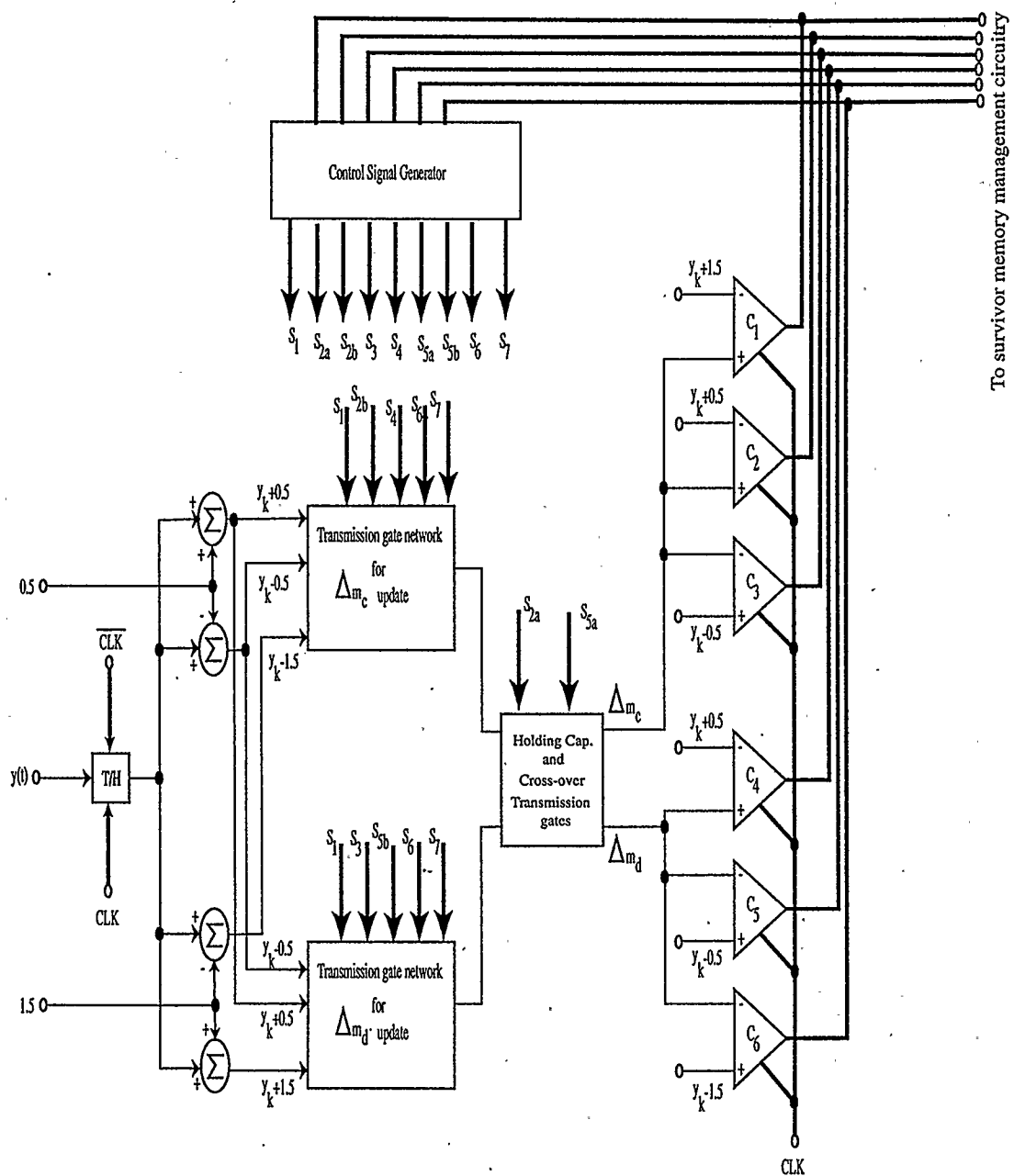


Figure D.5: Updated ternary dicode detector architecture

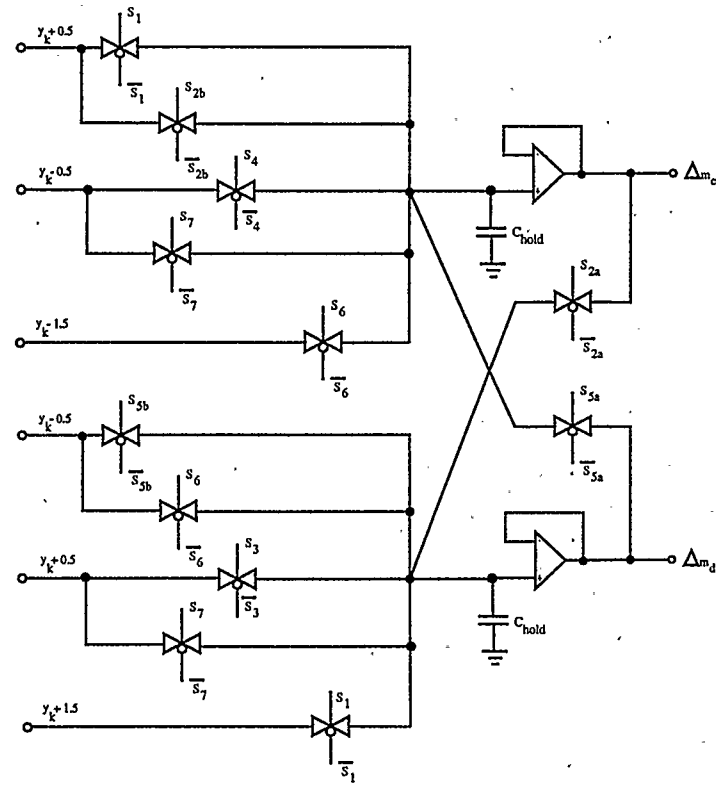


Figure D.6: Updated cross-over multiplexing T/H

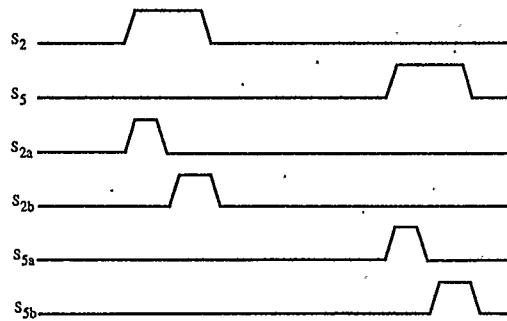


Figure D.7: Typical timing diagram for feed-forward and cross-over transmission gates

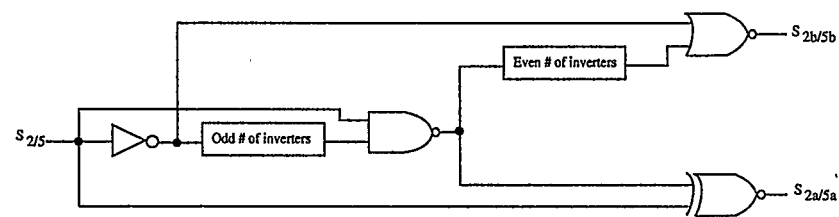


Figure D.8: Logic diagram for feed-forward and cross-over control signals