

THE UNIVERSITY OF CALGARY

HARDWARE IMPLEMENTATION OF THE

DIGITAL RAKE TRANSCEIVER

by

Edward Patton

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER

ENGINEERING

CALGARY, ALBERTA

AUGUST, 1993

© Edward Patton 1993



National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-88599-8

Canada

Name

Edward Patton

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

Engineering Electronics and Electrical

0544

U·M·I

SUBJECT TERM

SUBJECT CODE

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture 0729
 Art History 0377
 Cinema 0900
 Dance 0378
 Fine Arts 0357
 Information Science 0723
 Journalism 0391
 Library Science 0399
 Mass Communications 0708
 Music 0413
 Speech Communication 0459
 Theater 0465

EDUCATION

General 0515
 Administration 0514
 Adult and Continuing 0516
 Agricultural 0517
 Art 0273
 Bilingual and Multicultural 0282
 Business 0688
 Community College 0275
 Curriculum and Instruction 0727
 Early Childhood 0518
 Elementary 0524
 Finance 0277
 Guidance and Counseling 0519
 Health 0680
 Higher 0745
 History of 0520
 Home Economics 0278
 Industrial 0521
 Language and Literature 0279
 Mathematics 0280
 Music 0522
 Philosophy of 0998
 Physical 0523

Psychology 0525
 Reading 0535
 Religious 0527
 Sciences 0714
 Secondary 0533
 Social Sciences 0534
 Sociology of 0340
 Special 0529
 Teacher Training 0530
 Technology 0710
 Tests and Measurements 0288
 Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language 0679
 General 0289
 Ancient 0290
 Linguistics 0291
 Modern 0291
 Literature 0401
 General 0294
 Classical 0295
 Comparative 0297
 Medieval 0298
 Modern 0316
 African 0591
 American 0305
 Asian 0352
 Canadian (English) 0355
 Canadian (French) 0593
 English 0311
 Germanic 0312
 Latin American 0315
 Middle Eastern 0313
 Romance 0314
 Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy 0422
 Religion 0318
 General 0321
 Biblical Studies 0319
 Clergy 0320
 History of 0322
 Philosophy of 0469
 Theology 0323

SOCIAL SCIENCES

American Studies 0324
 Anthropology 0324
 Archaeology 0326
 Cultural 0327
 Physical 0310
 Business Administration 0272
 General 0770
 Accounting 0454
 Banking 0338
 Management 0385
 Marketing 0501
 Canadian Studies 0503
 Economics 0505
 General 0508
 Agricultural 0509
 Commerce-Business 0510
 Finance 0511
 History 0358
 Labor 0366
 Theory 0351
 Folklore 0578
 Geography 0578
 Gerontology 0578
 History 0578
 General 0578

Ancient 0579
 Medieval 0581
 Modern 0582
 Black 0328
 African 0331
 Asia, Australia and Oceania 0332
 Canadian 0334
 European 0335
 Latin American 0336
 Middle Eastern 0333
 United States 0337
 History of Science 0585
 Law 0398
 Political Science 0615
 General 0616
 International Law and 0617
 Relations 0814
 Public Administration 0452
 Recreation 0626
 Social Work 0627
 Sociology 0938
 General 0631
 Criminology and Penology 0628
 Demography 0629
 Ethnic and Racial Studies 0630
 Individual and Family 0700
 Studies 0344
 Industrial and Labor 0709
 Relations 0999
 Public and Social Welfare 0453
 Social Structure and 0537
 Development 0537
 Theory and Methods 0537
 Transportation 0537
 Urban and Regional Planning 0537
 Women's Studies 0537

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture 0473
 General 0285
 Agronomy 0475
 Animal Culture and 0476
 Nutrition 0359
 Animal Pathology 0478
 Food Science and 0479
 Technology 0480
 Forestry and Wildlife 0817
 Plant Culture 0777
 Plant Pathology 0746
 Plant Physiology 0306
 Range Management 0287
 Wood Technology 0308
 Biology 0309
 General 0379
 Anatomy 0329
 Biostatistics 0353
 Botany 0369
 Cell 0793
 Ecology 0410
 Entomology 0307
 Genetics 0317
 Limnology 0416
 Microbiology 0433
 Molecular 0821
 Neuroscience 0778
 Oceanography 0472
 Physiology 0786
 Radiation 0760
 Veterinary Science 0786
 Zoology 0760
 Biophysics 0425
 General 0996
 Medical 0996

EARTH SCIENCES

Biogeochemistry 0425
 Geochemistry 0996

Geodesy 0370
 Geology 0372
 Geophysics 0373
 Hydrology 0388
 Mineralogy 0411
 Paleobotany 0345
 Paleocology 0426
 Paleontology 0418
 Paleozoology 0985
 Palynology 0427
 Physical Geography 0368
 Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences 0768
 Health Sciences 0566
 General 0300
 Audiology 0992
 Chemotherapy 0567
 Dentistry 0350
 Education 0769
 Hospital Management 0758
 Human Development 0982
 Immunology 0564
 Medicine and Surgery 0347
 Mental Health 0569
 Nursing 0570
 Nutrition 0380
 Obstetrics and Gynecology 0354
 Occupational Health and 0381
 Therapy 0571
 Ophthalmology 0419
 Pathology 0572
 Pharmacology 0382
 Pharmacy 0573
 Physical Therapy 0574
 Public Health 0575
 Radiology 0575
 Recreation 0575

Speech Pathology 0460
 Toxicology 0383
 Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences

Chemistry 0485
 General 0749
 Agricultural 0486
 Analytical 0487
 Biochemistry 0488
 Inorganic 0738
 Nuclear 0490
 Organic 0491
 Pharmaceutical 0494
 Physical 0495
 Polymer 0754
 Radiation 0405
 Mathematics 0605
 Physics 0986
 General 0606
 Acoustics 0608
 Astronomy and 0748
 Astrophysics 0607
 Atmospheric Science 0798
 Atomic 0759
 Electronics and Electricity 0609
 Elementary Particles and 0610
 High Energy 0752
 Fluid and Plasma 0756
 Molecular 0611
 Nuclear 0463
 Optics 0346
 Radiation 0984
 Solid State 0984
 Statistics 0984
 Applied Sciences 0346
 Applied Mechanics 0984
 Computer Science 0984

Engineering 0537
 General 0538
 Aerospace 0539
 Agricultural 0540
 Automotive 0541
 Biomedical 0542
 Chemical 0543
 Civil 0544
 Electronics and Electrical 0348
 Heat and Thermodynamics 0545
 Hydraulic 0546
 Industrial 0547
 Marine 0794
 Materials Science 0548
 Mechanical 0743
 Metallurgy 0551
 Mining 0552
 Nuclear 0549
 Packaging 0765
 Petroleum 0554
 Sanitary and Municipal 0790
 System Science 0428
 Geotechnology 0796
 Operations Research 0795
 Plastics Technology 0994
 Textile Technology 0994

PSYCHOLOGY

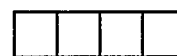
General 0621
 Behavioral 0384
 Clinical 0622
 Developmental 0620
 Experimental 0623
 Industrial 0624
 Personality 0625
 Physiological 0989
 Psychobiology 0349
 Psychometrics 0632
 Social 0451



Nom _____

Dissertation Abstracts International est organisé en catégories de sujets. Veuillez s.v.p. choisir le sujet qui décrit le mieux votre thèse et inscrivez le code numérique approprié dans l'espace réservé ci-dessous.

SUJET



CODE DE SUJET

U·M·I

Catégories par sujets

HUMANITÉS ET SCIENCES SOCIALES

COMMUNICATIONS ET LES ARTS

Architecture	0729
Beaux-arts	0357
Bibliothéconomie	0399
Cinéma	0900
Communication verbale	0459
Communications	0708
Danse	0378
Histoire de l'art	0377
Journalisme	0391
Musique	0413
Sciences de l'information	0723
Théâtre	0465

ÉDUCATION

Généralités	515
Administration	0514
Art	0273
Collèges communautaires	0275
Commerce	0688
Économie domestique	0278
Éducation permanente	0516
Éducation préscolaire	0518
Éducation sanitaire	0680
Enseignement agricole	0517
Enseignement bilingue et multiculturel	0282
Enseignement industriel	0521
Enseignement primaire	0524
Enseignement professionnel	0747
Enseignement religieux	0527
Enseignement secondaire	0533
Enseignement spécial	0529
Enseignement supérieur	0745
Évaluation	0288
Finances	0277
Formation des enseignants	0530
Histoire de l'éducation	0520
Langues et littérature	0279

Lecture	0535
Mathématiques	0280
Musique	0522
Orientation et consultation	0519
Philosophie de l'éducation	0998
Physique	0523
Programmes d'études et enseignement	0727
Psychologie	0525
Sciences	0714
Sciences sociales	0534
Sociologie de l'éducation	0340
Technologie	0710

LANGUE, LITTÉRATURE ET

LINGUISTIQUE

Langues

Généralités	0679
Anciennes	0289
Linguistique	0290
Modernes	0291

Littérature

Généralités	0401
Anciennes	0294
Comparée	0295
Médiévale	0297
Moderne	0298
Africaine	0316
Américaine	0591
Anglaise	0593
Asiatique	0305
Canadienne (Anglaise)	0352
Canadienne (Française)	0355
Germanique	0311
Latino-américaine	0312
Moyen-orientale	0315
Romane	0313
Slave et est-européenne	0314

PHILOSOPHIE, RELIGION ET

THÉOLOGIE

Philosophie	0422
Religion	
Généralités	0318
Clergé	0319
Études bibliques	0321
Histoire des religions	0320
Philosophie de la religion	0322
Théologie	0469

SCIENCES SOCIALES

Anthropologie	
Archéologie	0324
Culturelle	0326
Physique	0327
Droit	0398
Économie	
Généralités	0501
Commerce-Affaires	0505
Économie agricole	0503
Économie du travail	0510
Finances	0508
Histoire	0509
Théorie	0511
Études américaines	0323
Études canadiennes	0385
Études féministes	0453
Folklore	0358
Géographie	0366
Gérontologie	0351
Gestion des affaires	
Généralités	0310
Administration	0454
Banques	0770
Comptabilité	0272
Marketing	0338
Histoire	
Histoire générale	0578

Ancienne	0579
Médiévale	0581
Moderne	0582
Histoire des noirs	0328
Africaine	0331
Canadienne	0334
États-Unis	0337
Européenne	0335
Moyen-orientale	0333
Latino-américaine	0336
Asie, Australie et Océanie	0332
Histoire des sciences	0585
Loisirs	0814
Planification urbaine et régionale	0999
Science politique	
Généralités	0615
Administration publique	0617
Droit et relations internationales	0616
Sociologie	
Généralités	0626
Aide et bien-être social	0630
Criminologie et établissements pénitentiaires	0627
Démographie	0938
Études de l'individu et de la famille	0628
Études des relations interethniques et des relations raciales	0631
Structure et développement social	0700
Théorie et méthodes	0344
Travail et relations industrielles	0629
Transports	0709
Travail social	0452

SCIENCES ET INGÉNIERIE

SCIENCES BIOLOGIQUES

Agriculture	
Généralités	0473
Agronomie	0285
Alimentation et technologie alimentaire	0359
Culture	0479
Élevage et alimentation	0475
Exploitation des péturages	0777
Pathologie animale	0476
Pathologie végétale	0480
Physiologie végétale	0817
Sylviculture et faune	0478
Technologie du bois	0746
Biologie	
Généralités	0306
Anatomie	0287
Biologie (Statistiques)	0308
Biologie moléculaire	0307
Botanique	0309
Cellule	0379
Ecologie	0329
Entomologie	0353
Génétique	0369
Limnologie	0793
Microbiologie	0410
Neurologie	0317
Océanographie	0416
Physiologie	0433
Radiation	0821
Science vétérinaire	0778
Zoologie	0472
Biophysique	
Généralités	0786
Médicale	0760

Géologie	0372
Géophysique	0373
Hydrologie	0388
Minéralogie	0411
Océanographie physique	0415
Paléobotanique	0345
Paléocologie	0426
Paléontologie	0418
Paléozoologie	0985
Palynologie	0427

SCIENCES DE LA SANTÉ ET DE L'ENVIRONNEMENT

Économie domestique	0386
Sciences de l'environnement	0768
Sciences de la santé	
Généralités	0566
Administration des hôpitaux	0769
Alimentation et nutrition	0570
Audiologie	0300
Chimiothérapie	0992
Dentisterie	0567
Développement humain	0758
Enseignement	0350
Immunologie	0982
Loisirs	0575
Médecine du travail et thérapie	0354
Médecine et chirurgie	0564
Obstétrique et gynécologie	0380
Ophtalmologie	0381
Orthophonie	0460
Pathologie	0571
Pharmacie	0572
Pharmacologie	0419
Physiothérapie	0382
Radiologie	0574
Santé mentale	0347
Santé publique	0573
Soins infirmiers	0569
Toxicologie	0383

SCIENCES DE LA TERRE

Biogéochimie	0425
Géochimie	0996
Géodésie	0370
Géographie physique	0368

SCIENCES PHYSIQUES

Sciences Pures

Chimie	
Généralités	0485
Biochimie	487
Chimie agricole	0749
Chimie analytique	0486
Chimie minérale	0488
Chimie nucléaire	0738
Chimie organique	0490
Chimie pharmaceutique	0491
Physique	0494
Polymères	0495
Radiation	0754
Mathématiques	0405
Physique	
Généralités	0605
Acoustique	0986
Astronomie et astrophysique	0606
Électronique et électricité	0607
Fluides et plasma	0759
Météorologie	0608
Optique	0752
Particules (Physique nucléaire)	0798
Physique atomique	0748
Physique de l'état solide	0611
Physique moléculaire	0609
Physique nucléaire	0610
Radiation	0756
Statistiques	0463

Sciences Appliquées Et Technologie

Informatique	0984
Ingénierie	
Généralités	0537
Agricole	0539
Automobile	0540

Biomédicale	0541
Chaleur et ther modynamique	0348
Conditionnement (Emballage)	0549
Génie aérospatial	0538
Génie chimique	0542
Génie civil	0543
Génie électronique et électrique	0544
Génie industriel	0546
Génie mécanique	0548
Génie nucléaire	0552
Ingénierie des systèmes	0790
Mécanique navale	0547
Métallurgie	0743
Science des matériaux	0794
Technique du pétrole	0765
Technique minière	0551
Techniques sanitaires et municipales	0554
Technologie hydraulique	0545
Mécanique appliquée	0346
Géotechnologie	0428
Matières plastiques (Technologie)	0795
Recherche opérationnelle	0796
Textiles et tissus (Technologie)	0794

PSYCHOLOGIE

Généralités	0621
Personnalité	0625
Psychobiologie	0349
Psychologie clinique	0622
Psychologie du comportement	0384
Psychologie du développement	0620
Psychologie expérimentale	0623
Psychologie industrielle	0624
Psychologie physiologique	0989
Psychologie sociale	0451
Psychométrie	0632



THE UNIVERSITY OF CALGARY
FACULTY OF GRADUATE STUDIES

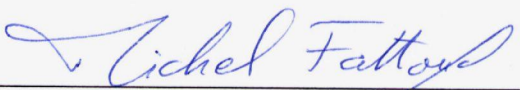
The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "Hardware Implementation of the Digital Rake Transceiver" submitted by Edward Patton in partial fulfillment of the requirements for the degree of Master of Science.



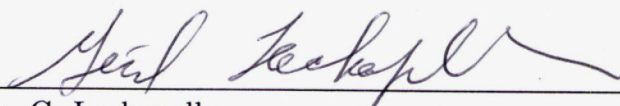
Supervisor, Dr. S. T. Nichols
Dept. of Electrical and Computer Engineering



Dr. L. E. Turner
Dept. of Electrical and Computer Engineering



Dr. M. Fattouche
Dept. of Electrical and Computer Engineering



Dr. G. Lachapelle
Geomatics Engineering Dept., Calgary

Date: Sept 18/93

ABSTRACT

Present digital cellular radio systems transmit data over 30 kHz multipath channels which can undergo deep signal fades (up to 50 dB). Unless some form of antenna diversity is implemented, the signal will be "wiped out". To combat deep fades, direct-sequence spread-spectrum transceivers are employed which can be used to resolve multipath signals for co-phasing and combining in the digital "Rake" (DRake).

The entirely digital DRake transceiver described in this dissertation spreads the data signal from 40.3 kHz to 1.25 MHz for ISM band transmission. It was implemented on field programmable gate-arrays using bit-serial, digit-serial, and parallel bit-level architectures to optimize clock speed and gate usage. It also employs a unique bandpass sampling / quadrature down-conversion (QDC) scheme which reduces the number of analog components compared to analog QDC.

The transceiver was tested using a real additive white Gaussian noise channel. A computer model determined performance over a time-varient multipath environment.

ACKNOWLEDGEMENTS

The author thanks Dr. S.T. Nichols for his enduring patience and guidance during the course of this work, to Dr. L.E. Turner for the use of the Xilinx hardware and SPARC computer, to Mr. P.J. Graumman for his unending suggestions and company during late evenings at the computer terminal, and to E.B. Olasz for her modified Dr. Hashemi channel model. The author also wishes to acknowledge the support and efforts of AMC, NSERC, and the secretarial staff at the University of Calgary.

DEDICATIONS

To Mom
for bringing me late night dinners,
to Fiona
for her ever loving support,
to Lori
for her understanding and patience, and
to My Friends
for waiting.

MEMORIES

"Its always a factor of two out."

Dr. S.T. Nichols

"It looks like that chips toasted!"

Peter Graumman

"Hey Peter, the done signal went high!"

Me

CONTENTS

APPROVAL PAGE	ii
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
DEDICATIONS	v
MEMORIES	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	xii
LIST OF FIGURES	xiii
1. INTRODUCTION	1
1.1 Overview	1
1.2 Scope of Thesis	5
2. OVERVIEW OF SPREAD-SPECTRUM AND RAKE DESIGNS	8
2.1 The Multipath Environment	8
2.1.1 Summary	13
2.2 Historical Background of Spread-Spectrum Systems	14
2.3 Spread-Spectrum systems	16
2.3.1 The basic Direct-Sequence Spread-Spectrum Transmitter	17
2.3.2 The basic Direct-Sequence Spread-Spectrum Receiver	20
2.3.3 Summary	23
2.4 The Rake Receiver	23
2.5 Introduction	23

2.5.1	Transversal filter realization of the Rake	25
2.5.2	Recent Developments of Rake Receivers	27
2.5.2.1	Kavehrad and Bodeep Direct-Sequence Spread-Spectrum Receiver	28
2.5.2.2	Grob et. al. N-Path Rake Receiver	30
2.5.2.3	Kaufmann et. al. Spread-Spectrum Multipath-Diversity Receiver	34
2.5.3	Summary	37
 3. DIGITAL IMPLEMENTATION OF A 1/4 WAVE SAMPLING RECEIVER		39
3.1	Introduction	39
3.2	The Quadrature Modulator	40
3.3	Down Conversion	41
3.3.1	Conventional Analog Quadrature Demodulator	41
3.3.2	Complex Demodulation	42
3.4	Bandpass Sampling	43
3.5	Quarter Wave Sampling	46
3.5.1	Halfband Filtering and Decimation	46
3.5.2	Halfband Filters	50
3.5.2.1	Complex and Real Bandpass Filters	50
3.5.2.2	Development of an Equivalent Baseband Filter System	53
3.6	Summary	57
 4. PROCESSING BLOCK DESCRIPTION OF THE INTEGRATED DIRECT-SEQUENCE SPREAD-SPECTRUM RAKE RECEIVER		59
4.1	Introduction	59
4.2	Constraints	59

4.3	The Direct-Sequence Spread-Spectrum Transmitter	61
4.3.1	Pseudo-Noise Code Generator	63
4.3.2	Transmit Filters	65
4.3.2.1	Reconstruction Filters, D/A's, and Interpolating Filters	65
4.3.2.2	Square-Root Nyquist Interpolating Filter	68
4.3.2.3	Predistortion	71
4.4	The Direct-Sequence Spread-Spectrum Receiver	74
4.4.1	Limiter	75
4.4.2	Square-Root Nyquist Matched Filter	76
4.4.3	M-sequence Matched Filter	77
4.4.4	DPSK demodulation	79
4.4.5	Diversity Combining with the Rake	80
4.4.6	Bit-Clock Recovery	82
4.5	Simulation	85
4.6	Summary	86
5.	CONSIDERATIONS FOR VLSI IMPLEMENTATION	88
5.1	Introduction	88
5.1.1	Binary Number representation	88
5.2	Non-Ideal Effects	90
5.2.1	Analog-to-Digital Converter	90
5.2.1.1	Quantization Noise	92
5.2.2	Coefficient Quantization	93
5.2.3	Overflow	94
5.2.4	Fixed-Point Arithmetic Errors	95
5.2.4.1	Addition and Subtraction	95
5.2.4.2	Multiplication	96
5.2.5	Modeling Quantization at the Output of a Filter	97

5.2.5.1	Quantization Noise at the Filter Input	97
5.2.5.2	Quantization Noise due to Multipliers	98
5.3	Hardware Considerations	98
5.3.1	Hardware Architectures	100
5.3.1.1	Parallel	101
5.3.1.2	Bit-Serial	103
5.3.1.3	Digit-Serial	106
5.3.1.4	A Comparison Between Parallel, Bit-Serial, and Digit Serial	106
5.3.2	Design Rules	108
5.4	Hardware Design of the Direct-Sequence Spread-Spectrum Receiver .	108
5.4.1	Hardware Development Tools	109
5.4.2	Xilinx Hardware Overview	111
5.4.3	Overall Chip Layout	112
5.4.4	Transmitter	114
5.4.4.1	M-Sequence Generator	114
5.4.4.2	Interpolating Filter	115
5.4.5	Receiver	118
5.4.5.1	Half-Band Filter and Decimate Chains	119
5.4.5.2	Square-Root Nyquist Matched Filter	120
5.4.5.3	M-Sequence Matched Filter	124
5.4.5.4	DPSK Demodulation	127
5.4.5.5	Rake	129
5.4.5.6	Bit-Clock Recovery	131
5.4.6	Chip Communication and Control	133
5.5	Summary	134

6. DRAKE TRANSCEIVER PERFORMANCE	137
6.1 Introduction	137
6.2 Theoretical, Simulated, and Hardware Waveforms	138
6.2.1 Nyquist-Pulse Matched Filter Output	138
6.2.2 DPSK, DRake, and PLL Output	140
6.2.3 DPSK Oscillations	143
6.2.3.1 Gain Offsets and Truncation Error Biasing	144
6.3 Theoretical and Hardware Probability of Bit-Error Curves	145
6.3.1 Theoretical DPSK	146
6.3.2 Signal-to-Noise Calibration	146
6.3.3 Experiment Setup	147
6.3.4 Additive White Gaussian Noise Channel - Sampling the DSPK Output	150
6.3.4.1 A Comparison Between Theory and Simulation	150
6.3.4.2 Non-ideal Effects - Simulation and Hardware	152
6.3.5 Additive White Gaussian Noise Channel - Sampling the DRake Output	155
6.3.5.1 DRake Output: Synchronous Sampling	155
6.3.5.2 DRake Output: Phase Locked-Loop Sampling	157
6.3.6 Multipath Channel Simulation	158
6.4 Summary	160
7. CONCLUSIONS	163
7.1 Introduction	163
7.2 Hardware Configuration and System Performance	163
7.3 Recommendations for Future Work	167
REFERENCES	169

LIST OF TABLES

4.1	Filter tap values for 1) ideal bandwidth, and 2) predistorted bandwidth.	70
4.2	Filter tap values for square-root Nyquist matched filter	76
5.1	Fixed-point addition/subtraction error statistics.	96
5.2	Fixed-point multiplication error statistics.	97
5.3	A comparison of adder structures	107
5.4	Delays for NAND gate (1.5μ drawn process).	123

LIST OF FIGURES

2.1	Illustration of a three path multipath channel	9
2.2	Hashemi time varying frequency profiles for a vehicle traveling at 100 km/hr	12
2.3	Basic DS-SS Transmitter	18
2.4	Basic DS-SS receiver	21
2.5	Autocorrelation output	22
2.6	Transversal Realization of the Rake	26
2.7	Transversal and SAW matched filters (MF): a) baseband binary signal input to transversal MF, b) transversal MF, c) transversal MF output, d) passband signal input to SAW MF, e) SAW MF, and f) SAW MF output.	29
2.8	Kavehrad and Bodeep Direct-Sequence Spread-Spectrum Differential Phase- Shift Keying Transceiver	31
2.9	Grob et. al. N-Path Rake Receiver	32
2.10	Kaufman et. al. Multipath-Diversity Receiver: a) hardware layout and b) time integrating correlator (TIC)	36
3.1	Conventional analog quadrature modulator.	40
3.2	Conventional analog quadrature demodulator.	42
3.3	Analog complex demodulator	43

3.4	Practical analog complex demodulator	44
3.5	Quadrature down conversion with bandpass sampling.	44
3.6	Complex demodulation with bandpass sampling.	45
3.7	Spectral illustration of complex demodulation: a) bandpass signal, b) sampling at $f_s = f_c$, and c) multiplying by $e^{j\omega_c t}$	47
3.8	Quadrature down-conversion using filter and decimation chains.	48
3.9	Spectral illustration of downconvert and decimate-by-4: a) bandpass sampled signal and complex filter spectrum; b) decimate-by-2 spectrum and real half-band filter; c) spectrum decimated to baseband	49
3.10	Halfband filtering to prevent aliasing: a) signal bandwidth small relative to sampling rate and b) signal bandwidth large relative to sampling rate	50
3.11	Digital structures for complex and real filters, and decimators	53
3.12	Decimation blocks for; a) complex and real filters, b) single complex filter, and c) complex demodulator and real filters	54
3.13	Baseband frequency response: a) signal spectrum and filter response after complex demodulation; b) signal spectrum after filtering and decimation	56
4.1	A binary differential phase-shift keying spread-spectrum transmitter. . .	62
4.2	Two linear feedback shift registers: a) low speed and b) high speed. . .	64
4.3	Ideal signal reconstruction.	65
4.4	Interpolating structure for increasing the sampling rate by N_T	67

4.5	Raised cosine pulse for rolloff = 0.0, 0.34, and 0.75.	69
4.6	Raised cosine frequency response for rolloff = 0.0, 0.34, and 0.75.	69
4.7	Distortion of square-root Nyquist frequency spectrum by the baseband equivalent of filter and decimate-by-four chain.	72
4.8	Eye diagram for interpolating filter bandwidth = 1/16.	73
4.9	Eye diagram for interpolating filter bandwidth = 0.070161.	73
4.10	Integrated Direct-Sequence Spread-Spectrum Rake Receiver.	74
4.11	Output from FIR M-sequence matched filter: 1) best case sampling, and 2) worst case sampling	77
4.12	M-sequence matched filter sample values	78
4.13	Phasor addition of the multipath autocorrelation values	79
4.14	Digital implementation of the DRake: a) DPSK output, b) DRake, and c) Drake output.	81
4.15	Phase-locked loops (PLL): 1) analog PLL, and 2) first-order digital PLL.	82
4.16	Type 4 Phase detector: a) digital structure, b) phase error, and c) fre- quency error.	84
5.1	Digitization of analog signals: a) real system, and b) theoretical system	90
5.2	Characteristics of serial/parallel architectures.	100
5.3	Four bit parallel adder.	102
5.4	Bit-serial adder.	104

5.5	Bit-Serial and Parallel Architecture Design Tools.	110
5.6	Allocation of Xilinx FPGAs to functional cells	113
5.7	Low level logic for M-sequence programmable insert.	115
5.8	Interpolation - by - 8 digital structure design	117
5.9	Halfband filter and decimate-by-two digital filter structure.	120
5.10	Halfband filter and decimate-by-two filter timing diagram.	121
5.11	Halfband filter and decimate-by-two filter timing diagram.	122
5.12	M-sequence matched filter with non-ideal quantization noise inputs . . .	125
5.13	DPSK hardware layout	128
5.14	DPSK timing diagram.	129
5.15	Rake threshold detector.	130
5.16	First-order bit-clock recovery phase locked-loop.	131
5.17	State representation of the I/D counter.	132
5.18	Control unit timing diagram.	136
6.1	Raised cosine frequency response for rolloff = 0.35.	139
6.2	Nyquist-pulse time waveform.	140
6.3	DPSK output: zero noise	140
6.4	Transmitted Nyquist pulse.	141

6.5	Received Nyquist pulse.	142
6.6	DRake output: threshold set at .016	142
6.7	Phase lock-loop output	143
6.8	Recovered data bits from 0/S	143
6.9	Non-ideal effects on DPSK output.	144
6.10	Experimental setup for AWGN experiment.	148
6.11	Probability of bit error for theoretical DPSK and simulated spread-spectrum receiver - 32-bit fixed-point arithmetic.	151
6.12	Non-ideal effects on bit-error rate performance.	152
6.13	Hardware and simulated BER curves for synchronous sampling of the DPSK output.	153
6.14	Effects of frequency offset of BER performance	154
6.15	Hardware BER curve for synchronous sampling of the DRake output. . .	155
6.16	Hardware BER curve for PLL sampling of the DRake output.	157
6.17	Multipath BER curve for synchronous sampling of the DRake output. . .	158
6.18	DPSK output illustration of self noise for a noiseless, single path channel.	160
6.19	DPSK output illustration of self-noise for noiseless, multipath channel. .	161

CHAPTER 1

INTRODUCTION

1.1 Overview

When cellular radio first made its appearance in the communication world, the consumer found it too expensive, not as mobile as they would like, and unreliable. As technology advanced, these became minor concerns, resulting in more consumers subscribing to cellular radio. As the number of subscribers increased, the cellular frequency band became overly crowded. Because of this, industry developed and promoted digital transceivers, instead of analog transceivers, as a way of increasing the number of subscribers in the limited bandwidth available. A channel sharing scheme was developed that permitted several digital cellular users to share a single 30 kHz band. This scheme is known as time-division multiple access (TDMA).

Unfortunately, the transmission of digital information over the 30 kHz band is prone to deep fades that cause significant bit errors. For voice, a bit error rate of 10^{-3} is sufficient. But subscribers want to use portable computers to communicate over the cellular bands. As a result, techniques have to be employed to reduce the number of bit errors that occur during channel fades.

Cellular channels undergo deep fades that can attenuate the regular 30 kHz cellular band by as much as 50 dB. Spread-spectrum systems are, however, viewed as a way of combating these deep fades while transmitting on adverse indoor and outdoor cellular channels at lower bit-error rates than the 30 kHz digital cellular radios. By spreading the information signal's bandwidth over a large bandwidth, only a small portion of its spectral components undergo a deep fade. The majority of the spectra remain intact to be demodulated at the receiver.

In DS-SS systems, the data signal's bandwidth is "spread" at the transmitter by modulating it with a periodic pseudo-random noise (PN) sequence. The PN sequence consists of a sequence of binary levels referred to as "chips", and has a spectrum similar to white noise, ie. it is spectrally flat over most of its signalling bandwidth. Since the PN sequence's chip's period is typically much smaller than the data bit period, the resulting modulated spectrum is much wider than the data spectrum.

An important characteristic of the PN sequences is that there is a finite set of sequences, or *preferred sequences*, that exist and can be transmitted in the same bandwidth. The individual set members can be recovered at the receiver without severe degradation in performance. There are several different families of PN sequences such as M-sequences, Kasami, or Gold sequences, that exhibit this characteristic but to varying degrees [46]. The M-sequences (or Maximal length sequences) are used in this thesis because they are easy to generate and possess good cross-correlation properties, ie. the correlation of one M-sequence set with another results in minimal interference. The other PN sequence sets have more *preferred sequences*, but show poorer cross-correlations. The advantage in using them is the increased number of users which can share the same bandwidth.

The signal at the receiver is demodulated by multiplying it by the same PN sequence used at the transmitter. If the two sequences are properly aligned, the signal de-correlates or "despreads"; which means that the spreaded signal's bandwidth is collapsed back into the same bandwidth of the original data signal thereby recovering the data. Difficulties arise in obtaining the proper alignment necessary to despread the signal. This thesis presents a despreading method that instantaneously aligns the sequences.

If the multipath environment causes several delayed versions of the signal to ar-

rive at the receiver, it is possible to implement a bank of "despreaders" to combine the signal from each path, thereby implementing a *time diversity* scheme and hence improving data recovery. This does imply, however, that there is sufficient signal bandwidth available to resolve the multipath signal before combining. The first hardware device to take advantage of the inherent time diversity of the channel was the F9C Rake implemented by Green [41] in 1958. The device was large and stood almost at ceiling height. Recent designs [30] [54] [44] are more compact due to surface acoustic wave (SAW) and very large scale integrated (VLSI) circuit technologies. In these receivers there exists designs consisting of analog, digital, and a hybrid of analog and digital components. The most optimum blend of components will make the design compact and easy to manufacture in order for it to be competitive in the global marketplace.

The earliest design using one of the above mentioned technologies is M. Kavehrad and G. Bodeep's SAW matched filter (MF) SS transceiver (1987) [30]. It did not, however, take advantage of the inherent time diversity of the channel, but it was immune to deep fades. The first hardware design that took advantage of time diversity since 1958 was the Rake transceiver designed by Grob et. al. [54]. It used a bank of analog despreaders and a digital signal processor for demodulation. Kaufmann [44] took the concept Grob used and implemented the despreading correlators on a single digital integrated chip (IC). Because of his digital approach, the design was more compact.

Kaufmann was able to implement despreaders in silicon because he implemented despreading at baseband. Attempting to despread at bandpass would result in high data rates that are difficult to handle at present CMOS speeds, and result in high power consumption.

The matched filter (MF) approach taken by Kavehrad has the advantage of instantaneously calculating the correlation of the received echoes without the need for an acquisition and tracking stage. The MF is unlimited in the number of echoes that it can process, however, its size is proportional to the number of chips used in the PN-code.

In the other two methods, the number of correlators required is directly proportional to the number of echoes received and inherently require a complex algorithm to scan for echo locations before locking onto them. Scanning results in an acquisition time that could possibly be longer than the coherence time of the channel. But a good characteristic of this method is the minimal amount of hardware needed to increase the PN-code length.

This dissertation describes the implementation of an all-digital transceiver spread-spectrum (SS) communication system operating in an outdoor cellular multipath environment. It incorporates:

- a digital MF to achieve instantaneous acquisition; therefore making it a good candidate for TDMA,
- a digital Rake for time diversity, and
- and bandpass sampling with digital quadrature down conversion; thereby reducing the number of analog components.

Being that its all-digital, it can be easily reprogrammed and reconfigured to suit different applications and new algorithms. It will be cost effective since it requires less tuning, and takes advantage of the high linearity of digital components. Since it can be implemented on a single integrated chip, it will also be compact.

An approach proposed by Lodge [26] is used to implement a quadrature down-conversion receiver with fewer analog components than the standard analog receiver. The technique involves bandpass signal sampling at a center frequency equal to one quarter the sampling frequency - thus termed the *1/4 wave sampling receiver*. This results in a simple digital implementation of the transceiver that eliminates the need for dual multiplexers, lowpass filters, and analog-to-digital converters (A/D). Also not needed is the manual tuning required to ensure that there are no imbalances in the quadrature down-conversion section.

Using the above technique, the entire transceiver downstream of the bandpass sampling was implemented on field programmable gate arrays (FPGA)s. The bit-level design used bit-serial, parallel, and a hybrid of both architectures to attain maximum gate usage densities on the FPGAs. Another consideration to implementing a particular architecture was the clock speed required to process the data. The final hardware design was tested on an additive white-gaussian noise channel. Simulations were used to determine performance in a multipath environment.

Where applicable, the transceiver prototype uses bit-serial, digit-serial, and parallel architectures. The system design utilizes high level CAD design tools presently being developed at the University of Calgary.

1.2 Scope of Thesis

In Chapter 2, the multipath environment is presented along with a mathematical model. Next, the historical developments of the SS systems is presented to understand the motivation behind SS research. Following this, the basic concepts of a DS-SS system are presented in a tutorial which covers the signal description of a simple DS-SS transceiver. A critique of existing hardware DS-SS transceivers is also presented.

In Chapter 3, the theoretical considerations in designing the *1/4 wave sampling receiver* are presented. The *1/4 wave sampling receiver* derivation follows the derivation of the equivalent analog system. A baseband equivalent system is also derived for computer simulation.

Chapter 4 presents the processing block description of the transceiver. It covers the theoretical considerations of designing the transceiver processing blocks without getting into the bit-level architecture design. Reconstruction filters, square-root Nyquist filters, and M-sequence matched filters are presented. Differential phase-shift keying (DPSK) is presented in conjunction with the Rake to assist in removing the channel phase from each multipath signal so that coherent combining of multipath signals is possible.

Bit-serial, digit-serial, and parallel digital architectures are presented in Chapter 5. The decision to implement a specific architecture is based on its physical characteristics such as processing speed, gate usage, and routing requirements. Non-ideal effects, such as quantization noise and overflow, are presented and also their relevance to design implementation. The Chapter also discusses the hardware implementation on Xilinx FPGAs.

The DRake transceiver performance is presented in Chapter 6. The Chapter first presents the output waveforms and frequency spectrum of the transmitted DS-SS signal. Non-ideal oscillations observed in the DPSK output are discussed. The bit-error rates (BER) are presented with a description of the experimental setup used to synchronously and non-synchronously sample the DSPK and DRake output. BERs are also shown in comparison to those generated using the computer model. All BERs are presented in reference to theoretical Binary DPSK.

Finally, Chapter 7 presents the conclusions that were made in observation of the

BER results and the experience gained in the hardware implementation of the design using several different bit-level design architectures.

CHAPTER 2

OVERVIEW OF SPREAD-SPECTRUM AND RAKE DESIGNS

This chapter presents the outdoor multipath problem, a tutorial of direct-sequence spread-spectrum (DS-SS) systems that combat this problem, and a review of three recent DS-SS Rake receivers that have been recently developed.

First, an introduction to the multipath environment is presented with a look at outdoor cellular time varying frequency profiles to substantiate the adverse effects of this environment. Second, a short historical background is presented to understand the motives that inspired SS research. Third, a basic SS transmitter and receiver pair is used as a tutorial to give the basic theory behind DS-SS systems and the realization of time diversity with the Rake. Their ability to transmit and receive over a line of sight channel is discussed and then expanded to include a multipath channel. Finally, a critique of three recent Rake systems is given: pros and cons relating to the hardware realizations are discussed.

2.1 The Multipath Environment

The 900 MHz channel for outdoor cellular radio communications is a *multipath channel*; a channel where the transmitted signal propagates over several paths to the receiving antenna. This phenomena, coupled with the digital trend to increase the number of channel users, results in *selective fading* and *intersymbol interference* (ISI). These undesirable conditions limit the rate at which digital information can be transmitted.

In a multipath channel, the signals arriving at the antenna are echoed versions of the transmitted signal; each echo having a different distance to travel and therefore

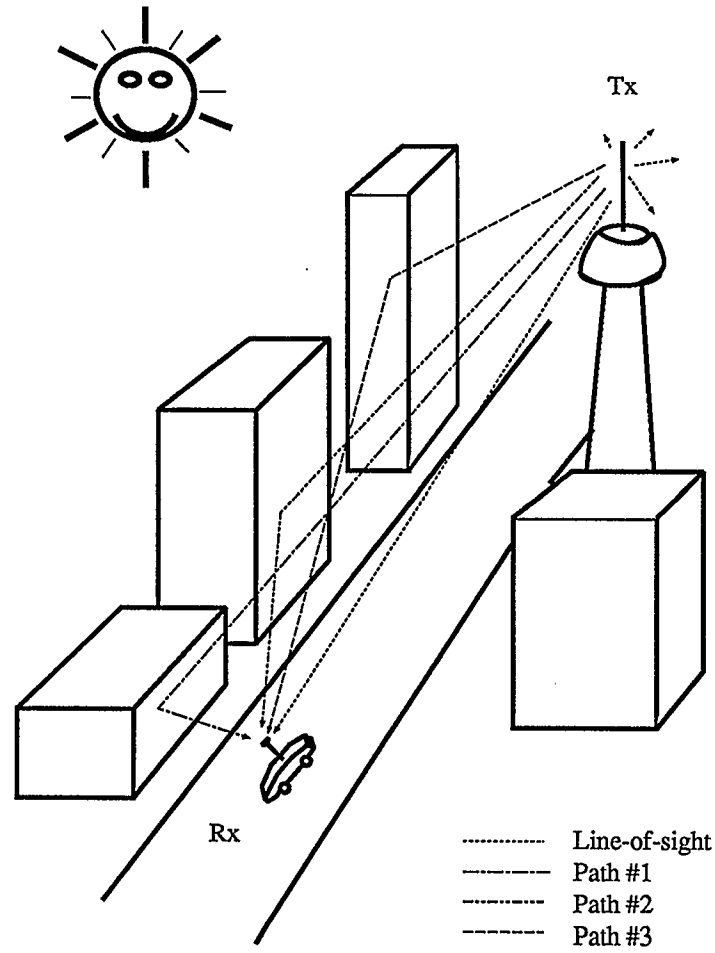


Figure 2.1 Illustration of a three path multipath channel

different time delay and attenuation. This is illustrated by a Fig. 2.1 where the automobile's antenna receives three echoed signals from the transmitting antenna. A line-of-sight path may or may not be present.

Consider the narrowband transmitted signal

$$s(t) = \text{Re}[\sigma(t)e^{j\omega_c t}], \quad (2.1)$$

where $\sigma(t)$ is the complex envelope of the transmitted signal and ω_c is the carrier frequency in radians/second. For N paths , the signal arriving at the receiver antenna

will be

$$r(t) = \text{Re}[\rho(t)e^{j\omega_c t}] + n(t), \quad (2.2)$$

where

$$\rho(t) = \sum_{n=0}^{N-1} \alpha_n(t) \sigma(t - \tau_n(t)) e^{-j\theta_n(t)}. \quad (2.3)$$

The parameters α_n , τ_n , and θ_n represent the n th path (or echo) attenuation, modulation delay, and carrier phase shift ($\theta_n(t) = \omega_c \tau_n(t)$), respectively. The waveform, $n(t)$, is the additive bandpass noise component. Details are given in [53].

At any one time, the total received signal, $r(t)$, is a vector sum of individually delayed signals, their relative phase angles depending on the frequency and the echo amplitudes and delays [41]. The amplitudes and phases are also varying in time. In the outdoor radio channel, this non-stationary phenomena is caused by the movement of the mobile unit.

If a single frequency is transmitted with constant envelope, multipath causes the envelope of the received signal to vary in time. This is referred to as *fading*. If all frequencies of a signal experience the same fading, the channel is referred to as *flat fading*. If different spectral regions of the signal are affected differently by the channel, then the channel is referred to as a *frequency selective* fading channel. The boundary between the *flat fading* and *frequency selective* fading channels is referred to as the *coherence* bandwidth of the channel - a term that is difficult to define since it largely depends on the multipath environment.

The time difference between the first received echo and the last echo is known as the *excess delay spread*, ΔT_{ds} , of the channel. If the time between successive data bits is in the order of the *delay spread*, then received echoes will overlap, thereby reducing the integrity of the signal. This type of distortion is known as ISI.

Transmitting data over the multipath channel, can be modeled by convolving the transmit waveform with a stochastic time-varying complex filter [50], where the baseband impulse response is taken from eqn. 2.3 as

$$c(\tau; t) = \sum_{n=0}^{N-1} \alpha_n(t) e^{-j\omega_c \tau_n(t)} \delta(\tau - \tau_n(t)), \quad (2.4)$$

where $c(\tau; t)$ is the impulse response, and $\delta(t)$ is the unit impulse function or Dirac delta function. The "t"-dependence indicates that the impulse response is time varying.

Applying the Fourier transform to eqn. 2.4 with respect to the delay variable τ gives

$$C(\omega; t) = \sum_{n=0}^{N-1} \alpha_n(t) e^{-j(\omega + \omega_c) \tau_n(t)}. \quad (2.5)$$

In 1977, Hashemi [20] developed a computer model that outputs the channel impulse response, $c(\tau; t)$, for an urban environment. The model was later modified in [32] to account for phase correlations for a cellular radio channel. The modified Hashemi model was used to generate the three-dimensional plot of $|C(\omega; t)|^{-1}$ shown on Fig. 2.2. The plot shows $|C(\omega; t)|^{-1}$ instead of $|C(\omega; t)|$ to illustrate the frequency of deep fades (there are several shown as spikes) as a mobile unit travels at 100 km/hr.

It is apparent from this plot that a 30 kHz narrowband channel, commonly used in outdoor cellular communications, would undergo several deep fades over a period of just 8 msec. This is because the signal's bandwidth is less than the coherence bandwidth channel. SS combats the fading by spreading the signal over a band that is much wider than the coherence bandwidth of the channel.

This spreading to combat deep fades is illustrated by the 1.25 MHz signaling band also shown on the figure. In this case, deep fades only wipe out portions of the entire bandwidth (BW). But because regions of the signal are undergoing different fading

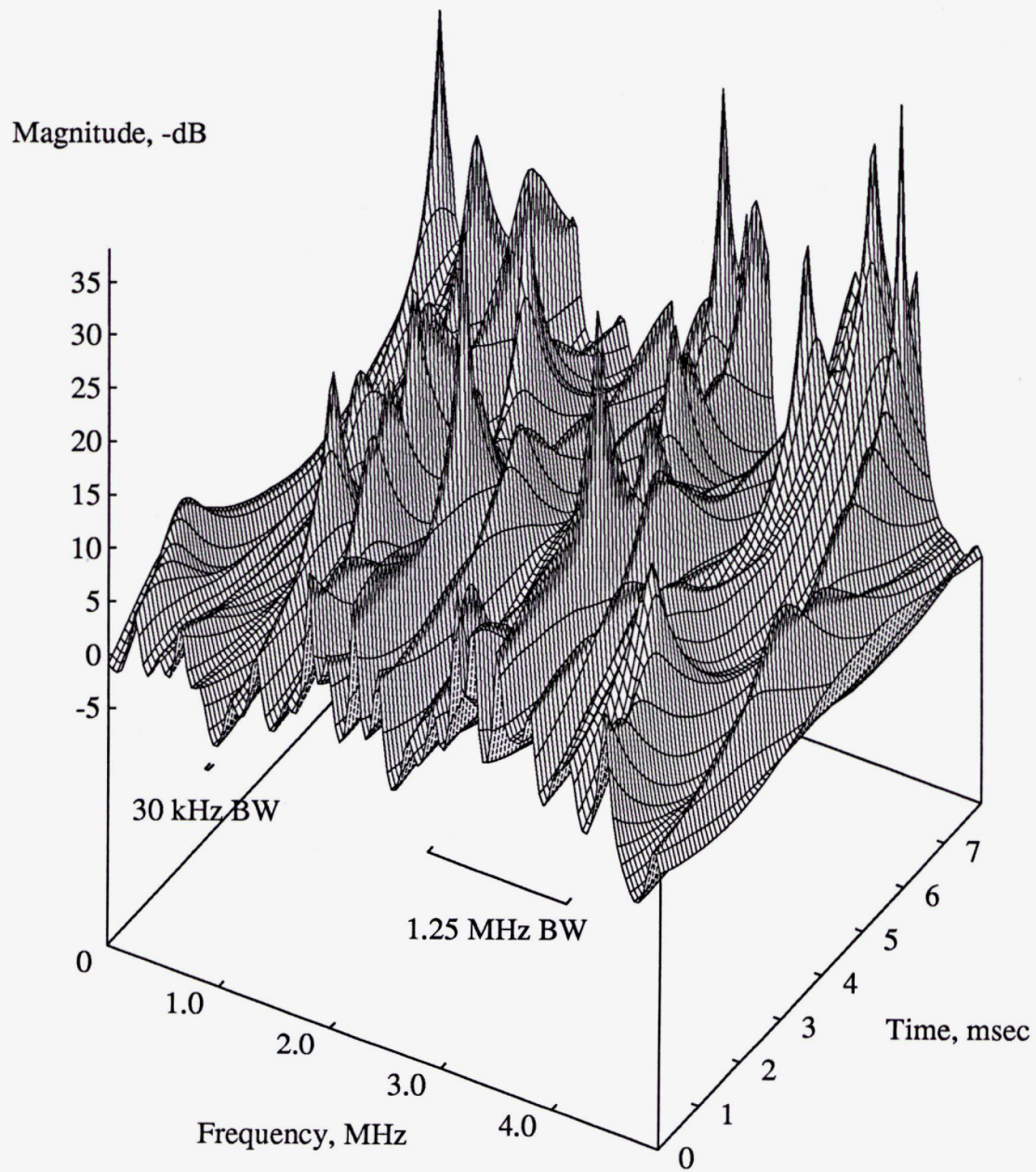


Figure 2.2 Hashemi time varying frequency profiles for a vehicle traveling at 100 km/hr

characteristics, the signal will be highly distorted.

K. Scott [47] shows how an equalizer can be used to estimate this time varying filter transfer function and then convolve its inverse to combat selective fading and ISI. Unfortunately, the shortcomings of this approach is the complexity involved in implementing an equalizer with sufficient convergence necessary to track the channels time-varying characteristics. Another disadvantage is that inverting spectral nulls results in large noise amplification.

Rather than negating the effects of multipath, SS systems can be used to resolve the multipaths echoes and combine them using time diversity. To resolve an echo, it is necessary that the transmit bandwidth, BW_{Tx} , be greater than reciprocal of the time between two successive echoes, ie.

$$BW_{Tx} > \frac{1}{|\tau_n - \tau_{n+1}|} \quad (2.6)$$

Echoes with a differential time delay less than $\frac{1}{BW_{Tx}}$ cannot be resolved and are seen as a single path [53]. Bandwidths in excess of 100 MHz have been used in channel measurements to achieve a 10 η sec resolution [14]. This, however, is an unrealistic bandwidth for outdoor cellular radio. A more realistic bandwidth of 1.25 MHz provides a 0.8 μ sec resolution.

The resolved multipath echoes can then be combined using the approach discussed in Sec. 2.4.

2.1.1 Summary

The multipath channel can be modeled as a linear time varying filter. The impulse response describes the attenuation, modulation delay, and phase of each of the paths that the signal travels from transmitter to receiver, and vice versa.

Signaling bandwidth's smaller than the *coherence* BW of the channel undergo deep fades, whereas signals with large bandwidth's experience *frequency selective* fading. If the signal bandwidth is sufficiently large, it can also be used to resolve the multipath echoes.

The above two factors form the reason for transmitting large signal bandwidth's in a multipath environment

2.2 Historical Background of Spread-Spectrum Systems

Did R. Price and P. Green [41] know that their Rake receiver (developed in 1958) would be in the limelight of digital communications thirty three years later? Perhaps not, but nonetheless it is presently viewed as a possible panacea for transmitting data over outdoor and indoor radio channels.

The first Rake prototype by Price et. al. towered to almost ceiling height over a two foot square area. Recent designs [30] [54] [44] are more compact due to surface acoustic wave (SAW) and very large scale integrated (VLSI) circuit technologies. In these receivers there exists designs consisting of analog, digital, and a hybrid of analog and digital components. The most optimum blend of components will make the design compact and easy to manufacture in order for it to be competitive in the global marketplace.

But how does a Rake receiver use SS signal to improve data communication? A look at the historical applications of SS systems and the development of the Rake will help to answer this question and to understand the motivation behind SS/Rake research.

Patented in 1935 by German engineers Paul Kotowski and Kurt Dannehl, and later in the US in 1940, the first SS systems were targeted for ranging of weather fronts. In

World War II, its usefulness grew to military applications such as antijamming tactical communications, guidance systems, and radio detection and ranging (RADAR)[43]. Today, it is used widely in short wave satellite communications such as the global position system (GPS).

In military antijamming, the signal is being spread so that an enemy's jamming tone - taking up only a limited amount of the spectrum relative to the SS BW - cannot totally wipe out the transmitted signal. World War II's radio controlled glide bombs, particularly the Pelican and Bat [51], were guided to their targets using radio signals from the mother plane. It was feared that the Germans would develop signals that could jam the controls. The work that ensued resulted in the procurement of antijamming transmitters and receivers in June 1944 (unfortunately near the end of the war). Signifying SS as an integral ally in World War II, the postwar Radio Research Laboratory at Harvard report [51] noted:

"In the end, it can be stated that the best anti-jamming is simply good engineering design and the spreading of the operating frequencies"

Spreading can be so extensive that the signal can be hidden under the noise floor and therefore not detected.

In the case of RADAR, "spreading" of the spectrum was used to resolve range echo versions of a transmitted SS signal. A target's range was calculated knowing the arrival time of the returned pulse; the greater the bandwidth of the transmitted signal, the finer the time resolution. Originally used in the mid-1920's by scientists to prove the existence of an ionized gas layer in the upper atmosphere [43], it was later developed for aircraft altimetry instrumentation and target tracking in World War II. Using SS spectrum to resolve time delayed echo's of the transmitted pulse forms

the basis for the Rake concept.

2.3 Spread-Spectrum systems

Although there exist several techniques to "spread" a signal, spread-spectrum describes the common characteristic of these techniques as:

Spread-spectrum is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information. [9]

The "spreading" is accomplished by using a code that is independent of the transmitted data. Detection of the code at the receiver is accomplished by a-priori knowledge of the transmitted code, and then synchronizing its reception to "despread" the signal for data recovery. For example, the codes can be either a pseudo-random signal that has a spectrum not unlike white noise, or a sequence that controls the "frequency hopping" (FH) of the modulated data signal. The former "spreading" technique employs a pseudo-random code that makes the signal appear similar to random noise and difficult to demodulate by receivers other than the intended ones [42]. It is commonly known as direct-sequence spread-spectrum (DS-SS). The latter spreading technique uses a pseudo-random sequence of numbers to control the frequency synthesizer that randomly "hops" the data spectrum over a prescribed bandwidth. There are also several other types of spreading techniques given in the literature [42] [45] [43].

Concerned about the detection of submarines, the Committee on Undersea Warfare, in 1950, urged the development of a system that would allow undetected communication with submarines [37]. The Hartwell report cited the possibility of using a different codes that appear as pseudo noise (PN), to simultaneously transmit on the

same band. This was possibly the birth of a code-division multiple-access (CDMA) SS systems. Today, CDMA systems are the topic of much research [2] [17] [34] [31] [36] [39] [53].

2.3.1 The basic Direct-Sequence Spread-Spectrum Transmitter

A simple DS-SS transmitter is shown on Fig. 2.3 ¹. The left hand side of the figure shows a basic flow diagram of the transmitter inputs while the right hand side shows the signal spectrums at operator outputs. At the transmitter input, the binary signal, $i(t)$ is a non-return to zero (NRZ) signal with levels $\pm\sqrt{\frac{2\varepsilon}{T}}$. The symbol period is T and ε is the energy in each symbol given by

$$\varepsilon = \int_{-\infty}^{\infty} |p(t)|^2 dt. \quad (2.7)$$

The rectangular pulse shape $p(t) = 1$ for $0 \leq t < T$ and $p(t) = 0$ otherwise. The continuous time expression of $i(t)$ is

$$i(t) = \sum_{k=-\infty}^{\infty} I_k p(t - kT). \quad (2.8)$$

I_k is a random variable taking values of ± 1 . $i(t)$ is a binary antipodal signal defined by

$$\begin{aligned} i(t) &= \sqrt{\frac{2\varepsilon}{T}}; & 0 < t \leq T, \\ i(t) &= -\sqrt{\frac{2\varepsilon}{T}}; & 0 < t \leq T, \end{aligned} \quad (2.9)$$

with power spectral density function (PSDF)

$$G_i^2(f) = 2\varepsilon T \left\{ \frac{\sin(2\pi fT)}{2\pi fT} \right\}^2 \quad (2.10)$$

$$= 2\varepsilon T \text{sinc}^2(fT). \quad (2.11)$$

¹Figure format taken with permission from David Dodd's Wireless 92 Calgary presentation

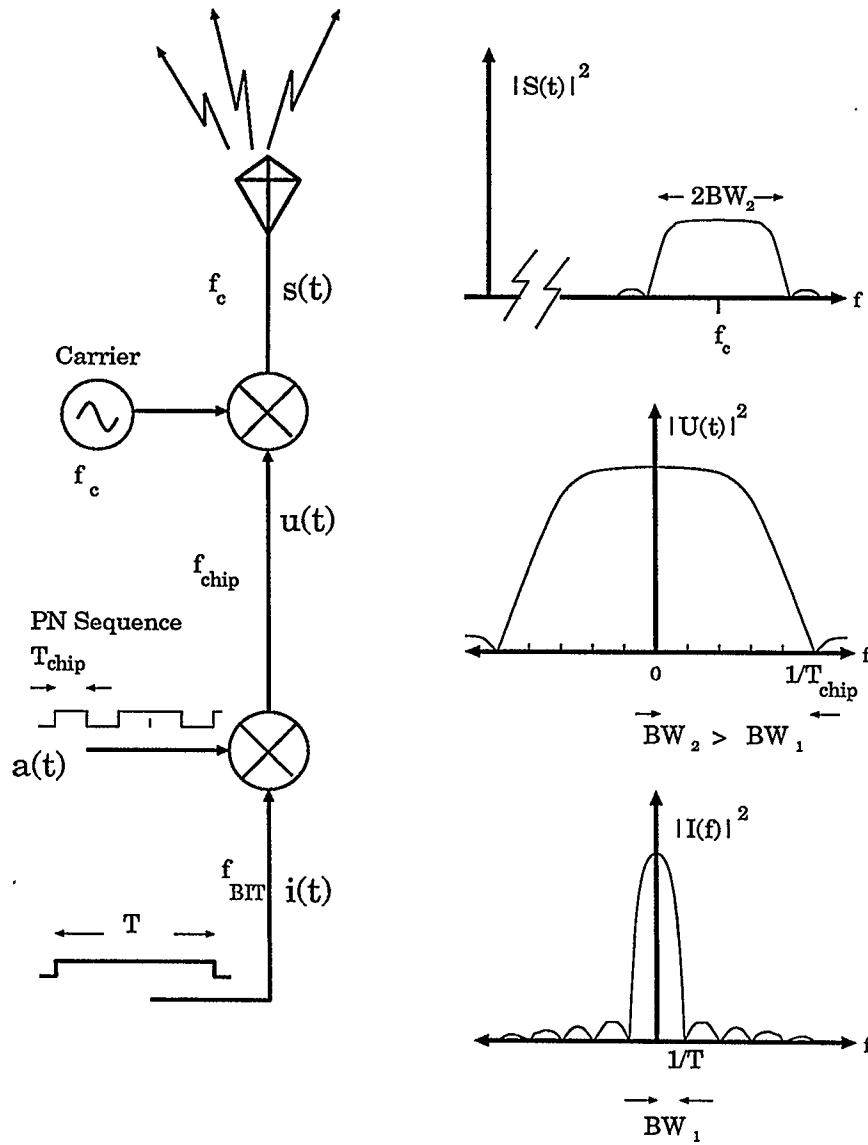


Figure 2.3 Basic DS-SS Transmitter

The *sinc* function in 2.11 has zero crossings at integer multiples of $1/T$ (as shown on Fig. 2.3 except at the origin where $\text{sinc}(fT)$ takes the value 1. The first zero at $1/T$ is commonly used as the bandwidth measure because most (90

In a standard phase shift-keying (PSK) transmitter, $i(t)$ would be directly modulated to a bandpass center frequency of f_c and with the same PSDF as 2.11 and double the bandwidth. Instead, in direct sequence spread spectrum, $i(t)$ is "spread"

by the PN sequence $a(t)$

$$u(t) = a(t)i(t), \quad (2.12)$$

where

$$a(t) = \sum_{k=-\infty}^{\infty} a_k p_c(t - kT_{chip}) \quad (2.13)$$

For a PN sequence of length N_c , the sequence a_k is periodic with period N_c and the signal $a(t)$ has period $T = N_c T_{chip}$. The width of each rectangular pulse, $p_c(t)$ is defined as T_{chip} .

By modulating the data signal, $i(t)$ with the PN signal, $a(t)$, the spectrum of the signal will have an envelope² similar to 2.11 given by

$$G_u(f)^2 = 2\epsilon T_{chip} \text{sinc}^2(fT_{chip}), \quad (2.14)$$

where the first zero crossing is now at $1/T_{chip}$.

If the signal's BW is defined as the first zero crossing, then BW has been spread by the amount

$$\Delta BW_{spread} = 1/T_{chip} - 1/T \quad (2.15)$$

or by the factor, G_p , given by

$$G_p = \frac{T}{T_{chip}} \quad (2.16)$$

where G_p is known as the *processing gain*.

Modulating $u(t)$ by a sinusoidal carrier, $\cos(\omega_c t)$, translates the baseband signal to a bandpass signal

$$s(t) = u(t)\cos(\omega_c t) \quad (2.17)$$

with center frequency $\omega_c = 2\pi f_c$. This can be written as

$$s(t) = \sqrt{\frac{2\epsilon}{T}} a(t) \cos[\omega_c t + \theta(t)], \quad (2.18)$$

²The line spectra due to the PN codes periodicity are not shown.

where the data is represented by phase $\theta(t)$.

The spreading sequence $a(t)$ acts only to spread the signals bandwidth by ΔBW (eqn. 2.15). Without the spreading code, $a(t)$, the transmitter shown on Fig. 2.3 transmits data by a means commonly known as narrowband binary phase shift-keying (BPSK). A more realistic transmitter will be presented in Sec. 3.2 with an indepth PN sequence discussion in Chapter 4 Sec. 4.3.

From eqn. 2.16 it is obvious that by increasing G_p , the signal's BW will also be increased. Fig. 2.3 shows the spreaded waveform $u(t)$ when G_p is equal to 5.

2.3.2 The basic Direct-Sequence Spread-Spectrum Receiver

In the case of a single path channel with delay ΔT and attenuation α , the signal received at the receiver (Fig 2.4) is

$$r(t) = \sqrt{\frac{2\varepsilon}{T}} a(t - \Delta T) \cos[\omega_c(t - \Delta T) + \theta(t - \Delta T)]. \quad (2.19)$$

Assuming that the receiver can coherently demodulate $r(t)$, the output from the mixer will be

$$\hat{u}(t) = \sqrt{\frac{2\varepsilon}{T}} a(t - \Delta T) \cos[\theta(t) + \phi], \quad (2.20)$$

where $\phi = \omega_c \Delta T$ is the phase error between the local-oscillators (the demodulator is assumed to have a peak-to-peak of 4. At this point, alignment of the receiver's PN code $\hat{a}(t)$ by the received code $a(t)$ is non-trivial; the codes must be synchronized and in-phase to prevent decorrelation. Assuming that the channel's time delay can be estimated as $\Delta \hat{T}$ such that $\Delta T - \Delta \hat{T} \approx 0$, then the combined function of the multiplier and integrator (together known as a correlator) computes the cross-correlation function defined as

$$R(\tau)_{a(t)\hat{a}(t)} = \frac{1}{T} \int_0^T a(t) \hat{a}(t - \tau) dt, \quad (2.21)$$

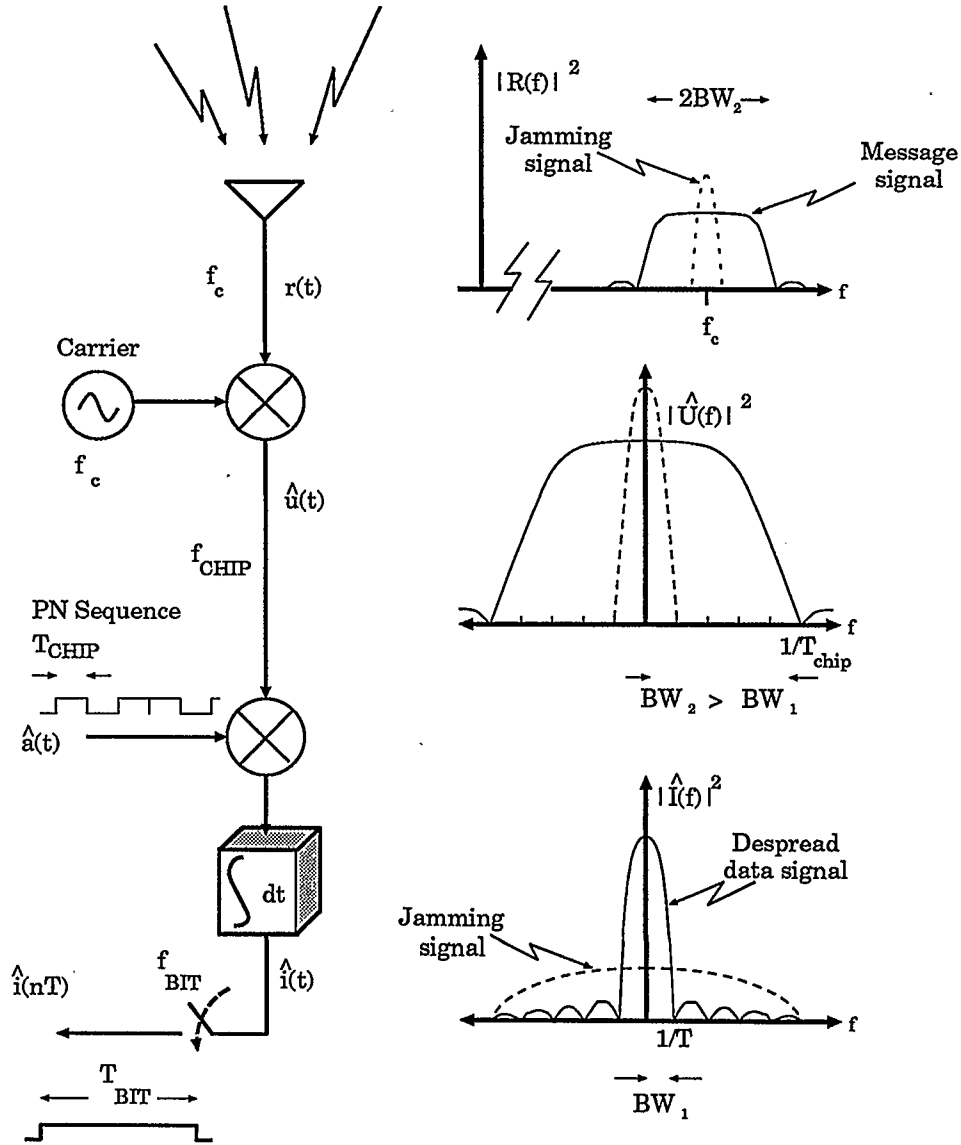


Figure 2.4 Basic DS-SS receiver

where τ is the alignment offset between the two codes. If $\hat{a}(t)$ is the same code as $a(t)$, then the autocorrelation will be calculated at the output of the integrator as depicted on Fig. 2.5 where the correlator output is given by [45]

$$R(\tau) = \begin{cases} 1 - \frac{|\tau|}{T_{chip}} & |\tau| < T_{chip}, \\ \frac{1}{N_c} & |\tau| \geq T_{chip}, \end{cases} \quad (2.22)$$

for an m-sequence of length N_c . Otherwise, if the codes are not the same, then the

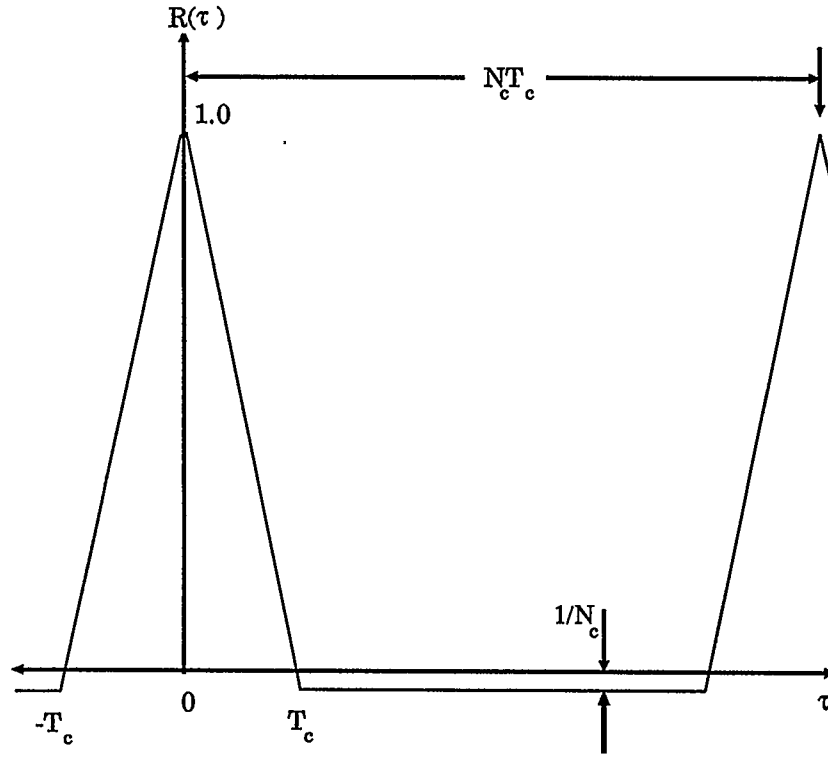


Figure 2.5 Autocorrelation output

output will depend on the cross-correlation properties of the two codes.

In the case where multiple users transmit over the same bandwidth, optimal codes can be chosen so that all codes used by the other transmitters appear as noise, ie. the cross correlations appear as noise, while the code intended for the receiver computes the autocorrelation. This type of channel bandwidth reuse is used in CDMA receivers.

If in the example, the code at the receiver is the same as the transmitter, ie. $\hat{a}(t) = a(t)$, then the correlator computes the autocorrelation (eqn. 2.22) only at zero.

From (Fig 2.4), the correlator output is

$$\begin{aligned}
 \hat{i}(t) &= \frac{1}{T} \int_0^T \hat{a}(t)a(t)i(t)dt \\
 &= \frac{1}{T} \int_0^T i(t)dt; \quad \text{if } \hat{a}_k = a_k
 \end{aligned} \tag{2.23}$$

Assuming that the bit clock is synchronized to sample at the apex of the autocorrelation function ie. at $R(0)$ (Fig 2.5), then the output after the sampler is given as

$$\hat{i}(nT) = \sqrt{\frac{2\varepsilon}{T}} \cos[\theta(nT)]. \quad (2.24)$$

The signal $r(t)$ 2.19 has now been *despread* by exploiting the autocorrelation properties of the spreading sequence $a(t)$ and the data recovered.

If a jamming signal is present at the input to the receiver (also shown on Fig. 2.4), its spectrum will be *spread* by the PN sequence to resemble low level noise at the sampler with a relatively flat spectrum. This is a result by the poor correlation properties between the jamming signal and the receivers PN sequence.

2.3.3 Summary

It was shown that a data signals bandwidth can be increased by multiplying the data signal with a periodic PN sequence. Conversely, it can be *despread* by correlating the transmitted signal with a delayed version of the same PN sequence at the receiver. For perfect data recovery, it was assumed that the receiver could accurately estimate the single path channel delay; otherwise, decorrelation would occur. Perfect bit clock recovery was assumed so that the correlator output could be sampled at the apex of the autocorrelation. Several different PN sequences could be used to transit data over the same bandwidth providing that they have excellent cross-correlation properties.

2.4 The Rake Receiver

2.5 Introduction

Ochsner [39] noted that conventional SS systems, or "one-path" receivers, discard the signal energy present in the other paths; rejecting this energy reduces the mul-

tipath processing gain of the system. A more prudent method would be to employ time diversity to combine this energy for improved signal recovery. Such a method was implemented by Paul Green in 1954 when he was in charge of building the first "NOise Modulation And Correlation" (NOMAC) system for the Army Signal Corps at Lincoln Laboratory, MIT.

The first NOMAC system was called the F9C. It consisted of a bank of correlators that could resolve multipath from ionospheric and tropospheric reflectors. The receiver was designed to select the strongest output from the bank of correlators. Field tests showed that the standard frequency shift-keying (FSK) teletype link (used for the reverse link) outperformed the F9C. It was concluded that the FSK receiver operated on the energy received over all paths; while the F9C lost a considerable amount of energy since it only selected one path [43]. Bob Price, who had synthesized a signal processing technique for receiving signals sent over multipath channels, met with Green to discuss the F9C problem. The fruit of this conversation was a method by which the taps of the F9C correlators could be adaptively controlled by the correlator outputs. This system was coined the "Rake" receiver by Green and the first prototype was developed by Price. Field tests of the "new and improved" Rake showed a 17 dB improvement of the F9C over the FSK against jamming in the presence of multipath [43].

Understanding how the Rake was born is of particular importance since thirty years later it is being reborn to improve signal reception in a cellular multipath channels.

2.5.1 Transversal filter realization of the Rake

A heuristic approach will be used to realize the Rake structure by looking at the multipath delay spread of the channel. In the single path channel, it was shown that knowing the channel's delay, ΔT , was required at the receiver a-priori to calculate the autocorrelation at zero (eqn. 2.22). Now suppose that the multipath channel consisted of n paths, each with delays $\Delta T + \tau_0, \Delta T + \tau_1, \Delta T + \tau_2, \dots, \Delta T + \tau_{n-1}$, then it would be possible to use n correlating receivers with appropriate code delays to *despread* the signal from each of the n paths. Doing so would increase the multipath processing gain by a factor, $\mathcal{G}_{mp}$, defined as

$$\mathcal{G}_{mp} = \frac{\sum_{n=0}^{N-1} \alpha_n}{\alpha_0}, \quad (2.25)$$

where α_n is the attenuation of the n th path.

The time between the arrival of the first path at time τ_0 to that of the last path at time τ_{n-1} , is known as the delay spread of the channel. If the sample values from the n correlators were input to n port adder, the multipath processing gain, $\mathcal{G}_{mp}$, is realized. The output of an infinite number of correlators would be the channel's impulse response (eqn. 2.4) depicted by its complex envelope on Fig. 2.6 [53]. for a 4 path channel.

The simplest realization of the Rake is called the digital Rake (DRake). Fig. 2.6 shows the transversal filter realization of the DRake as the output from an n -port adder travels down its delay tap line. One possibility of controlling the taps switches is to use a threshold detector at the end of the delay line that would detect the presence of the first path. Upon detection of the first path, the DRake would test all data values stored in the delay line against a threshold value. Those exceeding this value would get summed to the output. By implementing the Rake in this fashion,

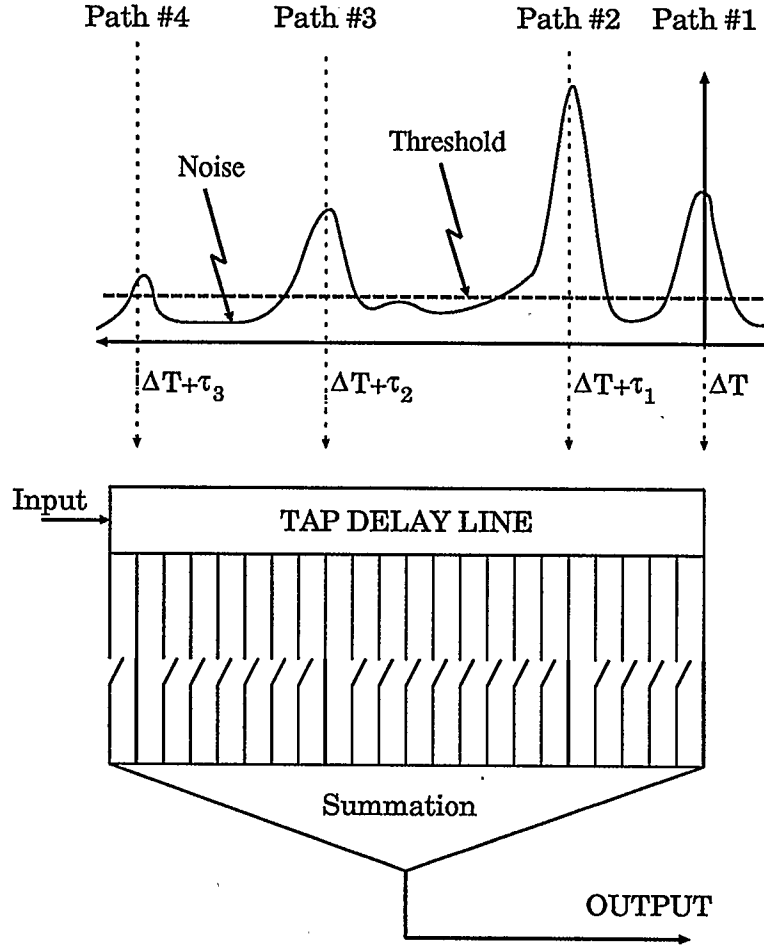


Figure 2.6 Transversal Realization of the Rake

the bit time cannot be any shorter than the delay spread of the channel, ie.

$$T > T_{ds}, \quad (2.26)$$

where $T_{ds} = \tau_3$ in this example. Each echo's energy can be summed to the output, thereby increasing the processing gain of the receiver. This scheme does not reduce the effect of ISI. Since the above scheme vectorially adds the energy from each path, some form of phase correction must also be executed before the DRake to achieve coherent addition of the signal phasors. This is discussed in Chapter 4 Sec. 4.4.4.

A more elaborate scheme was proposed by Turin [53] which uses a *sounding re-*

ceiver to determine the channel characteristics ie. attenuation and delay. In this scheme, the channel's phase is not estimated because of the complexity of the receiver and the rapid phase changes that render it impossible to track and hence used effectively [53]. The scheme is realized with the same transversal filter in Fig. 2.6 except that the delay line tap outputs are followed by multipliers which are weighted using the optimal linear diversity combiner [6]. The transversal filter is now an approximation of a filter matched to the channel. The output of the receiver is now the convolution of the received signal and the estimated channels impulse response. An undesirable artifact of this scheme is the loading and unloading of the transversal filter taps; the filter outputs echo correlation spikes from the time the first echo enters the filter (time $t_0 - t_{n-1}$) to the time when the last echo leaves the filter (time $t_0 + t_{n-1}$). Consequently, the maximum data rate is limited by

$$T > 2T_d. \quad (2.27)$$

An even more elaborate system was built by Green and Price [41] which uses an adaptive approach and difference-frequency correlation [13] to estimate the channel phase and attenuation. This system was implemented in the F9C Rake prototype described in Sec. 2.4.

2.5.2 Recent Developments of Rake Receivers

The advent of digital cellular radio has resurrected the development of Rake prototypes targeted for the indoor and outdoor cellular environments. Advancements in digital signal processors, application specific integrated circuits (ASIC), and surface acoustic wave (SAW) technologies, have given the development engineer the tools to develop DS-SS systems that are compact enough for consumer mobile systems and

yet have the processing capacity of the F9C Rake.

2.5.2.1 Kavehrad and Bodeep Direct-Sequence Spread-Spectrum Receiver

The earliest design using one of the above mentioned technologies is M. Kavehrad and G. Bodeep's SAW filter-based SS transceiver (1987) [30]. Unlike the correlator mentioned in Sec. 2.3.2, Kavehrad's SAW matched filter is analogous to a transversal filter structure - much like the one on Fig. 2.6 - with the tap weights equal to the transmitted PN sequence. When the transmitted code aligns with the filter taps, the correlation peak appears at the output. However, with non-alignment, the output is the *selfnoise* or cross-correlation of the transmitted code with the matched filter PN sequence.

A transversal and SAW matched filters are shown on Fig. 2.7 to illustrate their concept of operation. In this example, a hypothetical maximal length of $N = 5$ - hypothetical since $N \neq 2^k - 1$ for m-sequences (Sec. 4.3.1) - is shown as a baseband signal (Fig. 2.7 a)), and as a bandpass signal (Fig. 2.7 d)). When the baseband input signal aligns with the transversal filter taps, (Fig. 2.7 a) and b)), the filter reaches a maximum value shown on Fig. 2.7 c) (the maximum value is normalized to 1.0). For the SAW filter, the input signal is first converted to an acoustic waveform which then travels down the piezoelectric surface of the SAW. When the wave peaks align with the electrode as shown on Fig. 2.7 d) and e), a modulated correlation peak is produced (Fig 2.7). The advantage to this approach over correlating receivers is: 1) no *acquisition* and *tracking* loop is needed to align the PN sequences and 2) matched filtering will output the correlation of several echos without a cascaded set of correlators (as suggested in Sec. 2.3.2). The disadvantage is the length of the matched filter required when long PN codes are used. Specific problems associated

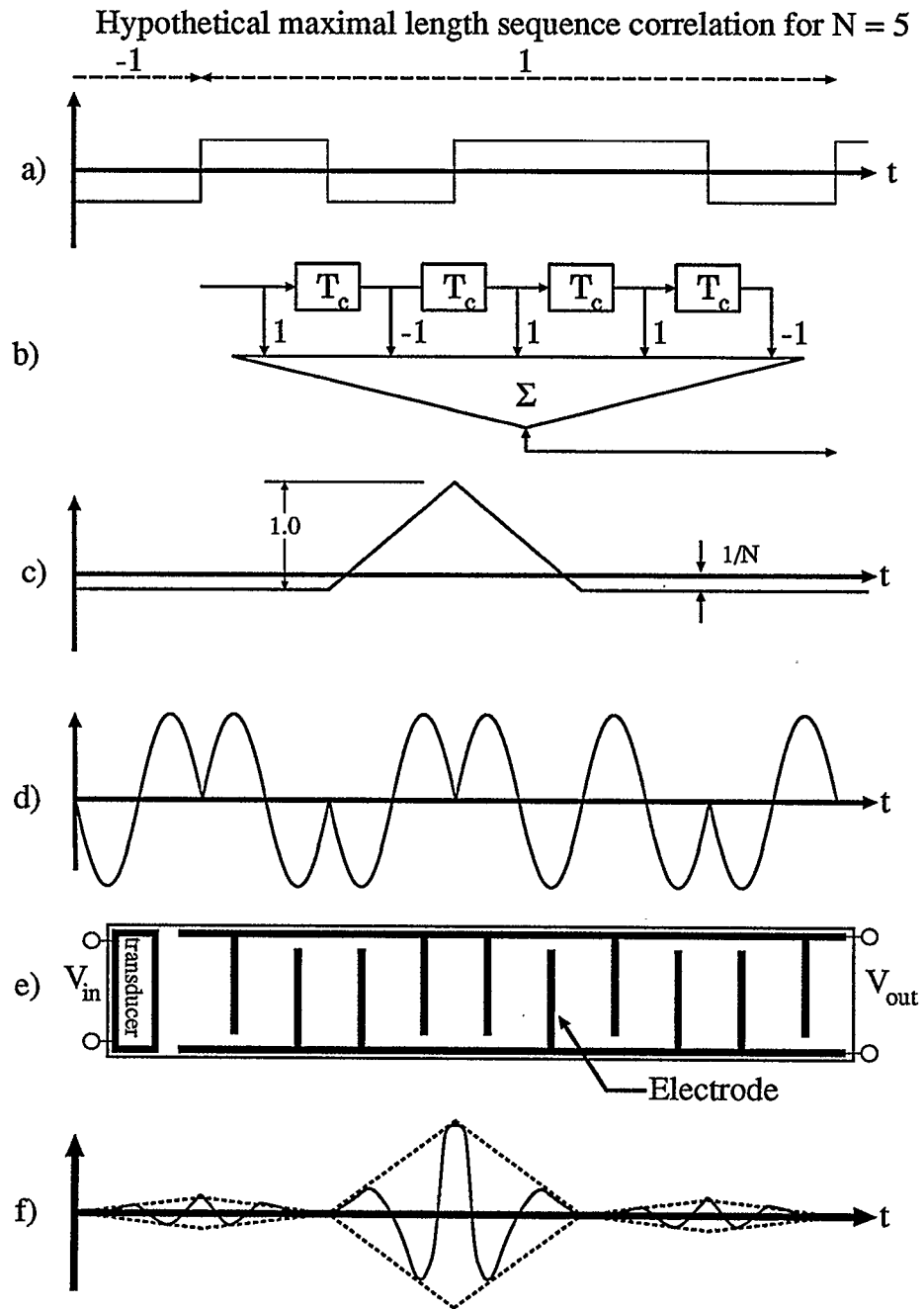


Figure 2.7 Transversal and SAW matched filters (MF): a) baseband binary signal input to transversal MF, b) transversal MF, c) transversal MF output, d) passband signal input to SAW MF, e) SAW MF, and f) SAW MF output.

with SAW filters are:

1. non-programmable³,
2. low dynamic range,
3. high signal attenuation, and
4. instability at high temperature.

These problems may be solved as SAW technology advances.

Kavehrad's system uses two SAW devices: one as a matched filter (MF) to compute the correlation; the other as a delay device (Fig. 2.8), to differentially detect the bandpass signal. The use of the SAW MF eliminates the need for an acquisition and tracking loop; thereby making the receiver simplistic in design. The multipath combining is implemented with a finite time integrating filter acting as an equal gain diversity combiner - a similar digital version would be the transversal Rake (Fig. 2.6) with all the taps closed. By using matched filters, the code synchronization time is instantaneous: a major advantage to this system. The disadvantage to this design is the amount of analog components and interconnects between devices that, in a commercial product, would mean more tuning and maintenance. Integration of this receiver onto a single chip would be difficult with both analog and acoustic components.

2.5.2.2 Grob et. al. N-Path Rake Receiver

The successor to the F9C DS-SS receiver was designed in 1990 by Grob et. al. [54]. This appears to be the first method of combining multipath echos with the

³Programability is important if the DS-SS receiver is to be used in CDMA systems.

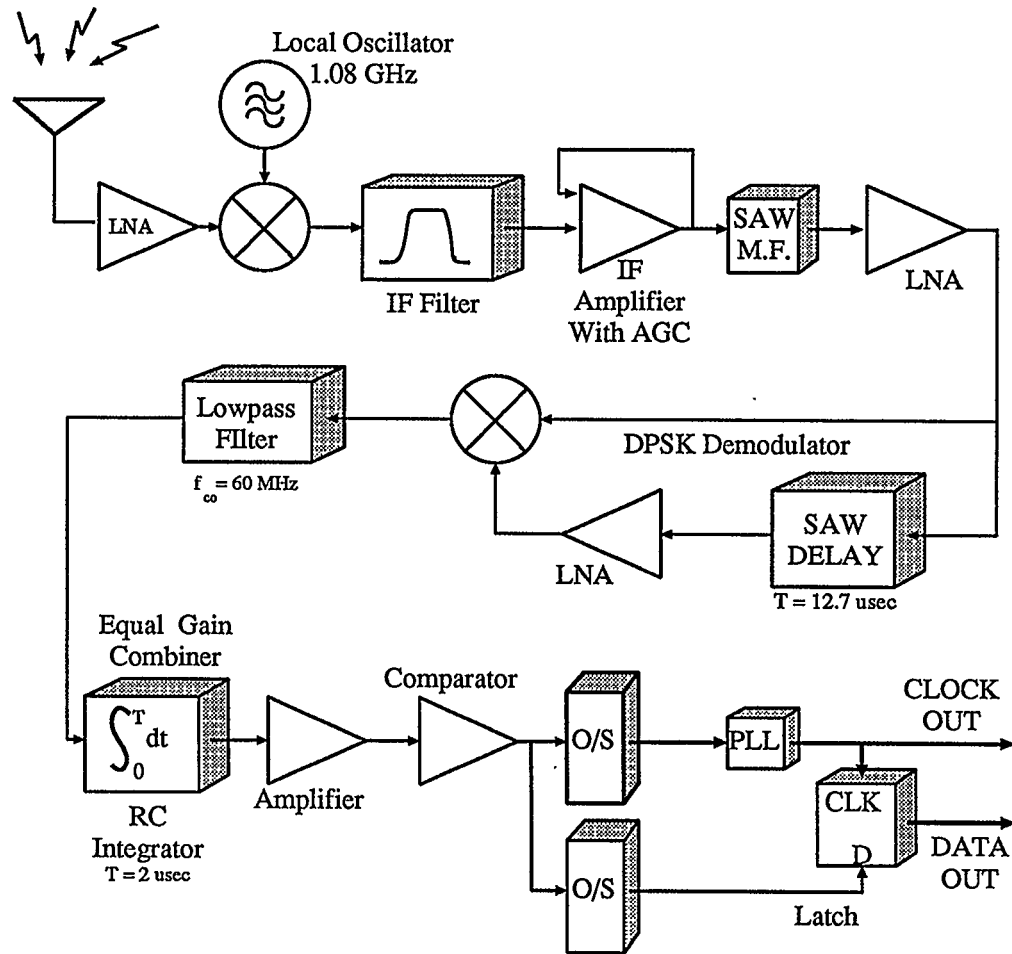
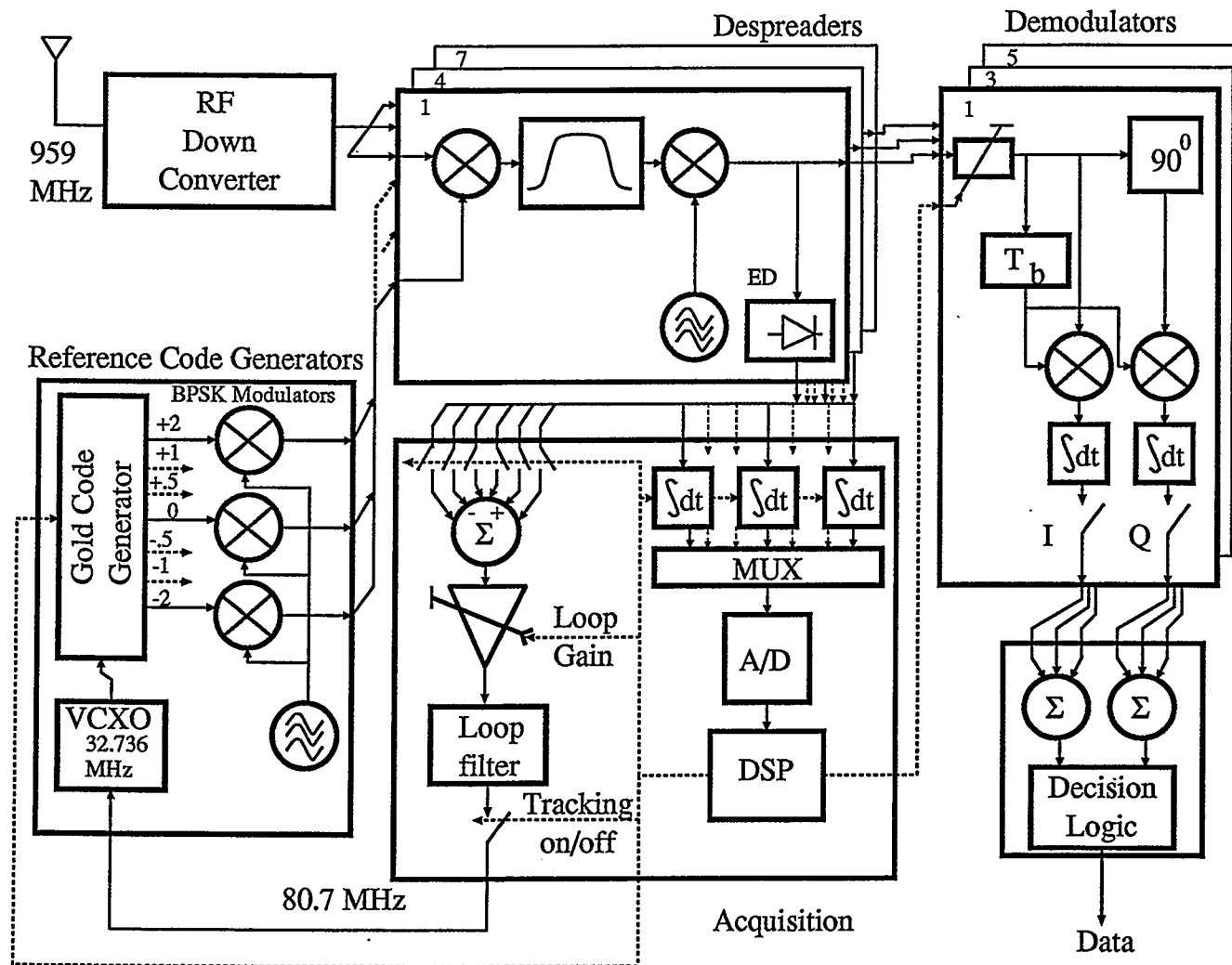


Figure 2.8 Kavehrad and Bodeep Direct-Sequence Spread-Spectrum Differential Phase-Shift Keying Transceiver

Rake concept since the 1950's. The hardware design essentially functions as a bank of correlators which Grob has termed *despreaders* (Fig. 2.9). After the radio frequency (RF) downconverter stage, the intermediate frequency (IF) signal is *despread* in a fashion similar to that described in Sec. 2.3.2, except that instead of a single correlator, a bank of five correlators are used to collect the energy from a maximum of five echos spaced one chip time apart (the window created by the despreading bank is 300 ns). Unlike the correlator in Sec. 2.3.2, the *despreaders* use bandpass filters instead of integrators (lowpass filters) to collapse the signal spectrum at band-

Figure 2.9 Grob et. al. N-Path Rake Receiver



pass. Bandpass despreading also facilitates code tracking and acquisition since at bandpass, the PN-sequence does not have to be data demodulated before entering the delay-lock loop⁴. The delay-locked loop discriminator S-curve is calculated using two despreaders spaced $\frac{1}{2}T_{chip}$ on either side of the center despreaders⁵. Consequently, there are a total of seven despreaders which input the received IF signal and seven locally generated IF versions of the delayed PN code.

The receiver PN code is synchronized with the transmitted codes by code *acquisition* and *tracking*. A characteristic of this method is the inherent delay in the time that the signal is first received at the antenna and the time that the data appears at the receiver output: unlike the Kavehrad receiver's instantaneous code acquisition. Acquisition is achieved by, either continuously or discretely, shifting the receivers code over the entire bit time, T . In Grob's system, the code is discretely shifted and the correlation calculated at each successive shift time ϵ . Assuming that the bit time adheres to eqn. 2.26, the maximum time⁶ to determine all possible correlation outputs is

$$T_{acq} = N_c \frac{T_{chip}}{\epsilon} T, \quad (2.28)$$

where N_c is the number of chips in the PN - sequence. For example, if $\epsilon = T_{chip}/2$ then the maximum time to acquisition is $N2T$. This time-to-lock could be intolerable in TDMA systems, where users have a fixed time frame to receive data.

The output of the despreaders is sampled by an analog-to-digital converter (A/D) before entering a DSP processor which controls acquisition, tracking, and the Rake attenuators. Upon successful acquisition, the DSP processor switches control to the

⁴Data demodulation would be very difficult since SS systems typically operate at very low signal-to-noise ratios in the transmission bandwidth [45].

⁵Delay-lock loops exploit the correlation properties of the PN-code whereby the sum of the two fractionally spaced despreaders will output a zero value when the loop is locked.

⁶Time-to-lock could be much larger for a multipath channel

tracking loop and monitors the despreaders outputs for out-of-lock detection. Should the loop loose lock, the DSP switches to acquisition mode.

The channel phase is removed by differentially decoding the signal after the variable attenuators. In this way, the multipath energy can be added coherently.

The design approach taken by Grob has the advantage that: 1) long PN codes can be implemented without any additional hardware, and 2) bandpass despreaders facilitates code tracking. The disadvantages are

1. the loss of multipath processing gain by only using two paths for tracking (Sec. 2.4),
2. the maximum multipath processing gain is five (there are only seven despreaders on which two are for tracking),
3. the extensive use of analog mixers and bandpass filters,
4. the hybrid of analog and digital components limits chip integration,
5. acquisition time, and
6. time diversity combining is limited to a 300 ns window.

2.5.2.3 Kaufmann et. al. Spread-Spectrum Multipath-Diversity Receiver

Certainly a step in the right direction is the multipath-diversity receiver designed by Kaufmann et. al. [44]. One of the main reasons that the Grob system used analog components for despreaders was the high speed digital components that would have to be used for processing data at bandpass rates⁷. On the other hand, Kaufmann et.

⁷Grob's sampling rate would have been twice the IF frequency of 70 MHz!

al. demodulated the spreaded signal down to baseband before sampling. As a result, the sampling rates are lower than at bandpass⁸.

Despreading is computed with eight digital correlators called the time integrating correlators (TIC's) (Fig. 2.10), (see Sec. 2.3.2 for correlator description) which are programmed onto an ASIC chip. The eight correlators (or Rake-arms) can be programmed to despread anywhere over a window of $1\mu s$ at spacings of $\frac{1}{2}$ a chip. During acquisition, all eight Rake-arms are used to test all possible delay positions for the maximum signal energy. By incorporating all Rake arms, the time to acquisition is now:

$$T_{acq} = \frac{2N_c}{L_{tot}}T, \quad (2.29)$$

where L_{tot} is the number of Rake arms. Notice that with the exception of L_{tot} , the acquisition time is similar to that of eqn. 2.28 for $\epsilon = T_{chip}/2$. Kaufmann's system reduces the acquisition time by a factor of L_{tot} .

When the system is in a tracking mode, the eight Rake arms are split into two functions: 1) four of the arms are positioned at the largest received path energies for data demodulation, and 2) the remaining four are used to estimate the channel impulse response. In this way the sytem is able to determine the maximum likelihood (ML) estimate of the channel impulse response [44].

Since the data is coherently demodulated, the system invokes the ML principle to make an estimate of each echo's carrier phase and attenuation. This estimate is then used to rotate and weight the correlation output; performing maximal-ratio combining [42].

The hardware consists of standard quadrature down conversion mixers, lowpass

⁸Kaufmann et. al. sample their baseband signal at twice the chip rate (32.736 MHz). while still satisfying the Nyquist criteria [42].

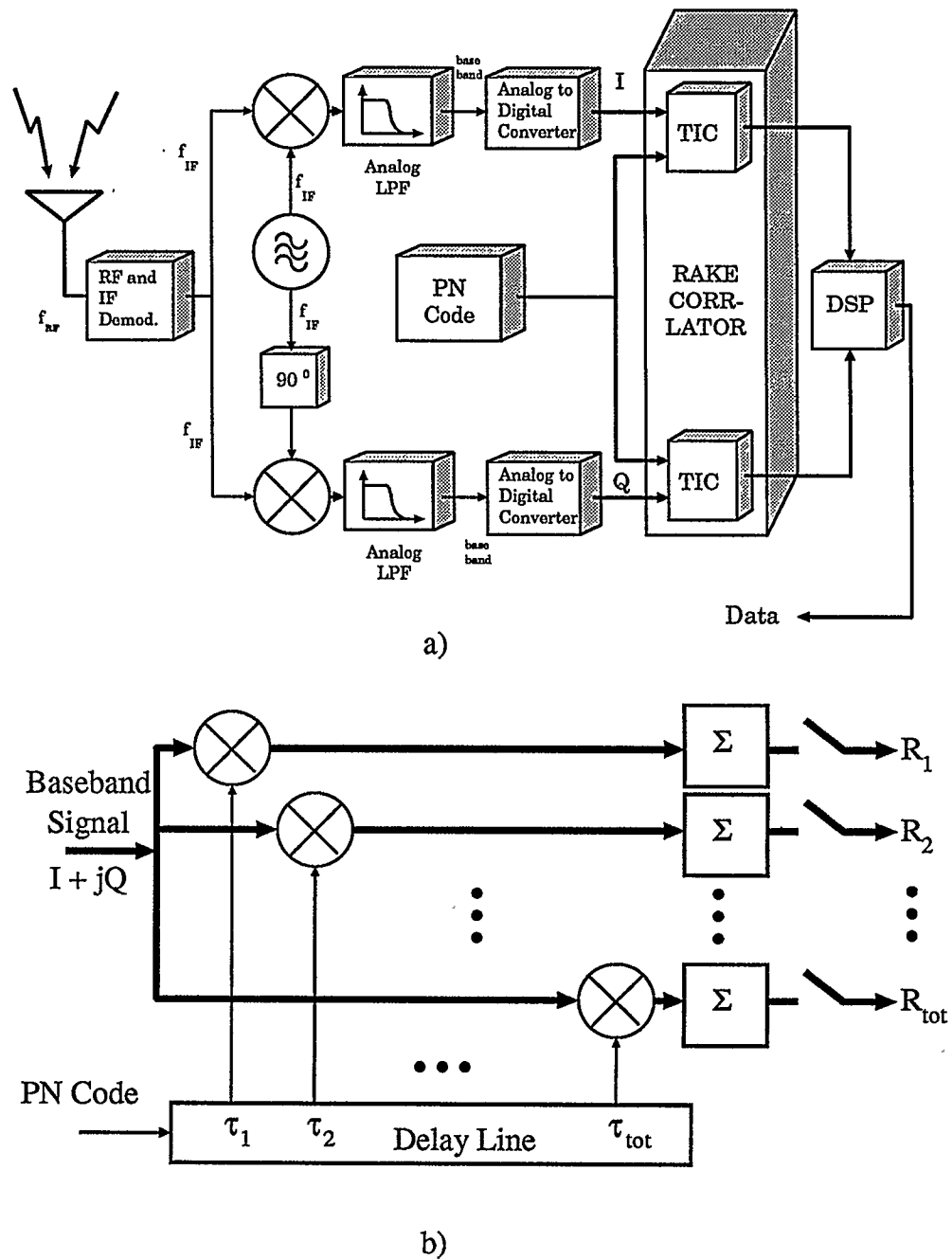


Figure 2.10 Kaufman et. al. Multipath-Diversity Receiver: a) hardware layout and b) time integrating correlator (TIC)

filters, and A/D's. The TIC performs the correlation multiplications⁹ and summations, while the DSP processor controls acquisition and tracking modes of the TIC and also coherent data demodulation.

Integration of the correlation functions onto a single chip has greatly reduced the hardware complexity of the system. It also benefits from the advantages of digital signal processing like high precision, drift-free operation, and no aging [44]. The use of correlators still has the unhealthy side-effect of acquisition time (the time required to align the received code with the code generated at the receiver); however, Kaufmann has found a way to decrease this time by a factor of L_{tot} . This is a definite advantage over the Grob system[54] since acquisition time may exceed coherence time of the channel¹⁰, especially for long PN-codes.

The digital processing of correlations brings Kaufmann's design a step closer to a receiver that may be integrated onto a single chip; integration of the DSP processor into an ASIC along with the TIC is viable although the structure would be very complex. Furthermore, digital structures can be used to eliminate the duplication of the I and Q quadrature down-conversion arms. This can be incorporated into the design by incorporating techniques in Chapter 3 Sec. 3.5.

2.5.3 Summary

The three systems discussed despread the transmit signal and resolve the echoes using two different methods: 1) the matched filter approach, and 2) the correlating bank approach. The correlating method has been implemented at both IF (as in Grob st. al. [54]) and baseband (as in Kaufmann et. al. [44]). Kavehrad's matched filter was implemented at IF.

⁹Multiplications at baseband are trivial since the multiplicand is either +1 or -1.

¹⁰Kaufmanns system transmits at an RF frequency of 2.44 GHz.

In two of the systems, either SAW or analog technologies were used to handle the high data rates and intensive computations indicative of passband signals. Whereas in the single baseband system, the data rates are much slower hence making processing viable with ASIC chips.

The matched filter has the advantage of instantaneously calculating the correlations of the received echoes without the need for an acquisition and tracking stage. It is unlimited in the number of echoes that it can process, however, its size is proportional to the number of chips used in the PN-code.

The number of correlators required is directly proportional to the number of echoes received and inherently require a complex algorithm to scan for echo locations before locking onto them. Scanning results in an acquisition time that could possibly be longer than the coherence time of the channel. A good characteristic of this method is the minimal amount of hardware needed to increase the PN-code length.

CHAPTER 3

DIGITAL IMPLEMENTATION OF A $1/4$ WAVE SAMPLING RECEIVER

3.1 Introduction

In this chapter, a technique is presented that utilizes simple digital structures to reduce the number of analog components typically used in a standard quadrature down-conversion receiver. The analog components, ie. linear multipliers, lowpass filters (LPF), analog-to-digital converter (A/D), are duplicated in the inphase and quadrature down-conversion chains of the receiver. The technique employs digital bandpass sampling to either eliminate an entire analog part, or reduce duplication. Advantages of digital replacement of analog components are largely economical although system performance can also be improved.

An overview of an analog quadrature up-conversion and down-conversion system is given to develop the fundamentals of baseband/bandpass modulation and demodulation theory. Next, a digital receiver is realized by replacing analog components with digital counterparts. Finally, complex and real filters with decimation chains are presented as an alternative to conventional down-conversion demodulation. The baseband equivalent system is derived for simulation purposes.

Although the above mentioned techniques can be easily applied to quadrature up-conversion, the transmitter developed for this thesis only transmits standard binary phase shift-keying and therefore does not suffer from duplication of analog components - symptomatic of the quadrature up-conversion transmitter.

Matched filtering with square-root Nyquist (SRN) filters is briefly discussed in the quadrature transceiver. Although not necessary in the $1/4$ wave sampling derivation,

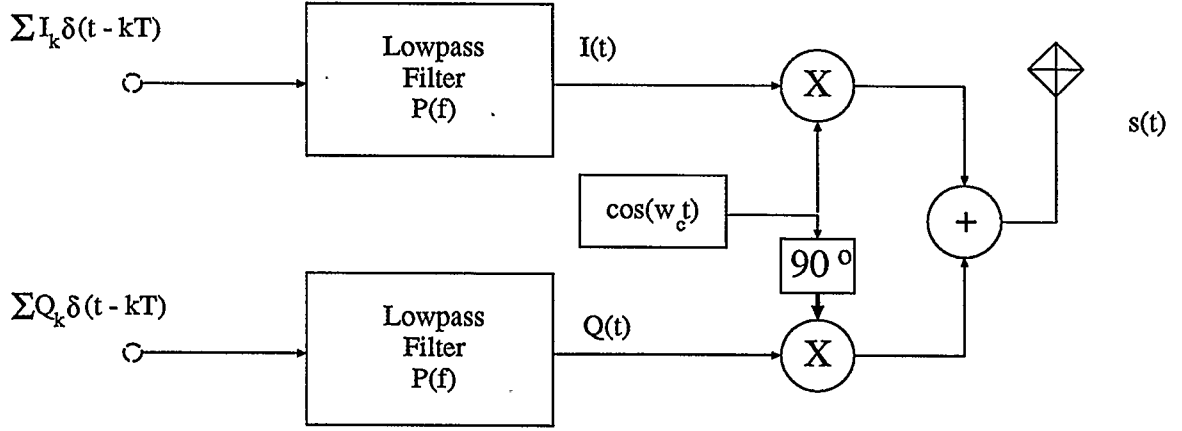


Figure 3.1 Conventional analog quadrature modulator.

they are included to reduce the re-derivation of quadrature matched filtered signals in the following chapters.

3.2 The Quadrature Modulator

An example of a quadrature modulator is shown on Fig. 3.1. The continuous-time in-phase and quadrature components at the output of the lowpass filter are

$$I(t) = \sum_{k=-\infty}^{\infty} I_k p(t - kT) \quad (3.1)$$

and

$$Q(t) = \sum_{k=-\infty}^{\infty} Q_k p(t - kT), \quad (3.2)$$

where I_k and Q_k represent the k th transmitted symbol in the complex plane.

Ideally the overall system is designed to bandlimit the signal and have zero inter-symbol interference (ISI). A class of pulses that meet this criteria is given in Proakis [42]. These are commonly referred to as raised-cosine pulses.

It can be shown, at least for an AWGN channel, that the receive filter should be matched to the transmitted pulse $p(t)$. This means that the lowpass filter in the

transmitter should have a square-root raised cosine response. In this thesis, these filters are referred to as square-root Nyquist filters (SRN).

From Fig. 3.1, the narrowband bandpass signal, $s(t)$, at the output of the two port adder is

$$s(t) = I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t) \quad (3.3)$$

Alternatively, $s(t)$ can be expressed by either

$$s(t) = \text{Re} \{ \tilde{w}(t)e^{j\omega_c t} \} \quad (3.4)$$

or

$$s(t) = w(t)\cos[\omega_c t + \phi(t)]. \quad (3.5)$$

The complex envelope is defined by

$$\tilde{w}(t) = I(t) + jQ(t) \quad (3.6)$$

$$= w(t)e^{j\phi(t)}, \quad (3.7)$$

where the envelope $w(t) = |\tilde{w}(t)|$, while the phase $\phi(t) = \arg[\tilde{w}(t)]$.

It is the task of the receiver to demodulate the signal to baseband and recover the quadrature signals $I(t)$ and $Q(t)$.

3.3 Down Conversion

3.3.1 Conventional Analog Quadrature Demodulator

Down converting a received signal to baseband can be accomplished using a variety of demodulation structures. The conventional analog quadrature demodulator is shown in Fig. 3.2. Illustrated is the duplication of analog components, ie. LPF's and linear multipliers, that are necessary to down convert the signal to baseband.

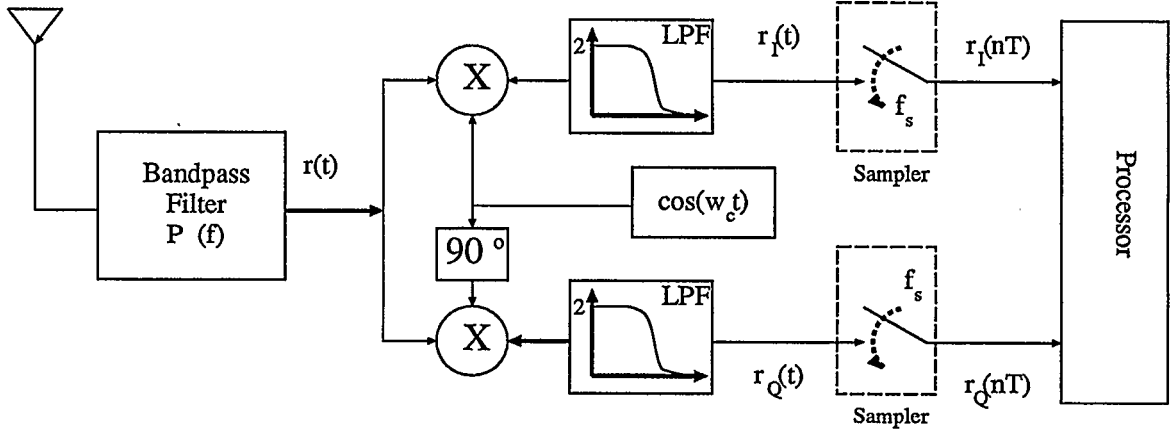


Figure 3.2 Conventional analog quadrature demodulator.

After passing through the channel, the narrowband received signal at the output of the bandpass filter is given by

$$r(t) = \text{Re} \{ \tilde{r}(t) e^{j\omega_c t} \} \quad (3.8)$$

$$= r_I(t) \cos(\omega_c t) - r_Q(t) \sin(\omega_c t), \quad (3.9)$$

where the complex envelope $\tilde{r}(t) = r_I(t) + jr_Q(t)$.

The bandpass filter is a SRN matched filter matched to the transmit pulse. The quadrature mixers down-convert the signal to baseband while the lowpass filters remove the unwanted double frequency terms. Following the down-conversion, the inphase and quadrature signals, $r_I(t)$ and $r_Q(t)$, are sampled at the symbol rate. The sampled values are then processed to recover the binary information.

3.3.2 Complex Demodulation

Another approach to demodulating the bandpass signal to baseband is to use the complex demodulator shown on Fig. 3.3. In this demodulator, the narrowband signal,

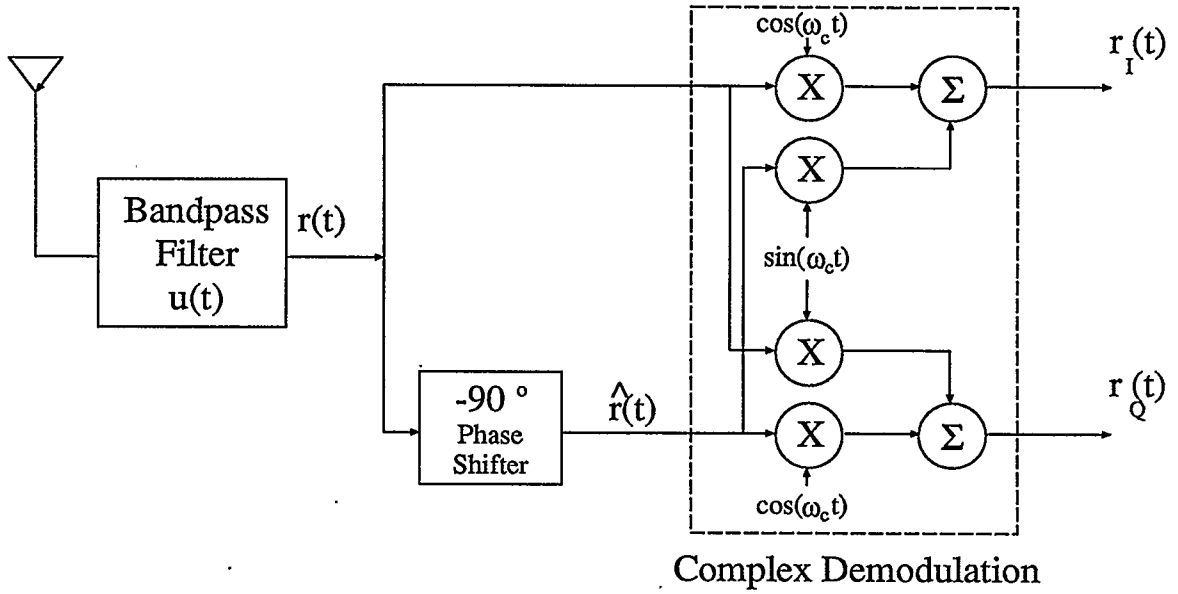


Figure 3.3 Analog complex demodulator

$r(t)$, combined with the -90° phase shifted version $\hat{r}(t)$, gives the analytic signal

$$r(t) + j\hat{r}(t) = \tilde{r}(t)e^{j\omega_c t} \quad (3.10)$$

Multiplying by $e^{-j\omega_c t}$ complex demodulates the signal giving $\tilde{r}(t) = r_I(t) + jr_Q(t)$. In this case, the lowpass filters of the conventional system are not required. This is advantageous since the lowpass filter may cause unwanted distortion of the received pulse, but now two additional multipliers are required.

In practice, the system is usually implemented with two bandpass filters as shown in Fig. 3.4. The two bandpass filters have identical gain and group delay characteristics, but bandpass filter #2 has an additional constant phase of -90° .

3.4 Bandpass Sampling

If the received narrowband signal is uniformly sampled at bandpass at a sampling rate $f_s = 1/T_s$ Hz, then the sample values at the output of the bandpass filter are

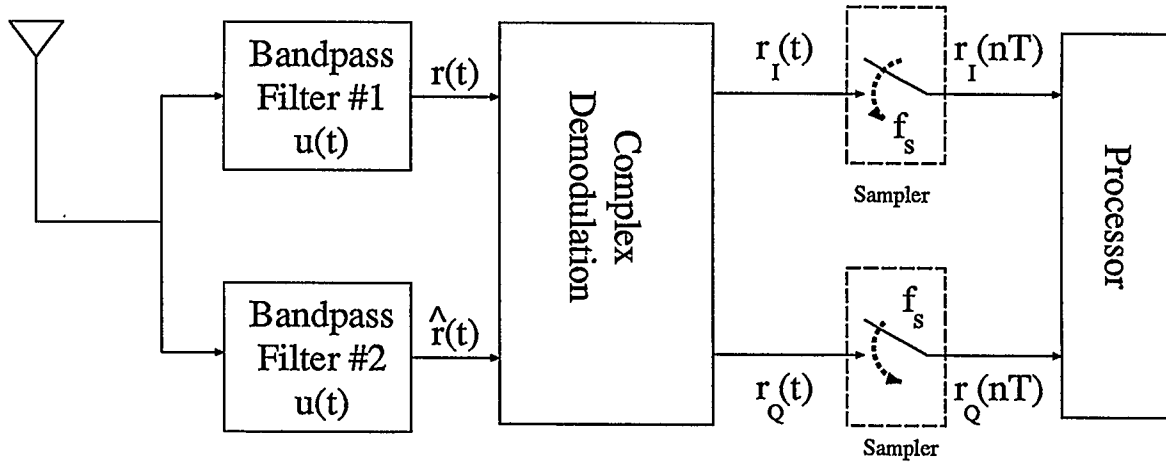


Figure 3.4 Practical analog complex demodulator

given by

$$r(nT_s) = \text{Re} \left\{ \hat{r}(nT_s) e^{j\omega_c nT_s} \right\} \quad (3.11)$$

In this case, the digital demodulation structures shown in Fig. 3.5 and 3.6, are the same as the analog structures, provided the received signal is sampled at the Nyquist

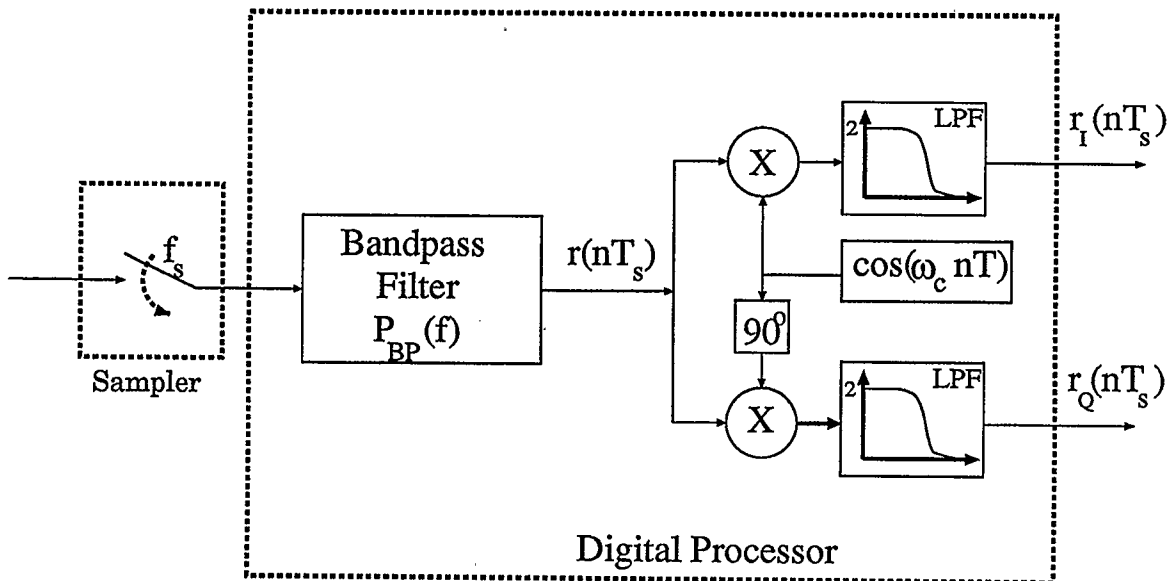


Figure 3.5 Quadrature down conversion with bandpass sampling.

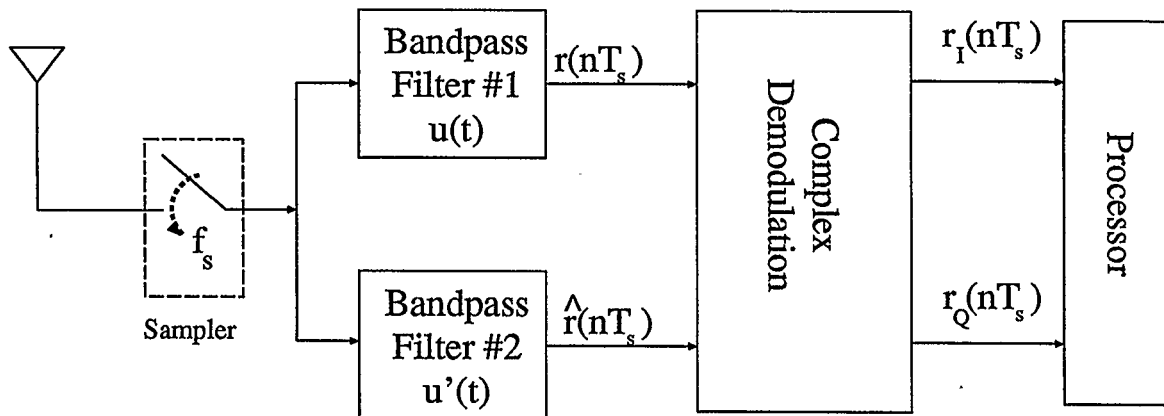


Figure 3.6 Complex demodulation with bandpass sampling.

rate.

The complex demodulator structure is particularly convenient since complex operations are easily implemented using digital processors.

Moving the sampling to bandpass has the immediate benefit of one less sampler. In hardware this translates to one less S/H and A/D converter. The other benefit is that the multipliers and SRN matched bandpass filters can now be implemented digitally, thus taking advantage of the high degree of linearity possible with digital components. This also insures that the phase of the quadrature oscillators will be exactly 90° apart. Lodge [26] noted that the degree of linearity required by the bandpass A/D is similar to that for analog components (ie. radio frequency (RF), and intermediate frequency (IF) amplifiers, mixers, etc.) in the receive chain. Whereas lowpass sampling requires a higher degree of linearity.

Not only is the reduction of analog down-conversion components possible, but it may also be possible to reduce the number of IF components by sampling the signal at higher carrier frequencies. The downside is that in order to meet the Nyquist sampling criteria, the S/H and A/D must sample at a very high rate. At present, the

availability of fast yet low power consumption S/H and A/D's is limited [26].

3.5 Quarter Wave Sampling

The use of bandpass sampling to reduce the number of analog components in the receiver is described in the previous section (Sec. 3.4). This of course, is at the expense of an increased burden on the digital signal processing facilities. A technique highlighted by Lodge [26], greatly simplifies the processor's computations if the received signal is sampled at four times the carrier frequency (referred to as *quarter wave sampling*). In practice, this may not be possible if the carrier frequency is too high because of present limitations on A/D converter rates.

For quarter wave sampling of the complex demodulator in Fig. 3.6 becomes

$$e^{j\omega_c n T_s} = e^{j2\pi n/4} = c_n + j s_n, \quad (3.12)$$

where c_n and s_n only take values of 0 and ± 1 . No multiplications are required in the complex demodulating, only sign inversion and addition.

It will be shown in the next section that demodulation occurs naturally if filtering with half-band filters and decimation chains are incorporated with *quarter wave sampling*.

3.5.1 Halfband Filtering and Decimation

A sampled signal can be mathematically expressed as the multiplication of the continuous time signal by a periodic impulse train,

$$t(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s), \quad (3.13)$$

where $\delta(t)$ is the Dirac delta function. The sampled signal, $s_s(t)$

$$s_s(t) = \sum_{n=-\infty}^{\infty} s(nT_s)\delta(t - nT_s) \quad (3.14)$$

has the Fourier transform

$$S_s(j\Omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} S(j\Omega - nj\Omega_s), \quad (3.15)$$

where $\Omega_s = 2\pi/T_s$.

For the bandpass signal shown in Fig. 3.7 a), the sampled spectrum is given by

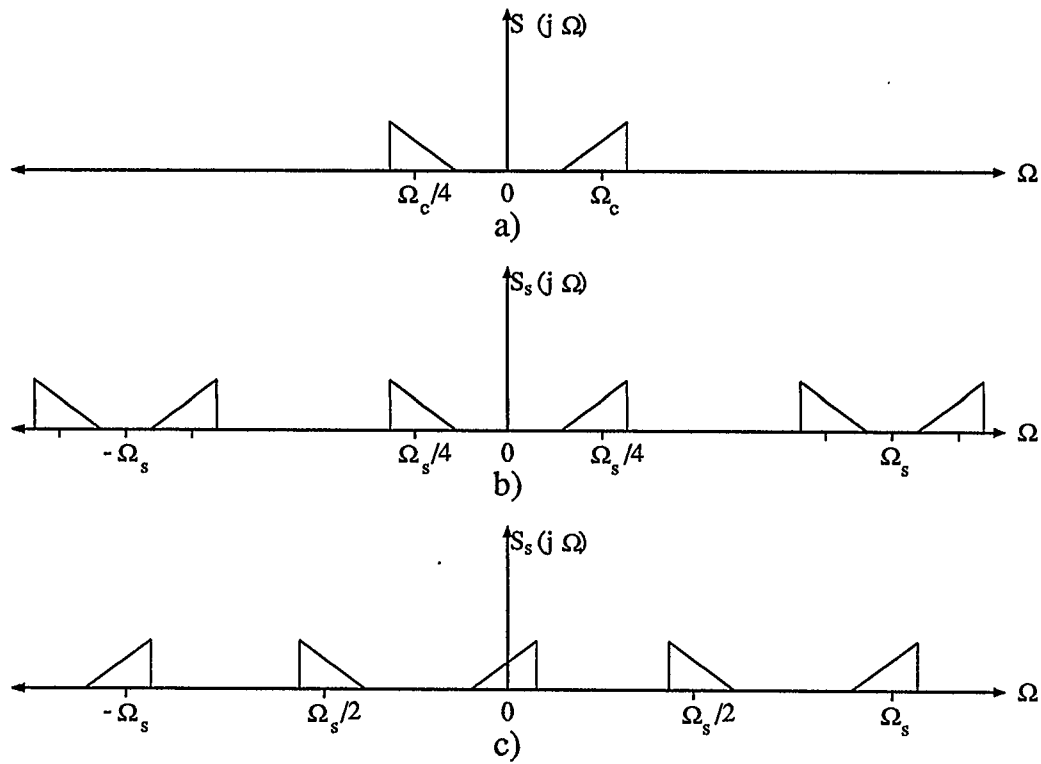


Figure 3.7 Spectral illustration of complex demodulation: a) bandpass signal, b) sampling at $f_s = f_c$, and c) multiplying by $e^{j\omega_c t}$.

Fig. 3.7 b) when $\Omega_s = 4\Omega_c$. Complex demodulating translates the spectrum shown in Fig. 3.7 b) to the left by Ω_c giving Fig. 3.7 c).

A down-converter using 1/2 band filters and decimation is shown on Fig: 3.8. It is shown that this system is equivalent to complex demodulation. A brief description

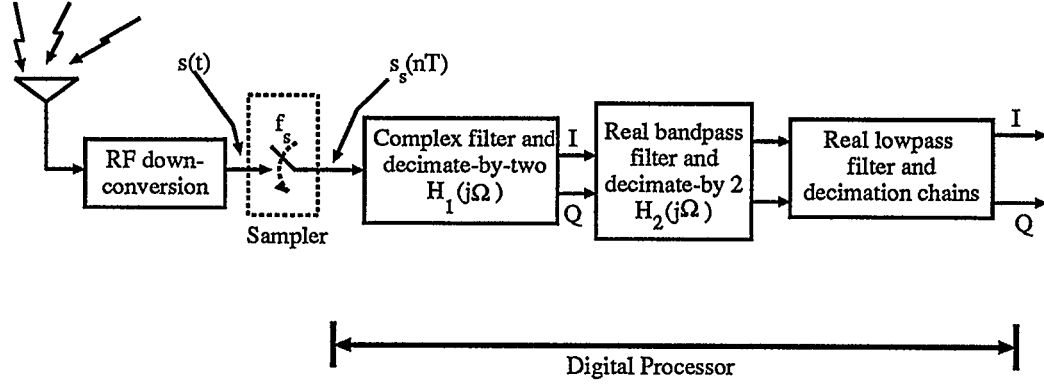


Figure 3.8 Quadrature down-conversion using filter and decimation chains.

of the system is shown below. For more details on variable rate systems, the reader is referred to DeFatta et. al. [27].

The filter $H_1(j\Omega)$ removes negative spectral components located at $-\Omega_s/4$ and their periodic repetitions. This avoids the aliasing of these components when the filter output is decimated by two. This is shown in Fig. 3.9. In the filtering technique presented by Lodge [26], the first stage complex filter, $H_1(j\Omega)$, removes the lower image spectra of $S_s(j\Omega)$. The image (referring to the positive frequency axis) is removed so that the following decimation-by-two stage does not alias the lower image into the desired upper spectra. An example of this is aliasing that will occur when the spectrum at $\Omega_s + \Omega_s/4$ is translated to $\Omega_{s2} + \Omega_{s2}/2$ (Fig. 3.9 b)).

Next, the real bandpass filters, $H_2(j\Omega)$, are used to filter frequency bands near integer values of the sampling rate Ω_{s2} . Frequency components in these bands are predominantly related to quantization noise since front-end RF or IF bandpass analog filters typically remove the out-of-band channel noise. If the spectral density is

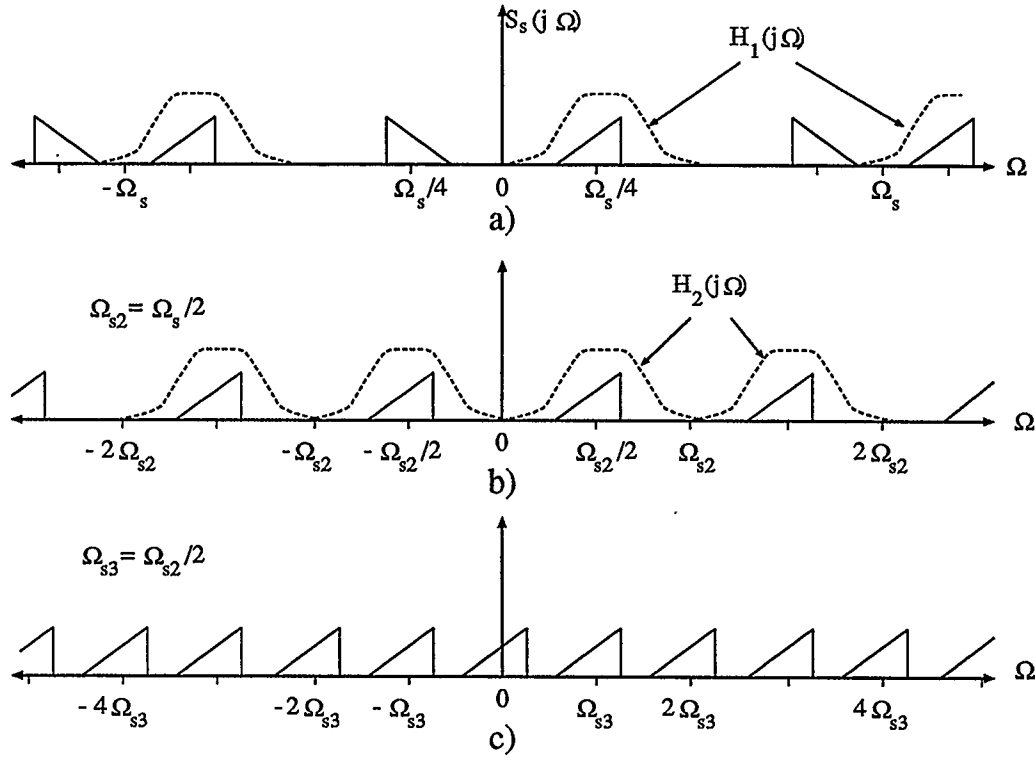


Figure 3.9 Spectral illustration of downconvert and decimate-by-4:
a) bandpass sampled signal and complex filter spectrum;
b) decimate-by-2 spectrum and real half-band filter; c)
spectrum decimated to baseband

assumed to be flat (Chapter 5 Sec. 5.2.1.1), each stage of halfband filtering would reduce the noise power by 3 dB. The second decimate-by-two translates the sidebands to integer multiples of the sampling rate Ω_{s3} .

If the final sampling rate at the output of the filter and decimate-by-4 chain is much larger than the bandwidth of the signal, then it would be possible further reduce the sampling rate by using additional cascaded sections of lowpass filters and decimation chains, thereby reducing quantization noise power and decreasing the sampling rate.

Halfband filters presented by Lodge [24] alleviate the processing burden with efficient digital structures. An efficient design of halfband filters is presented in

Sec.3.5.2.1.

3.5.2 Halfband Filters

3.5.2.1 Complex and Real Bandpass Filters

The half-band filter mentioned in Sec. 3.5.1 describes any filter response, ie. square-root Nyquist, Butterworth, etc., that has a bandwidth defined as $1/4$ the sampling rate. Fig. 3.10 shows the desired and aliased signal spectrums when the signal's BW is small relative to the sampling rate, and 2) when the signal's BW is

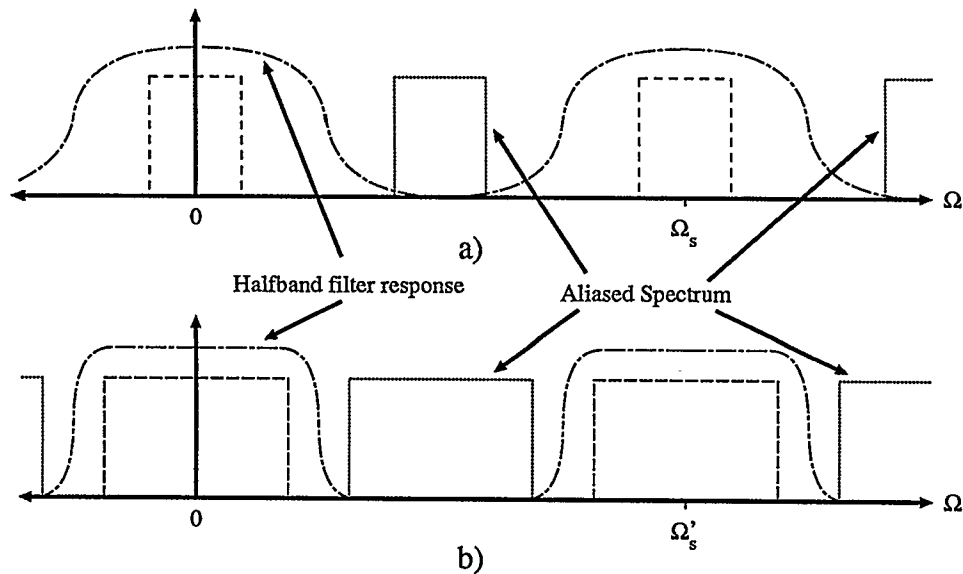


Figure 3.10 Halfband filtering to prevent aliasing: a) signal bandwidth small relative to sampling rate and b) signal bandwidth large relative to sampling rate

large relative to the sampling rate. In both cases, the halfband filters are used to remove the aliased spectra with minimal distortion of the desired sideband. It is obvious that the order of the filter in Fig. 3.10 b) will have to be larger than that of Fig. 3.10 a). In cascaded filter and decimate-by-2 chains, it may be necessary

to increase the order in downstream filters as the bandwidth of the signal increases relative to the sampling rate. Fortunately, the decreased sample rate will allow for more time to complete the computations of higher order filters.

Due to their simplicity in design and linear phase, Lodge proposes the use of finite impulse-response (FIR) filters of the form

$$H(z) = 1 + \sum_{k=1}^{(N+1)/4} h_{2k-1}(z^{-2k-1} + z^{2k-1}); N = 3, 5, 7, \dots \infty, \quad (3.16)$$

where N is the order of the FIR filter. A filter of this type requires $N - 1$ multipliers; however, the number of multipliers can be reduced to $(3N - 1)/4$ due to the symmetry of coefficients¹. Goodman [10] showed that several filter stages are more efficient than using a single filter to change sampling rates.

Of particular interest, is the third order 100% raised cosine FIR filter with z transform function

$$H(z) = \frac{1}{2}z^{-1} + 1 + \frac{1}{2}z^1 \quad (3.17)$$

The coefficients form trivial multiplications: the $1/2$ multiplicand is simply a binary shift of one bit. The only hardware necessary will be the implementation of the adder. This not only reduces chip area, but with less multipliers, it may also be possible to pipeline the addition structures for higher data rates, ie. in the case of the real halfband filter, a three port adder could be implemented instead of scheduling additions through a two port adder.

The complex halfband filter $H_1(z)$ is realized applying the Fourier transform shifting property to eqn. 3.17 at a shifting frequency of $f = \frac{1}{4}f_s$,

$$H_1(z) = H(ze^{j2\pi/4}) \quad (3.18)$$

¹It will be shown later in Sec. 5.4.5.2 that using $(3N - 1)/4$ multipliers results in a decrease in the dynamic overflow range by a factor of 2.

$$= -j\frac{1}{2}z^{-1} + 1 + j\frac{1}{2}z^1 \quad (3.19)$$

This filter is positioned as shown on Fig. 3.9 a).

The next step is to design the real halfband filter $H_2(z)$. Applying the shifting property again with $f = \frac{1}{2}f_s$ gives

$$H_2(z) = H(ze^{j2\pi/2}) \quad (3.20)$$

$$= -\frac{1}{2}z^{-1} + 1 - \frac{1}{2}z^1 \quad (3.21)$$

As expected, this filter is positioned as shown on Fig. 3.9.

The digital realization of eqn. 3.19 and 3.21 with decimation chains is shown in Fig. 3.11. As expected, the three FIR's have multiplicands that form trivial multiplications; two three port adders, 1 two-port adder, and three memory registers constitute the majority of the hardware requirements.

Assuming that the quantization noise from the A/D converter (see Sec. 5.2.1) is spectrally flat over the sampled spectra, then each of the halfband filtering stages will reduce the quantization noise by 3 dB, which effectively translates to a 1/2 bit/stage increase in the signals binary word length.

The lowpass half power point for the above filters is $0.182f_s$. A signal's spectrum of passband bandwidth $BW = 2(0.182f_s)$, would have an unacceptable stopband attenuation at $\frac{1+BW}{2}$ of -4.6 dB, ie. the attenuation of the upper and lower frequencies of the lower sideband. For a more acceptable attenuation, ie. 20 dB, the signal's lowpass bandwidth would be approx $\frac{1}{15.5}f_s$. Obviously, the signals bandwidth must be small relative to the sampling or severe aliasing will occur; not to mention the distortion of the desired signal before aliasing.

Should distortion become a problem, there are two alternatives: 1) increase the order of the halfband filter, or 2) predistort the transmitted signal so that distortion

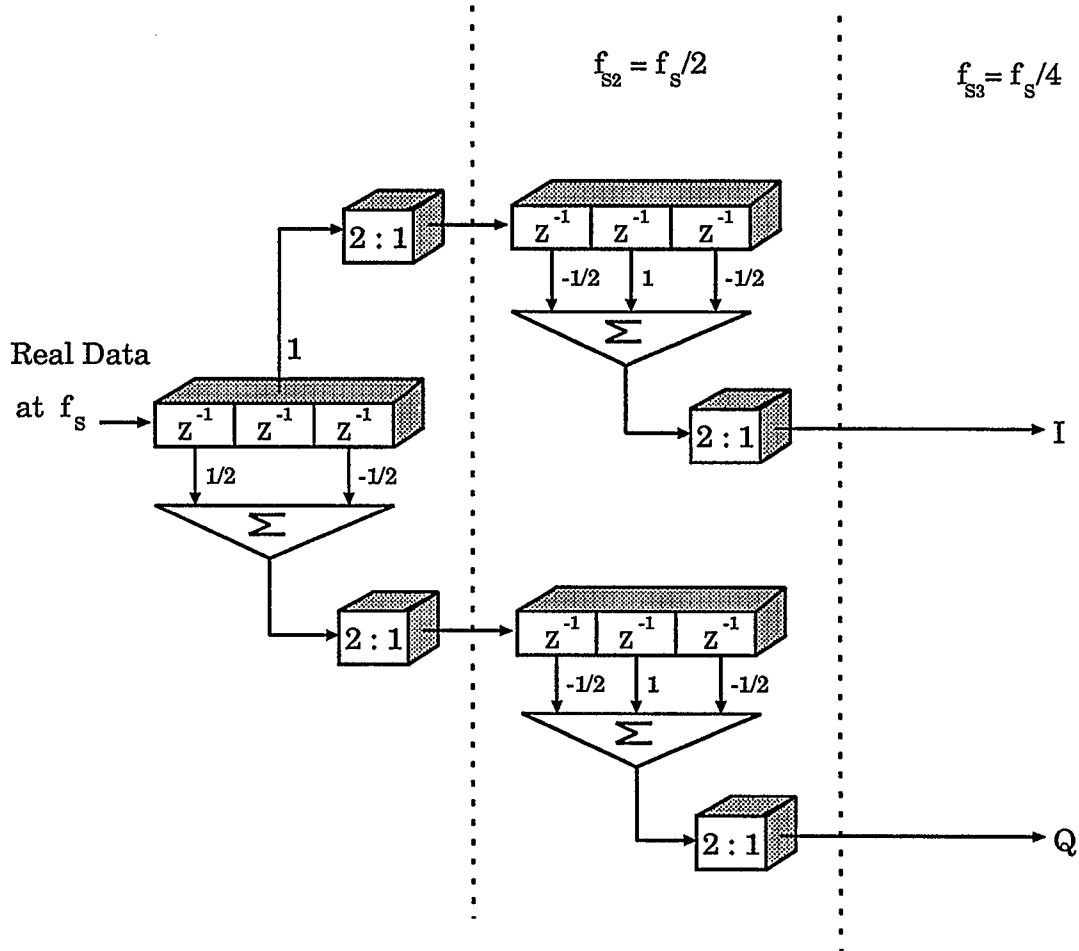


Figure 3.11 Digital structures for complex and real filters, and decimators

caused by the halfband filters only acts to recover to desired spectrum. The latter was used in the thesis receiver since predistortion can be implemented at the transmitter by simply changing the filter coefficients.

3.5.2.2 Development of an Equivalent Baseband Filter System

The purpose of this derivation is to develop an equivalent baseband filter representation of $H_1(z)$ and $H_2(z)$. The baseband equivalent filter system is used in a computer simulation discussed in (Sec. 4.5). This eliminates the need to modu-

late/demodulate the transmitted BPSK signal.

Fig. 3.12 a) shows the halfband filter blocks developed in Sec. 3.5.2.1 In this

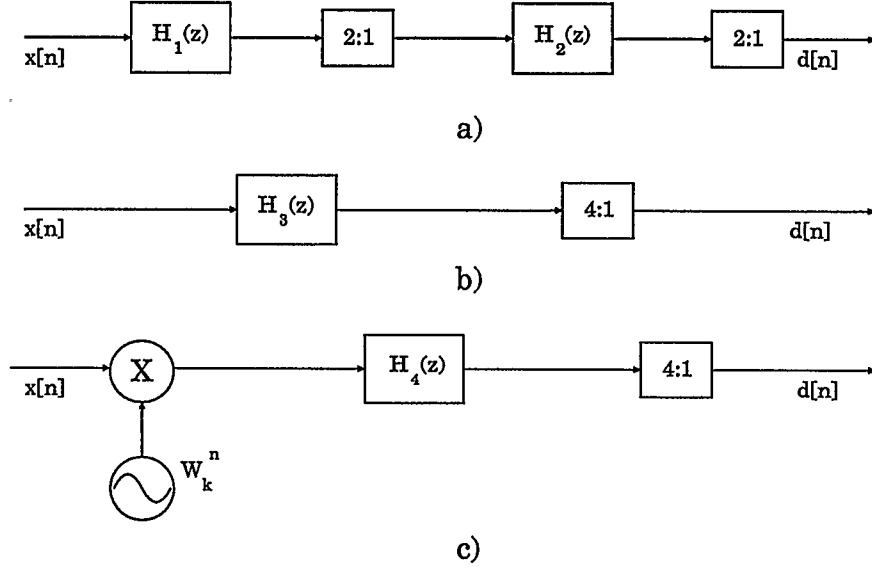


Figure 3.12 Decimation blocks for; a) complex and real filters, b) single complex filter, and c) complex demodulator and real filters

figure, the discrete samples of $s_s(nT_s)$ are defined as

$$x[n] = x(nT_s); -\infty < n < \infty \quad (3.22)$$

Using the z-transform, it can be shown that the transfer functions of the equivalent system shown in Fig. 3.12 b) and c) are given by

$$H_3(z) = H_1(z)H_2(z^2) \quad (3.23)$$

and

$$H_4(z) = H_3(zW_4) \quad (3.24)$$

respectively, where $W_4 = e^{-j\frac{2\pi}{4}}$ is the complex demodulation term. Substituting eqn.

3.19 and 3.21 into eqn. 3.23 and 3.24 gives the filter's transfer function

$$H_3(z) = -2z^2 + 4 - 2z^{-2} + j(-z^3 + 3z - 3z^{-1} + z^{-3}) \quad (3.25)$$

and

$$H_4(z) = z^3 + 2z^2 + 3z + 4 + 3z^{-1} + 2z^{-2} + z^{-3} \quad (3.26)$$

The filter and decimate-by-4 chain in Sec. 3.5.2.1 can now be implemented as a baseband equivalent filter (eqn. 3.26) with frequency response

$$H_4(f) = H_4(z)_{z=e^{j2\pi f}} \quad (3.27)$$

$$= \frac{\sin^2(4\pi f)}{16\sin^2(\pi f)}, \quad (3.28)$$

where $H_4(f)$ has been normalized to unity gain at $f = 0$.

Fig. 3.13 a) shows the complex demodulated sample spectrum (normalized frequency) and the baseband filter response, $H_4(f)$. The x-axis is the frequency response normalized to f_s . Before decimation, the sideband centered at $f_n = 1/2$ must be removed in order to prevent aliasing. This is illustrated by the gray regions on Fig. 3.13 b), where decimation has caused the upper sideband to be contaminated with the aliased lower sideband that was partially removed by $H_4(f)$.

Equation 3.29 is used to calculate the sideband attenuation as

$$H_4\left(\frac{1 \pm BW}{2}\right) = -10\log\left(\frac{\sin^2\left(2\pi\left(\frac{1 \pm BW}{2}\right)\right)}{16\sin^2\left(\pi\frac{1 \pm BW}{2}\right)}\right), \quad (3.29)$$

where $\frac{1-BW}{2}$ and $\frac{1+BW}{2}$ are the upper and lower frequencies of the lower sideband spectrum, and BW is the passband bandwidth. To attenuate stopband frequencies to 20dB, the signals bandwidth must be less than $\frac{1}{15.5}f_s$. The passband ripple for the

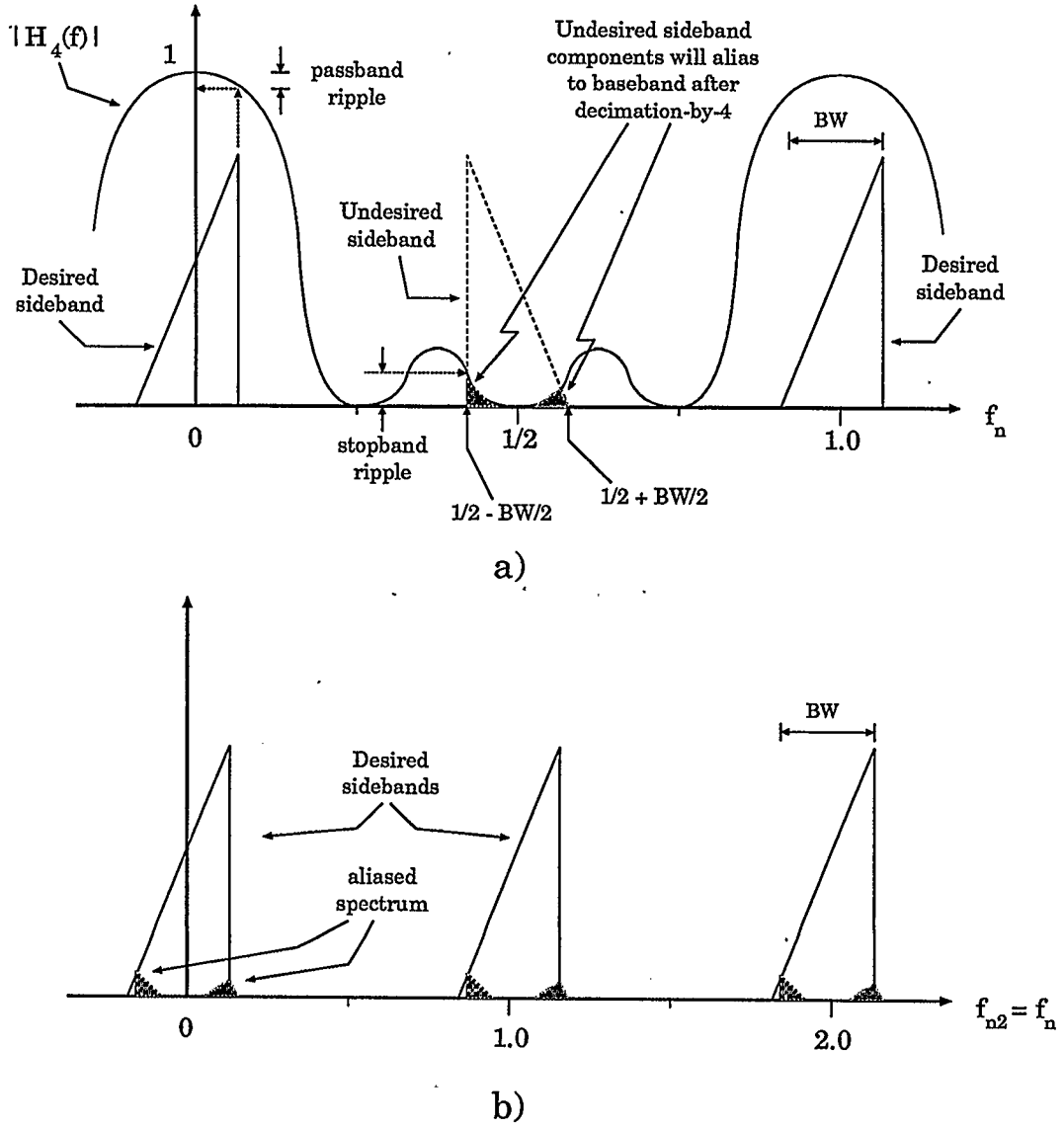


Figure 3.13 Baseband frequency response: a) signal spectrum and filter response after complex demodulation; b) signal spectrum after filtering and decimation

baseband filter $H_4(f)$ is given by

$$H_4\left(\frac{\pm BW}{2}\right) = -10 \log \left(\frac{\sin^2\left(2\pi \left(\frac{\pm BW}{2}\right)\right)}{16 \sin^2\left(\pi \frac{1 \pm BW}{2}\right)} \right) \quad (3.30)$$

which represents a 0.22 dB attenuation at $BW = \frac{1}{15.5} f_s$.

The calculated passband ripple and stopband attenuations for both the baseband filter derived above and a single 100% raised cosine filter (Sec. 3.5.2.1), substantiate the need for the bandwidth to be much less than f_s . The other alternative is to increase the order of the halfband filter. The cost in terms of hardware is either the increased chip area needed for the non-trivial multiplications of higher order filter coefficients, or faster A/D's and upstream digital hardware to accommodate the a higher sampling rate.

3.6 Summary

In this chapter, a bandpass sampling technique was presented to reduce the number of analog components in a standard quadrature receiver. In transferring the analog functions to digital, it becomes necessary to implement efficient digital structures to minimize processing load.

Sampling the bandpass signal at four times its center frequency results in trivial multiplications of the cosine/sine terms. For the quadrature down-converter, it simplifies the multiplier structure.

With *quarter wave sampling*, it is possible to alias the bandpass signal to baseband using half-band filter and decimate-by-two chains. The chains - consisting of complex and real filters - attenuate unwanted sideband spectra (they will also reduce out-of-band quantization noise power as shown in Chapter 4. Their design can be implemented with simple digital structures.

To prevent distortion of the desired signal and adequately attenuate the aliased bands (20 dB attenuation), it is necessary that the signals bandwidth be at least $1/15.5^{th}$ the sampling frequency when 100% raised cosine filters are used².

²It will be shown later in Chapter 4 Sec. 4.3.2.3 that passband ripple distortion can be overcome

An equivalent baseband filter was derived for the computer simulation model. Its transfer function shows similar signal bandwidth requirements to assure sufficient attenuation of aliased sidebands.

CHAPTER 4

PROCESSING BLOCK DESCRIPTION OF THE INTEGRATED DIRECT-SEQUENCE SPREAD-SPECTRUM RAKE RECEIVER

4.1 Introduction

This chapter presents the processing blocks of the direct-sequence spread-spectrum (DS-SS) transmitter/receiver pair (commonly referred to as a transceiver). The discussion covers the communication theory and basic digital structure design of the processing blocks without getting into in-depth topics related to lower-level digital architectures - a discussion that is better pursued in conjunction with the hardware implementations in Chapter 5. Finite arithmetic effects - such as quantization noise, coefficient quantization, finite word length, and overflow - are also described later in Chapter 5.

The chapter begins by first discussing the modulation scheme and the pseudo-noise code length imposed by the design constraints of the transceiver. With this defined, a spread-spectrum (SS) transmitter is realized which employs square-root Nyquist (SRN) pulse shaping and interpolation. Next, a receiver that matched filters the SRN pulse and the PN code, and coherently combines the multipath signal using the time diversity Rake, is discussed. Finally, a computer simulation model of the hardware design is described.

4.2 Constraints

In May 1991, Novatel staff met with the author and supervisor, Dr. S.T. Nichols, to discuss Rake receivers. From this discussion, it was noted that the proposed spread-

spectrum system would occupy a 1.25 MHz channel in the 902 MHz to 928 MHz ISM outdoor cellular band. The minimum bit rate was set at 8 kbps for linear predictive coded voice. With these constraints, it was decided that the minimum data rate could be achieved by modulating a 127 chip M-sequence code with binary phase-shift-keying (BPSK) data. In practice, however, Kasami or Gold sequences would be used since the cross-correlation properties of the 18 different 127 chip M-sequences have poor cross-correlation properties [46] limiting the number of users per channel. With 127 chips, the aggregate data rate would be 9.84 kbps; allowing for a protocol overhead of 1.84 kbps.

The hardware constraints were 1) the availability of Xilinx¹ field-programmable gate-arrays (FPGA's) for prototyping, 2) the maximum clock rate of the Xilinx chips, and 3) the conversion rate of the analog-to-digital converter.

A preliminary gate count showed that a 127 chip M-sequence would barely leave enough Xilinx FPGA's for design contingencies. In order to reduce the gate count, it was decided that a 31 chip sequence would be used for the prototype since it possessed the same number of M-sequences, more preferred sets, better cross-correlation properties, and half the memory requirements as the next lower 63 chip M-sequence [46]. The drawback to using a smaller M-sequence is a lower processing gain (Chapter 2 Sec. 2.3.1). The final design should, however, allow the code to be easily increased to 127 - or for that matter, any other length.

Past experience with the Xilinx prototyping modules² has shown that the maximum clock rate achievable without errors is 20 MHz. Although the internal toggle rate of a single chip is higher (approx. 50 MHz), the rate appears to be limited by

¹Xilinx is a trademark of Xilinx Inc. which manufactures commercial FPGA's

²The modules were developed by Dr. L. Turner and P. Graumann at the University of Calgary, and are individual printed circuit boards (PCB) that carry a single Xilinx chip

external factors such as board layout, ground planes, coupling, etc.. Since the design incorporates several chips with distributed clock signals, it was decided that to ensure successful toggling, the maximum sampling rate would be 10 MHz. This would also relax the requirements on the A/D converter.

The sampling rate, after the decimation chain, was determined to be twice the Nyquist rate or $2f_{chip} = 2.5\text{MHz}$. This meant that only the filter and decimate-by-4 chain could be used; it would not be possible to use several more cascaded sections of lowpass filter-and-decimate chains to further increase the signal-to-quantization noise ratio (SQNR) (Sec. 3.5.1) .

Past experience also showed that the Xilinx routing resources favoured the use of bit-serial architectures. Implementation of large algorithms in bit-serial resulted in up to 90% usage of Xilinx configurable logic blocks (CLB's), whereas equivalent parallel designs achieved up to 60%. Since the size of bit-serial structures are an order of magnitude less than parallel designs, more bit-serial arithmetic logic units (ALU's) could be used in an algorithm, thereby greatly reducing the complexity of the design. Also, the bit serial designs are inherently optimal for "stuck-at-low" fault scan testing. The drawback to bit serial designs is that the clock rate increases linearly with the data word size. For a maximum clock rate of 20 MHz, and a minimum sampling rate of 2.5 MHz (two samples / chip), the maximum data word size is 8 bits. The computer simulation showed that 8 bit words provided sufficient dynamic range to prevent non-ideal quantization noise from becoming a hindrance to performance.

4.3 The Direct-Sequence Spread-Spectrum Transmitter

The transmitter block diagram is shown on Fig. 4.1. The data, whose input can be selectively programmed by a MUX, is differentially modulated to overcome phase

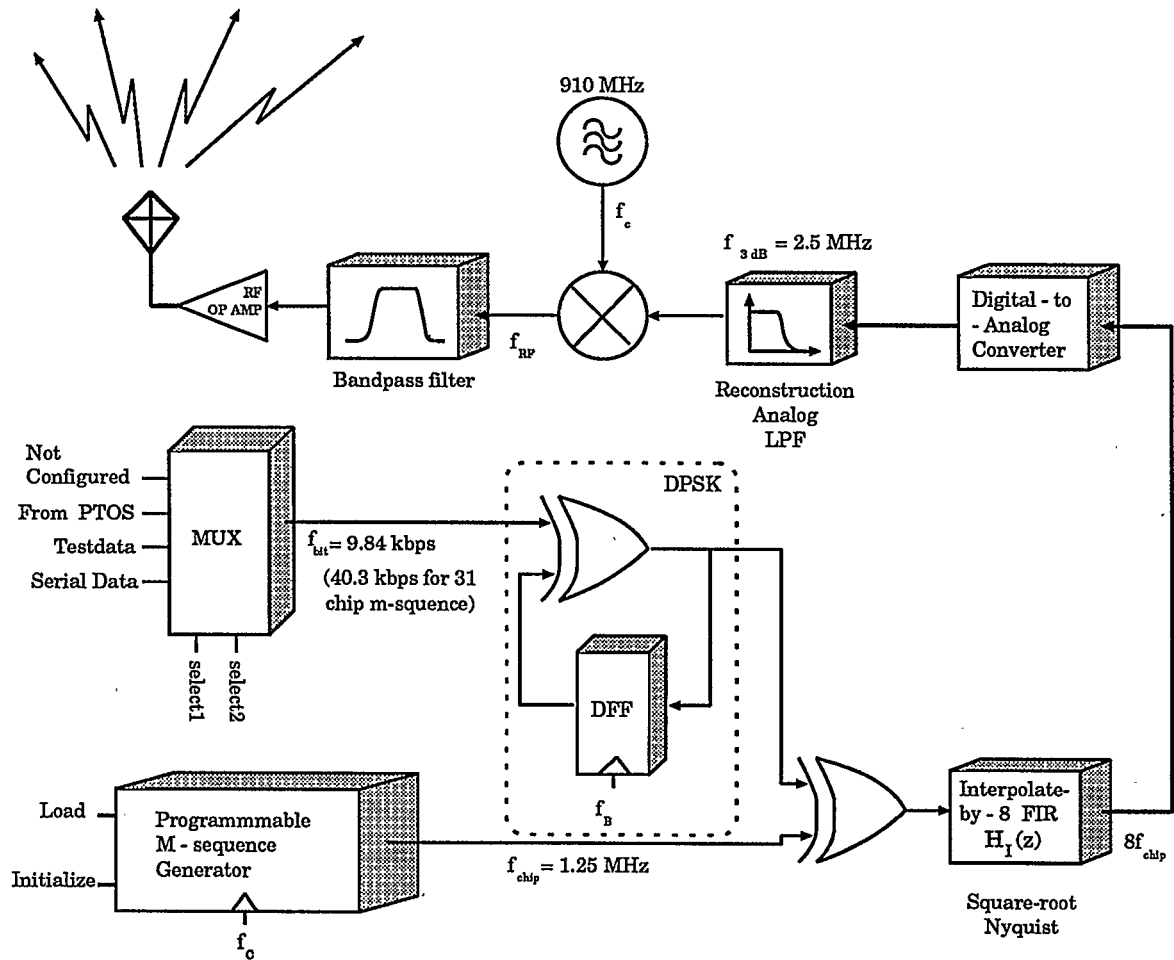


Figure 4.1 A binary differential phase-shift keying spread-spectrum transmitter.

ambiguities at the receiver. Phase ambiguities are introduced by non-synchronized transmitter/receiver carriers and the multipath channel. It will be shown later in Sec. 4.4.4 that the demodulation of differentially encoded data provides coherent combining of multipath echoes.

A modulo-2 adder (exclusive-or (XOR) gate) is used to modulate the data with the M-sequence coming from the programmable pseudo-noise (PN) generator. Modulation by a 31 chip M-sequence spreads the data's bandwidth (Chapter 2 Sec. 2.3.1) of the signal from 40.3 kHz to 1.25 MHz. The modulated data is then passed through

a square-root Nyquist (SRN) interpolate-by-8 filter. This filter bandlimits the spectrum to 625 kHz and increases the sample rate to 10 MHz, before it is converted to an analog signal, lowpass filtered and mixed to a radio frequency (RF) of 910 MHz.

4.3.1 Pseudo-Noise Code Generator

The pseudo-noise generator can be defined in terms of the binary generator polynomial

$$g(x) = \sum_{m=0}^M g_m x^m \quad (4.1)$$

where M is the degree of the polynomial, $g_0 = g(M) = 1$ and the other g 's take the terms 0 or 1. The polynomials are conventionally referred to in octal form. For example, the octal notation 36, represented in binary as 10110, would be the polynomial $x^4 + x^2 + x + 1$. The binary M-sequence, b , can be generated from $g(x)$ using a shift register given by Sarwate [46].as

$$b_{j+M} = g_M b_j \oplus g_{M-1} b_{j+1} \oplus g_{M-2} b_{j+2} \oplus \dots \oplus g_1 b_{j+M-1} \quad (4.2)$$

where $\oplus$ denotes the modulo 2 addition (or exclusive or (XOR)).

There exists a set of polynomial $g(x)$ that outputs the maximum length periodic sequence with period $N_c = 2^M - 1$. Of particular interest are the primitive polynomials. Examples of the primitives (in octal notation) for a fifth order polynomial are 45, 75, and 67 [45]. These polynomials possess good cross-correlation and out-of-phase autocorrelation properties [46], but only a small fraction of the M-sequences are primitives. For telecommunication systems such as CDMA, a better choice of code would be the Kasami codes [46] [30].

Another characteristic of the M-sequence is that it will generate a single sequence of 1's that equal the order, M , of the polynomial. This was used in generating a

timming pulse for input data latching.

Eqn. 4.2 can be implemented with simple shift registers shown in Fig. 4.2 [45]. For an M-sequence of length N_c , the propagation delay through the generator in Fig.

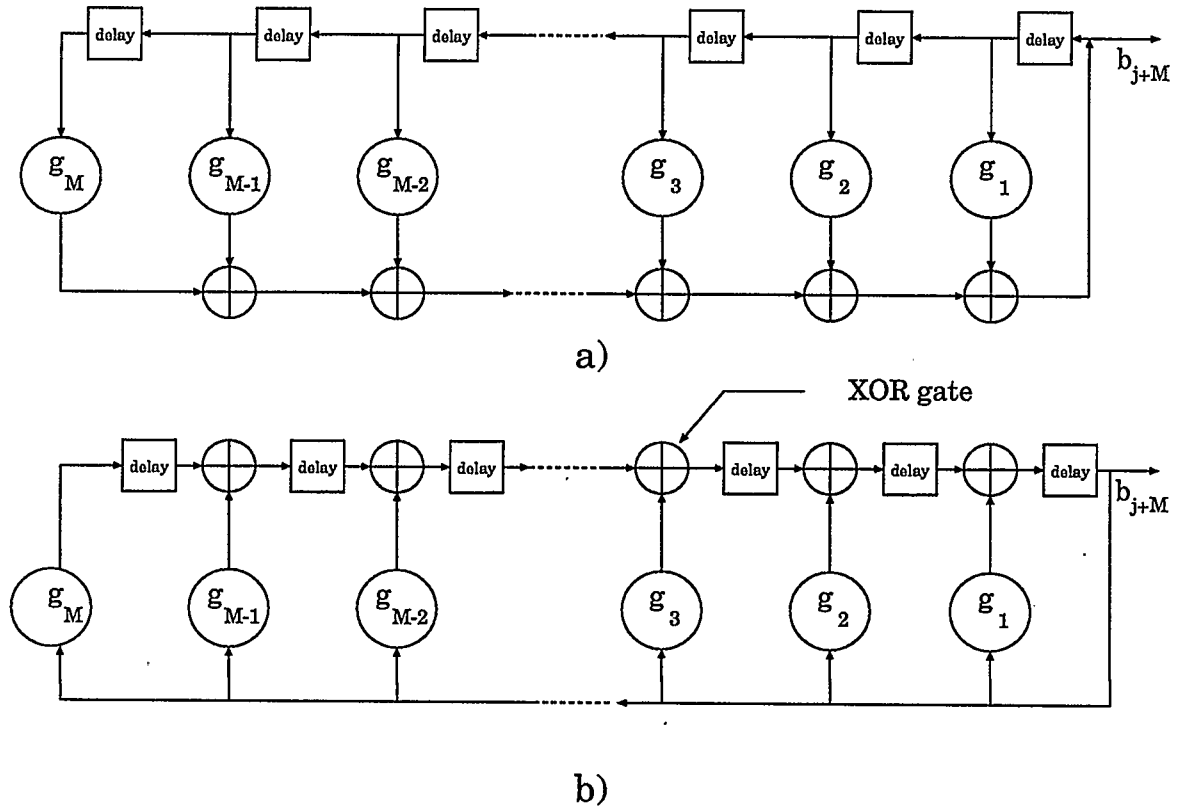


Figure 4.2 Two linear feedback shift registers: a) low speed and b) high speed.

4.2 a) is $\log_2(N_c + 1) - 1$ XOR gates, whereas the generator on Fig. 4.2 b) has a propagation delay of only one XOR gate regardless of the M-sequence length N_c . In high speed applications with large chip sequences, the latter design is definitely more desirable.

4.3.2 Transmit Filters

The transmitter shown in Fig 4.1 shows the modulated M-sequence entering a filter at a sample rate of f_{chip} , and leaving it at a rate eight times f_{chip} . This type of filter is known as an interpolating filter. The purpose of the filter is 1) to reduce the rolloff requirement on the reconstruction analog lowpass filter (LPF), 2) to bandlimit the modulated M-sequence, and 3) to reduce intersymbol interference (ISI). A lower rolloff means that the analog LPF will not have to be a high order filter and therefore more economical to produce. Bandlimiting the signal prevents spectral spillover into adjacent channels and reduces the sampling rate of the receiver. Reducing ISI increases the noise margin, providing a greater immunity to additive noise.

4.3.2.1 Reconstruction Filters, D/A's, and Interpolating Filters

The D/A converter and analog LPF (Fig. 4.1) are a practical method of approximating the ideal reconstruction discrete-to-continuous (D/C) system shown on Fig. 4.3.

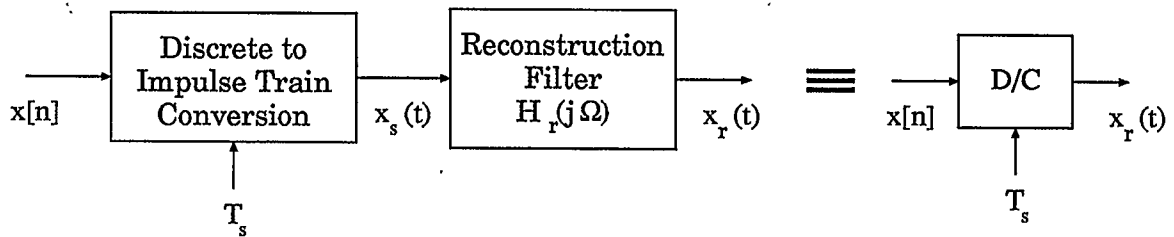


Figure 4.3 Ideal signal reconstruction.

Following the approach given by Oppenheim and Shafer [3], It can be shown that

the output of the ideal reconstruction filter, $H_r(j\Omega)$,

$$H_r(j\Omega) = \begin{cases} T_s, & |\Omega| < \pi/T_s = \Omega_s/2 \\ 0, & |\Omega| > \pi/T_s \end{cases} \quad (4.3)$$

is the reconstructed signal given by the well known sinc function interpolation

$$\begin{aligned} x_r(t) &= \sum_{n=-\infty}^{t=\infty} x[n] \frac{\sin[\pi \frac{(t-nT_s)}{T_s}]}{\pi \frac{(t-nT_s)}{T_s}} \\ &= \sum_{n=-\infty}^{t=\infty} x[n] \text{sinc}(2W(t - nT_s)), \end{aligned} \quad (4.4)$$

where $2WT = 1$, and T_s is the uniform sample spacing associated with the sample values $x[n]$. Consequently, $x_r(t)$ is equal to the original continuous-time signal if $x[n]$ is bandlimited with bandwidth Ω_x and the D/A operates at a rate

$$\Omega_s = \frac{2\pi}{T_s} \geq 2\Omega_x. \quad (4.5)$$

In the frequency domain, the ideal filter removes the high-frequency components of the sampled spectrum of $x_s(t)$ given by

$$X_s(j\Omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(j\Omega - nj\Omega_s) \quad (4.6)$$

(the sampled spectrum of eqn. 4.6 is similar to Chapter 3 Fig. 3.9). To completely remove the unwanted spectra, a very high ordered filter is needed to approximate the steep rolloff of $H_r(j\Omega)$. If a practical filter is employed, some leakage of the high frequency terms above $\Omega_s/2$ will occur - depending of course on the non-ideal filter's stopband attenuation. The passband ripple will also cause distortion of the desired spectra.

To reduce this leakage and distortion, a common approach is to split the reconstruction into two steps: a digital interpolation filter followed by a low order analog interpolation. The approach simply over samples the digital signal by a factor N_t ,

thereby providing a large rolloff region for the analog filter. The first step is to insert $N_T - 1$ zero's between each sample $x[n]$. A discrete ideal interpolating LPF (shown in Fig 4.4) "fills in" the zero values so that the interpolator's output, $x_i[n]$, has the same samples at an interpolated sampling rate $f_i = N_T f_s$.

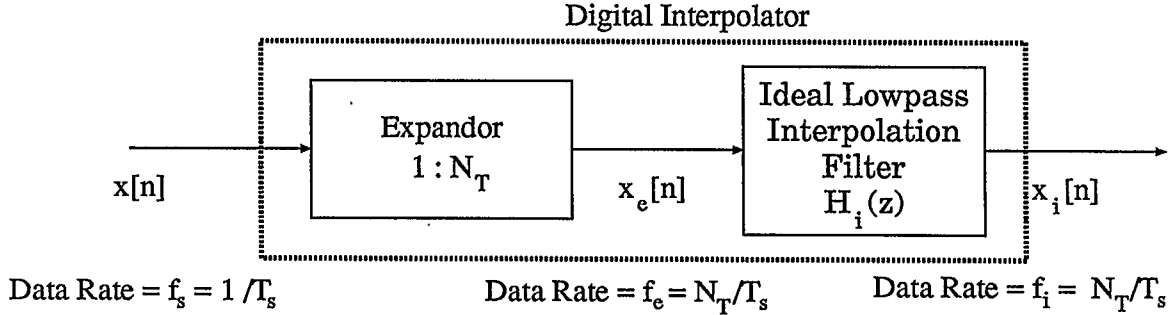


Figure 4.4 Interpolating structure for increasing the sampling rate by N_T .

The ideal interpolation filter transfer function, $H_i(z)$ where $z = e^{j\omega N_T T_s}$, is similar to the ideal reconstruction filter transfer function (eqn. 4.3) except that the gain is equal to N_T and stopband equal to $\frac{\pi}{N_T}$. As in the case of $H_r(j\Omega)$, the ideal filter $H_i(z)$ is unrealizable and must be approximated.

The z transform of a signal $x_e[n]$ with zeros inserted, has a repeating spectra at integer multiples of $2\pi/N_T$ given by

$$X_e(z) = X(z^{N_T}) \quad (4.7)$$

The ideal interpolating LPF then passes only the spectra at baseband and integer multiples of 2π . The output z -transform is given by

$$X_i(z) = H_i(z)X(z^{N_T}) \quad (4.8)$$

After digital filtering by $H_i(z)$, the interpolated signal spectra bandwidths are now small relative to the new sampling rate. This means that the reconstruction

filter used to approximate $H_r(j\Omega)$, can have a wider transition band (lower rolloff) with acceptable stopband attenuation and passband ripple. This results in a lower filter order for the analog reconstruction filter.

4.3.2.2 Square-Root Nyquist Interpolating Filter

It was shown in Chapter 3 Sec. 3.2, that the quadrature transmitted symbols are pulse shaped by the analog filter $p(t)$. This pulse is detected at the receiver using the bandpass matched filter $p_{BP}(t)$ with lowpass equivalent frequency response $P^*(f)$. The overall frequency response of the transmitter pulse shaping filter and receiver matched filter is

$$Q(f) = P(f)P^*(f) \quad (4.9)$$

$Q(f)$ is designed to bandlimit the signal's spectrum with zero ISI. Proakis [42] presents several waveforms that possess these spectral properties. A frequently used waveform in telecommunications is the Nyquist pulse (also known as the raised cosine pulse) shown on Fig. 4.5 and frequency spectrum shown on Fig. 4.6. The mathematical representation of $q(t)$ and $Q(f)$ can be found in [42] and [11].

Since $P(f)$ is the square-root of the Nyquist response $Q(f)$, the pulse wave form $p(t)$ is known as the SRN pulse and is given by

$$\begin{aligned} p(t) = & (1 - \alpha)\text{sinc}((1 - \alpha)2Bt) \\ & + \alpha \left[\cos(2\pi Bt - \frac{\pi}{4})\text{sinc}(2\alpha Bt - \frac{1}{4}) \right. \\ & \left. + \cos(2\pi Bt + \frac{\pi}{4})\text{sinc}(2\alpha Bt + \frac{1}{4}) \right] \end{aligned} \quad (4.10)$$

where B is the half-power Nyquist pulse bandwidth and α is the rolloff parameter. The specification of $\alpha = 0.35$ is chosen from the first generation digital cellular [1].

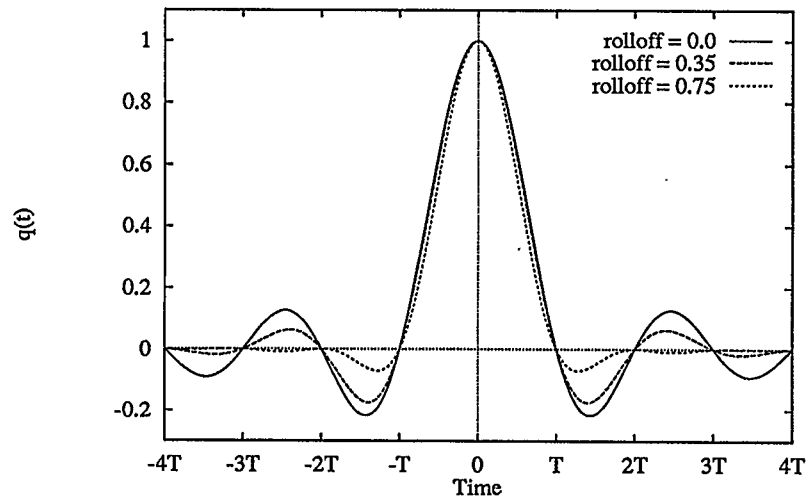


Figure 4.5 Raised cosine pulse for rolloff = 0.0, 0.34, and 0.75.

The Fourier transform of eqn. 4.10 is

$$P(f) = \begin{cases} \frac{1}{2B}, & |f| \leq B_1 \\ \frac{1}{2B} \cos \left[\frac{\pi(|f| - B_1)}{4\alpha B} \right], & B_1 < |f| \leq B_2 \\ 0, & \text{otherwise} \end{cases} \quad (4.11)$$

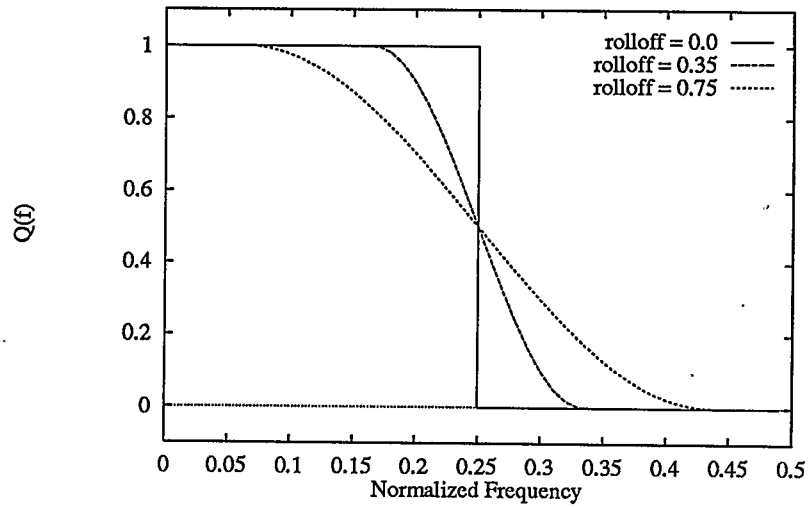


Figure 4.6 Raised cosine frequency response for rolloff = 0.0, 0.34, and 0.75.

where $B_1 = (1 - \alpha)B$ and $B_2 = (1 - \alpha)B$. The transmit digital filter taps, h_n , of a SRN finite-impulse response (FIR) filter are obtained by sampling the impulse response (eqn. 4.10) uniformly at a rate $f_s = 1/T_s$. The transmitter's pulse shaping filter and the receiver's matched filter will both have the same pulse shape due to the symmetry of $p(t)$.

Transmit filter taps		$B_N = \frac{1}{16}$	$\tilde{B}_N = 0.070161$
h_0	h_{30}	1	3
h_1	h_{29}	-1	2
h_2	h_{28}	-3	0
h_3	h_{27}	-6	-2
h_4	h_{26}	-7	-5
h_5	h_{25}	-8	-7
h_6	h_{24}	-6	-7
h_7	h_{23}	-3	-6
h_8	h_{22}	2	-3
h_9	h_{21}	8	3
h_{10}	h_{20}	16	11
h_{11}	h_{19}	24	20
h_{12}	h_{18}	31	29
h_{13}	h_{17}	38	36
h_{14}	h_{16}	42	41
h_{15}		43	43

Table 4.1 Filter tap values for 1) ideal bandwidth, and 2) predistorted bandwidth.

A rule-of-thumb for determining the number of filter taps, L , is

$$L = \frac{4}{2B_N} \quad (4.12)$$

where $B_N = \frac{B}{f_s}$ is the normalized bandwidth. For an interpolation-by-8 SRN filter, $B_N = \frac{1}{16}$, giving $L = 32$. A filter length of $L = 31$ was chosen to give an odd ordered filter. Table 4.1 shows the transmit FIR filter tap values (the notation $p(t)$ is replaced by $h_i(t)$ to be consistent with Sec. 4.3.2.1).

Referring back to Fig. 4.4, the expander and interpolating filter can be combined into one single block by computing only the tap values that do not have a zero at their input, ie.

$$\begin{aligned}
 y_{8n} &= x_n h_0 + x_{n-1} h_8 + x_{n-2} h_{16} + x_{n-3} h_{24} \\
 y_{8n+1} &= x_n h_1 + x_{n-1} h_9 + x_{n-2} h_{17} + x_{n-3} h_{25} \\
 y_{8n+2} &= x_n h_2 + x_{n-1} h_{10} + x_{n-2} h_{18} + x_{n-3} h_{26} \\
 &\quad * \\
 &\quad * \\
 &\quad * \\
 y_{8n+6} &= x_n h_6 + x_{n-1} h_{14} + x_{n-2} h_{22} + x_{n-3} h_{30} \\
 y_{8n+7} &= x_n h_7 + x_{n-1} h_{15} + x_{n-2} h_{23}
 \end{aligned} \tag{4.13}$$

In this way, the interpolating filter only requires four multiplications to compute each output. But, since the DS-SS transmitter (Fig. 4.1) input to the FIR takes values of ± 1 , no multiplications in fact are necessary.

4.3.2.3 Predistortion

Fig. 4.7 shows the distortion caused by the decimate-by-four chain on the SRN filter's frequency response. The decimate-by-four chain's passband ripple is -0.85 dB at the Nyquist pulse bandwidth of $B_N = \frac{1}{16}$. The effect of this seemingly negligible distortion was determined with the use of a baseband computer simulation (Sec. 4.5). By sending binary data through the interpolating filter, and viewing the detected pulse at the output of the cascaded decimate chain and square-root Nyquist matched filter (Sec. 4.4.2), the discrete values were plotted in repeating chip periods, T_c , to produce the eye diagram shown on Fig. 4.8. The ISI, shown as superimposed signals

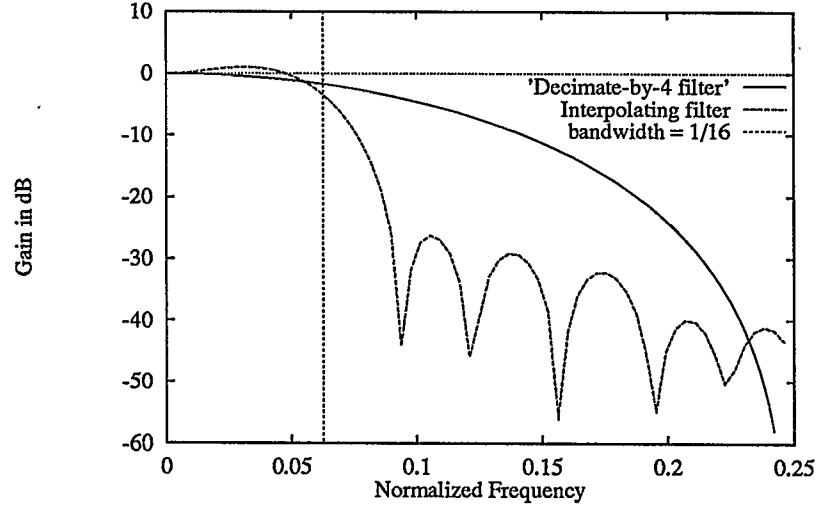


Figure 4.7 Distortion of square-root Nyquist frequency spectrum by the baseband equivalent of filter and decimate-by-four chain.

at the sampling instant, T_c , represents approx. 30% of the nominal sample value. This results in an appreciable decrease in the noise margin. The decreased distance between the sample and threshold value increases the probability that additive noise will cause an error.

To overcome the distortion, the interpolating filter is increased by an amount ΔB to account for the decimate chain's passband ripple. Eqn. 4.11 and baseband response of the filter and decimate chain (Chapter 3 3.28) are used to determine ΔB at the half power point as

$$\Delta B = \frac{\alpha B - \frac{4\alpha b}{\pi} \cos^{-1}(\psi)}{(1 - \alpha) + \frac{4\alpha}{\pi} \cos^{-1}(\psi)} \quad (4.14)$$

where

$$\psi = \frac{16 \sin^2(\pi B)}{\sqrt{2} \sin^2(4\pi B)} \quad (4.15)$$

For $B_N = \frac{1}{16}$, the predistortion bandwidth of the SRN pulse is $\hat{B}_N = B_N + \Delta B =$

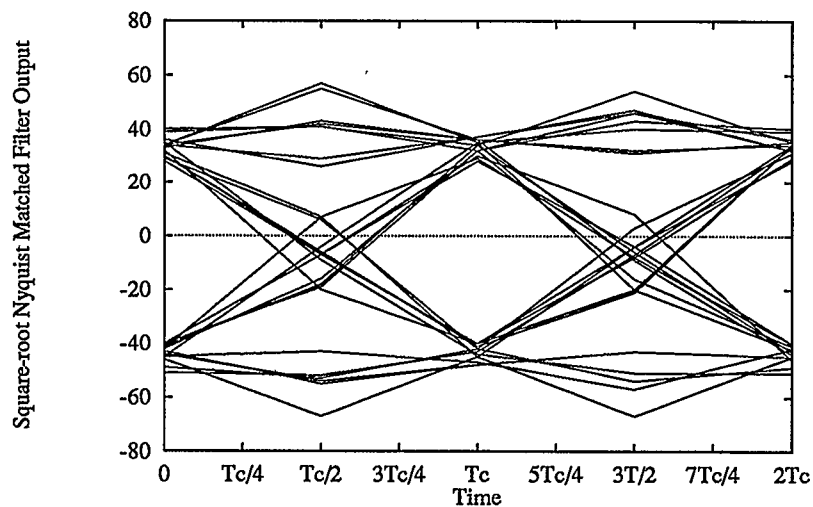


Figure 4.8 Eye diagram for interpolating filter bandwidth = $1/16$.

0.070161. The tap values corresponding to $\hat{B}_N$ are shown on table 4.1. The eye diagram using the predistorted pulse is shown on Fig. 4.9. It is apparent from this

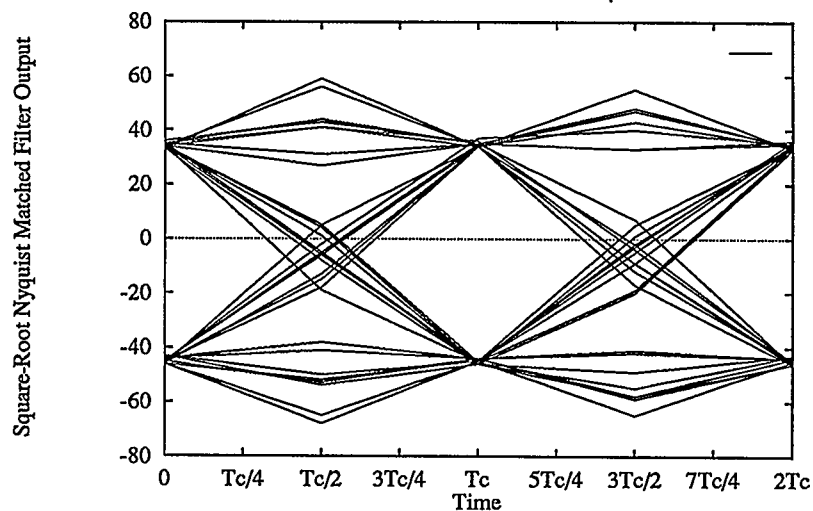


Figure 4.9 Eye diagram for interpolating filter bandwidth = 0.070161.

figure that predistortion of the waveform causes the eye to open, thereby reducing the errors caused by additive noise.

4.4 The Direct-Sequence Spread-Spectrum Receiver

The functional blocks of the receiver's digital structure are shown on Fig. 4.10. The entire system downstream of bandpass filter # 2 has been implemented and tested

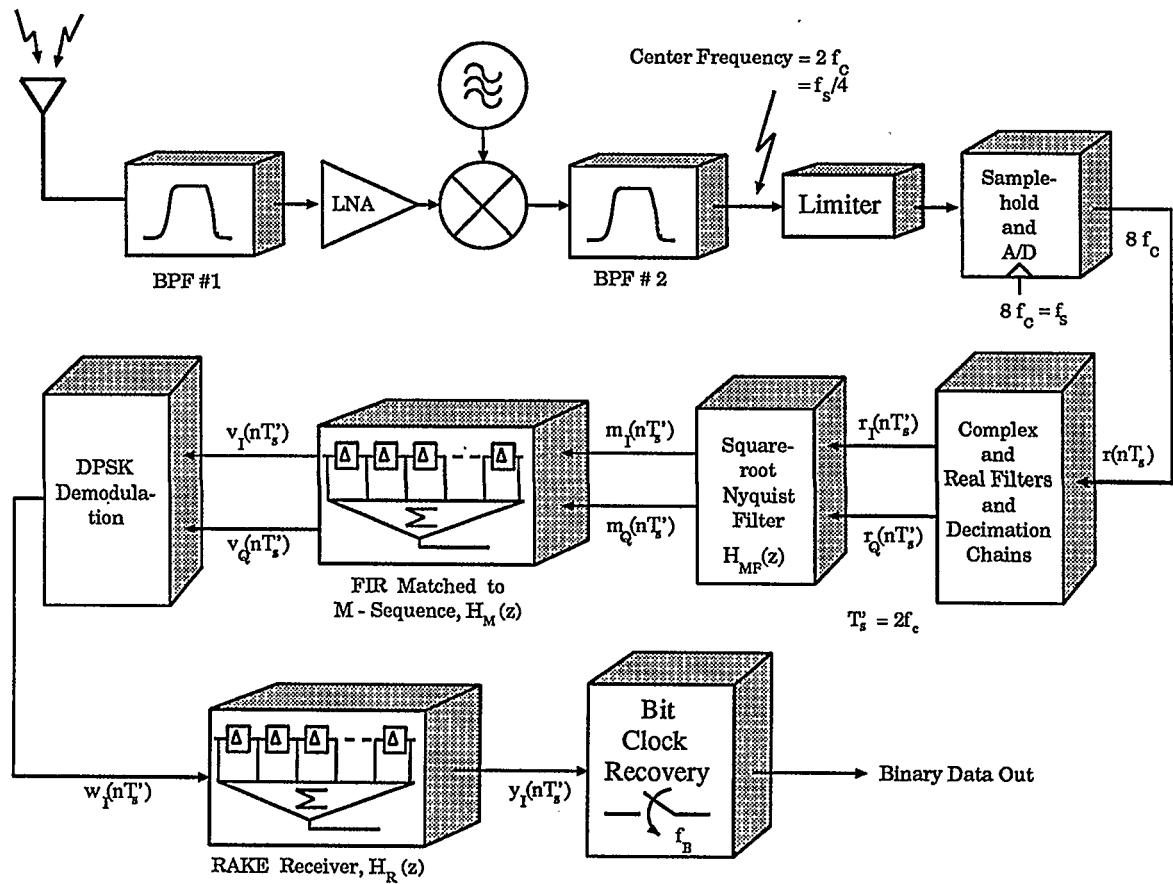


Figure 4.10 Integrated Direct-Sequence Spread-Spectrum Rake Receiver.

for an additive white Gaussian noise channel. Xilinx³ field-programmable gate-arrays

³Xilinx is a trade name of Xilinx Inc.

(FPGA's) were used to implement the digital portion of the receiver downstream of the A/D converter.

The radio frequency (RF) down-converter consists of 1) a front-end BPF to prevent imaging of out-of-band noise, 2) a low-noise amplifier, and 3) a mixer and BPF to demodulate the signal down to an intermediate frequency (IF) of 2.5 MHz. The signal then enters a limiter before being sampled at 10 MHz - four times the center frequency (see quarter-wave sampling Chapter 3 Sec. 3.5). Down-conversion by halfband-filters and decimation chains moves the received signal to baseband sampled at 2.5 MHz. This baseband signal is then filtered with a square-root Nyquist filter and an FIR filter matched to the M-sequence. DPSK demodulation removes the channel phases from the multipath echoes allowing coherent addition in the Rake. The Rake output enters a first-order digital phase-lock-loop (PLL) to derive the bit clock for sampling.

All of the above components of the DS-SS will be discussed in greater detail with the exception of the halfband-filter and decimate chains which were already presented in Chapter 3 Sec. 3.5.1.

4.4.1 Limiter

Hardware emulation without the limiter showed that noise spikes, which exceeded the A/D input, resulted in poor A/D performance. Also, simulation showed that an automatic gain control (AGC) device resulted in low correlation peaks (Chapter 6 Sec. 6.3.3). It is possible that at the low experimental SNR's (there is a processing gain of 31 due to the M-sequence), the noise signal "swamps" the data signal resulting in large data signal attenuation due to the AGC (see Chapter 6 Sec. 6.3.4.1 for a discussion on the effects of signal attenuation versus performance).

4.4.2 Square-Root Nyquist Matched Filter

The square-root Nyquist filter block consists of two real identical FIR matched filters given by the z-domain expression.

$$H_{MF}(z) = H_i^*(1/z^*), \quad (4.16)$$

where $H_i(z)$ is the transmit interpolating filter. The matched filter, $H_{MF}(z)$, is the optimum filter for detecting a pulse in the presence of additive white Gaussian noise [4] [42]. Due to the symmetry of the Nyquist pulse, the matched filter is the same as the pulse-shaping filter given by eqn. 4.10 except that $B_N = 1/4$ to account for decimation-by-4. Using eqn. 4.12, the number of taps $L = 8$. However, based on eye diagram analysis and out-of-band noise rejection, it was decided to implement the filter with 13 taps. The tap values, shown on table 4.2, were calculated using NOMAD⁴ which uses an *annealing* algorithm to find the best set of coefficients that

Receive filter taps		$B_N = \frac{1}{4}$
h_{MF0}	h_{MF12}	-2
h_{MF1}	h_{MF11}	2
h_{MF2}	h_{MF10}	4
h_{MF3}	h_{MF9}	-8
h_{MF4}	h_{MF8}	-6
h_{MF5}	h_{MF7}	40
h_{MF6}		72

Table 4.2 Filter tap values for square-root Nyquist matched filter

match the SRN frequency response. Annealing and the non-ideal effects of coefficient word-length and quantization are discussed in Chapter 5, Sec. 5.4.5.2.

⁴NOMAD, a CAD design for FIR and IIR filters, is available from Dr. L. Turner. The development of the tool was supported by MICRONET.

4.4.3 M-sequence Matched Filter

For a single path noiseless channel, the sampled autocorrelation output from the M-sequence matched filter is shown on Fig. 4.11. Straight-line interpolation is shown

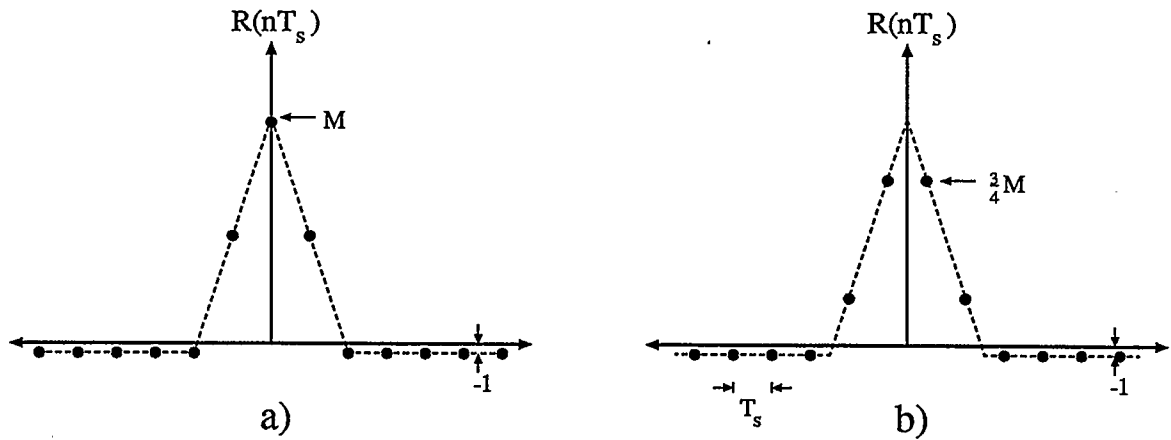


Figure 4.11 Output from FIR M-sequence matched filter: 1) best case sampling, and 2) worst case sampling

to simplify the diagram, although the actual shape of the autocorrelation would resemble the Nyquist pulse. Two sampling scenarios are shown: 1) the best case sampling when the sample values are synchronized at times $\frac{n}{2}T_c$; $\{n = 0, 1, 2, 3, \dots\}$, and 2) the worst case sampling when the samples are at time $\frac{2n+1}{4}T_c$; $\{n = 0, 1, 2, 3, \dots\}$. The latter case represents a worst case 1.2 dB drop in the processing gain when sampling at two times the Nyquist rate. Worst case Nyquist rate sampling with a 3 dB attenuation would be unacceptable

An illustration of multipath channel detection at the M-sequence matched filter's output is shown on Fig. 4.12. The output is simplified in that it only shows the sample values of the detected echoes and not the odd or even periodic auto correlations of the codes.

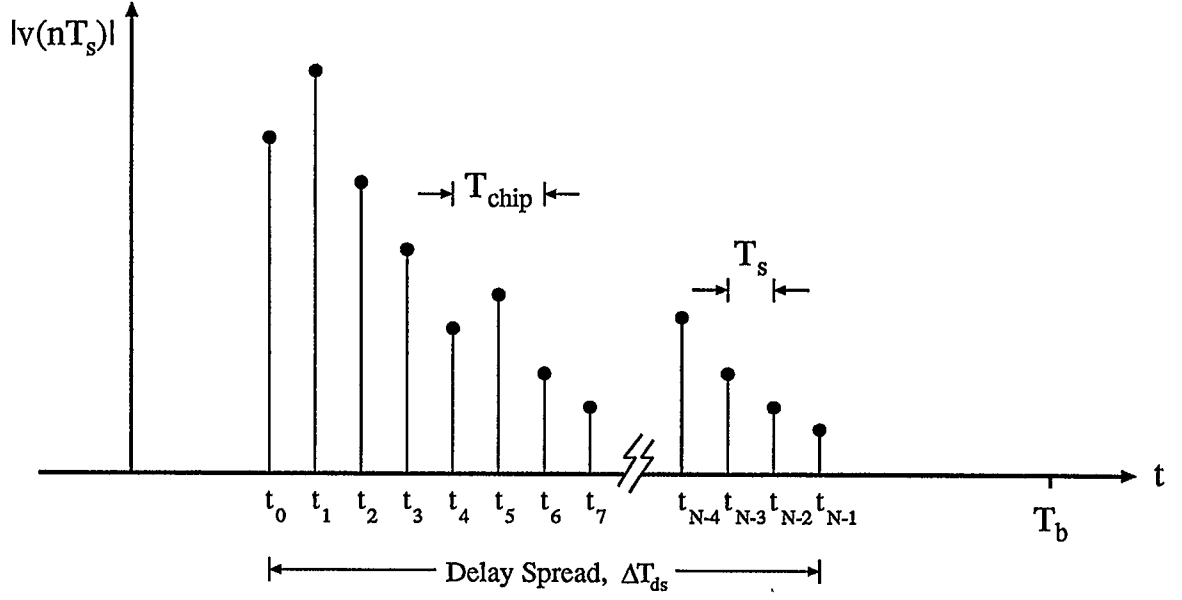


Figure 4.12 M-sequence matched filter sample values

The matched filter output is

$$\tilde{v}(nT_s) = \sum_{j=0}^{N-1} \alpha_j R(nT_s - t_j) e^{j(\theta(nT_s) + \phi_j)}, \quad (4.17)$$

where $\tilde{v}(nT_s)$ is the complex envelope, α_j and ϕ_j are the multipath attenuation and phase at time t_j , $R(nT_s)$ is the correlation between the transmit and receive M-sequences, $\theta(nT_s)$ is the phase of the transmitted complex envelope, and N is the number of paths. The channel phase in eqn. 4.17 will cause non-coherent addition of the autocorrelation phasors as depicted by the six phasors shown on Fig. 4.13. In this case, the channel phase has caused a 145° rotation of the resultant phasor from the desired response. Obviously, the channel phase must be removed to coherently combine the channel phase; doing so will realize the multipath diversity processing gain, $\mathcal{G}_{MP}$ (Chapter 2 Sec. 2.5.1).

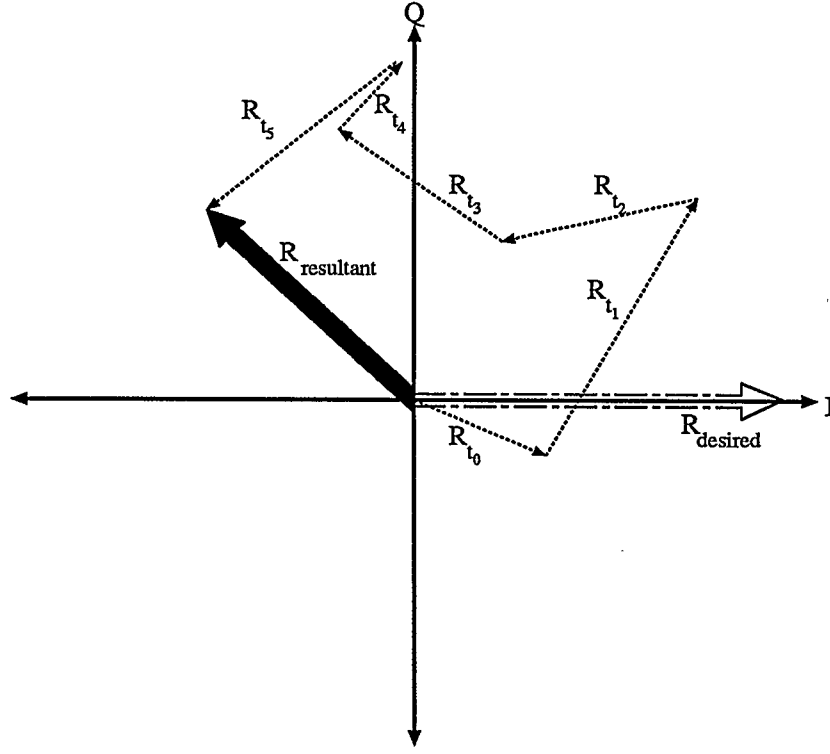


Figure 4.13 Phasor addition of the multipath autocorrelation values

4.4.4 DPSK demodulation

Two of the latest DS-SS systems [54] [30] , and those proposed by [53] and [15], employ a DPSK demodulation to rotate the autocorrelation phasors. In DPSK demodulation, the output of the in-phase and quadrature M-sequence matched filters (eqn. 4.17) is multiplied by the previous output delayed by one symbol time, T_b . The complex multiplication is

$$\begin{aligned}\tilde{w}(nT_s) &= \tilde{v}(nT_s)\tilde{v}^*(nT_s - T_b) \\ &= \left| \sum_{i=0}^{N-1} \alpha_i R(nT_s - t_i) e^{j\phi_i} \right|^2 e^{j\Delta\theta(nT_s)}\end{aligned}\quad (4.18)$$

where $\Delta\theta(nT_s)$ is the differentially demodulated phase of the transmit data. Eqn. 4.18 valid only if the channel does not change over one symbol period, and ISI is

neglected.

If the multipath can be resolved perfectly, there is no overlap of the correlations, and

$$\left| \sum_{i=0}^{N-1} \alpha_i R(nT_s - t_i) e^{j\phi_i} \right|^2 = \sum_{i=0}^{N-1} \alpha_i^2 R^2(nT_s - t_i). \quad (4.19)$$

It follows that the DPSK output is

$$\tilde{w}(nT_s) = \sum_{j=0}^{N-1} \alpha_j^2 R^2(nT_s - t_j) e^{j\Delta\theta(nT_s)}. \quad (4.20)$$

Eqn. 4.20 shows that the detected signal at the DPSK output no longer has the phasor rotation caused by the channel phase. Coherent combining of the N phasors is now possible.

4.4.5 Diversity Combining with the Rake

Turin [53] presents the digital Rake (DRake) to combine the energy of the multipath echoes. Turin's system uses a sounding receiver's estimate the channel's path delays to open or close the transversal filter's tap switches. A realization of the transversal filter is described in Chapter 2 Sec. 2.5.1. The convolution of the transversal filter response and the DPSK output (eqn. 4.20) extends channel delay spread, ΔT_{ds} , to $2(N-1)T_s$. To prevent ISI, the maximum symbol period, T_b , would be given by

$$T_b > 2\Delta T_{ds}. \quad (4.21)$$

The Rake designed in this thesis computes the convolution by implementing a time-invariant transversal filter that "barrel" shifts the tap control variable with its associated correlation value. It is essentially equivalent to an RC equal gain combiner except that integration is not continuous over the entire integration period. Fig. 4.14 shows the channel estimator, the DRake, and its output. Because of the delay line,

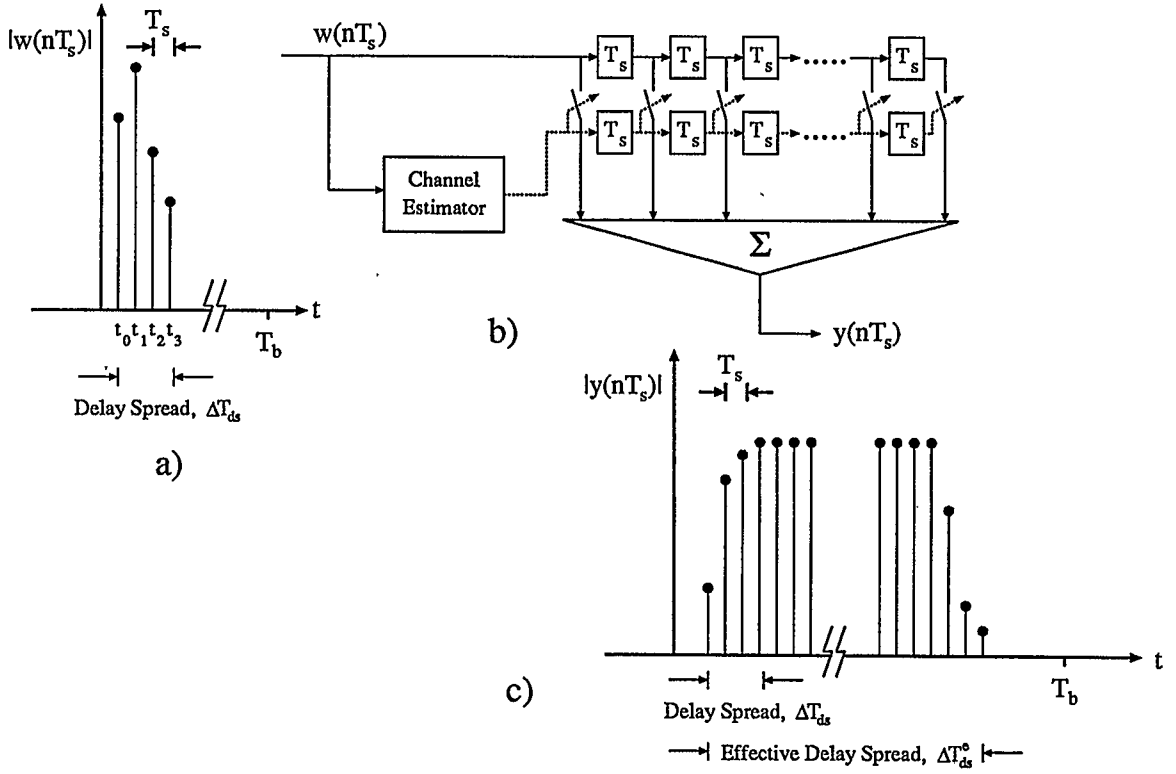


Figure 4.14 Digital implementation of the DRake: a) DPSK output, b) DRake, and c) Drake output.

the ISI criteria of Eqn. 4.21 is no longer valid. ISI will be prevented if the symbol period T_b is greater than the effective delay spread, ΔT_{ds}^e , given by

$$T_b > \Delta T_{ds}^e = (N_{DRake} + N - 2)T_s \quad (4.22)$$

where N_{DRake} is the number of DRake taps. Given the maximum delay spread of 7 μsec . [20], a 16 taps DRake would have an effective delay spread of approx 14 μsec . Although larger delay spreads have been measured [47], the symbol period of $T_b = \frac{127}{1.25\text{MHz}} = 101.6 \mu\text{sec}$ should prevent ISI from occurring (127 is the number of chips and 1.25 MHz is the sample rate).

The output from the DRake will be wider than the output from a typical correlating receiver. The wider output makes it easier to sample in the presence of multipath

phase jitter.

In the DRake, the channel estimator in Fig. 4.14 is simply a threshold detecting device. The estimator compares the incoming signal to a threshold value and outputs a control signal to the transversal tap switches. A more elaborate scheme [15] would be to compute tap value for the i th symbol as

$$\bar{h}_i = \rho \bar{h}_{i-1} + (1 - \rho) \bar{w}_i \hat{d}_i \quad (4.23)$$

where $\bar{h}$ and $\bar{w}$ are the DRake tap and DPSK output vector representations, $\hat{d}_i$ is the sampled data value, and ρ is the exponential weighting factor. Unfortunately, this algorithm was not implemented due to the inavailability of hardware.

4.4.6 Bit-Clock Recovery

A phase-locked loop (PLL) is used to derive the bit-clock from the DRake output. Two PLL's are shown on Fig. 4.15. The conventional analog PLL shown on Fig. 4.15 a) consists of three basic functional blocks: 1) a phase detector, 2) a loop filter, and

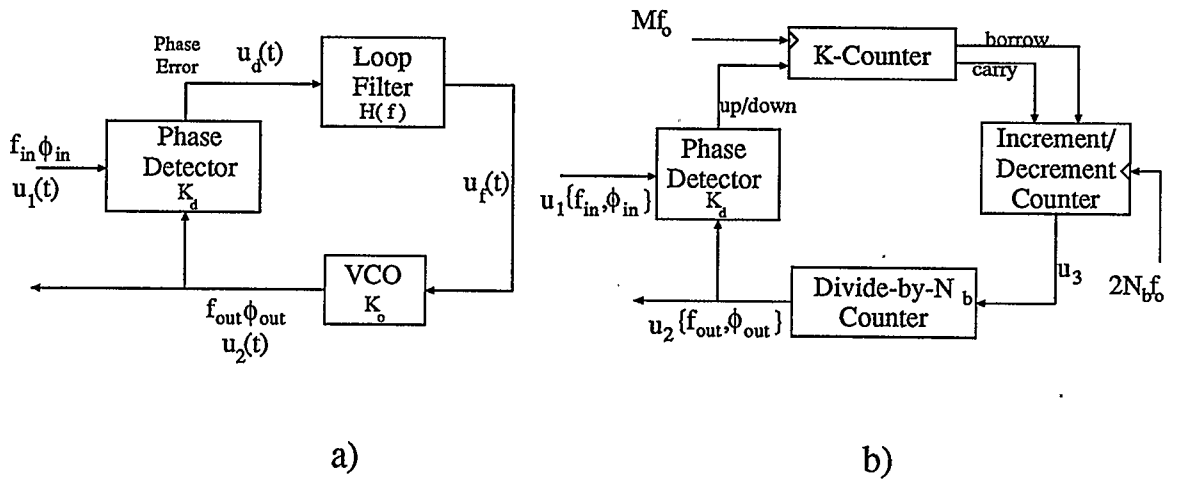


Figure 4.15 Phase-locked loops (PLL): 1) analog PLL, and 2) first-order digital PLL.

3) a voltage controlled oscillator. A discussion of its operation can be found in [5], [33], [49], [42], [45]. By implementing $H(f) = 1$ as the loop filter, the conventional analog PLL can be realized by a simple first-order digital equivalent shown on Fig. 4.15 b). At the expense of a higher degree of complexity (and hence more hardware components), higher order phase lock loops can be implemented using a cascade of first-order digital equivalent systems [52], or implementing the Hilbert transform phase detector PLL [5].

Besides keeping the DS-SS receiver completely digital, another reason for using a digital PLL over an analog PLL is its insensitivity to voltage and temperature changes, higher free running frequency (32 MHz vs 10M MHz), and bandwidth programmability [55].

The PLL on fig, 4.15 operates at a center frequency of f_o when the input signal u_1 is the same frequency and phase as the reference signal u_2 ; ie., $f_o = u_1 = u_2$. In this case, the K-counter does not produce any *carry* or *borrow* pulses therefore the increment/decrement (I/D) counter divides exactly by two. If the frequency of u_1 increases, the K-counter produces *carry* pulses which causes the I/D counter to add pulses to u_3 ; causing u_2 's frequency to increase. Conversely, *borrow* pulses are generated from the K-counter when the frequency of u_1 is less than u_2 . A detailed analysis of the first-order PLL is found in [52].

The phase detector (PD) implemented in the bit-clock recovery circuit is a type 4 phase detector shown on 4.16. The type 4 phase detector outperforms type 2 PD's (XOR gate) and type 3 PD's (edge triggered JK flip-flop) in both frequency and phase sensitivity and its performance is independent of duty cycle [5].

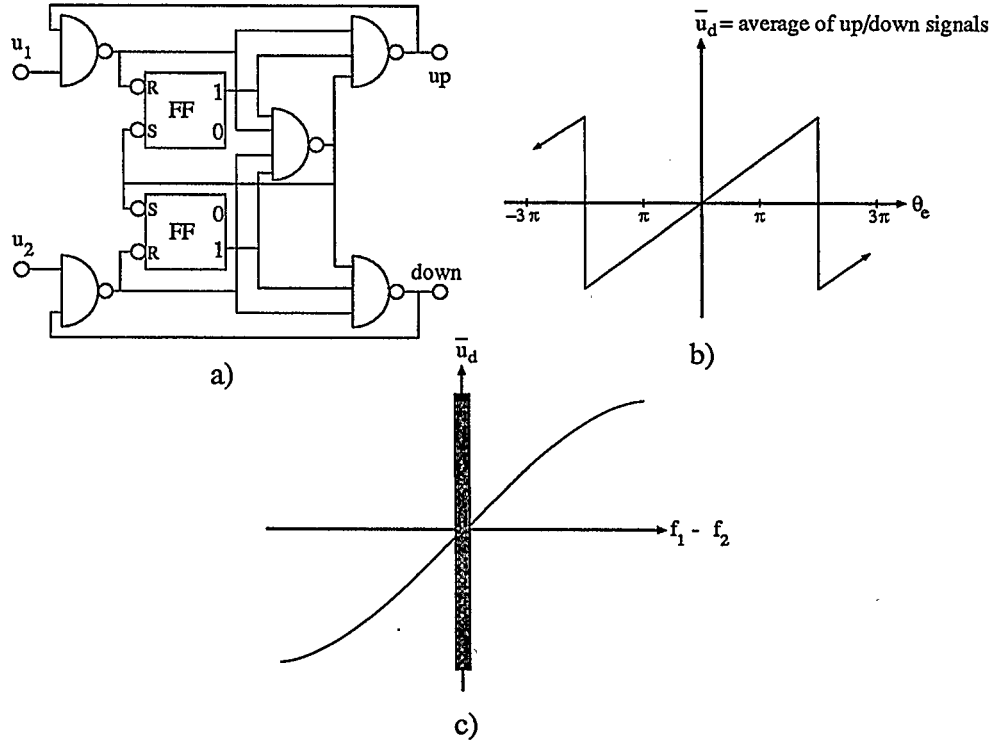


Figure 4.16 Type 4 Phase detector: a) digital structure, b) phase error, and c) frequency error.

The lock-in range of the PLL is given by

$$\Delta f_{max} = \frac{M f_o}{2K N_b} \quad (4.24)$$

where M is the scale factor for the I/D counter clock input, K is the modulus of the K-counter, and N_b is the divide-by-N counter modulus. Based on a system clock stability of 200 ppm, the maximum frequency deviation of the bit clock is

$$\Delta f_{max}^{200ppm} = f_{sys} \frac{1}{4N_d N_c} * 200ppm \quad (4.25)$$

where f_{sys} is the system clock, N_d is the receiver decimation factor and N_c is the number of chips. For $N_d = 4$, $N_b = 248$, $f_{sys} = M f_o = 20MHz$, and $N_c = 31$, eqn.s 4.24 and 4.25 give $K = 5000$.

Since this is a first-order PLL, a phase error, ϕ_e , will be present when the frequency of u_1 equals u_2 so long as $f_{in} \neq f_o$. This phase error is given by

$$\phi_e = 2\pi \frac{2KN_b(f_{in} - f_o)}{K_d M f_o} \quad \text{radians} \quad (4.26)$$

which as expected represents one cycle for $f_{in} - f_o = \Delta f_{max}$. This phase error is unacceptable for the sampling of the DRake output which is a narrow pulse relative to the bit period, but less critical for the DRake output.

Assuming that the transmission medium is a single path noiseless channel, the data output train of the DRake would have a 26% duty cycle given a 31 chip M-sequence and a 16 tap DRake. If at f_o the sample point is at the center of the pulse, the maximum phase error acceptable is $\frac{1}{8}$ th of a cycle to assure optimum sampling. Using eqn. 4.26, K is computed to be 625 - note that this factor is much less than that computed by equations 4.24 and 4.25. Experimentation showed that the number of bit errors (and therefore the "time-to-lock" period) was reduced by using a smaller K value as discussed in Chapter 6 Sec. 6.3.5.2.

These phase error considerations could be avoided by implementing a higher order PLL like the ones mentioned earlier in this section. The decision to use this particular PLL was made to minimize the complexity and hence the number of transistors in the design.

4.5 Simulation

Several *C* programmes were developed to simulate the baseband and passband transmitter and receiver pairs with both floating-point and fixed-point arithmetic. Their functional blocks were implemented as described in the above sections with the exception of the bit-clock recovery block which was not simulated. In all bit error

simulations, it was assumed that the received signal is synchronously sampled at the symbol rate.

The floating-point model was used to test algorithm performance without the non-ideal effects of fixed-point arithmetic. The algorithms were then implemented in fixed-point to determine effect of quantization, word-length, and overflow.

The fixed-point model was developed for two different word lengths: 1) a 32 bit model, and 2) an 8 bit model. The 32-bit model was used to determine the dynamic range requirements of the system. In Chapter 6 Sec. 6.3.4.2, this program was used to determine bit error curves for various A/D input levels. This program was also used to flag overflows in the DS-SS functional blocks.

The effect of overflows in the DS-SS receiver were investigated with the 8-bit model. Although most of the functional blocks of the DS-SS receiver were designed to prevent overflow, two of the blocks - the SRN matched filter and the DPSK demodulator - were allowed overflows. Overflow was allowed in these instances to prevent downstream dynamic range of the signal from being pushed into the quantization noise; thereby grossly affecting the receiver's performance.

The passband models were developed to simulate aliasing and frequency offsets between the transmitter and receiver carrier signals.

4.6 Summary

Each functional block of the DS-SS transmitter and receiver pair was presented. The transmitter, by using an interpolating square-root Nyquist filter, sends a differentially modulated spreaded waveform of the binary data sampled at 8 times the symbol rate. The spreading waveform is a 31 chip M-sequence that occupies 1.25 MHz of spectral bandwidth. The interpolating filter coefficients are designed to predistort

the signal to account for distortion caused by the receiver half-band filter chain, and to reduce the rolloff requirements of the analog reconstruction filter.

The signal, sampled at a rate four times its center frequency of 2.5 MHz, is quadrature down-converted using a half-band filter and decimate-by-4 chain. The demodulated signal is over-sampled at two times the Nyquist rate to avoid a 3 dB loss on the correlation peak magnitude. The M-sequence matched filter outputs the impulse response of the channel convolved with the autocorrelation function of the M-sequence. It is passed through a DPSK demodulator to differentially decode the symbol and to remove the channel phase for coherent addition in the DRake. The DRake is a simple channel estimator whose algorithm compares the incoming signal with a pre-programmed threshold level. The "barrel shifting" design of this block results in an increased duty cycle for improved clock recovery and sampling.

CHAPTER 5

CONSIDERATIONS FOR VLSI IMPLEMENTATION

5.1 Introduction

In this chapter, the Xilinx¹ implementation of the direct-sequence spread-spectrum (DS-SS) receiver is presented. The completely digital receiver design, from the analog-to-digital (A/D) converter to (and including) the phase-locked loop, was implemented on 4-3000 and 2-4000 series Xilinx field-programmable gate-arrays (FPGA's). Only a single 3000 series FPGA was required for the transmitter.

The chapter begins by reviewing the non-ideal effects such as quantization noise, coefficient quantization, and finite word-length arithmetic errors, that are inherent to the digital representation of analog signals and filter structures. Following this, parallel, bit-serial, and digit-serial digital architectures are introduced and their relationship to processing speed, layout, testing, and hardware utilization is discussed. With this background in non-ideal effects and digital architectures, the DS-SS functional blocks (presented in Chapter 4) are designed in conjunction with special considerations that were made for Xilinx FPGA prototyping. Finally, the gross digital design layout of the functional blocks, the distribution of clock signals and the communication protocols between Xilinx modules is discussed.

5.1.1 Binary Number representation

The implementation of a digital processing system requires that the analog signals, represented theoretically by real numbers, be represented by some finite digital

¹Xilinx is a trademark of Xilinx Inc.:manufacturer of commercial FPGA's

numbering system. A rational number N_r can be represented with finite precision [4] as

$$N_r = \sum_{i=-M_{LSB}}^{M_{MSB}} c_i r^i; \quad 0 \leq c_i \leq (r-1), \quad (5.1)$$

where c_i is the i th coefficient, r is the radix of the representation. M_{MSB} and M_{LSB} are integer values whose sum will be defined by $B + 1$ as

$$B + 1 = M_{MSB} + M_{LSB}. \quad (5.2)$$

From eqn. 5.1 N_r is represented in decimal if $r = 10$ and in binary if $r = 2$. Thus for a binary number, c_i takes the values 0 and 1. In this case the coefficients are referred to as binary bits and their aggregate makes a binary word.

The binary representation of a negative number depends on the fixed-point arithmetic used. There are three commonly used types: 1) straight binary, 2) one's complement, and 3) two's complement. In two's complement arithmetic, a negative number is represented by assigning the most significant bit (MSB), $c_{M_{MSB}}$, as a sign bit. It will be the convention of this thesis to assign the binary point to a fixed place between the first and second most significant bits. In this way, the maximum positive value is given by

$$N_{max_pos} = \sum_{i=-M_{LSB}+1}^{M_{MSB}-1} r^i, \quad (5.3)$$

and the maximum negative value by

$$N_{max_neg} = c_{M_{MSB}} r^{M_{MSB}}, \quad (5.4)$$

where $c_{M_{MSB}} = -1$ and $M_{MSB} = 0$. For example, if the binary word (or binary code) is 3 bits (ie. $B + 1 = 3$), the maximum positive value fractionally given by eqn. 5.3 is $N_{max_pos} = (2^{-2}) + (2^{-1}) = 3/4$, while the negative number is $N_{max_neg} = (-1)(2^0) = -1$.

5.2 Non-Ideal Effects

5.2.1 Analog-to-Digital Converter

An A/D converter is a physical device that converts an analog signal amplitude into a binary code representation. A typical converter is shown on Fig. 5.1 a). The converter can be mathematically described as a continuous-to-discrete converter (C/D) (analogous to the discrete-to-continuous converter (D/C) in Chapter 4, Sec. 4.3.2.1) cascaded with a quantizer as shown on Fig. 5.1 b).

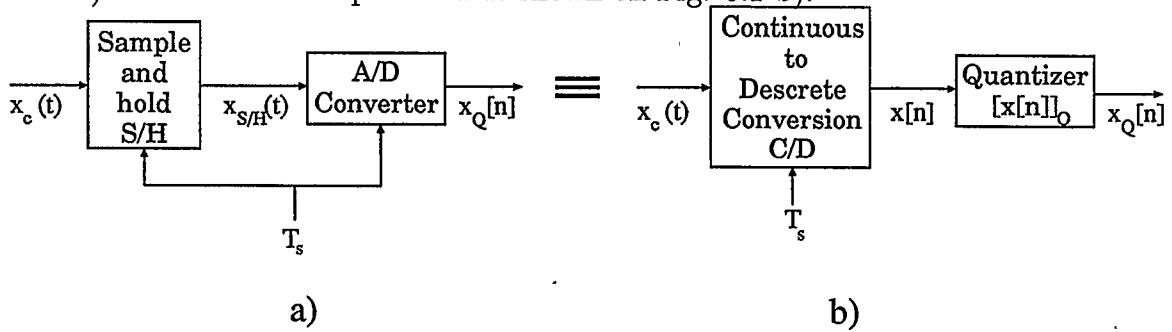


Figure 5.1 Digitization of analog signals: a) real system, and b) theoretical system

High performance A/D's use a sample-and-hold (S/H) circuit to hold the sampled signal for T_s seconds while the A/D completes the conversion process. In some cases, the A/D's conversion speed is very fast relative to the change of the signal and therefore do not require a S/H. These A/D's are typically *flash* A/D's which can operate up to 300 MHz², and even faster A/D's will be available as technology advances [18].

Nevertheless, sampling of signals means that the signal will be held for a finite period of time. Thus, the S/H output can be expressed as

$$x_{S/H}(t) = \sum_{n=-\infty}^{\infty} x_c(nT_s)h_{S/H}(t - nT_s), \quad (5.5)$$

²Analog Devices AD9038

where the continuous time signal $x_c(t)$, sampled at intervals T_s , is convolved with a square pulse, $h_{S/H}(t)$, given by

$$h_{S/H}(t) = \begin{cases} 1, & 0 < t < T_s \\ 0, & \text{otherwise.} \end{cases} \quad (5.6)$$

The Fourier transform of eqn. 5.6 - aside from a linear phase term - gives the well known *sinc* function distortion given by:

$$H_{S/H}(\Omega) = T \frac{\sin\left(\Omega \frac{T_s}{2}\right)}{\Omega \frac{T_s}{2}}. \quad (5.7)$$

This distortion can be compensated for by applying the inverse filter of eqn. 5.7 downstream of the A/D.

The binary value representation of , $x_Q[n]$, will be an approximation of $x_{S/H}(t)$ (as in the case of the A/D output) or $x[n]$ (as in the case of the quantizer output). This is because $x_Q[n]$ is represented with a finite set of values.

Assuming that the bandlimited waveform $x_c(t)$ is sampled at or above the Nyquist rate, then for the real system:

$$x_{S/H}(t) = x_c(nT_s), \quad (5.8)$$

and for the theoretical system:

$$x[n] = x_c(nT_s). \quad (5.9)$$

Unlike $x_{S/H}(t)$, $x[n]$ is the result of multiplying $x_c(t)$ by a train of discrete time impulses spaced at period T_s , and therefore is not distorted by the sinc function. If the sinc distortion is removed from $x_{S/H}(t)$, then it can be said that the sampled value of the continuous-time signal is the same for both systems. The discrete-time signal in turn is represented by the quantized signal, $x_Q[n]$. The error in approximating $x[n]$ with a finite set of values is given by

$$x[n] = x_Q[n] + \epsilon[n]. \quad (5.10)$$

The error $\epsilon[n]$ is commonly known as the *quantization error* and results in *quantization noise* inherent to digital systems.

5.2.1.1 Quantization Noise

Errors that occur in the finite value representation of $x_c(t)$ have maximum values of $\epsilon = \pm\frac{\Delta}{2}$ for rounding and $\epsilon = 0$ and $-\Delta$ for truncation, where

$$\Delta = \frac{1}{2^B}. \quad (5.11)$$

This error, or *quantization noise*, can be statistically defined as a uniformly distributed white-noise sequence with variance

$$\sigma_\epsilon^2 = \frac{\Delta^2}{12} \quad (5.12)$$

and mean, μ_ϵ . The mean is zero for rounding and $\mu_\epsilon = -\frac{\Delta}{2}$ for truncation. The above is true if the following assumptions are made [3]:

1. The error sequence $\epsilon[n]$ is a sample sequence of a stationary random process.
2. The error sequence is uncorrelated with the sequence $x[n]$.
3. The random variables of the error process are uncorrelated.
4. The probability distribution of the error process is uniform over the range of quantization error.

These assumptions are valid if the continuous signal changes by an amount greater than the quantization step and in a fashion that is random [26] [4]. A signal that does not possess these characteristics can be changed to one that does by adding *dither* [23]. *Dither* is a broadband, yet bandlimited, noise-like signal, or pseudo-noise signal, that is added to the signal at the input to the A/D. The *dither* spectra need not be

added to portions of the spectrum that interfere with the signal spectra. The spectra should however be placed where they will be filtered out by downstream digital filters: half-band filter and decimate chains could be used for this purpose.

The signal-to-quantization noise ratio is

$$\begin{aligned} SNR_Q &= 10 \log \left(\frac{\sigma_x^2}{\sigma_\epsilon^2} \right) \\ &= 6.02B + 10.8 - 20 \log(\sigma_x) \quad dB, \end{aligned} \quad (5.13)$$

where σ_x is the variance of the signal $x_c(t)$. Oppenheim [3] shows that if $x_c(t)$ is gaussian distributed, then the amplitude of $x_c(t)$ will be greater than $4\sigma_x$ approximately 0.06 % of the time. Hence eqn. 5.13 can be written as³

$$SNR_Q \approx 6B - 1.24 \quad dB. \quad (5.14)$$

An 8-bit A/D converter ($B = 7$) has a SNR_Q of 40.76 dB. It was shown in Chapter 3 Sec. 3.5.2.1 that halfband filters can be used to increase the SNR_Q by 3 dB per filter and decimate chain stage; representing an increase the signal's binary word, B , by one-half bit per stage.

5.2.2 Coefficient Quantization

Whenever finite word-length coefficients are used to represent a desired frequency response of a digital filter, the quantization of the real coefficients results in a shifting of the pole and zero locations. This inturn results in an error between the desired and implemented response. In infinite-impulse response (IIR) filters, this is of particular importance when poles are close to the unit z-plane circle. Quantization may cause the poles to be shifted outside the unit circle causing the IIR filter to become unstable.

³Eqn. 5.14 is noted as a *rule-of-thumb* for broadband signals in [23].

Finite-impulse response (FIR) filters do not suffer from this side-effect; however, they are not immune to the distortion of the ideal response.

The effect of coefficient quantization is more significant for a high ordered filter realized in direct form than the corresponding cascade or parallel realizations [4]. This lends support for half-band filter-and-decimate chain implementations over the use of a single high-ordered filter.

To combat the effect of coefficient quantization, the filter designer has three options: 1) increase the number of coefficient bits, 2) choose a filter structure that is less sensitive to the coefficient truncation/rounding (such as cascaded sections), and 3) use an algorithm that selects filter coefficients in such a way as to minimize the difference between the desired and actual response [28]. One such algorithm is *simulated annealing*. This algorithm was used in the selection of the square-root Nyquist matched filter coefficients (Chapter 4, Sec. 4.3.2.2).

5.2.3 Overflow

In two's complement finite-precision systems, the addition of two values can cause the sum to exceed or *overflow* the maximum number representable by the system word length. Such an instance can also occur in subtraction too. When an *overflow* occurs, the bits that exceed the word length are ignored. For example, when a positive number increases past its maximum, the number wraps around to start counting up from the most negative number (two's complement) or from zero (unsigned arithmetic). These "wraparounds" create large opposite-direction transitions which have broadband harmonic content (perhaps introducing aliasing) and are difficult to filter [23].

A way of combating overflows is to use saturation arithmetic. Although harmonics

will also be introduced, their effect will be less severe as the harmonics introduced by *overflow*.

Yet another option is to design the system so that overflows will not occur. For the k th node of a filter with unit sample response $h_k(n)$, it can be shown that the upper bound to an input signal $x(n)$ is

$$x_{max} < \frac{1}{\sum_{n=0}^{\infty} |h_k(n)|}, \quad (5.15)$$

where $\sum_{n=0}^{\infty} |h(n)|$ is the L_1 norm. Eqn. 5.15 guarantees that all nodes in a filter will not overflow, But in reality, few signals will cause the output to approach this upper bound. Another less stringent "rule-of-thumb" is given by Higgins [23] as

$$x_{max} < \frac{1}{\max [|H_k(e^{j\omega T_s})|]}, \quad (5.16)$$

where $H_k(e^{j\omega T_s})$ is the Fourier transform of $h_k(n)$.

5.2.4 Fixed-Point Arithmetic Errors

The following two sections provide the statistics necessary to evaluate the noise power at the filter outputs in Sec. 5.4.4.1 and 5.4.5.4. In all arithmetic computations, the DS-SS transceiver uses truncation rounding to reduce system complexity. Although the following results show that other rounding schemes offer better statistics, the cost of additional gates to implement rounding outweighed the increase in SNR_Q .

5.2.4.1 Addition and Subtraction

Scaling is used to prevent addition or subtraction overflow in fixed-point systems. Scaling is treated as a right shift operation that mathematically represents a divide

by the factor two. The addition of two numbers, x and y , can be greater than the finite-word representation of their sum z . Scaling z to $B + 1$ bits yields $\hat{z} = [z/2]_S$ where $[]_S$ is the scaling operation. The difference between $\hat{z}$ and $z/2$ is the rounding error, ϵ_a . The statistics of ϵ_a have been studied by Coldham [8] for various methods of rounding $z/2$ to $B + 1$ bits. The results are shown in Table 5.1. There are four

	error mean, μ_{ϵ_a}	error variance, σ_{ϵ_a}	MSE
truncation	$-\Delta/4$	$\Delta^2/16$	$\Delta^2/8$
up-rounding	$\Delta/4$	$\Delta^2/16$	$\Delta^2/8$
random bit add or signed rounding	0	$\Delta^2/8$	$\Delta^2/8$

$\Delta = \frac{1}{2^B}$

Table 5.1 Fixed-point addition/subtraction error statistics.

rounding schemes: 1) truncation, where the least-significant bit is dropped; 2) up-rounding, where 1 bit is added before truncation; 3) random bit addition, where a bit is added at random before truncation; and 4) signed rounding, where a 1-bit is added if the $x + y$ is negative. Although the latter two schemes introduce the least error, they require increased system complexity - as does up-rounding - to implement.

5.2.4.2 Multiplication

The multiplication of two $B + 1$ bit numbers gives a $2(B + 1)$ bit product. Unfortunately, it is not always practical to use $2(B + 1)$ bits - particularly in IIR cases where the multiplication is in a recursive loop. Obviously, if the product is to be represented in anything less than $2(B + 1)$ bits, then truncation is necessary. The truncation and then rounding of the product introduces an error whose mean, vari-

ance, and mean-square error have been studied by Coldham [8] and are listed in Table 5.2. The rounding schemes are similar to Sec. 5.2.4.1 with the exception of signed

	error mean, μ_{ϵ_m}	error variance, σ_{ϵ_m}	MSE
truncation	$-\Delta/2$	$\Delta^2/12$	$\Delta^2/3$
up-rounding	0	$\Delta^2/12$	$\Delta^2/12$
random bit add	$-\Delta/4$	$7\Delta^2/48$	$5\Delta^2/24$
signed truncation	0	$19\Delta^2/64$	$19\Delta^2/64$

$\Delta = \frac{1}{2^B}$

Table 5.2 Fixed-point multiplication error statistics.

truncation where a 1-bit is added if the product is positive and if the least significant bits are non-zero.

The preferred method for rounding after multiplication is up-rounding if the error variance, $\sigma_{\epsilon_m}^2$, was a concern. If it is not, then truncation is preferred.

5.2.5 Modeling Quantization at the Output of a Filter

5.2.5.1 Quantization Noise at the Filter Input

In Sec. 5.2.1 it was shown that the output after the quantizer is

$$x_Q[n] = x[n] + \epsilon[n]. \quad (5.17)$$

The filter output, $y[n]$, is given by

$$y[n] = \sum_{m=0}^{\infty} h[m]x_Q[n-m], \quad (5.18)$$

where $h[m]$ is the unit sample response. Assuming the filter is linear time-invariant, the output will be the sum of two components: one due to the $x[n]$, and the other

due to $\epsilon[n]$. Also assuming that $\epsilon[n]$ is white noise with zero mean and variance σ_ϵ^2 , then the output noise power is given by

$$\sigma_{out}^2 = \sigma_\epsilon^2 \sum_{m=0}^{\infty} |h[m]|^2. \quad (5.19)$$

5.2.5.2 Quantization Noise due to Multipliers

Eqn. 5.19 can also be used for calculating noise power output due to quantization noise generated at the filter's internal nodes. Given $h_k[m]$, the unit impulse response from a k^{th} multiplier node to the output, the noise power out is

$$\sigma_{out}^2 = \sigma_{\epsilon_m}^2 \sum_{m=0}^{\infty} |h_k[m]|^2. \quad (5.20)$$

It follows that the total noise power out from N_m multipliers is

$$\sigma_{Total}^2 = \sum_{k=0}^{N_m} \sigma_{\epsilon_{mk}}^2 \sum_{m=0}^{\infty} |h_k[m]|^2. \quad (5.21)$$

The above equations also apply to nodes where scaling occurs.

Evaluation of eqn. 5.20 can be difficult to compute for IIR filters. But for an FIR filter with N_{tap} , the noise power out is simply expressed as

$$\sigma_{Total}^2 = \sum_{k=0}^{N_{tap}-1} \sigma_{\epsilon_{mk}}^2. \quad (5.22)$$

Thus, for an FIR filter, doubling the number of taps requires one more bit to retain the same signal-to-quantization noise ratio if the noise power is the same for each tap.

5.3 Hardware Considerations

Custom integrated designs, application-specific intergrated circuits (ASICs), and field-programmable integrated circuits (FPGA's), can be fully optimized for a particular design. All have several common limitations , such as finite transistor density,

finite routing resources, and finite switching speed, that must be considered for each application.

Data throughput is typically the reason why an ASIC, FPGA, or fully custom implementation is chosen over the *general purpose* digital-signal-processor (DSP). The general purpose DSP, being designed to meet the needs of several different applications, is not an optimal design for a particular application; hence, economy's of speed, power consumption, and chip area are not realized [35]. More flexibility can be achieved using an *application oriented* DSP; but not as much flexibility as with ASIC's, FPGA's, and full custom designs.

On the other hand, unless the fully custom designed chips are ordered in large quantities, their cost will be prohibitive in competitive markets. In these instances, DSP's become more attractive.

The assumption is made that the transceiver designed in this thesis would be produced for the next generation of spread-spectrum digital outdoor cellular communicators. Thus, the fully custom designed transceiver would be produced in quantities that would make it economical.

Another factor is the compactness of the design. The estimated number of arithmetic computations per second for the DS-SS transceiver totals $f_s(85 + 1.5N_c)$, where N_c is the number of chips and f_s is the sample rate. For a 2.5 MHz sample rate and a 31 chip M-sequence, at least six general purpose DSP's are required if each operates at 50 million instructions per second (50 MIPS) (assuming that a multiply, shift, add, etc., can each be completed in one instruction and no memory read/writes are needed). Estimates show that the DS-SS custom receiver design can be implemented with 40000 gates (160000 transistors), and the transmitter with 8000 gates. (chip length = 127). This is an acceptable size for ASIC chips and would be appreciably

compact compared to six DSP chips.

5.3.1 Hardware Architectures

There are three types of digital architectures that are available to the engineer: 1) parallel, 2) serial-parallel, and 3) bit-serial. These architectures create a wide spectrum of designs that exhibit certain characteristics which make each suited for a specific application. The arrows on Fig. 5.2 show the increasing degree of each characteristic. At one end of the spectrum (Fig. 5.2), parallel architectures realize

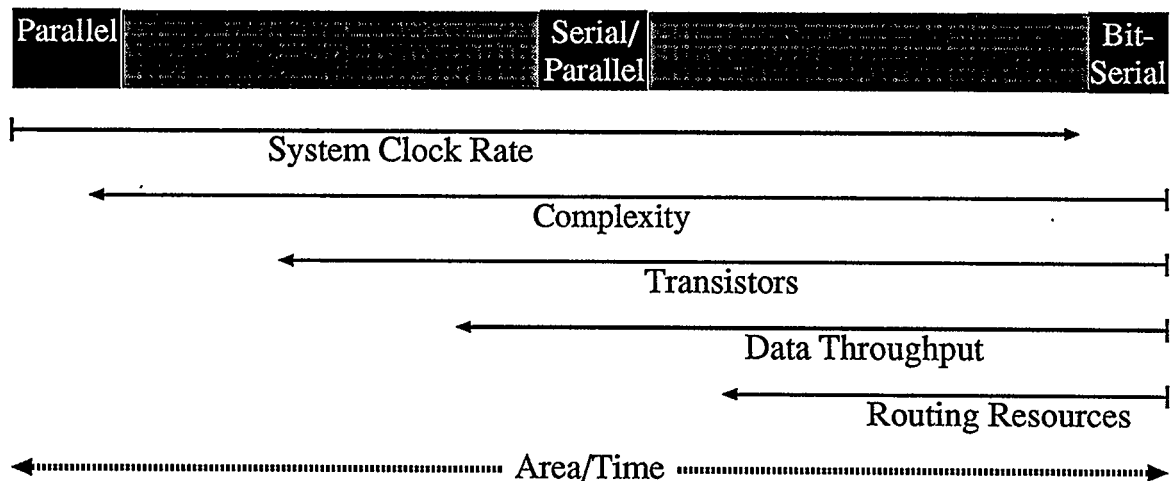


Figure 5.2 Characteristics of serial/parallel architectures.

high data rates at the expense of complexity, transistors, and routing resources. At the other end of the spectrum lies bit-serial architectures which realize low chip area at the expense of an increased system clock rate. A hybrid that possesses both characteristics lies between fully parallel and bit-serial designs.

Analysis of the above observations are made simple when looking at a system scheduling arithmetic computations around a single function block. For example, a

bit-serial adder would take up much less area and routing resources, but require a system clock that is equal to the data word length times the data word rate. But when several bit-serial function blocks are implemented - the smaller chip area of a single bit-serial function block might allow the designer to layout several as opposed to one - analyzing architectures into categories of speed, chip area, etc., becomes a perplexing problem. Instead of multiplexing all computations around a single large parallel multiplier, several bit-serial multipliers (these could also be serial/parallel multipliers) can be allocated in the system data flow paths where bottlenecks occur. Also, with more multipliers comes reduced scheduling complexity. An example of this approach is the realization of a fast-Fourier transform (FFT) machine given by Swartzlander [12].

Overshadowing the optimism of using bit-serial is the time required for data computations. Bit-serial function blocks require several clock cycles to compute the output; consequently, the system clock must be higher than for parallel designs.

In general, bit-serial architectures consume less chip area but require more clock cycles (or more time) per data sample, while parallel designs take less time but require more chip area. Thus, the designs are typically described by their area-time relationships. Sections 5.3.1.1 and 5.3.1.2 present the logic level layout of bit-serial designs in order to provide a better understanding of parallel/bit-serial designs and their area/time relationships (Fig. 5.2).

5.3.1.1 Parallel

Parallel architectures are used extensively in DSP and microprocessor design. The data flow is essentially of a parallel nature where the bits making up a binary word travel in *parallel* along a *bus* from memory to ALU, to I/O, etc. The *bus*, like the ALU

- or any other functional block for that matter - is typically not a "dedicated" device since all data flow must be scheduled to prevent collisions. Of course, parallelism is becoming prevalent in *massive parallel processors* (MPP's), but interconnecting the large matrix of parallel buses becomes a routing nightmare.

On the micro scale, the placement of say a single multiplier involves careful attention to long routing lines that may cause set-up and hold time violations. Although the use of long routing lines is not recommended, parallel designs are typically routing intensive (compared to bit-serial) and may force the designer to utilize unfavourably long routes.

Fig. 5.3 [38] shows the functional blocks of a four-bit ripple carry parallel adder.

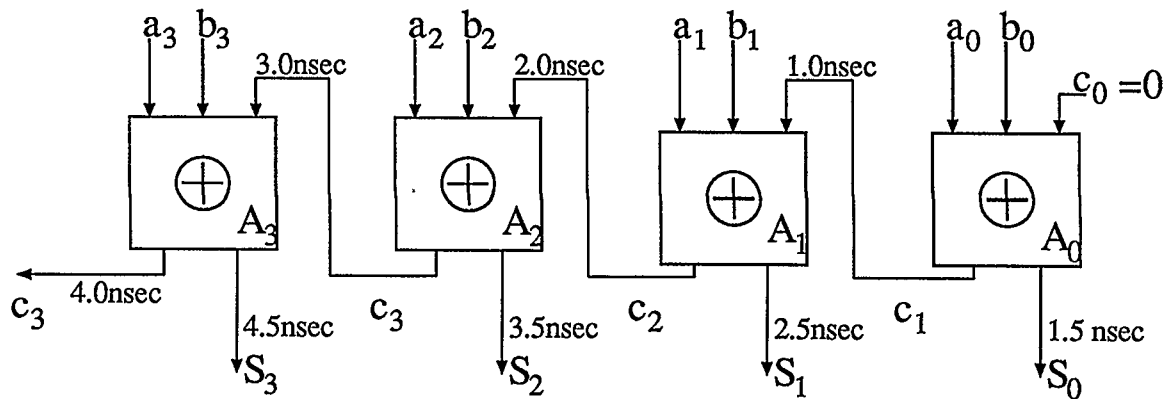


Figure 5.3 Four bit parallel adder.

The block can be cascaded to make higher order adders. It is presented here to make a comparison with the bit-serial adder presented in Sec. 5.3.1.2. The functional blocks labeled A_n output the *sum*, S_n , and *carry*, c_{n+1} , given the binary inputs a_n , b_n and c_n . This type of block will be referred to as a *computational* block since it performs arithmetic addition⁴. Associated with the parallel design is the delay time between the arrival of valid data at the input, and the propagation of the result through logic

⁴The gate level logic components can be easily derived using Carnaugh maps [38]

levels to the output. Shown in Fig. 5.3 is the propagation time of 0.7 μm CMOS. Obviously, the propagation delay time increases proportionately with the number of cascaded computational blocks. For example, the ripple carry output takes 4.0 nsec to propagate from the input to the output at the instance that c_0 is valid. On top of this is the increased in routing resources necessary not only for the interconnects between the computational blocks, but also for the interconnects inside the computational block itself.

The area/time relationship is summarized as:

- Gates = $N_{word} * 11.5$,
- Propagation delay = $(N_{word} + .5)$ nsec,
- Interconnects = $N_{word} * 18$, and
- System clock = f_s ,

where N_{word} is the bits that constitute a binary word and f_s is the system sampling rate. It is obvious that the area (ie. gates and interconnects) increases with the word size while the system clock rate does not.

5.3.1.2 Bit-Serial

In relation to parallel designs, designs that employ bit-serial architectures typically allow more of the chip area to be devoted to computation. This is because bit-serial design use less routing resources since they only require a single flow path, or channel, for the bits to travel in a *serial* fashion. The central channel is analogous to a large shift register with latches as its main building blocks to control the synchronous propagation of data. Between each latch are functional blocks that perform Boolean

logic functions where the inputs and outputs to the blocks are latched on the clock cycle. Fig. 5.4 shows the bit-serial implementation of a variable word-length adder. A description of its operation follows. The least-significant bit (LSB), a_0 and b_0 , enter

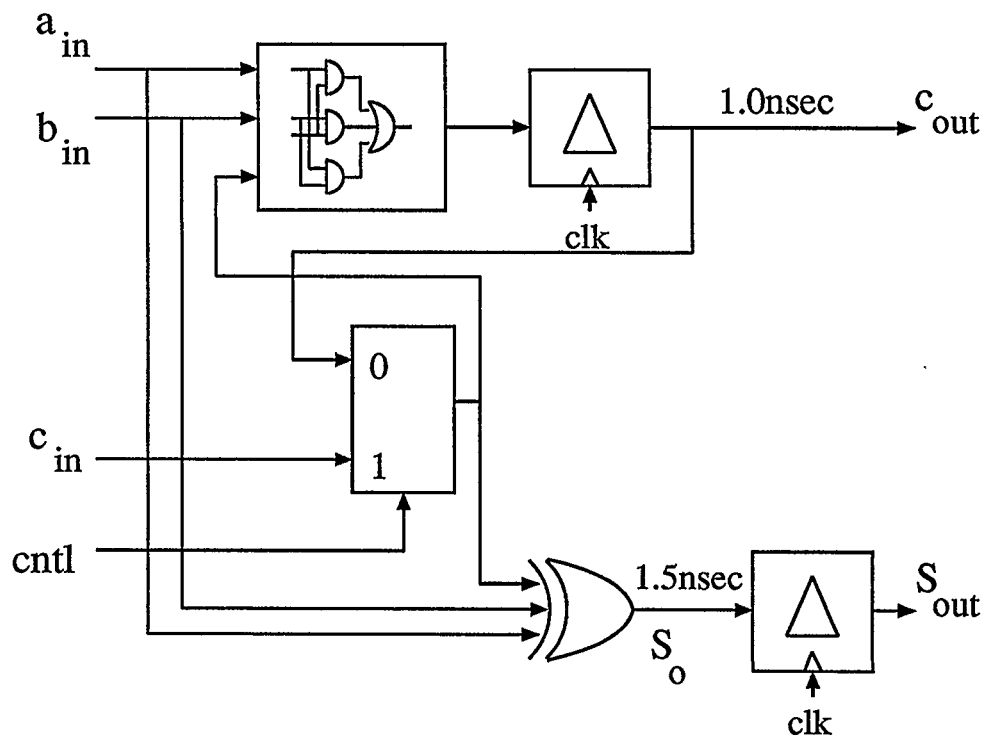


Figure 5.4 Bit-serial adder.

the adder as the control signal, $cntl$, goes high. This clears any carry data bits at the c_{out} latch, and enables the propagation of the carry-in signal, c_0 , to the carry logic. On the clock cycle, $cntl$ is low and c_1 (the carry bit from a_0 and b_0 addition) is latched to c_{out} , while at the same time the sum of a_0 and b_0 , indicated as S_0 , is latched to the output S_{out} . The $cntl$ will now stay low as the data bits, ie. a_1 to a_{n-1} , are clocked through the adder. When a_0 and b_0 of the next data word are at the input, $cntl$ will go high and the process repeats itself.

The key component to the adder's serialism is the latch placed at the output of the carry and sum logic. Because of the latch, it takes one clock cycle for the sum LSB to travel through the adder functional block, and is therefore said to have a latency equal to one. Functional blocks can have latencies that are either independent of the data (as in the case of the adder), or dependent (as in the case of a multiplier). Timing of the word's LSB at the input to a functional block is accomplished by upstream control signals much like *cntl* in Fig. 5.4. These control signals are generated from simple shift registers. In parallel implementation of serial functional blocks, generating *cntl* signals can be a strenuous task⁵.

In comparison to the parallel adder in Sec. 5.3.1.1, which has a propagation delay of 4.5 *nsec*, the longest propagation of the bit-serial adder is 1.5*nsec*.. From this observation, it could be said that the bit-serial adder is therefore a faster device; but for a 4-bit word, the adder must be clocked four times to complete the addition. Therefore, the time for the bit-serial adder to compute the four bits of the sum is $4 * 1.5nsec = 6.0\ nsec$ (assuming ideal propagation and no fanout effects). Obviously, the parallel adder is faster, but at the cost of almost four-times the hardware. On the other hand, the bit-serial adder requires a clock rate N_{word} times faster than the data rate, but less hardware.

The above 4-bit adder illustrates the area/time relationship that must be studied in-order to decide on which of the architectures is suitable for implementation. The problem becomes more complicated when the adders are being multiplexed with several data streams. In this case, not only are gate counts and clock speeds a concern, but also routing and the complexity of the control system.

⁵In Sec. 5.4.5.6, the task of generating control signals for a complex filter was orchestrated by the CAD development tool SNAFU.

The area/time relationship is summarized as:

- Gates = 19.5,
- Propagation delay = 1.5 nsec,
- Interconnects = 22, and
- System clock = $N_{word} * f_s$.

Note that only the system clock is a function of the data rate, f_s .

5.3.1.3 Digit-Serial

Digit-serial systems, where bits travel in pairs (or digits) along two channels instead of one, are the first step to parallelism from entirely bit-serial designs. In relation to area/time, the digit-serial device requires a clock $N_{word}/2$ times faster than the data rate at the cost of slightly more hardware and routing compared to the bit-serial system.

The area/time relationship is summarized as:

- Gates = 31,
- Propagation delay = 3 nsec,
- Interconnects = 41, and
- System clock = $(N_{word}/2) * f_s$.

5.3.1.4 A Comparison Between Parallel, Bit-Serial, and Digit Serial

Table 5.3 shows a comparison between parallel, bit-serial, and digit-serial designs of a 16-bit adder. With this table it is easy to see that the bit-serial design uses

Table 5.3 A comparison of adder structures

Architecture	gates		inter-connect	clock speed	Prop. delay (nsec)	Normal. delay (nsec)	Power Factor	
	min	max					min	max
Parallel	184	184	288	1x's	16.5	16.5	184	184
Bit-ser.	13.5	19.5	22	16x's	1.5	24.0	216	312
digit-ser.	25	31	41	8x's	3.0	24.0	200	248

considerably less gates (the minimum values represents no latch on the carry out) and interconnects (routing) than the parallel adder. But based on the normalized speed (clock speed x 's propagation delay), the parallel design is faster.

Another factor that has not been mentioned is the power consumption of the design. The *power factor* in Table 5.3 is an attempt to normalize the number of gates and clock speed (ie. area/time) in order to give a power consumption factor. There are two values given: the minimum, which assumes that the data is such that the carry out bit does not toggle; and the maximum, which assumes full toggling. This should give a good indication of relative power usage if say a parallel adder is replaced by a single serial adder. True power consumption would have to be obtained by experiment.

As mentioned before with parallelism of serial components, area/time relationships become complex; power consumption is not immune to this complexity either. Experimental work comparing parallel to serial power consumption by [19] shows that for bit-serial/parallel multipliers, all designs consumed almost the same power; whereas, [35] found that a bit-serial speech codec system consumed less power than

a DSP speech codec system. Nonetheless, more research is needed in this area to characterize power consumption in theory rather than by experiment.

5.3.2 Design Rules

A discussion of designs rules is an indepth topic. Details are given in [16]. Four of the key design rules are

1. Avoid excessive fanout: increased loads degrade signal rise and fall times, which in turn can result in improper latching.
2. Do not gate clock signals: gated clock signals, that pass through several layers of logic, can accumulate substantial delays which can violate timing between control signals and data.
3. Design must be fully synchronous: driving latches with a system wide clock provides a unit of time with which propagation delays, due to routing, capacitance, and non-ideal rise and fall times, can be accounted for with relative ease (ie., the minimum clock period is the sum of the worst case delays).
4. Two phase clock communication between chips: two phase clocks will make a time buffer of one-half a clock period; therefore the worst case clock skew would have to be greater than this amount to cause an error.

5.4 Hardware Design of the Direct-Sequence Spread-Spectrum Receiver

As mentioned in the above sections, limitations of speed, size, and routing, play a significant role in the decision to choose a particular architecture. A fourth limitation - power consumption - was assumed to be less for bit-serial designs compared to

parallel design. Even if future studies prove this not to be the case, it is strongly believed that the complexity and large chip area of parallel designs would outweigh any power consumption issues.

The limitations of the design are largely a function of the hardware platform. Another concern not addressed, is the availability of highlevel design tools to reduce implementation time. How the tools utilize special hardware features is also a feature that can ease implementation. Consequently, the tool's ability to utilize these features, and the choice of an architecture that takes advantage of them, were major concerns in designing the DS-SS transceiver.

The following sections first present hardware development tools and Xilinx prototyping hardware that influenced the choice of design. Next, the different functional blocks of the transceiver are presented with reference to hardware and design tool limitations.

5.4.1 Hardware Development Tools

There are essentially three paths shown on Fig. 5.5 that a design can go from concept to hardware prototyping: 1) filter synthesis using Nomad⁶ and Digicap⁷; 2) brute-force gate level description of parallel designs; and 3) high-level language bit-serial design using the FIRST silicon compiler⁸. Each path passes through the module

⁶NOMAD, a CAD design for FIR and IIR filters, is available from Dr. L. Turner. The development of the tool was supported by MICRONET.

⁷Digicap, a analysis and implementation tool for one-dimensional filters, is available from Dr. L. Turner.

⁸FIRST emerged from the University of Edinburgh Scotland in 1982 and the version used at the University of Calgary has been revised by Dr. L. Turner and P. Graumann.

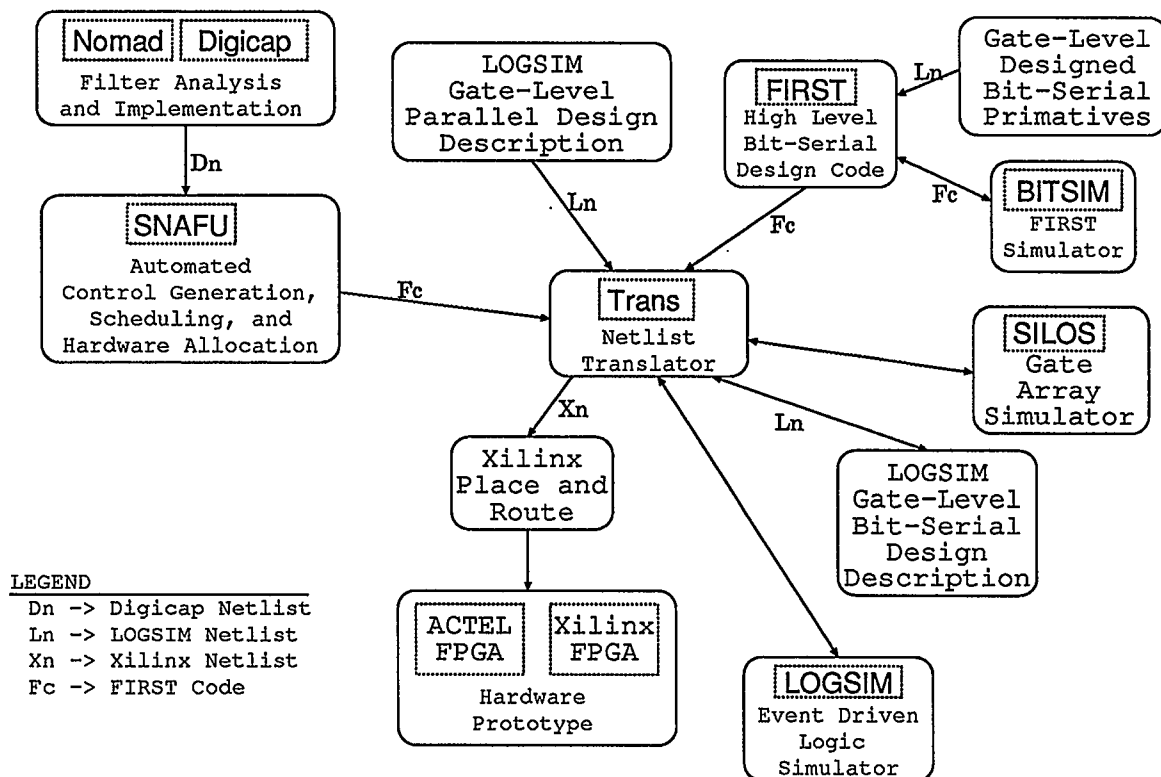


Figure 5.5 Bit-Serial and Parallel Architecture Design Tools.

Trans⁹, which translates netlists from Nomad, Digicap, SNAFU¹⁰, Logsim¹¹, and FIRST into XILINX or ACTEL netlists for hardware prototyping. It also translates Digicap netlist into Logsim netlist for logic simulation, and Digicap netlist into SILOS netlists for gate-level simulation. As a utility program, its functions include redundant logic removal, gate counting, and fanout specification. In the case of Xilinx 4000 series FPGA's, the translator can identify registers that can be transformed from latches into RAM cells and addressing units, thereby increasing the capacity of the chip (details in Sec. 5.4.2).

⁹Trans, a netlists translator and utility program, is available from Dr. L. Turner. The development of the tool was supported by MICRONET.

¹⁰SNAFU, a annealing utility program for FIRST code optimization, is available from Dr. L. Turner.

¹¹Logsim, an event driven logic simulator, is also available from Dr. L. Turner.

The module SNAFU, uses an annealing program to schedule data between shared bit-serial functional blocks while optimizing the system in terms of latency and hardware requirements. It generates FIRST code from Nomad and Digicap netlist descriptions - this program was used in the IIR lattice filter design (Sec. 5.4.5.6).

Unfortunately, high level design tools for parallel system layout were not available at the time of the transceiver design; however, at the time of this writing, tools for digit-serial implementations were nearing completion. This is the first step towards high level design tools with greater parallelism.

5.4.2 Xilinx Hardware Overview

The Xilinx XC3020-70, XC3090-50, XC4005-7, and XC4010-6 FPGA's were used to implement the transceiver. The chips architecture consists of 1) perimeter I/O blocks (IOBs), that provide a programmable interface between the chip programmable logic, and external output pins; and 2) an internal array of configurable logic blocks (CLBs), that perform user-specified logic functions. Three characteristics [25] that make the 4000 series chips distinct from the 3000 series chips are:

1. eight instead of two globally distributed clock signals,
2. on-chip programmable memory, and
3. independent inputs and outputs for the CLB flip-flops.

All play a significant role in the design of synchronous systems and therefore make this chip more ideally suited to bit-serial and digit-serial implementations. For example, an n -bit shift register would require $\frac{n}{2}$ XC3000 series CLB's, while the same register could be implemented on N_{CLB} XC4000 series CLBs, where

$$N_{CLB} = (((n)MOD16)MOD2), \quad (5.23)$$

where MOD is the modulus operator. For example, a 20-bit register would require 10-XC3000 series CLBs or 1-XC4000 series CLB's. This is because each XC4000 series CLB has 16×2 or 32×1 read/write memory cells¹². Although this configuration also requires an addressing unit, the CLB savings can be substantial. As a result, larger bit-serial designs can be implemented since they rely heavily on shift registers.

Each chip has been mounted on a stand-alone PC-board with I/O pins leading to headers for easy communication of several chips via ribbon connectors. Although this was not optimal in the sense of shielding wire connectors from switching noise, ground loops, and stray inductance and capacitance, it was advantageous in the initial stages of testing and flexibility of interchanging components. The high frequency operation of the design would be greatly enhanced if all chips resided on a single PC-board.

5.4.3 Overall Chip Layout

The design philosophy was to try to keep the number of external analog hardware components to a minimum, and to use only a single clock at the transmitter and receiver. In this way, less tuning, maintenance, and layout would be required, thereby making the design more economical to produce.

The transmitter was implemented on a single XC3090-50 FPGA as shown on Fig. 5.6 a). An external crystal provides the 10 MHz clock. The estimated number of gates for implementing a 127-chip m-sequence in a custom circuit or ASIC is approximately 8,000 gates.

Unlike the transmitter, the receiver requires several chips to realize the high-level functional blocks shown on Fig. 5.6 b). Shown are the data buses and clock signal distribution. To facilitate communication between chips, the 8-bit data words are

¹²The two XC4000 Xilinx memory tables per CLB can be programmed to be either 1-bit wide RAMs or logic function generators.

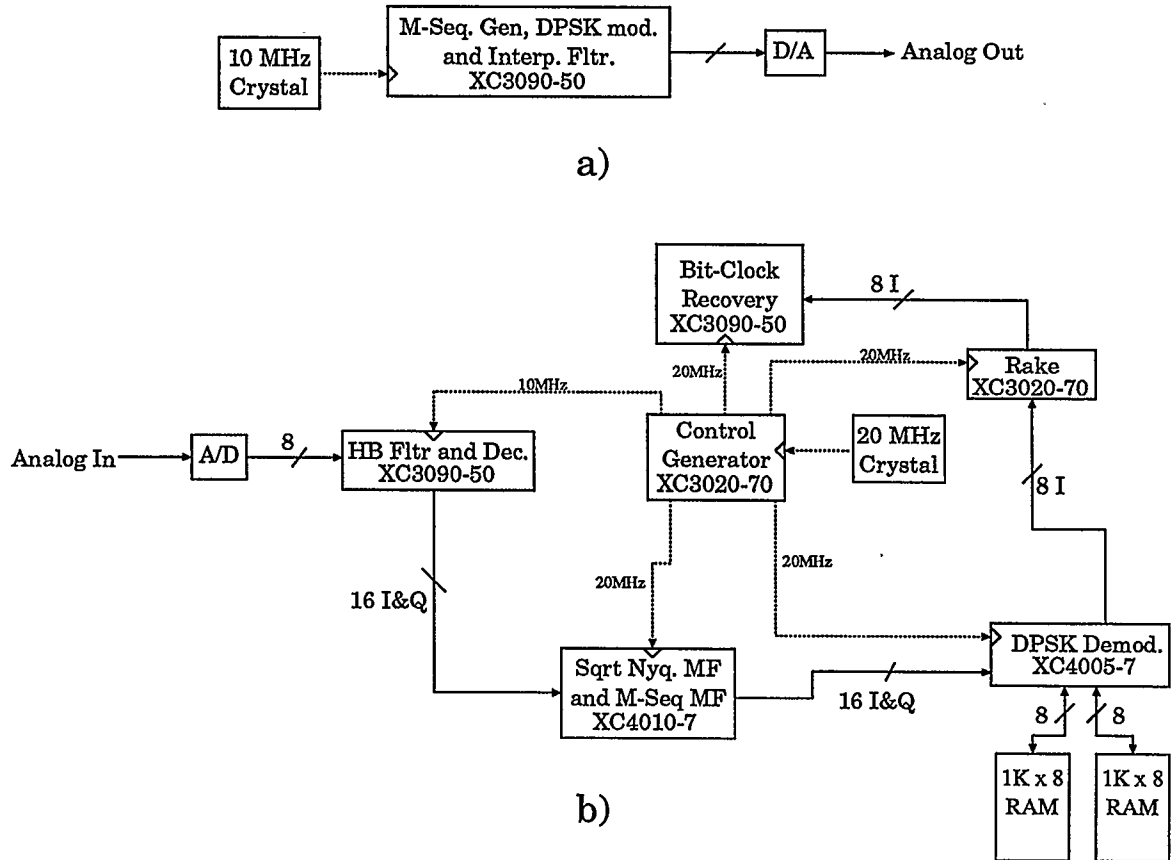


Figure 5.6 Allocation of Xilinx FPGAs to functional cells

transferred via parallel data buses at 2.5 MHz - rather than at a serial rate of 20 MHz. This meant that except for the half-band filters and decimation chains, which are a parallel design to begin with, additional hardware was required to convert from serial-to-parallel, and parallel-to-serial format. Parallel communication allowed for clock and control signal skews due to I/O buffer delays and routing, and non-ideal effects such as control and clock-edge degradation (details in section 5.4.6).

Data flow control signals (not shown on Fig. 5.6 b)) are generated from the central control generator. This approach, rather than generating *hand-shaking* signals in each individual functional block, was viewed as being optimum since it would be easier to pay particular attention to placement and routing of several sequential circuits

on several chips, than it would be to a single sequential circuit of several chips. Of course, it would not be prudent to choose this implementation if the entire design was placed on a single ASIC or custom designed chip; serial communication between the high level functional blocks would be more immune to signal skew/edge degradation since no external connections are needed.

To implement the design on a single chip, the total number of gate was estimated to be 40,000 for a 127 chip sequence.

5.4.4 Transmitter

In Chapter 4, Sec. 4.3, the DS-SS transmitter was shown to have four functional blocks: 1) an input selector with external select pins, 2) a DPSK modulator, 3) a programmable M-sequence generator, and 4) a interpolation-by 8 FIR. Not shown on the figure was another M-sequence generator that was connected to the test data input for bit-error rate experiments, and is the data latching circuitry that synchronizes the data with the M-sequence by latching the data in on the occurrence of M 1's in the M-sequence (see Chapter 4 Sec. 4.3.1).

All parts were designed using gate level descriptions and then were tested on the *Logsim* event-driven simulator.

5.4.4.1 M-Sequence Generator

The M-sequence generator, and the input test sequence generator, were implemented using the high-speed design discussed in Chapter 4, Sec. 4.3.1. To make the device programmable, the logic shown in Fig. 5.7 was inserted between each shift register delay unit to control the feedback taps g_n . Also shown is the initialization tap values given by h_n . The critical propagation path is g_n at the MUX_1 input, to

silicon area for reduced processing time. In choosing the parallel design, the assumption is made that propagation times through several layers of logic and routing delays do not limit the parallel design to any speed less than 10 MHz. It can be shown that the interpolating filter in Chapter 4 Sec. 4.3.2.2, it was shown that the interpolation process requires N_{mults} given by

$$N_{mults} = \left\lceil \frac{L}{I} \right\rceil_{int} + 1, \quad (5.24)$$

where L is the number of taps (odd numbers only), I is the interpolation factor, and $\lceil \cdot \rceil_{int}$ represents integer rounding. For the interpolation-by-8 filter, there are a maximum of four multiples per sample; consequently, for the full parallel design shown on Fig. 5.8, the propagation time has to be less than eight times the input chip period. This design represents a substantial decrease in the probability of incorrect data latching, compared to a parallel design that schedules multiplications and additions around a single adder,

Eight control signals were generated to orchestrate the flow of data through each of the parallel multiplier and adder units. Each unit is selected by the MUX just prior to new values being latched into the units delay chain, thus realizing the eight times chip-period processing time. The data is latched at the MUX output to prevent any *glitches* from propagating to the D/A converter.

The drawback to this high degree of parallelism is the number of multipliers. The technique used to reduce the gate count associated with multipliers was to take advantage of the ± 1 values of the input data stream by only having to pass the stored tap value if the data value is $+1$, or its negated value if the data value is -1 . The negated value - which is typically calculated from an exclusive-OR (XOR) and "add-one" operation - was approximated by only performing the XOR operation. Thus,

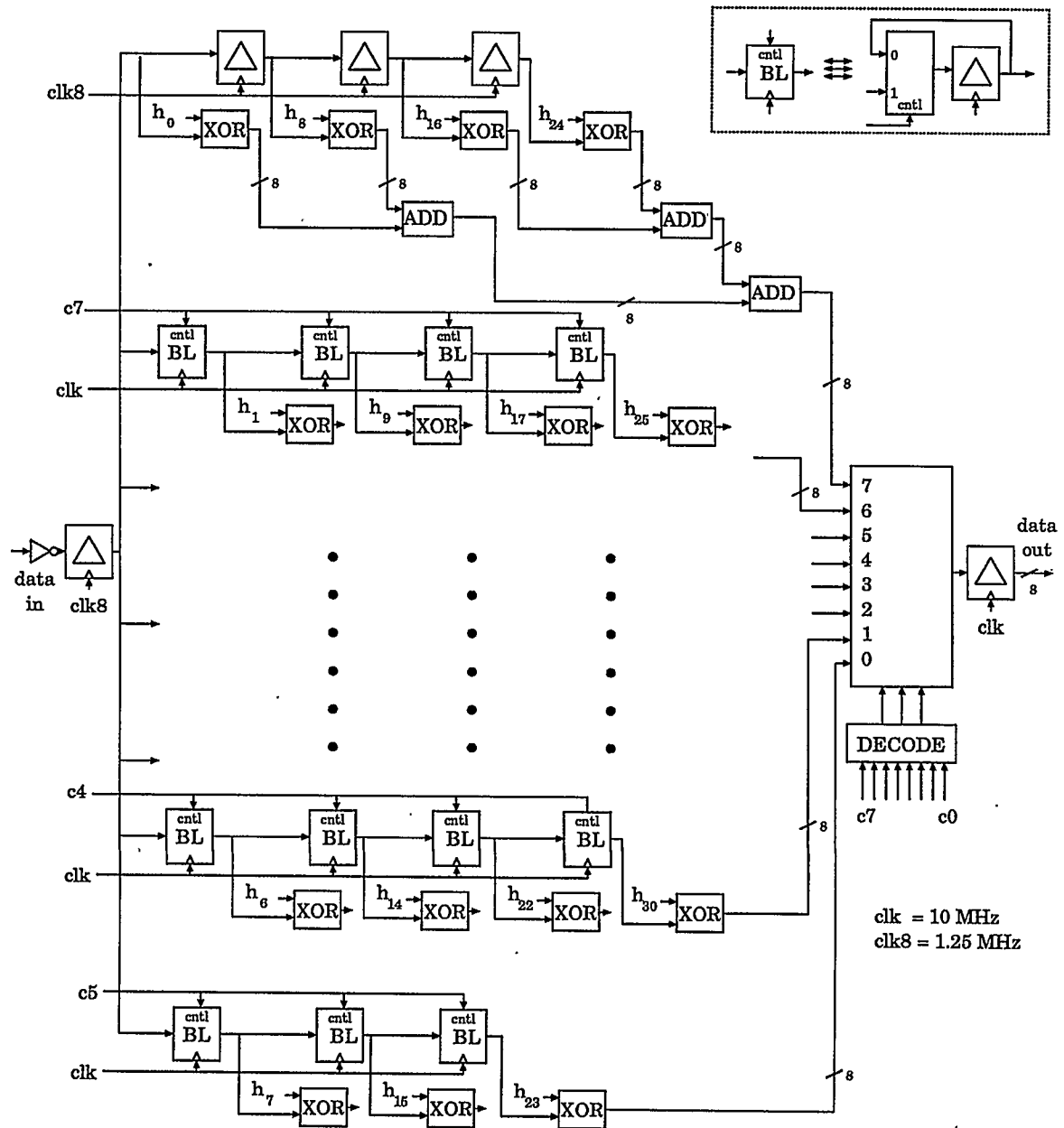


Figure 5.8 Interpolation - by - 8 digital structure design

silicon area was conserved by removing the add-one logic. It was shown by simulation that this approximation did not degrade system performance.

The total propagation through gates from a unit's input data register to its MUX input was estimated to be 18 nsec - representing a theoretical data rate of 55.6 MHz

(neglecting routing delays). With half-worst-case routing delays¹⁵, the maximum clock rate would be 23.3 MHz.

To prevent overflow, eqn. 5.25:

$$|x_k|_{max} < \frac{1}{\sum_{n=0}^{N_{mult}} |h(n * I + k)|} \quad (5.25)$$

was derived from eqn. 5.15 to represent the the fact that there are only N_{mult} summations at a single time. The maximum value for the k th shift is then used to determine the scaling of the filter coefficients.

5.4.5 Receiver

The basic receiver functional blocks - their functional description is discussed in Chapter 4, Sec. 4.4 - consist of several different design architectures that were implemented based on the most favourable area/time relationships. It was found that front-end functional blocks favoured high-area over long processing times, while the downstream blocks favoured low-area over short processing times. This was due to two factors:

1. High data rates (10 and 5 Msamples/sec.) in the front-end filter-and-decimation chains would exceed maximum processing rate limitations of bit-serial designs (8-bit words); however, lower downstream data rates (2.5 Msamples/sec.) are within bit-serial designs processing speeds.
2. The front-end filter and decimation chains are not chip area intensive due to the techniques highlighted by Lodge [26], whereas downstream structures have been determined to be chip area intensive.

¹⁵The worst-case routing delays were observed to be 50 nsec. on average for the XC4010 chip.

Two options for implementing the downstream functional blocks are to either implement complex control systems for scheduling data through single (or several depending on the processing bandwidth) parallel architecture arithmetic functional blocks, or to use low-complexity control signals in a highly parallelized bit-serial design. Since high-level design tools were available for bit-serial design, it was decided to implement the latter option and hopefully reduce the amount of time to finish this thesis.

5.4.5.1 Half-Band Filter and Decimate Chains

The halfband filter and decimation chains were built with two different structures as shown on Fig. 5.9. The complex filter was designed (probably over designed) to operate at a data rate twice the speed of the two real FIR filters. There are three control signals and three clock signals that are delivered from off chip. Multiphase and multirate clock signals are used to ensure that the control signals are properly latched (see $c0$ and $c1$ waveforms on Fig. 5.10), and also to reduce the number of gates, control signals and interconnects that would be required if the data was latched with a single global clock: a multi-rate system would have to use holding latches [48] which in-turn mean more gates. The signals are delivered to the chip by the control unit which ensures that the off-chip control signals, $c0$, $c1$, and $c2$, are switching on the falling edge of the clock. In this way, control signals may have approximately a $\frac{1}{2}$ clock period of propagation delay¹⁶ before there signals are improperly latched on chip.

The 10 MHz clock signal provided off-chip by the control generator, is routed to an external I/O pin to act as the A/D sample/convert signal. Chip internal buffer and routing delays plus the A/D convert time provided sufficient delay so as not to

¹⁶The maximum amount of propagation time is $\frac{1}{2}$ the clock period minus the latch setup time.

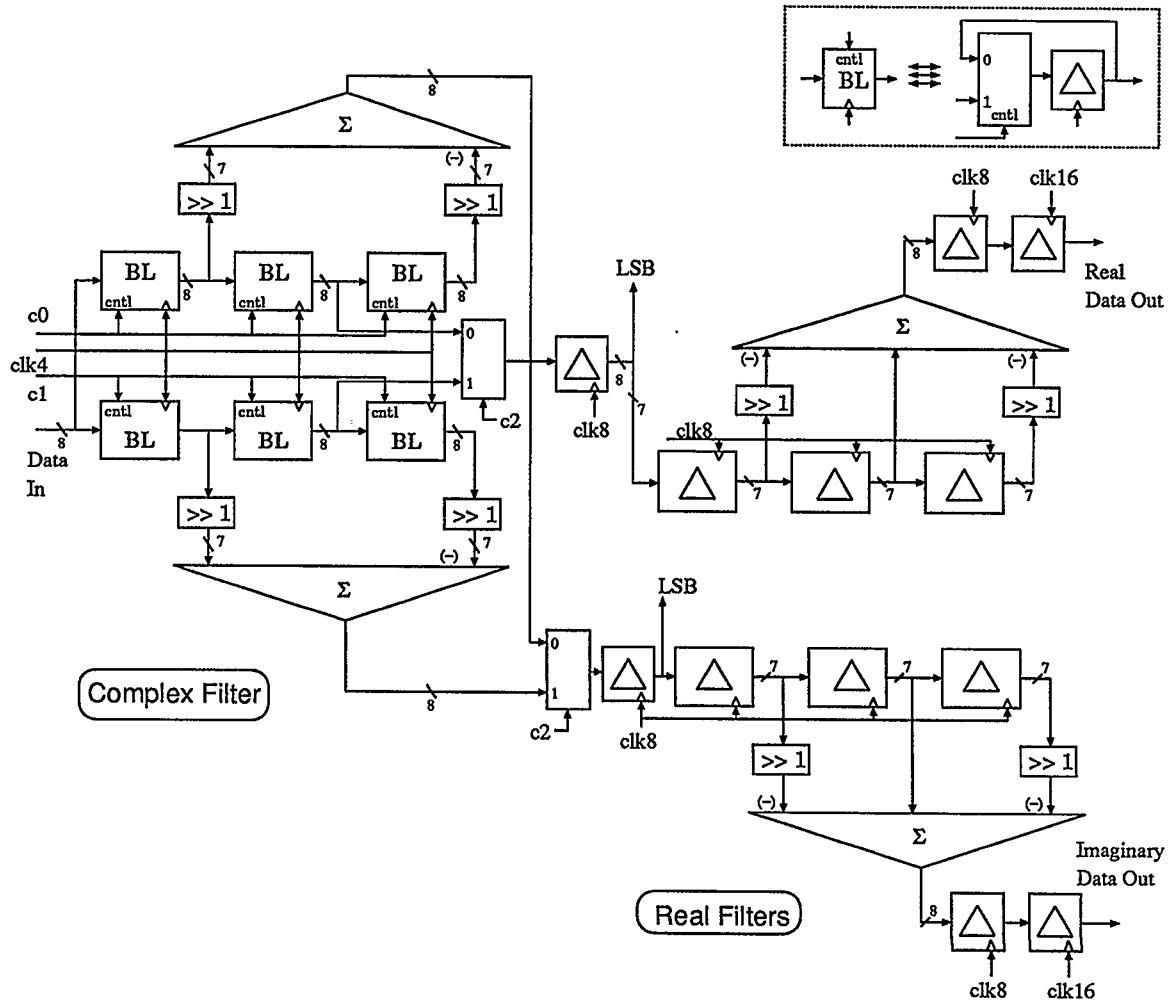


Figure 5.9 Halfband filter and decimate-by-two digital filter structure.

violate latch hold times.

5.4.5.2 Square-Root Nyquist Matched Filter

The square-root Nyquist matched filter was designed using Nomad to anneal the filter response to the desired frequency response. The program reduces the error function between the desired and actual response (caused by the quantization of filter coefficients), and also chooses the filter coefficients from a limited set of multiplier values that allow multiplication with canonic-sign-digit (CSD) multipliers and

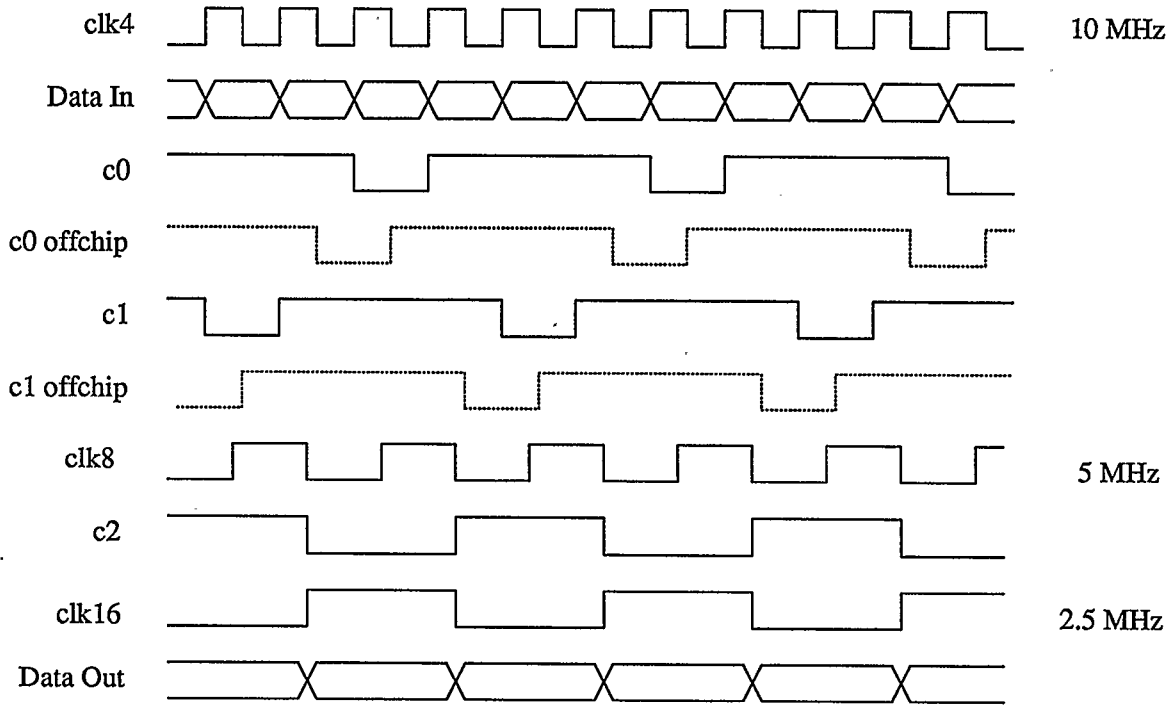


Figure 5.10 Halfband filter and decimate-by-two filter timing diagram.

down-shifters (DSs). It was found that selection of these filter multipliers resulted in substantial savings (almost 40%) in the number of gates required to implement this 13 tap filter.

The bit-serial filter was designed using a FIR tree structure as shown on Fig. 5.11. The signals $c_{n+x}; x = 0, 1, 2, \dots$ are used to control the flow of the eight-bit data word through the adder and divide-by-2 functional blocks. Only one signal, c_{n-2} , is supplied off-chip by the control generator to latch in the external parallel bus data to the chip internal. A single shift-register then generates subsequent control signals. Each functional block shows its latency, and the blocks labeled X are CSD multipliers while the D blocks are simple down-shift with sign extension operators.

The number of multiplies could have been reduced from $N - 1$ to $(3N - 1)/4$ by taking advantage of the filter tap symmetry. But adhering to eqn. 5.15 would have

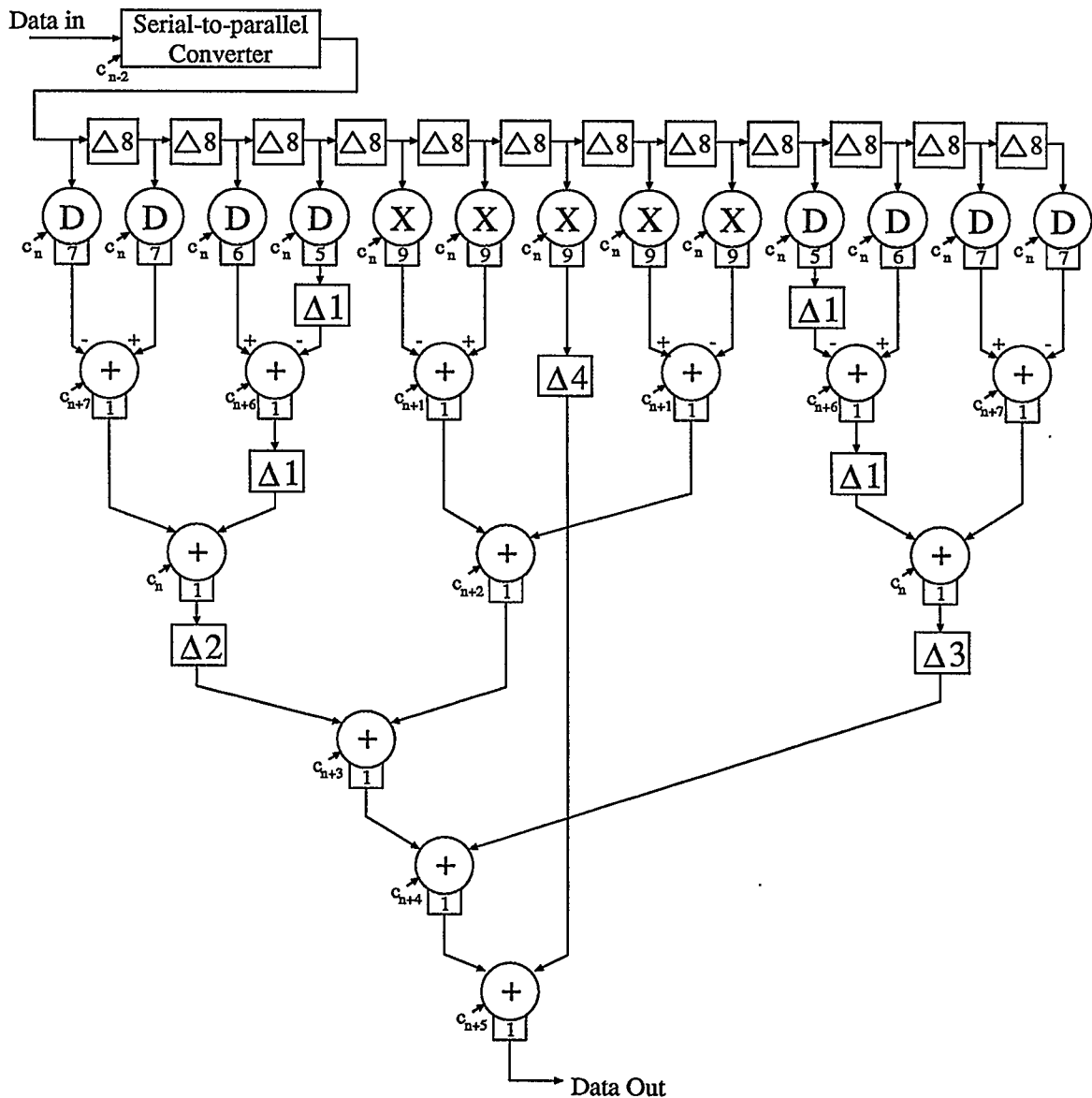


Figure 5.11 Halfband filter and decimate-by-two filter timing diagram.

resulted in a factor of two reduction in the signals dynamic range input.

The structure signal flow was written in the FIRST high-level design language before translated into Xilinx file format. In this translation, the delay-by-8 blocks are automatically implemented using address generators and configurable RAM blocks inside the FPGA cell array (Sec. 5.4.2). In this way, chip flip-flops and CLB-to-

CLB routing lines were conserved. Also, the translator automatically determines the fan-out of the address unit and will duplicate these units as requested. Because of the limited number of 4000 series chips, it was decided that a minimum fanout of ten could be used. Any smaller fanout required that the Square-root Nyquist filter and the M-sequence matched filter be configured on separate chips. A typical delay time of a logic element NAND gate was used to determine the maximum toggling frequency the address unit's output under several fanout conditions. Since the Xilinx equivalent information could not be obtained, the following information was obtained from LSI Logic Corp. [7]. Table 5.4 shows the fanout for a 1.5μ process NAND

Table 5.4 Delays for NAND gate (1.5μ drawn process).

low-high (nsec)	high-low (nsec)	Output Loading
0.6	0.2	1
1.1	0.5	4
1.6	0.8	8
2.7	1.5	16

gate. Using a worst case process variation factor of 1.74, the maximum delay of a NAND gate with fanout=16 is 4.7 nsec . This represents a maximum toggling speed of 212 MHz. For the M-sequence matched filter, it was found that the maximum system clock rate observed was 5 MHz (a 20 MHz rate was needed for a data input rate of 2.5 samples/sec). The large difference between the LSI estimated rate and that observed on the Xilinx FPGA could be due to the input loading factor for a

DFF. For example, an XOR gate has an input loading factor of 2. A DFF would have an input loading factor of at least two (the loading factor was not available) which means that the actual fanout is at least 20. Also, the Xilinx fanout delays for a CLB will be different than the LSI NAND gate. Another possibility is the routing delays which were observed to reach almost 50 *nsec* (Sec. 5.4.4.2).

The coefficient taps were not designed for overflow using equation 5.15 since it was felt that the "worst-case", for which eqn. 5.15 was derived, would not occur due to prefiltering by the halfband filter and decimate chains, and the divide-by-2 block inserted in front of the real half-band filter. It was also found that the signal's downstream dynamic range would also have been greatly reduced after the DPSK multiplications. The less stringent criteria of eqn. 5.16 was used instead. The actual frequency response slightly violates this criteria since it was difficult to adhere to with the finite coefficient data set for CSD multipliers.

The signal flow design supports the theory that the addition of opposite signed coefficient outputs, and then summing from small to large coefficient outputs, would also reduce the possibility of overflow.

A definite advantage would be to use saturation adders in this block. In this way, the information of an overflow would be maintained and then passed to the M-sequence matched filter.

5.4.5.3 M-Sequence Matched Filter

The M-sequence matched filter was implemented as a tree'd FIR structure shown on Fig. 5.12. The input data, x_n , enters the M-sequence matched filter at twice the chip rate. Consequently, there are two chip delays between each filter tap. Since the design was implemented using bit-serial 8-bit words, the delay between the taps would

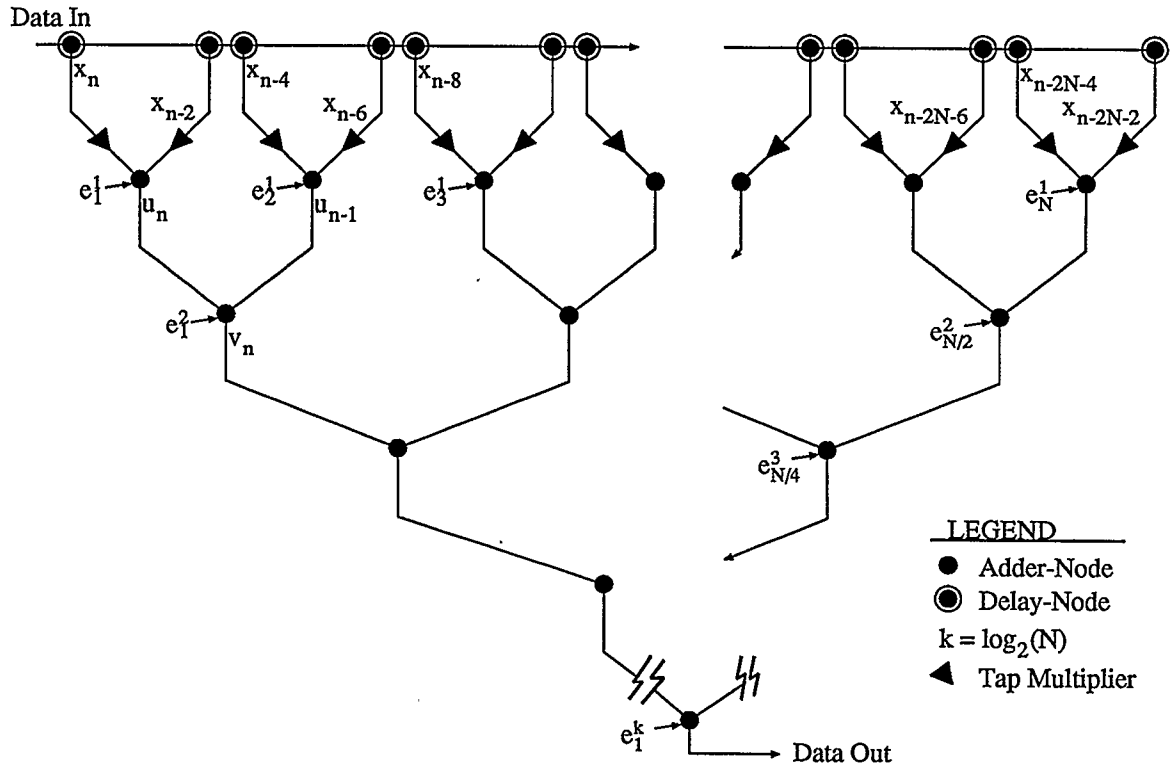


Figure 5.12 M-sequence matched filter with non-ideal quantization noise inputs

simply be implemented as sixteen DFFs connected as a serial shift register. With the Xilinx RAM configuration (see Sec. 5.4.5.2), these delays were implemented as 16 x 1 RAM cells with addressing units, thereby reducing the number of CLBs per tap from eight down to two. As in Nyquist matched filter, addressing units for these RAMs were duplicated to limit fanout.

The programmable ± 1 filter taps were implemented with adder units and an external XOR gate to the input *sum*. The coefficient bit (ie. 0 or 1) inputs to the adder carry-in and XOR gate to compute *invert and add one* two's complement multiplication by -1.

After each of the summation units, scaling was used (ie. a right shift or a divide-by-two) to prevent overflow at the next summation node. This results in an finite

arithmetic error, e_{ij} , that can be modelled as white noise with variance σ^2 and mean μ . From Fig. 5.12, the total error, e_T , at the output of the last tree summation node is given by

$$e_T = \sum_{i=1}^k \sum_{j=1}^{2^{k-i+1}} e_{ij} \frac{1}{2^{k-i}}, \quad (5.26)$$

where $k = \log_2(N)$ and N is the number of M-sequence chips, N_c plus one¹⁷ The mean, is derived using expectation principles as

$$\begin{aligned} \mu_T &= E[e_T], \\ &= 2k\mu_{\epsilon_a}, \end{aligned} \quad (5.27)$$

where μ_{ϵ_a} is the addition error mean given by Table 5.1. The variance of the error at the output of the tree is given by

$$\begin{aligned} \sigma_T^2 &= E[e_T^2] \\ &= \frac{1}{2^{2n}} \sigma_{\epsilon_a}^2 2^{n+2} (2^n - 1), \end{aligned} \quad (5.28)$$

where $\sigma_{\epsilon_a}^2$ is error variance given by Table 5.1. Given the fractional maximum value of the Nyquist Pulse at the SRN filter output, A , the signal-to-scaling noise ratio is given as

$$\begin{aligned} SNR_s &= \frac{\frac{(2^k-1)^2}{(2^k)^2} A^2}{\sigma_T^2}, \\ &= \frac{(N-1)A^2}{\sigma_{\epsilon_a}^2 4N} \end{aligned} \quad (5.29)$$

For B -bit data words and N number of taps (where N is large), equation 5.29 is reduced to

$$SNR_s = 20\log_{10}(A) + 10\log_{10}(2^{2t'(B-1)+2}) \quad dB. \quad (5.30)$$

¹⁷The development of the statistical error is greatly simplified by using N instead of N_c , where $N = N_c + 1$. This difference is negligible when N_c is large.

It is apparent from Equ. 5.30 that with the tree'd structure, the scaling noise is independent of the number of taps in the M-sequence matched filter.

Another option to implementing the M-sequence matched filter is to use a single B -bits wide accumulation register, similar to the approach used in DSP micro processors. This method has the advantage of using less hardware; however, the work by Coldham [8] has shown that this method does not perform as well as the tree'd method. A possibility that may be attractive, but was not implemented, is the use of saturation arithmetic at each adder node. This would eliminate the need of the scaling units and allow increased dynamic range of the input. Although the correlation peak will be clipped, its sign is still intact whereas if overflow arithmetic is used, the sign is inverted, resulting in a bit error.

5.4.5.4 DPSK Demodulation

Gate level descriptions of the DPSK demodulator functional blocks shown on Fig. 5.13, were generated using FIRST. The block connections were made using logsim netlist code - thereby allowing simulation of the RAM DPSK delay units. The real and imaginary data streams from the two M-sequence matched filters are stored in circular buffered RAMs which realize a delay of one symbol period (refer to Chapter 4, Sec. 4.4.4). In each chip period, the parallel input data bus is latched into the parallel-to-digit (PTOD) converters at the same time the tri-state buffer (resident on the Xilinx chip) goes into high impedance mode; thus allowing the delayed data values to be latched from the RAM to the second PTOD converter. This operation occurs in the first half of the chip period while the latter half is used to write the input data values to the RAM. The PTOD converters implement a divide-by-2 internally to prevent an overflow at the DPSK adders. As in the case of the matched filter, it

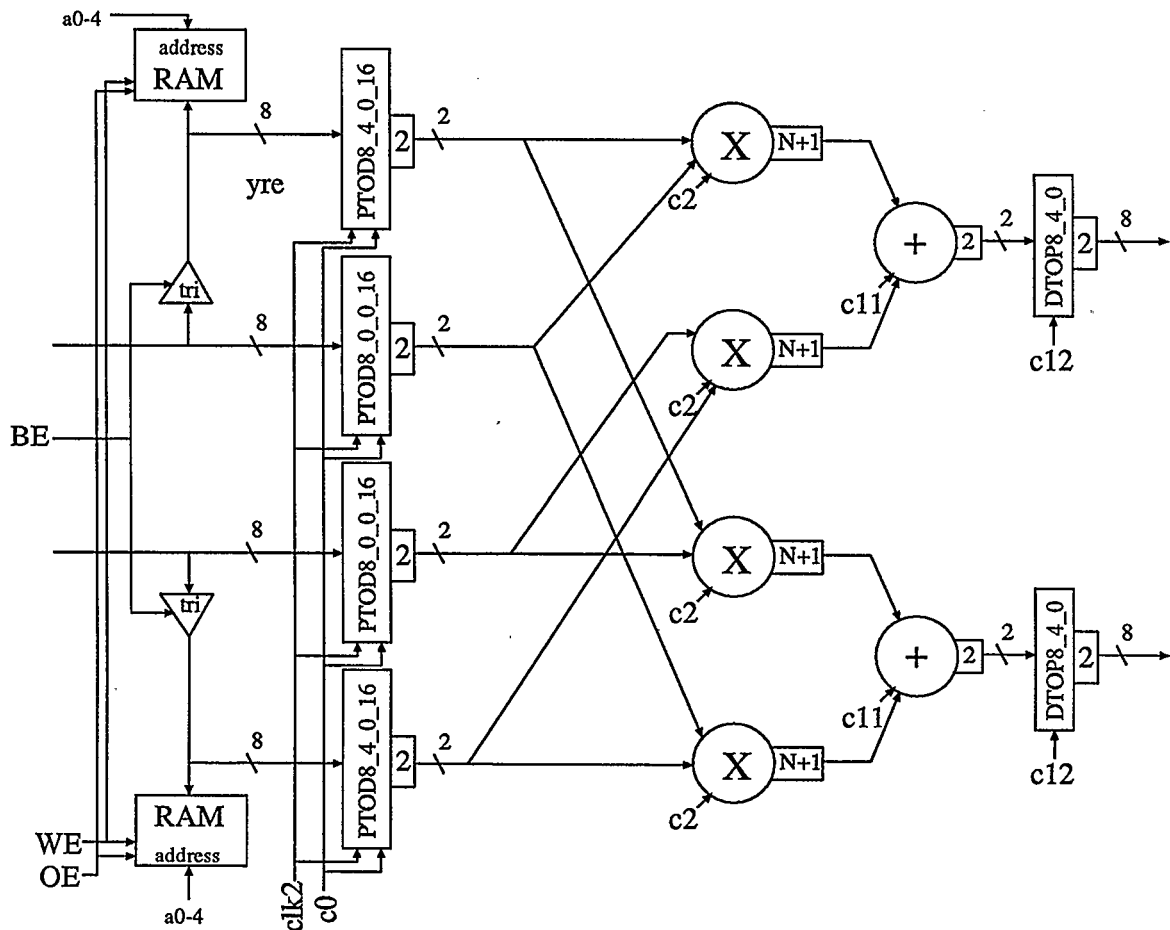


Figure 5.13 DPSK hardware layout

would be more advantageous to use a saturation adder thus eliminating the need for the divide-by-2 and hence maintaining a better SNR_Q .

Fig. 5.14 shows the timing diagram for the control signals supplied by the control unit. For the 2.5 MHz data throughput data rate, the RAMs must be fast enough so as not to violate the 50 nsec setup time. The setup time is measured from when the chip enable (CE) signal goes high to when the first rising clock edge goes high after $c0$ goes high. Non-ideal routing delays CE could lessen the period; hence, it would be wise to choose a RAM after the routing delays are known. The buffer enable (BE) signal was also designed to account for any possible routing delays that result in the

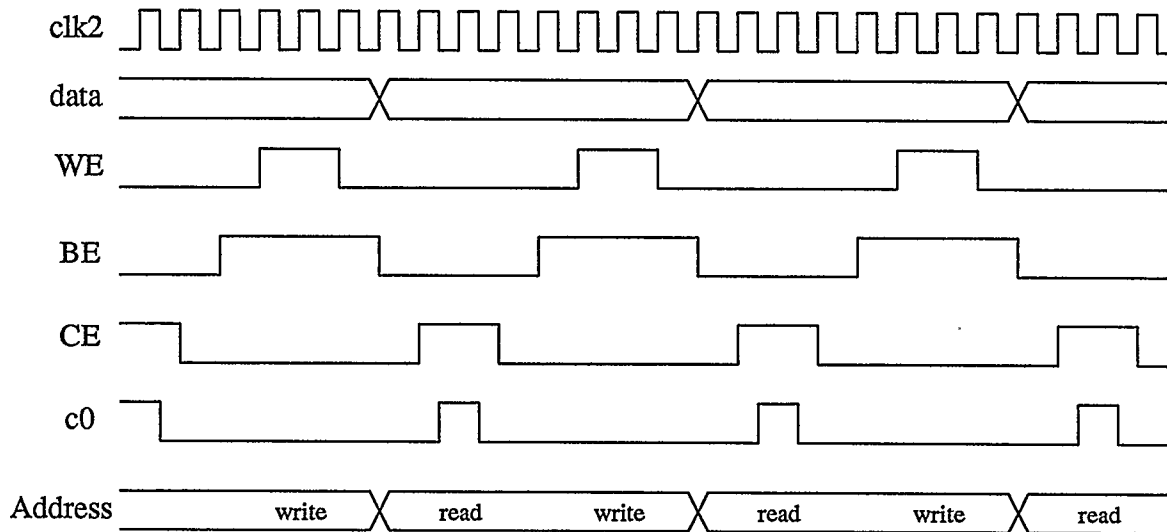


Figure 5.14 DPSK timing diagram.

data buses from driving the RAM during a read cycle. The BE signal could have up to a 50 nsec delay before this occurs.

Although the external RAM implementation of the DPSK delay means more external connections to the chip, the advantage to this scheme is that the M-sequence length can be easily increased without reconfiguration of the internal chip layout. It also means that slower RAMs could be used when tailoring the system to lower speeds, thereby realizing the cost savings of low-speed RAMs. Implementing the RAMs on-chip is a possibility; however, this would increase chip area (particularly for large M-sequences) and thus increase the chip price.

5.4.5.5 Rake

The Rake was designed as a sixteen-tap FIR filter with programmable tap values. The taps take on the values 1 and 0 as determined by the threshold detector unit shown on Fig. 5.15. The threshold detector compares the absolute value of the input data stream against a threshold value. The output is a logic 1 or 0 that controls a

5.4.5.6 Bit-Clock Recovery

The bit clock recovery system was implemented using the completely digital phase locked-loop (PLL) shown on Fig. 5.16. The input signal f_1 is the most significant

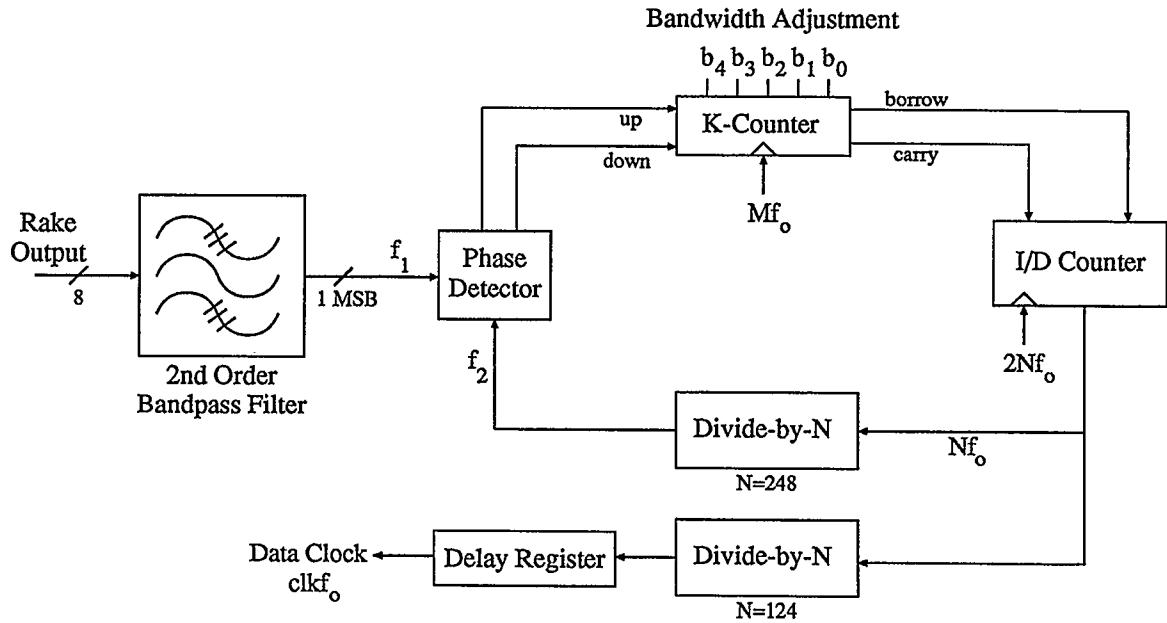


Figure 5.16 First-order bit-clock recovery phase locked-loop.

bit of the second order lattice bandpass filter output. The lattice filter, tuned to the bit symbol frequency of 40.322 kHz, is used to isolate the bit symbol fundamental frequency and to remove dc. Since the MSB represents the fundamental as a 50% duty cycle square wave, it is important that the low frequencies around dc and dc itself be removed to prevent jitter from causing the sampling point to drift.

The lock-in range (see Chapter 4, Sec. 4.4.6) is adjusted by the external pins b_4 to b_0 . The K-counter and Divide-by-N counters are both implemented using Johnston counters with additional logic to provide a logic "0" to the DFF inputs when N is reached. The logic is segmented into several layers, with each layer separated by

DFFs, to prevent large propagation delays.

The increment/decrement (I/D) counter is the only functional block in the entire transceiver design where the clock signal is gated. The clock signal $2Nf_o$ is gated by a signal supplied from a sequential circuit clocked by $\overline{2Nf_o}$. The state representation of the gating signal b_g is shown in Fig. 5.17. By clocking the sequential circuit with

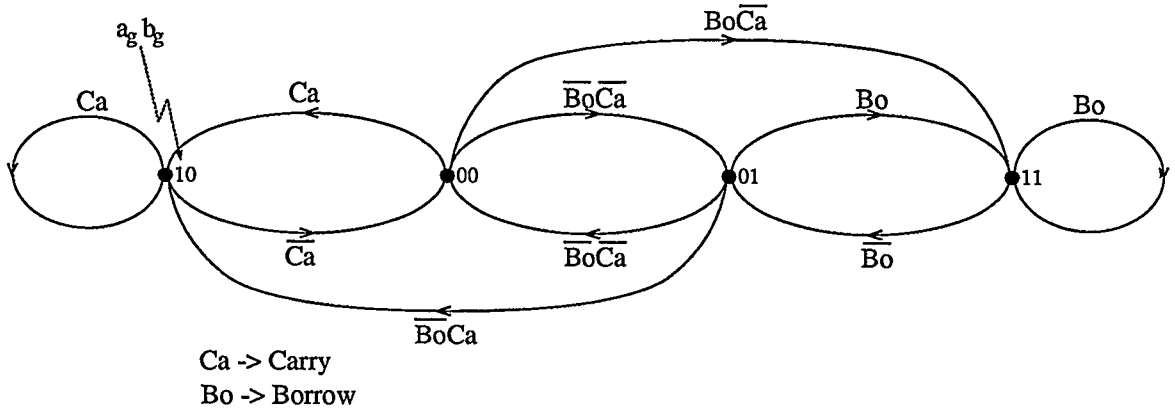


Figure 5.17 State representation of the I/D counter.

$\overline{2Nf_o}$, the gating signal can have a maximum routing delay of 20 nsec (half the clock period) without causing any adverse glitches to propagate to the divide-by-N counter.

The second order bandpass filter is a lattice realization of a IIR filter [21]. The lattice was chosen over the direct form realization of the IIR because the tap values are less than 1, a characteristic of such filters. A *Digicap* netlist description (see Sec. 5.4.1) of the lattice was generated and then input into *SNAFU* to automatically generate a FIRST (digit-serial) code description. The filter is the only high-level functional block that has an asynchronous operation: the control unit issues a *wake-up* signal to the filter, the filter processes the data before the next data value is available from the Rake, and then issues itself a *sleep* command to wait for the next

data value. All control signals are supplied by the lattice internal logic.

The delay register is used to "tune the sampling location" (or rising edge of the $clkf_o$ signal) so that it aligns with the Rake output. This method assumes that the difference in the two clock frequencies, f_1 and f_2 , will not exceed a prescribed frequency difference that would cause adverse movement of the "tuned sampling location". It should be noted that a better suited bit-clock recovery system would be a second-order PLL (like those mentioned in Chapter 4, Sec. 4.4.6) because the frequency and the phase errors would both be corrected when the loop is locked.

5.4.6 Chip Communication and Control

All clock and control signals are provided by a single control generation functional block. The block itself is clocked by a single 20 MHz crystal oscillator. The input clock drives several sequential circuits that are initialized by a single global reset line. Initialization is realized with AND or OR logic gates located at the input to the sequential circuit DFFs. Divided-down versions of the input 20 MHz signal are derived straight from the DFF outputs while control signals were generated with additional logic connected to the DFF outputs when needed. The control signals were latched 180°s out-of-phase to the external chip outputs to ensure that glitches would not be mistaken as authentic control signal levels.

A single starting point in time was used as a reference to design the AND and OR logic. If a functional block was altered such that its total latency was different than previously defined, the AND and OR gates were manually changed to reflect the new latency. Fig. 5.18 illustrates this approach. In Fig. 5.18, all the control sequences were initialized with respect to the shaded area. The shaded area is referenced as time zero for $clk2$. The Rake control signal, $c0$, is generated on time three of $\overline{clk2}$ to

latch data into the Rake at approximately the middle of the DPSK data valid region. If the DPSK unit was altered so that latencies cause the data to switch at or close to c_0 , then the reset logic (ie. the AND and OR gates) would be manually relocated at the DFF inputs to adjust c_0 back to an optimum location. In this way, any non-ideal effects caused by the chip-to-chip bus interconnects would have subsided before the data is latched.

The initialization point was carefully picked in relation to the required control signals in order to minimize the logic layers needed to generate such signals from the sequential logic DFF outputs.

5.5 Summary

The entire layout of the functional blocks have been described in relation to non-ideal effects due to hardware (ie. fanout, routing, etc.), and those due to finite arithmetic (ie. quantization noise, scaling, overflow, etc.). The methodology of control and clock signal generation has also been described.

Due to the size of the receiver, the design was segmented into several separate functional blocks that were implemented on separate Xilinx FPGA's for prototyping. The designs used parallel, bit-serial, and digit-serial architectures to accommodate speed and the limited number of Xilinx chips that were available. It was found that the Xilinx 4000 series FPGAs are more suited to bit-serial designs.

Interchip communication used parallel data buses. With this protocol, the bus data rate is lower than if the data was communicated at bit-serial data rates; consequently, the data latching between chips is more immune to bit errors. The interchip communication was orchestrated by an single control unit that resided on one of the more faster 2000 series chips.

The gate-level and high-level design tools made implementation of the receiver possible. Design of such a receiver from grass roots would be a labourous task. It would not be possible to take advantage of gate reduction and optimization algorithms that allow fast and efficient translation of algorithms to silicon.

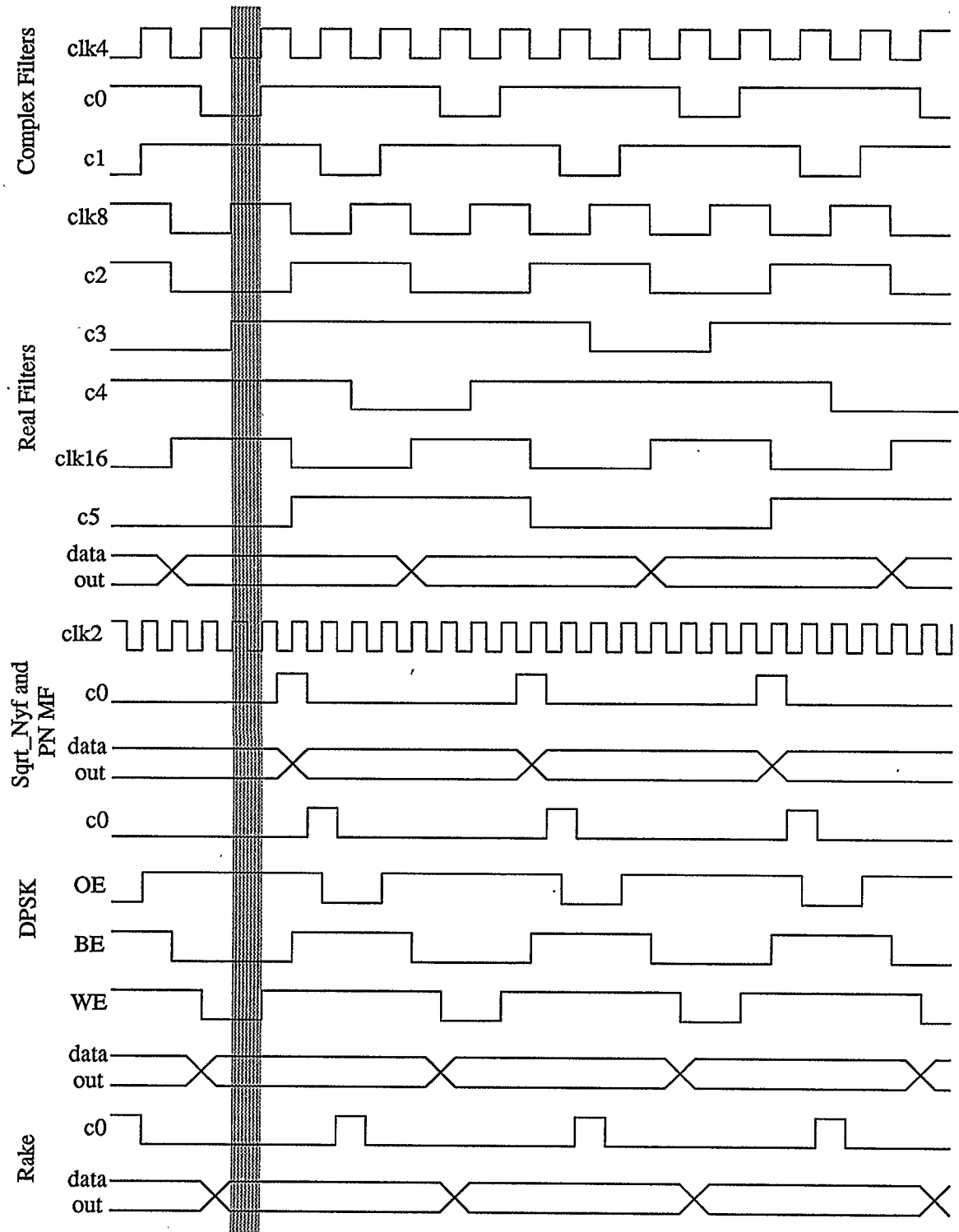


Figure 5.18 Control unit timing diagram.

CHAPTER 6

DRAKE TRANSCEIVER PERFORMANCE

6.1 Introduction

In this chapter, an experimental comparison is made between the DRake transceiver hardware, a simulation package, and theoretical results. Output waveforms and probability of bit-error rates (BER) are presented. The methodology in designing the DRake transceiver was first to develop a floating-point and fixed-point model. Next, the model's operation was verified by comparing the BERs of the M-sequence matched filter output (after the DPSK block) to the well known BER curve for binary differential phase shift-keying (BDPSK). The fixed-point model was then used as a verification tool for testing non-ideal effects such as modulation frequency offset, attenuation of input signal, clipping, and overflow. With these non-ideal effects, the model's BERs were compared to the hardware prototype. The model was instrumental in the development of each hardware functional block. As each block was added, the blocks output observed on the oscilloscope was visually compared to the model's output. BER tests for the "validation process" were carried out for an additive white gaussian noise (AWGN) channel.

With the exception of the PLL, all the hardware components were mirrored in the fixed-point model to accurately simulate the effects of finite-arithmetic errors. The PLL was not simulated.

Multipath channel results are presented for the model only since there was insufficient time and resources to develop the radio frequency (RF) equipment required. However, the results should be characteristic of the hardware's actual performance

in a multipath environment since similar performance was observed for the AWGN channel.

6.2 Theoretical, Simulated, and Hardware Waveforms

The purpose of looking at the output waveforms was to trouble shoot any discrepancies between theory, simulation, and hardware before the BER measurements were taken. The first step was to look at the transmit and receive Square-Root Nyquist spectrums to verify pulse shaping filter operation and to check finite arithmetic overflow (overflow would result in high frequency spectra components). The next step was to compare the simulation output to the hardware M-sequence matched filter, DPSK, and DRake output amplitudes. This ensures that correct arithmetic computations are executed and also that no overflows are occurring. Finally, the PLL output was observed under a noiseless channel to verify its operation and to tune the sampling point to an optimal position on DRake output. With these tasks complete, the BERs were then measured.

6.2.1 Nyquist-Pulse Matched Filter Output

A Fourier analysis of the overall transmit and receive square-root Nyquist filter, and the halfband filters (see Chapter 4, Sec. 4.3.2.2, 3.5.1, and 4.4.2, respectively) was determined for the hardware. Fig. 6.1 shows the measured magnitude response and the 6 dB point. The sampling rate was decreased by a factor of 200 to accommodate the spectrum analyzer; consequently, the repeating spectrum is shown at 12.5 kHz instead of at 2.5 MHz. The bandwidth of the received signal was designed to be one quarter the sampling frequency, f_s , as presented by Lodge [26]. Fig. 6.1 shows the receive square-root Nyquist filter response with the designed 6 dB attenuation

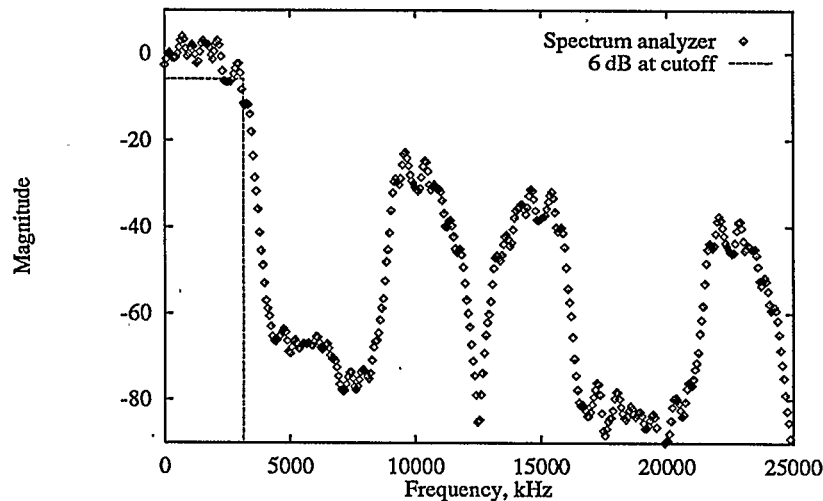


Figure 6.1 Raised cosine frequency response for rolloff = 0.35.

bandwidth at $\frac{1}{4}f_s$ or 3125 kHz.

Note that this plot demonstrates the limitations of a practical hardware system. Since the signal is sampled at two times the Nyquist rate, the designed filters will be more than adequate to prevent aliasing. If the signal was sampled at the Nyquist rate (or 6250 kHz in Fig. 6.1), higher rolloff filters would be necessary to prevent any frequencies just outside the 3125 kHz band from aliasing. Such a filter would have to be an unrealistic "brick walled" filter or "ideal" filter as described in Chapter 4, Sec. 4.3.2.1.

The time waveform of the Nyquist-pulse is shown on Fig. 6.2. The time period from zero to 1600 represents one 31 chip sequence as described in Chapter 4, Sec. 4.3.1. Also note that the sample levels appear to oscillate over the period of one M-sequence. This is due to the frequency offset between the transmitter's modulator and the receiver system clock. These oscillations will be shown to effect the BERs of the system (see Sec. 6.3.4.2).

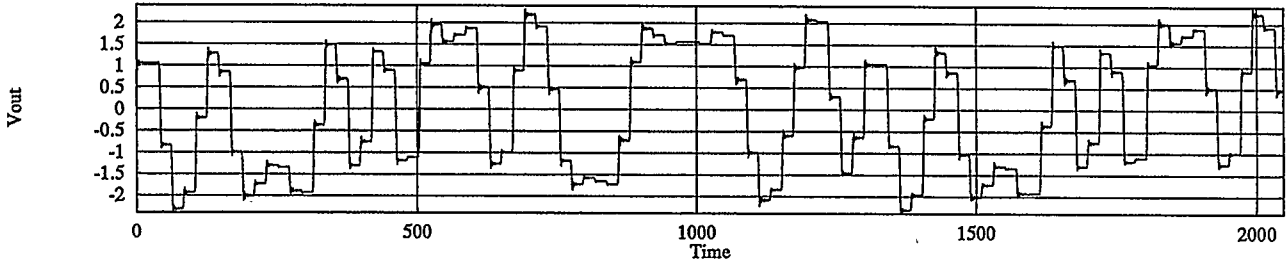


Figure 6.2 Nyquist-pulse time waveform.

6.2.2 DPSK, DRake, and PLL Output

The DPSK output is shown on Fig. 6.3. Based on the output magnitude of

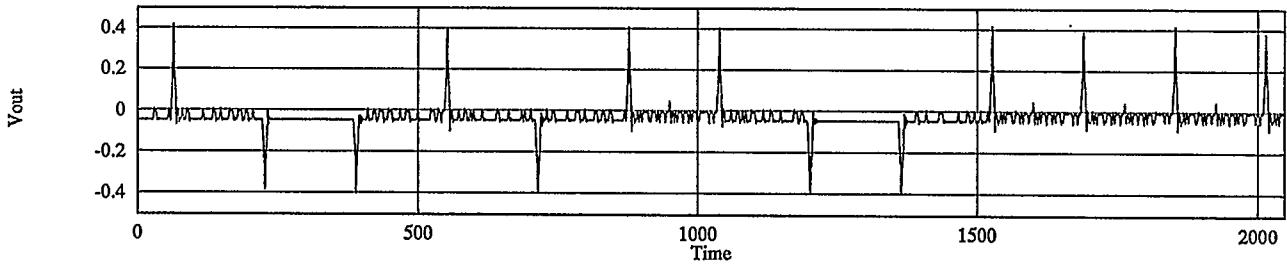


Figure 6.3 DPSK output: zero noise

approximately 2 volts for the Nyquist Pulse, the expected correlation pulse magnitude for the DPSK output is

$$Vout_{DPSK} = \left(\frac{Vout_{NYF}}{DAC_{max}} \right)^2 * \frac{31}{32} * \frac{1}{2} \quad (6.1)$$

which gives $Vout_{DPSK} = 0.375$ volts (normalized to the maximum DAC output (DAC_{max}) of 5 volts). These results compared favourably with the values shown on Fig. 6.3. The factors $\frac{31}{32}$ and $\frac{1}{2}$ represent the M-sequence matched filter and the DPSK gain (Chapter 5 Sec. 5.4.5.3 and 5.4.5.4).

There is also a dc biasing of approximately 0.03 volts which can be attributed to fixed-point arithmetic truncation. This effect is demonstrated in Figures 6.4 and 6.5 where truncation has resulted in an offset in the received Nyquist pulse. This offset

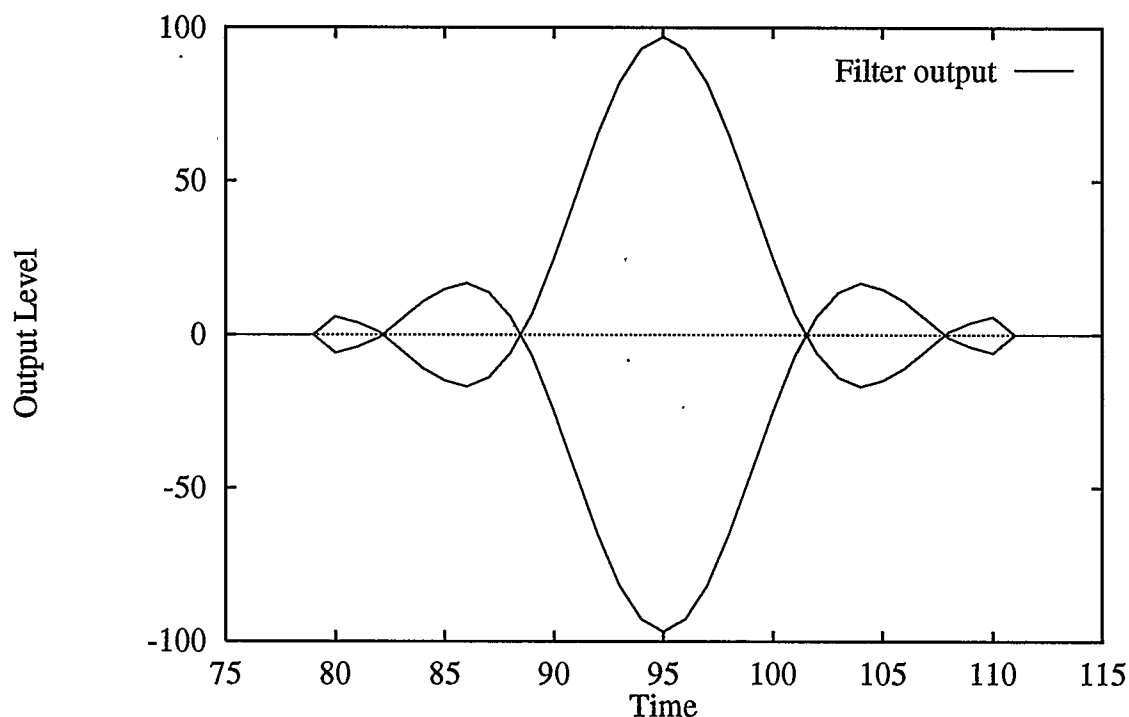


Figure 6.4 Transmitted Nyquist pulse.

should be taken into account when setting the threshold detectors for bit recovery.

The noise between the correlation peaks on the abscissa represents self-noise which occurs at instances where the M-sequence modulated bits switch polarity. In instances when they do not, the cross-correlation noise is constant.

The DRake, PLL, and sampled output bit stream are shown on Figures 6.6, 6.7, and 6.8. With manual adjustment of the DRake threshold, most of the cross-correlation noise (shown in Fig. 6.3) can be removed from the signal as demonstrated

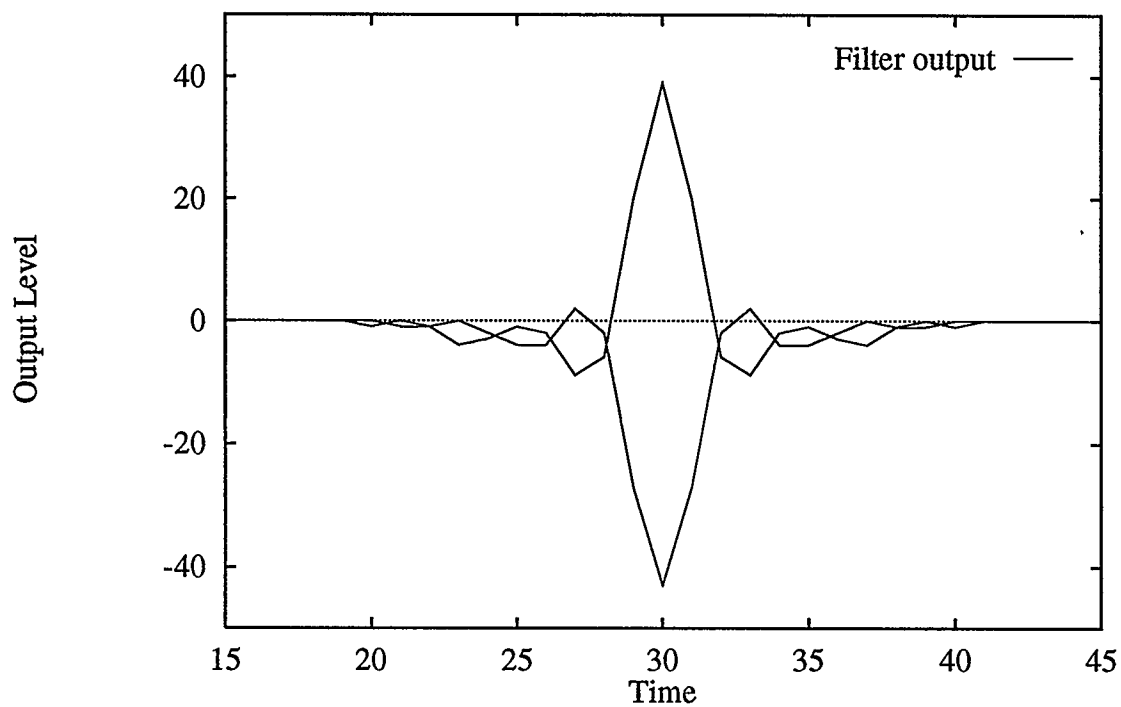


Figure 6.5 Received Nyquist pulse.

in Fig. 6.6. Fig. 6.6 also shows the effect when cross-correlation noise exceeds the threshold (this occurred in the period 1500 to 2000).

The DRake output goes to the PLL which outputs a sampling clock (Fig. 6.7) to

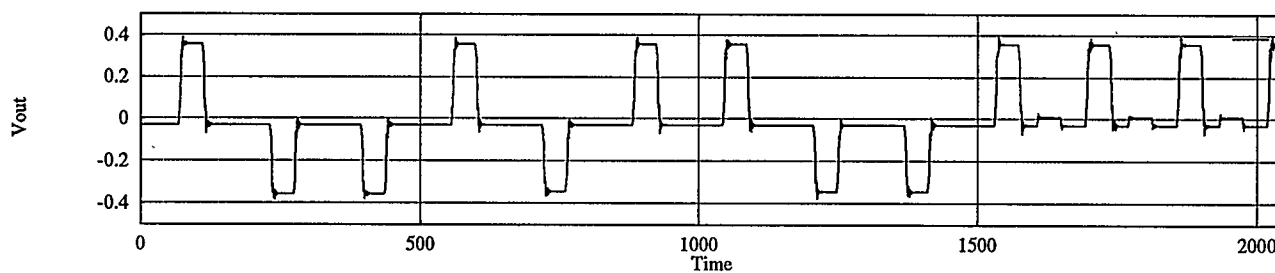


Figure 6.6 DRake output: threshold set at .016

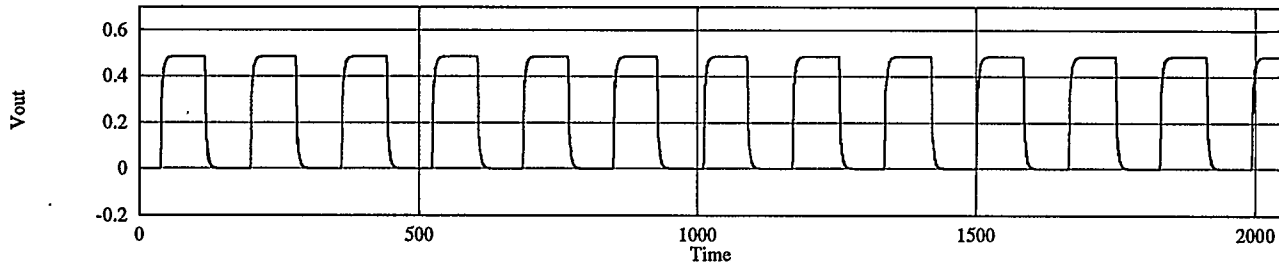


Figure 6.7 Phase lock-loop output

recover the data from the Rake output. The recovered data is shown on Fig. 6.8.

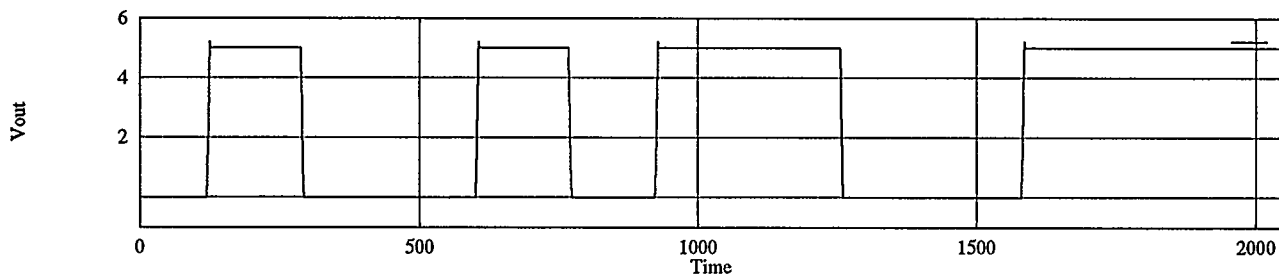


Figure 6.8 Recovered data bits from 0/S

6.2.3 DPSK Oscillations

Not obvious from Fig. 6.3 are the oscillations that occur due to a combination of gain offsets between the inphase and quadrature filtering stages, and truncation error biasing. Another non-ideal effect is the frequency offset between the transmitter and receiver clocks. Such frequency offsets were determined by Kavehrad [30] to attenuate the signal with a sinc pulse shape where the zeros occur at frequency offsets that are multiples of the bit rate (ie $1/T_B$). He proposed that this could be a method

of reusing like codes in adjacent cells.

6.2.3.1 Gain Offsets and Truncation Error Biasing

Oscillations first observed on the prototype were investigated using the fixed and floating-point models. At first, it was believed that the oscillations were due to the truncation noise. As shown on Fig. 6.9, the truncation noise is modeled as additive

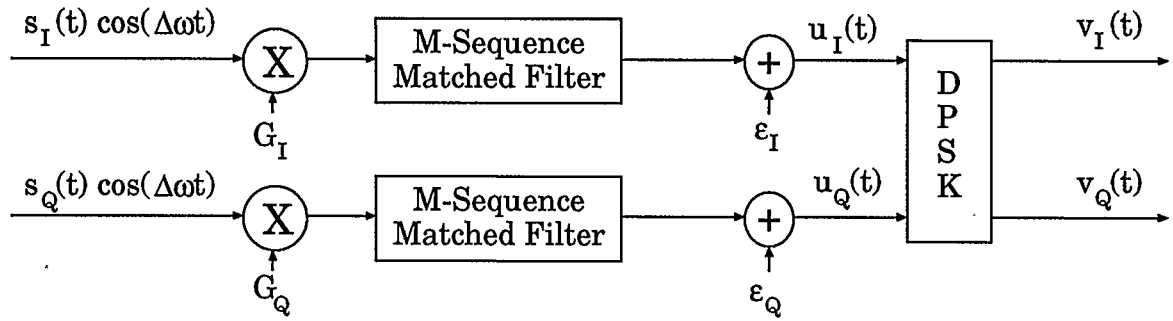


Figure 6.9 Non-ideal effects on DPSK output.

noise. After correlation, the M-sequence matched filter output, $u(t)$, is

$$u_I(t) = G_I R_I(t) \cos(\Delta\omega t) \quad (6.2)$$

$$u_Q(t) = G_Q R_Q(t) \sin(\Delta\omega t) \quad (6.3)$$

where $R(t)$ is the correlation of the input signal $s(t)$, $\Delta\omega$ is the transmitted data, G_I and G_Q are the in-phase and quadrature system gains, and ϵ is the truncation noise. Assuming that the inphase and quadrature gains are equal, ie. $G_I = G_Q = 1.0$, and the truncation noise is equal, ie. $\epsilon = \epsilon_I = \epsilon_Q$, the in-phase DPSK output, $v_I(t)$ can be approximated as

$$v_I(t) \approx R(t)R(t-T)\cos(\Delta\omega T) + \epsilon R(t)[1 + \cos(\Delta\omega T) - \sin(\Delta\omega T)] + 2\epsilon^2 \quad (6.4)$$

Obviously, the second term will attribute to variations in the DPSK output.

The floating-point model was used to verify eqn. 6.4. It was found that oscillations were prevalent in the floating-point model - but to a lesser degree- although the effects due to truncation were virtually eliminated. Further analysis showed that in the floating-point model, when $\omega_I = \omega_Q = 0$, the gains, G_I and G_Q , are not equal. This was due to the halfband filter chain. Consequently, eqn. 6.3 can be written as

$$\begin{aligned} v_I(t) = & R(t)R(t-T) \left\{ \cos(\Delta\omega T) \left[G_I^2 \cos^2(\Delta\omega t) + G_Q^2 \sin^2(\Delta\omega t) \right] \right. \\ & \left. + \cos(\Delta\omega t) * \sin(\Delta\omega t) * \sin(\Delta\omega T) \left[G_I^2 - G_Q^2 \right] \right\} \end{aligned} \quad (6.5)$$

It is obvious that unless $G_I = G_Q$, the DPSK output will oscillate.

Both eqn. 6.4 and 6.5 show that the non-ideal gain offsets and truncation will result in undesirable oscillations in the DPSK peak amplitude. These oscillations will in turn result in an increased BER at high SNRs.

6.3 Theoretical and Hardware Probability of Bit-Error Curves

This section presents the BERs for the model, hardware, and theory. The BERs for BDPSK transmission over an AWGN environment was first used to calibrate the computer simulation to the theoretical equations. The simulation was then used to verify the hardware performance under the same environment. Since there was not enough time to develop radio frequency (RF) hardware, the hardware performance in a multipath environment could not be compared to theory. However, since the hardware performance was verified for a AWGN channel, it is assumed that the hardware's performance would not deviate greatly from the simulation.

6.3.1 Theoretical DPSK

The theoretical BER vs. SNR for BDPSK transmission over an additive white Gaussian noise channel is given by Proakis [42] as

$$P_b = \frac{1}{2}e^{-\gamma_b}, \quad (6.6)$$

where the SNR, γ_b , is equal to

$$\gamma_b = \alpha^2 \frac{\varepsilon_b}{N_o}, \quad (6.7)$$

where α is the channel and processing attenuation, N_o is the channel noise spectral density function, and ε_b is the bit energy at the input to the DPSK functional block. Equation 6.7 can be re-written in terms of the peak Nyquist pulse signal level, $\mathcal{A}$, at the point of sampling, and the noise variance, σ_N^2 :

$$\gamma_b = \alpha_b \frac{\mathcal{A}^2}{2\sigma_N^2} \quad (6.8)$$

Eqn. 6.8 provided the bases for comparing the simulation and hardware results to theory.

6.3.2 Signal-to-Noise Calibration

The first task was to compare the simulation BER curve to the theoretical curve given by eqn. 6.6. To do this, the signal was synchronously sampled at the input to the Rake functional block (ie. at the DPSK output was sampled at the correlation peak). The additive white Gaussian channel was simulated by summing 24 uniformly distributed random variables (central limit theorem). The standard deviation applied to the noise generator was calculated from eqn. 6.8 where $\mathcal{A}$ was determined by observing the M-sequence matched filter peak output, and α_b^2 was determined from root-mean-square (RMS) calculations at the noise input to the receiver and noise

output at the M-sequence matched filter. The abscissa SNR values were chosen to give adequate representation of the BER curve.

In the hardware, the SNR was calculated by

$$SNR = \left(\frac{V_{Rx}}{V_N} \right)^2 \frac{B_N}{R} \quad (6.9)$$

where V_{Rx} is the received signal RMS, V_N is the RMS of the additive noise, B_N is the noise noise bandlimiting filter bandwidth, and R is the bit rate. The calculation of B_N for the experimental setup is discussed in Sec. 6.3.3.

6.3.3 Experiment Setup

The experimental setup is shown on Fig. 6.10. and was used to measure the BERs for the AWGN channel hardware simulation. Due to the limited bandwidth of the Gaussian noise generator, the Wavetek and Rockland variable filters, and spectrum analyzer, a factor of 200 had to be used to scale the spectral bandwidth of the spread-spectrum signal to a measurable range. Therefore, the carrier and sampling signal supplied by the Hewlett Packard frequency synthesiser was 12.5 kHz and 50 kHz instead of 2.5 MHz and 10.0 MHz, respectively.

Three experiments were run to determine the BER performance of the hardware with 1) synchronous sampling of the DPSK output, 2) synchronous sampling of the DRake output, and 3) sampling of the DRake output by the phase locked-loop. For each of the experiments, the carrier oscillator frequency, f_{osc} , was adjusted to give an offset of 0.2502 of the sampling frequency. No experiments were performed using a zero frequency offset as this would have required the use of a 12.5 kHz PLL that was locked onto the system clock's frequency in order to prevent drift.

The amplitude limiter with back-to-back Zener diodes presented in [22] was used

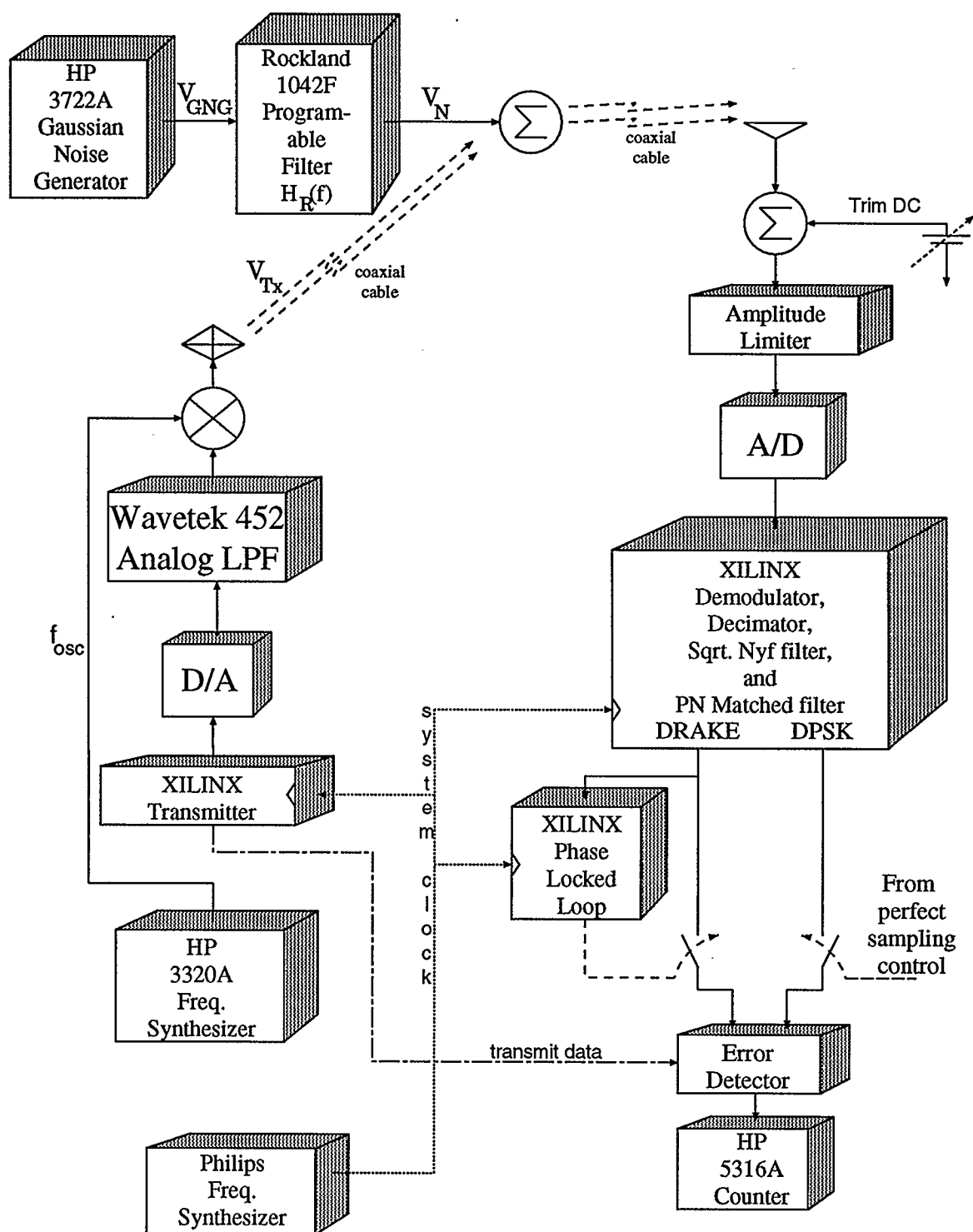


Figure 6.10 Experimental setup for AWGN experiment.

to prevent saturation of A/D input buffers. In practice however, the limiter would be replaced by an automatic gain control device. The reason for implementing the limiter was discovered by simulation. It was observed that the high input SNR's required to compare BERs to theoretical made the spread-spectrum signal power small relative to the aggregate received signal power. Thus, as the SNR was decreased, the correlation peaks were driven down into the quantization noise. Input SNR's were in the range of -3 to -10 dB, which is far lower than the expected 10 dB in the outdoor cellular radio environment.

Distortions that were not simulated were those due to the non-ideal effects of D/A and A/D conversion, A/D input buffer distortion, distortion caused by the non-ideal limiter, mixer distortion, and distortion caused by the noise bandlimiting filter. Simulation of these non-idealities would have required response measurements for characterization and modelling. Since these results would vary from component to component, it was decided not to pursue this task.

The noise bandwidth, B_N for the system was determined from

$$B_N = \frac{B_{GNG}}{\mathcal{G}} \left(\frac{V_N}{V_{GNG}} \right)^2, \quad (6.10)$$

where B_{GNG} is the bandwidth of the HP Gaussian Noise Generator, $\mathcal{G}$ is the Rockland bandpass filter (BPF) gain, V_n is the RMS noise measured at the BPF output, and V_{GNG} is the noise RMS at the generator output. To determine $\mathcal{G}$, the BPF gain was measured over the nominal bandpass bandwidth (9 - 16 kHz), and then averaged. The BPF gain was observed to vary by 0.8 dB over the nominal bandwidth.

6.3.4 Additive White Gaussian Noise Channel - Sampling the DSPK Output

The following probability of BER curves for the simulation and hardware were generated statistically by counting the number of errors that occurred at a particular signal-to-noise ratio. The sampled data values are taken at the peak of the M-sequence matched filter output. It can be shown that the probability of bit error P_b is a random variable whose sample value will vary by its standard deviation, σ_P for 63 % of the time where

$$\sigma_P = \frac{1}{\sqrt{P_b \mathcal{N}}}, \quad (6.11)$$

and $\mathcal{N}$ is the number of times P_b occurs. In all the following BER curves, $\mathcal{N}$ has been chosen using eqn. 6.11 such that the standard deviation is less than 0.1.

6.3.4.1 A Comparison Between Theory and Simulation

A comparison between theory (eqn. 6.6) and the simulation was made to verify the calculation of the BER curve-SNR abscissa values using equation 6.8, and to observe the performance with fixed point arithmetic. Figure 6.11 shows the simulation results for the case where there is no clipping, no overflow (the receiver uses 32 bit fixed-point words), and no frequency offset. The spread-spectrum signal level entering the receiver is represented by 8 bits, and the other case by 7 bits. The results show that the fixed-point arithmetic has little effect on the BER curve provided that the signal input to the receiver is represented by 8 bits or more. The graph indicates that quantization noise effects may be causing the measured BER curve to deviate from theory at high SNRs where the AWGN channel effects are not as dominant. This deviation could also be due to the biasing of the signal due to quantization.

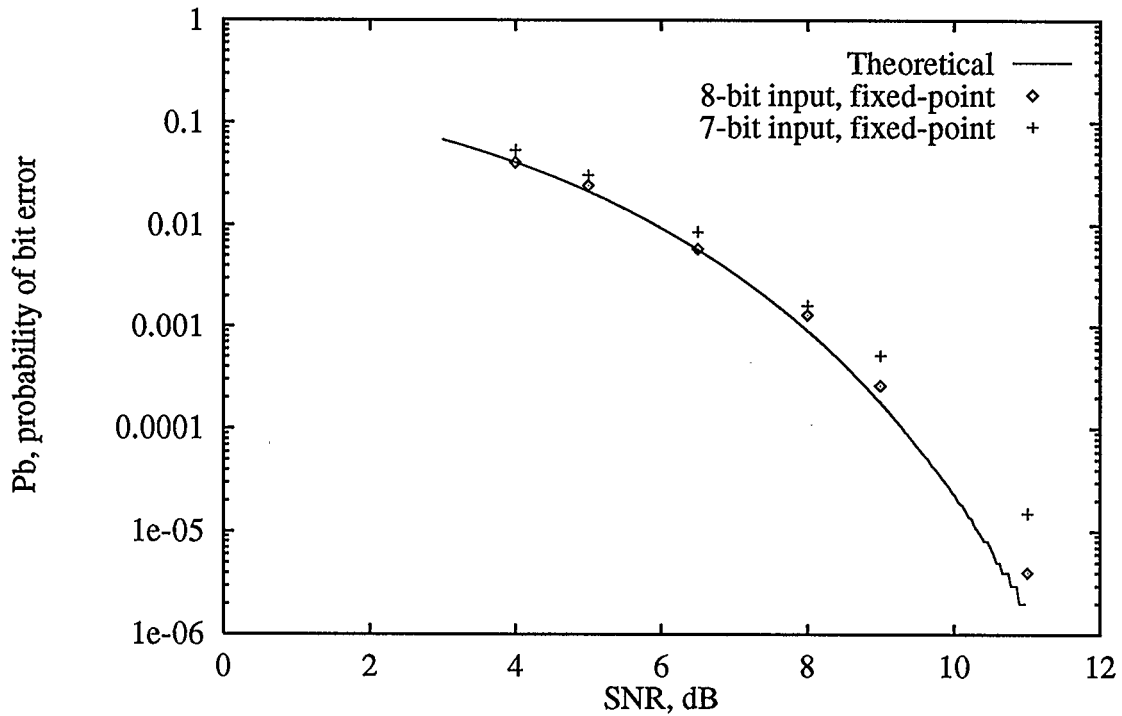


Figure 6.11 Probability of bit error for theoretical DPSK and simulated spread-spectrum receiver - 32-bit fixed-point arithmetic.

It is important to note that although the signals dynamic range is 2^n where n is the number of bits (Chapter 5 Sec. 5.1.1), the rolloff of $\alpha = 0.35$ results in roughly a 25 % overshoot in the transmitted square-root Nyquist pulse (see eye diagram Chapter 4 Fig. 4.8). Consequently, to prevent overflow, the signal is attenuated thereby reducing the signal-to-quantization noise level by 2.5 dB. Higher rolloffs were not investigated, however, this could definitely improve performance and also reduce the number of pulse-shaping filter taps.

6.3.4.2 Non-ideal Effects - Simulation and Hardware

The BER curves representing the non-ideal effects of frequency offset, clipping, and overflow are presented in Fig. 6.12. In the overflow case, the receiver model used

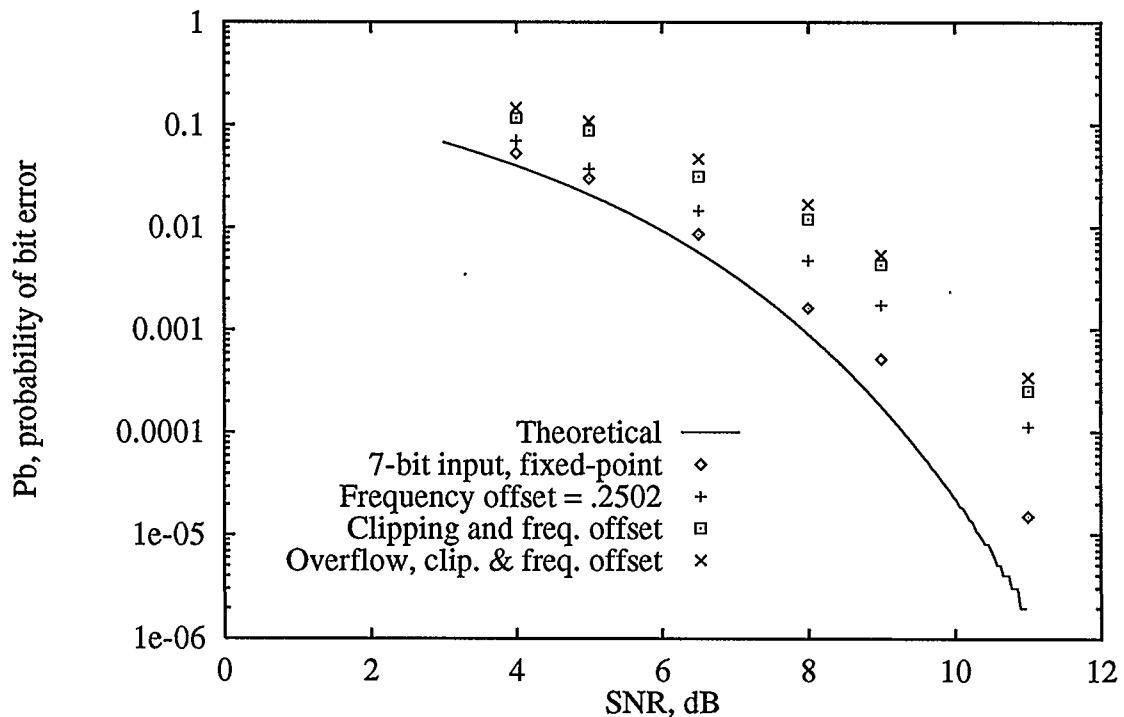


Figure 6.12 Non-ideal effects on bit-error rate performance.

8-bit fixed-point words. Synchronous sampling was performed in all cases.

The results show that in the frequency offset case, higher bit-errors - relative to the theoretical case - are expected as the SNR increases. Quantization noise, and the threshold offset could be the cause of the phenomena. Also, as the SNR increases, the non-ideal cases begin to merge. This is because the occurrences of overflow and clipping diminish with a reduced noise component of the signal.

Figure 6.13 shows the prototype BER curve compared to the non-ideal BER curve. Also shown is the simulated effects of bandlimiting the noise signal (all non-idealities are also modeled in this case). Bandlimiting the simulation noise signal was investi-

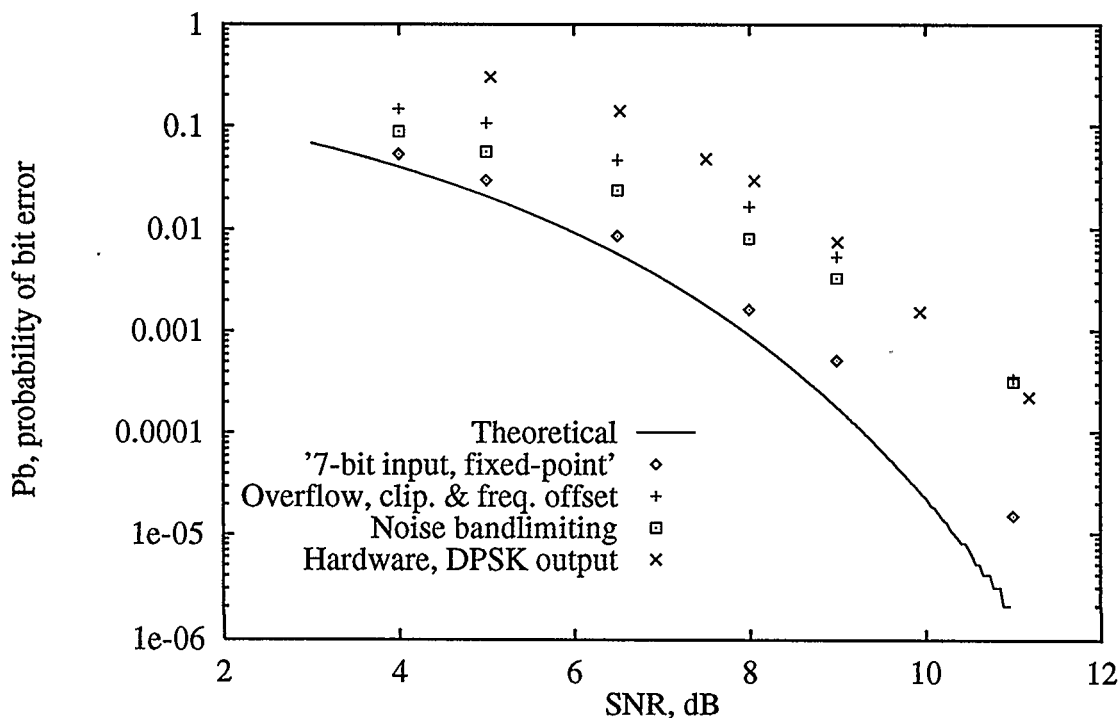


Figure 6.13 Hardware and simulated BER curves for synchronous sampling of the DPSK output.

gated since this operation would reduce clipping and also better simulate the Rockland BPF. As shown, bandlimiting improved the BER performance.

The hardware performance was not good as expected at low SNRs. This is most likely due to the non-ideal effects that were not simulated such as D/A, A/D, and mixer non-linearities, clipping circuit non-idealities, and A/D input buffer and noise bandlimiting BPF distortions. At higher SNRs, the BER's converge.

Figure 6.14 shows the effects of the carrier offset on receiver performance. The

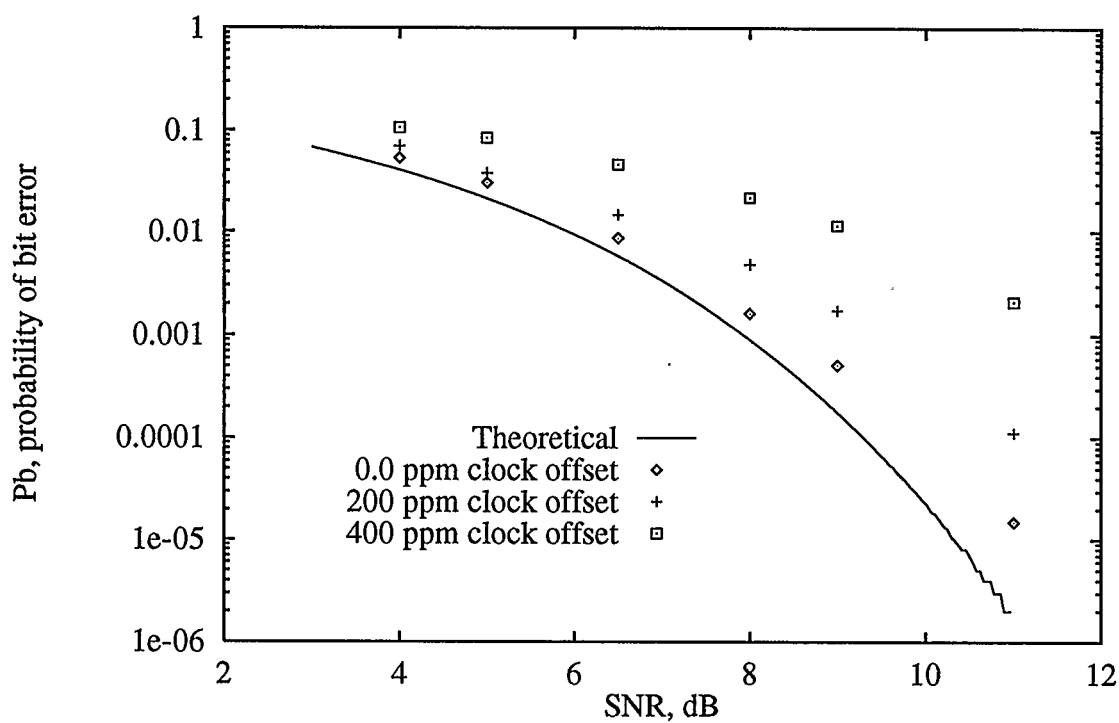


Figure 6.14 Effects of frequency offset of BER performance

200 ppm case is the same as the *frequency offset = 0.2502* case shown on Fig. 6.11. It is apparent from these results that 1 dB or more decrease in performance will occur if greater than 200 ppm stability oscillators are used.

6.3.5 Additive White Gaussian Noise Channel - Sampling the DRake Output

6.3.5.1 DRake Output: Synchronous Sampling

To synchronously sample the output from the DRake hardware, the signal that was used to sample the DPSK output, was delayed and positioned approximately near the end of the DRake output pulse (the DRake output pulse is shown on Fig. 6.6).

Figure 6.15 shows simulated and hardware BER curves for synchronously sampling the DRake output. The results show that there is approximately a 1.5 dB decrease in

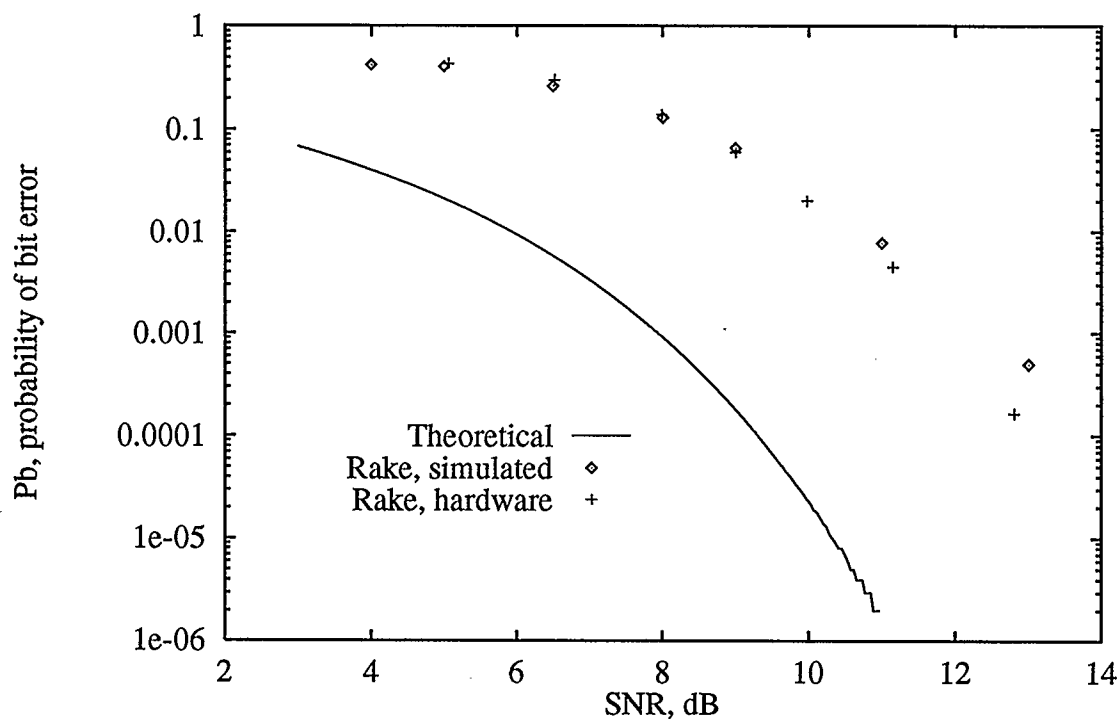


Figure 6.15 Hardware BER curve for synchronous sampling of the DRake output.

performance (at $\text{BER} = 10^{-2}$) over the DPSK sampling case (Fig. 6.13 - hardware, DPSK output). This is largely due to the non-linear integration of the noise signal - "non-linear" describes the partial summations of the DRake threshold detector tap outputs. The BER curves begin to deviate at high SNRs. It is apparent that the AWGN channel is dominant in dictating the systems performance at low SNRs, but as the SNR increases, small dissimilarities between the hardware and simulation prevail. These could be attributed to difficulties mirroring signal amplitudes between the simulation and hardware, and also the setting of the SNR ratios.

The DRake threshold was set at 0.016 (maximum is 1.0). This value was chosen to be approximately the same as the quantization noise level. Performance curves were not generated for other threshold level settings.

It is apparent from these results that the simple threshold detector DRake does not perform as well as sampling the DPSK output. Kamil [29] noted that the difference in performance between a first-path receiver (which in this case would be the same as sampling the matched filter output) and a DRake receiver decreases as multipath decreases. These results are an extension to that statement in that they show that if there is no multipath (assuming synchronous sampling) the first-path receiver performs better. But since a DRake is primarily used to take advantage of multipath diversity, it is unlikely that it would be implemented in a AWGN channel. Furthermore, it is believed that the DRake output is more immune to PLL jitter. Although the BER performance for PLL sampling of the DPSK output was not measured for comparison to the DRake output case, the above statement was based on visual observation of the PLL sampling pulse and the DPSK signal on the oscilloscope.

6.3.5.2 DRake Output: Phase Locked-Loop Sampling

The sampling signal, which triggers a one/shot for sampling in the error detection block (Fig. 6.10), was not positioned exactly at the end of the DRake pulse, but more towards the $\frac{3}{4}$ position. This was done to reduce the sampling misses caused by PLL jitter. The PLL's bandwidth was adjusted (by changing the K factor Chapter 4 sec. 4.4.6) at 13 dB SNR to try to reduce jitter and time-to-lock. This manual adjustment was made in conjunction with observation of the bit-error counter and oscilloscope trace of the PLL output clock.

The results for the PLL sampled DRake output are shown on Fig. 6.16. These

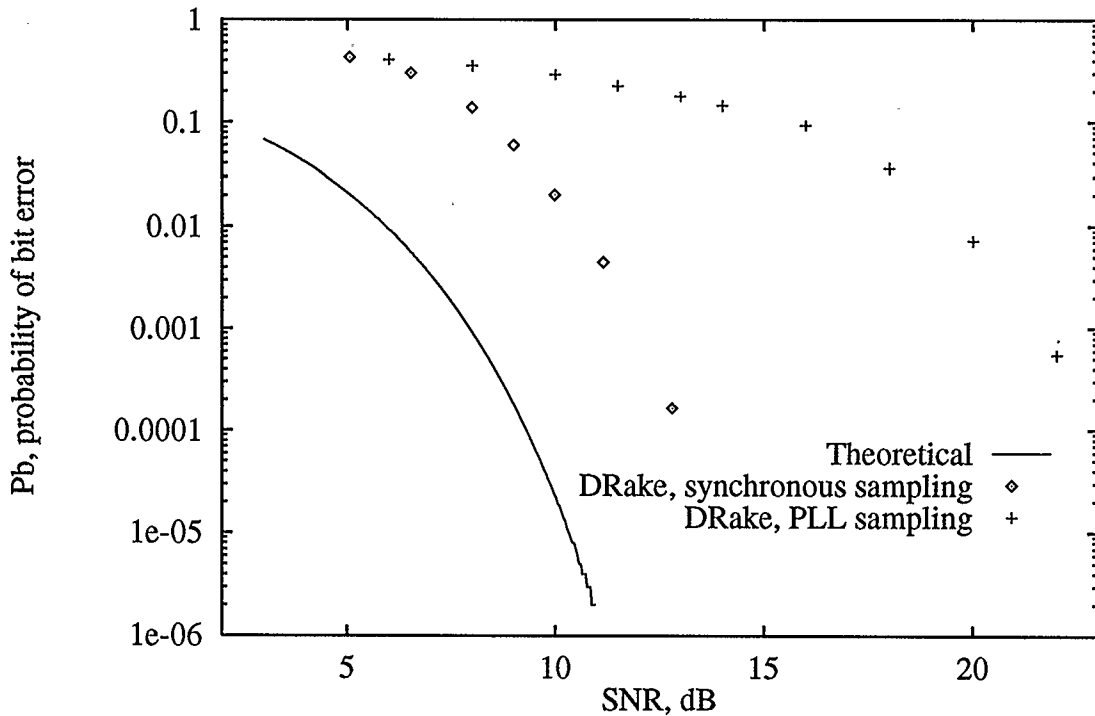


Figure 6.16 Hardware BER curve for PLL sampling of the DRake output.

results show that the first order PLL does not perform well. The performance degradation due to time-to-lock performance and phase jitter resulted in a 9 dB increase in SNR for an equivalent BER of 2×10^{-3} in the DRake synchronous sampling case. These results necessitate the need for a higher ordered PLL to remove jitter, phase offset, and thereby improve performance.

6.3.6 Multipath Channel Simulation

The multipath channel simulation results are shown on Fig. 6.17. The BER curve

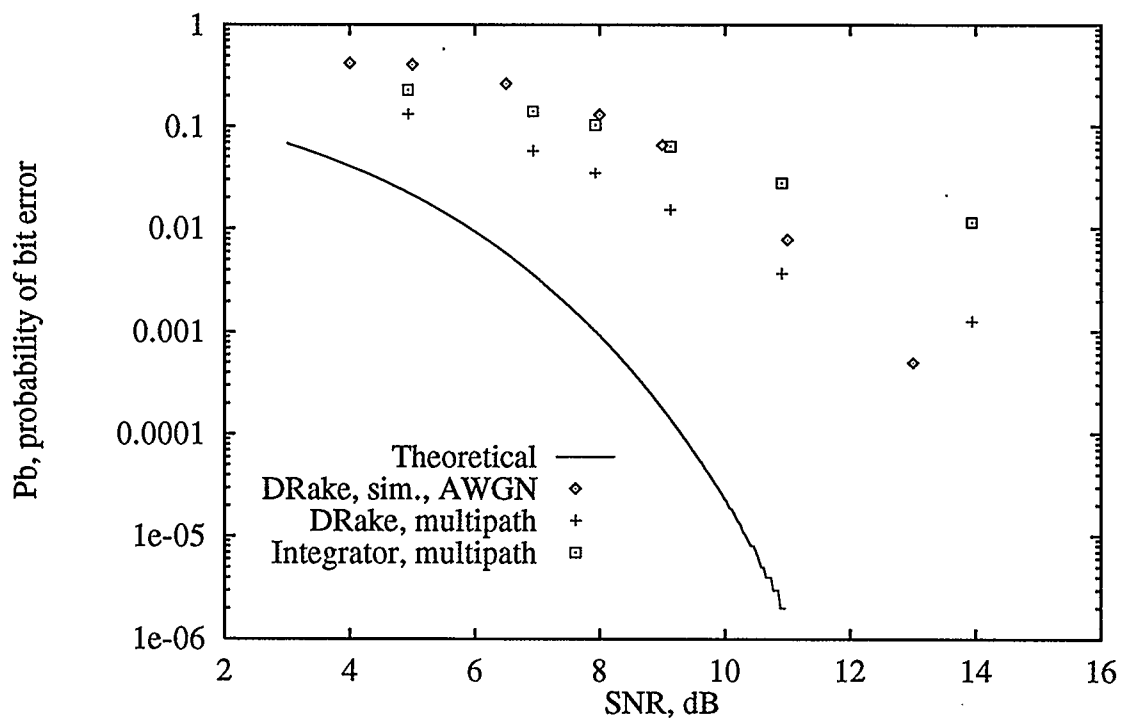


Figure 6.17 Multipath BER curve for synchronous sampling of the DRake output.

for the DRake and finite time integrator reflect the receiver's operation in a multipath channel when there is no frequency offset, no scaling of the input (8-bit input),

and no clipping. For the DRake case, the threshold is set at 0.016 (normalized). The integrator is essentially a DRake with the threshold set at 0.0. Also shown for comparison are the results for the simulated AWGN channel, synchronously sampled DRake output that was presented in Fig. 6.15. The channel impulse responses for the simulation were derived from the Hashemi profiles [20] for a vehicle traveling at 100 km/hr through urban area "B" [40]. Chapter 2 Sec. 2.1 presents a brief theoretical discussion of the channel model.

The results show that at low SNRs (less than 10 dB), a 2 dB increase in performance can be realized by using the DRake as opposed to the integrator. Also, at low SNRs, the multipath channel results parallel the AWGN channel results, but as the SNR increases, the BER curves for the hardware DRake and integrator begin to "flatten out". This is believed due to the noise component of the channel becoming less dominant as SNR increases while the multipath phenomena becomes more dominant.

The multipath results reflect the use of a 31 chip sequence transmitted over a 1.25 MHz bandwidth. Based on the waveforms shown in Figures 6.18, and 6.19, it is apparent that neither the M-sequence gain (see Chapter 2 Sec. 2.3.1) or the resolution of the 1.25 MHz bandwidth is adequate to significantly distinguish between multipath paths and the self-noise (see Chapter 2 Sec. 2.5.2.1). The self noise is present in the bit interval that scans from time 70 to 130 approximately, while the correlation output without self-noise effects is present in the pre and post intervals.

The maximum delay spread is $\approx 7\mu\text{sec}$. [29], which means that in relation to the 62 sample bit intervals (2 samples /chip), the multipath spreading would occur in $7 * 1.25\text{MHz} = 8.75$ samples or 14 % of the bit period. From the figures, it is apparent that multipath spreading occurred as expected; however, the single multipath channel magnitudes - other than the main path channel - do not have sufficient gain

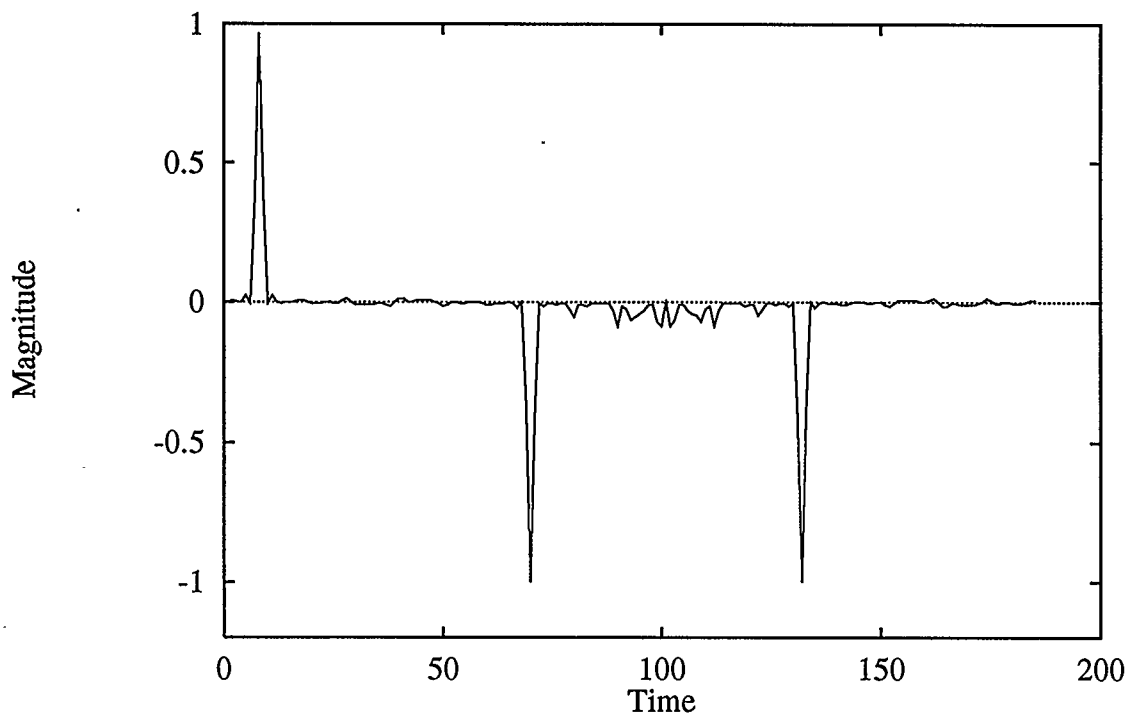


Figure 6.18 DPSK output illustration of self noise for a noiseless, single path channel.

to distinguish their presence from that of the self-noise.

6.4 Summary

The prototype waveforms and the BER performance curves have been presented along with simulation results. It was found that they performed essentially the same as the simulation with the exception of non-ideal effects that were not simulated (ie. A/D and D/A non-linearities, A/D buffer distortion, and clipping circuit non-idealities). The simulation's performance was verified by comparing the results to theoretical binary DPSK.

The results shown were for a relatively harsh AWGN environment. To work at

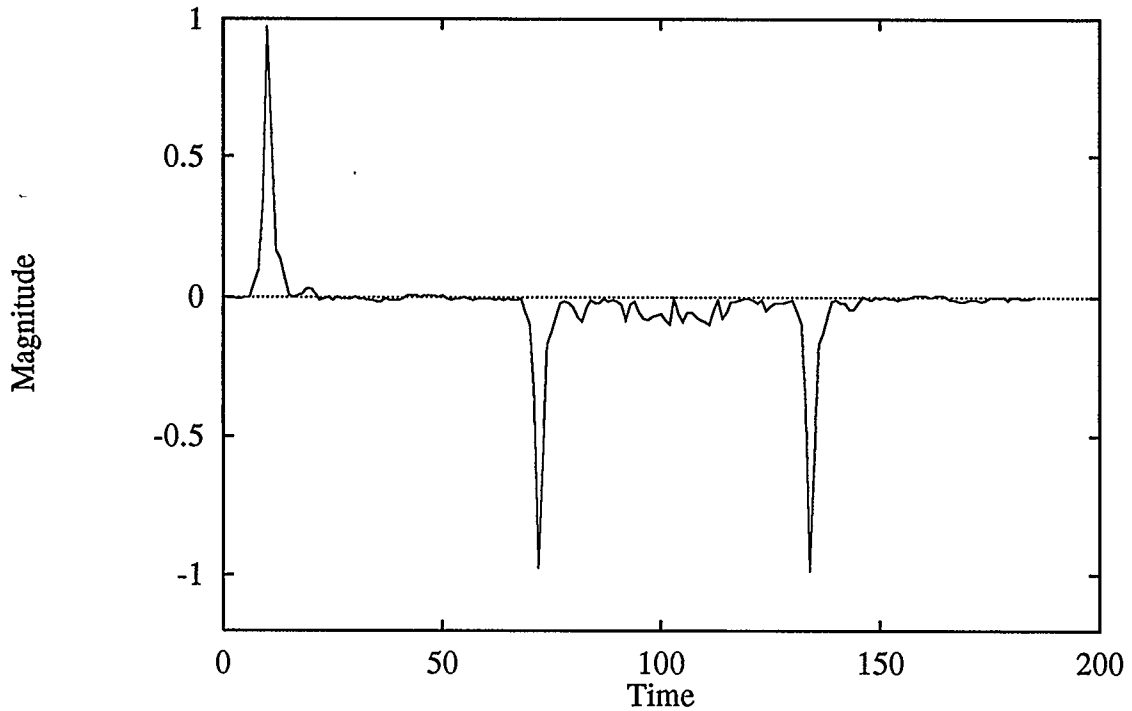


Figure 6.19 DPSK output illustration of self-noise for noiseless, multipath channel.

these low SNRs, it was found that a clipping circuit, instead of an AGC, would provide adequate A/D saturation protection without drastically reducing performance.

Frequency offsets, clipping, scaling and overflows attributed to 3 dB loss in performance over the 8-bit input, DPSK output sampling case. Performance improvements could be realized by using stable oscillators to reduce frequency drift, and by implementing saturation arithmetic (Chapter 5 Sec. 5.2.3) to reduce the need for scaling and the effects of overflows. Losses due to clipping would be lessened for reasons described in the previous paragraph.

The DRake was found to be 1.5 dB poorer in performance than sampling the DPSK output (synchronous sampling in both cases). This is due to the summation

of signals that pass the threshold level test but are only attributed to noise. In these instances, the DRake could not differentiate between noise and signal.

Eight-bit representation of the received signal was adequate to process the spread-spectrum signal and realize performance similar to a floating-point binary DPSK system. Improved performance could be realized by decreasing the Nyquist pulse rolloff, which relates to less overshoot, and hence increasing the magnitude of the signal sample values.

Gain offsets between the real and imaginary *halfband filter and decimate chains* were observed to cause oscillations in the DPSK output, as will quantization noise and oscillator offset frequencies.

The DRake was simulated with the multipath channel - as was a finite time integrator. The results showed that the integrator's performance was inferior to the DRake at combining the multipath channel energy. The 31 chip sequence was not found to provide sufficient gain to greatly distinguish multipath channels from the M-sequence self-noise. Furthermore, the 1.25 MHz signal does not provide sufficient resolution of multipath signals.

CHAPTER 7

CONCLUSIONS

7.1 Introduction

This dissertation describes the implementation of an all-digital transceiver spread-spectrum communication system operating in an outdoor cellular multipath environment. Where applicable, the transceiver prototype uses bit-serial, digit-serial, and parallel architectures. The feasibility of these structures as they pertain to spread-spectrum applications is investigated. The system design utilizes high level CAD design tools presently being developed at the University of Calgary.

The transceiver was prototyped on Xilinx field programmable gate arrays (FPGAs), and was tested on a additive white gaussian noise channel. A "like" computer simulation was also developed to verify the prototype's performance and to isolate non-ideal sources of error such as fixed-point arithmetic errors, frequency offsets, and signal clipping, etc.

7.2 Hardware Configuration and System Performance

The digital transceiver implemented used the matched filter (MF) approach described in Chapter 2 to despread the transmitted M-sequence. The matched filter approach provides instantaneous acquisition at the cost of increased silicon area. Other techniques using correlation receivers are easier to implement but typically have a long acquisition time - a disadvantage for time-division multiple access communication systems. Increased silicon in the MF approach is a limiting factor if implemented entirely with parallel architectures. But with bit-serial architectures, the gate counts

can be reasonable enough to warrant the use of this design - particularly as new silicon processes decrease gate size and power consumption.

With MFs, any number of multipath signals can be decorrelated as long as there is sufficient bandwidth to resolve the spreading. Correlator despreading requires a correlator for each signal path, requiring an elaborate scheme to track the changes in the multipath environment. The MF will track these changes instantaneously.

The transceiver designed uses efficient hardware to down-convert the received signal using digital filter decimation chains. The theory is presented in Chapter 3. The decimation chains were more effectively implemented as parallel structures since multipliers (which typically use the majority gates in digital filters) were replaced by simple shift operators. Furthermore, system clock rates needed to process the 10.0 MHz sampled signal would be too high for bit-serial implementation on Xilinx FPGA's.

Pre-distortion of the square-root Nyquist pulse was required to prevent the half-band down-convert and decimate filter chains from introducing signal intersymbol interference. At the transmitter, the added complexity of pulse shaping was minimal since multiplications were replaced by simple negation of the tap values. At the receiver, the square-root Nyquist pulse matched filter required multipliers and was therefore implemented using bit-serial designs. Simplicity and reduced silicon area was realized with this implementation. The tap values were derived from an annealing process that selected canonic bit-serial signed multipliers. These multipliers require less gates than standard bit-serial multipliers.

Distortion caused by the halfband filter and decimation chains could have been eliminated if higher ordered halfband filters were used. This, however, would be at the expense of filter complexity and gate usage which may be difficult to design -

particularly at high sampling rates (it is therefore easier to use pre-distortion).

The transmitter uses an interpolation-by-8 digital filter to reduce the rolloff requirements on the transmit lowpass analog filter described in Chapter 4. Consequently, a 10 MHz digital-to-analog converter (DAC) could be used. At the receiver, sampling at four-times the received signal's center frequency (ie. sampling at 10 MHz) provided sufficient stopband bandwidth for antialiasing analog bandpass filters (BPFs). In a practical system, it would be more advantageous to sample at higher rates for two reasons: 1) higher rates would reduce the rolloff requirements on the analog BPFs, and 2) adding more digital halfband filters to the down conversion chain will increase the signal to quantization noise levels.

The final design consisted of three different architectures: 1) bit-serial, 2) digit-serial, and 3) parallel. Each functional block in the transceiver was assessed for gate counts and speed to determine the best suited architecture. Parallel designs were found to be better suited for high speed blocks, while bit-serial designs were better suited to applications where a large amount of computation units can be pipelined (as in the M-sequence MF). Digit-serial designs were somewhere between these two architectures and were used where it was essential to accommodate for bit growth (as in the DRake) and multiplications (as in the DPSK).

Gain offsets between the real halfband filter and decimation chains, truncation noise, and carrier frequency offsets were found to cause the DPSK output to fluctuate (Chapter 6). In particular, it was shown that frequency offsets can substantially degrade performance and should be eliminated whenever possible by implementing stable oscillators (less than 200 ppm stability).

The prototype transceiver's performance was verified and assessed in relationship to a binary differentially phase shift keyed (DPSK) system transmitting over an ad-

ditive white gaussian noise (AWGN) channel. Probability of bit-error rate (BER) curves were measured for sampling the DPSK and digital Rake (DRake) outputs. It was found that for an AWGN channel, the synchronously sampled DPSK output case performed better than the DRake. This is expected since the DRake combines not only signal levels, but also combines noise levels that are greater than the threshold detector level (Chapter 5). The DRake performance would improve if a larger PN sequence is used.

Carrier frequency offsets, signal clipping, and finite-arithmetic overflows were simulated and compared with the prototype's performance with an AWGN channel. It was found that the BER curves did not align at low signal-to-noise ratios (SNR). But at high SNRs, the BER curves converge. It is possible that the non-ideal clipping circuit, non-ideal analog-to-digital input buffers, non-linear analog-to-digital and digital-to-analog converters, and the non-ideal noise bandlimiting filter were responsible for this result.

Multipath channel performance could not be assessed for the prototype. Since the prototype's performance was verified with the simulation (and the simulation was in turn verified with theory), it was decided that the Hashemi channel model and the DRake simulation model be used to evaluate the prototype's performance in an outdoor cellular environment. As expected, an irreducible BER is reached for the multipath simulation.

The simulation showed that there was insufficient M-sequence gain to differentiate low power multipath signals from the M-sequence self-noise (31 chips). Also, it was observed that the 1.25 MHz bandwidth provided insufficient resolution of multipath signals where delay spreads range $7\mu sec$.

7.3 Recommendations for Future Work

In Chapter 5, gate usage, system clock speeds, routing requirements, and power consumptions of the three different design architectures were presented. Based on simple gate times clock speed calculations, it was found that bit-serial designs will consume more power than parallel designs. More investigation is needed in this area to determine, and to better define, not only the power cost for these architectures, but also to incorporate the costs of routing resources, design time, systems speeds, and complexity. These areas are of concern to production of digital systems, such as the one presented in this thesis, since its implementation using either architectures is contingent on these economics factors.

A large number of gates for this system are due to the receiver square-root Nyquist taps and the M-sequence matched filter (MF). The Nyquist filter taps could be reduced by using a lower rolloff on the Nyquist pulse. A lower rolloff would also have the added advantage of reducing the "overshoot" thereby increasing the signal level at the MF input. This was not investigated, but could prove advantageous in simplifying the design. In relation to the M-sequence MF where the signal is presently represented by 8-bits, it is possible that this structure could perform as well with less bits representing the signal. The advantages are less gates (and therefore larger M-sequences could be used), and cheaper A/D converters (assuming that say a 4-bit, or even a 1-bit converter is cheaper than an 8-bit converter).

Saturation adders in the M-sequence MF were not investigated; however, it is believed that performance would greatly improve with a minimal cost associated with implementation.

Extensions to the DRake design should be investigated. Since its structure would

allow easy adaptation of channel estimator to control the open/close switches, it would be logical to implement this as an upgrade. Performance concerns could be easily addressed with the present simulation software.

REFERENCES

- [1] EIA/TIA Project Number 2215. , "*Dual-mode subscriber equipment - network equipment compatability standard*". Electronic Industries Association, 1989.
- [2] M. Moher A. Fapojuwo, A. Shen and J. McRory. , "Bit Error Rate Performance of a DS-SS Multiple-Access Cellular System in Urban Multipath Mobile Radio Environments". Report compiled for the dept. of Communications Research Center Ottawa.
- [3] A. V. Oppenheim and R. W. Schaffer. , "*Discrete-Time Signal Processing*". Prentice-Hall, Englewood cliffs, New Jersey 07632, 1989.
- [4] H. Baher. , "*Analog and Digital Signal Processing*". John Wiley and Sons, New York, 1990.
- [5] R. E. Best. , "*Phase-Locked Loops*". McGraw-Hill Inc., R. R. Donnelley and Sons Inc., 1984.
- [6] D.G. Brennan. , "On the Maximum Signal-to-noise Ratio Realizable from Several Noisy Signals". *Proc. IRE*, 43:p. 1530, Oct 1955.
- [7] Provided by the Alberta Micro Electronics Center. , "*LSI Logic Databook and Design Manual*". LSI Logic.
- [8] D. B. Coldham. , "Block Floating-Point Arithmetic". *Dr. Thesis*, Dept. of Elec. and Comp. Eng.(Univ. of Calgary):Alberta, 1977.
- [9] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein. , "Theory of Spread-Spectrum Communications - A Tutorial". *IEEE Trans. Commun.*, COM-30:855-884, May 1982.

- [10] D.J. Goodman and M.J. Carey. , "Nine Digital Filters for Decimation and Interpolation". *IEEE Trans. Acoust., Speech, and Signal Processing*, ASSP-25(no. 2):pp. 121–126, April 1977.
- [11] S. Benedetto, E. Biglieri, and V. Castellani. , "*Digital Transmission Theory*". Prentice-Hall, Inglewood Cliffs, N.J., 1987.
- [12] E. E. Swartzlander Jr. and G. Hallnor. , "Fast Transform Processor Implementation". *Proc. IEEE ICASSP'84*, pages pp. 25A.5.1 – 4, March 1984. San Diego.
- [13] T.L. Eckersley. , "Multiple Signals in Short-Wave Transmission". *Proc. of IRE*, 18:pp. 106–122, January 1930.
- [14] R. W. Hubbard et al. , "Measuring Characteristics of Microwave Mobile Channels". *Nat. Telecommun. and Inform. Admin*, Rep. 78-5, June 1978. Boulder, CO.
- [15] J. Ruprecht, F.D. Neeser. and M. Hufschmid. , "Code Time Division Multiple Access: An Indoor Cellular System". pages pp. 736–739, February 1992.
- [16] S. Freeman. , "Logical Design Rules and Testability Assessment". *30th Midwest Symposium on Circuits and Systems*, pages pp. 344–349, 1988.
- [17] E. A. Geraniotis. , "Performance of Noncoherent Direct-Sequence Spread-Spectrum Multiple-Access Communications". *IEEE J. Select. Areas Commun.*, SAC-3(5):687–694, September 1985.
- [18] F. Goodenough. , "ISSCC Analog Technology". *Electronics Design*, pages pp. 65–75, February 1992.
- [19] P. J. Graumann. , "Design and Implementation of Serial Multipliers and Dividers". *Graduate ENEL611 report*, Dept. of Elec. and Comp. Eng.(University of Calgary), December 1992.

- [20] H. Hashemi. , "Simulation of the Urban Radio Propagation Channel". *Ph.D. thesis*, Dep. Elec. Comput. Sci.(Univ. California):Berkeley, 1977.
- [21] S. Haykin. , "*Adaptive Filter Theory*". McMaster University, Prentice-Hall, Englewood cliffs, New Jersey, 1986.
- [22] R. J. Higgins. , "*Electronics with Digital and Analog Integrated Circuits*". Prentice-Hall Inc, New Jersey, 1983.
- [23] R. J. Higgins. , "*Digital Signal Processing in VLSI*". Prentice-Hall Inc, New Jersey, 1990.
- [24] E. B. Hogenauer. , "An economical class of digital filters for decimation and Interpolation". *IEEE Transactions ASSP*, ASSP-29(2):155–162, April 1981.
- [25] Xilinx Inc. , "*Technical Data: XC3000 and XC4000 Series*". Xilinx Inx, 2100 Logic Drive, San Jose, CA. 95124, 1991.
- [26] J. H. Lodge and V. Szwarc. , "Digital Implemetation of Narrowband Radio". *Wireless 92 Proceedings*, July 1992.
- [27] D. DeFatta, J. Lucas, and W. Hodgkis. , "*Digital Signal Processing*". John Wiley and Sons, 1988.
- [28] R. V. Kacelenga. , "Simulated Annealing: Digital Filter Design". *MSc. Thesis*, Dept. of Elect. and Comp. Eng.(Univ. of Calgary):Alberta, 1990.
- [29] M. A. Kamil. , "Simulation of Digital Communication through Urban/Suburban Multipath". *Ph.D dissertation*, Dep. Elec. Eng. Comput. Sci.:Univ. California, Berkeley, June 1981.
- [30] Kavehrad, M. and Bodeep, G. E. , "Design and Experimental Results for a Direct-Sequence Spread-Spectrum Radio Using Differential Phase Shift Keying

- Modulation for Indoor, Wireless Communications". *IEEE J. Select. Areas Commun.*, SAC-5(5):815–823, June 1987.
- [31] Kavehrad, M. and Ramamurthi, B. Direct-sequence spread-spectrum with dpsk modulation and diversity for indoor wireless communications. *IEEE Trans. Commun.*, COM-35(2):224–236, February 1987.
- [32] M. Fattouche, L. Petherick, and A. Fapojuwo. , "Diversity for Mobile Radio Communications". *Proc. 15th Biennial Symp. Comm.*, pages pp. 196–199, June 1990.
- [33] W. C. Lindsey. , "*Synchronization Systems in Communication and Control*". Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1972.
- [34] M. Kavehrad and P. J. McLane. , "Spread-Spectrum for Indoor Digital Radio". *IEEE Communications Magazine*, 25(6), 1987.
- [35] M. Renfors et. al. , "Speech Codex Architectures for Pan-European Digital Mobile Radio using Bit-serial Signal Processing". *Department of Electrical Engineering*, Tampere University of Technology(P.O. Box 527, SF-33101 Tampere, Finland):pp. 50–60. Chapter 6.
- [36] J. E. Mazo. Some theoretical observations on spread-spectrum communications. *Bell Syst. Tech. Journal*, 58:2013–2023, November 1979.
- [37] MIT. , "A Report on Security of Overseas Transport". *MIT, Cambridge*, ATI(205035, 205036), September 1950.
- [38] R. M. Oberman. , "*Digital Circuits for Binary Arithmetic*". MacMillan Press Ltd., London, 1987.

- [39] H. Ochsner. , "Direct-Sequence Spread-Spectrum Receiver for Communication on Frequency-Selective Fading Channels". *IEEE J. Select. Areas Commun.*, SAC-5(2):188–193, February 1987.
- [40] E.B. Olasz. , "Antenna Diversity in Analog Cellular Radio". *MSc. thesis*, Dep. Elec. Comput. Sci.(Univ. Calgary):Calgary, Canada, 1991.
- [41] P. E. Green and R. Price. , "Communication Technique for Multipath Channels". *Proceedings of the IRE*, pages 555 – 569, 1958.
- [42] J. G. Proakis. , "*Digital Communications*". Second Edition. McGraw-Hill Book Company, 1989.
- [43] R. A. Scholtz. , "The Origins of Spread-Spectrum Communications". *IEEE Trans. Commun.*, COM-30:pp. 822–854, May 1982.
- [44] H. Kaufmann, R. King and U. Fawer. , "Digital Spread-Spectrum Multipath-Diversity Receiver for Indoor Communications". *Proc. of IEEE Vehicular Tech. Conf., Denver*, pages pp. 1038–1041, May 1992.
- [45] R.E. Ziemer and R.L. Peterson. , "*Digital Communications and Spread Spectrum Systems*". Macmillan Publishing Company, 866 Third avenue, New York, 10022, 1985.
- [46] Sarwate, D. V. and Pursley, M. B. , "Crosscorrelation properties of pseudorandom and related sequences". *Proc. IEEE*, 68(5):593 – 619, 1980.
- [47] K.E. Scott. , "Diversity with Multichannel Equalization". *Ph.D. thesis*, Dep. Elec. Comput. Sci.(Univ. Calgary):Calgary, Canada, 1991.
- [48] S.G. Smith and P.B. Denyer. , "*Serial-Data Computation*". Kluwer Academic Publishers, The Netherlands, 1988.

- [49] W. C. Lindsey M. K. Simon. , "*Phase-Locked Loops & Their Application*". IEEE Pres, 345 East 47 Street, New York, NY 10017, 1978.
- [50] Seymour Stein. , "Fading channel issues in system engineering". *IEEE J. Select. Areas Commun.*, SAC-5(2):68-89, February 1987.
- [51] F. E. Termin. , "Administrative History of the Radio Research Laboratory". *Radio Res. Lab., Harvard Univ., Cambridge, MA*, pages Rep. 411-299, March 1984.
- [52] Donald G. Troha. , "*Digital Phase-Locked Loop Design Using SN54/74LS297*". Applications Report SDLA0005A. Texas Instruments, Dallas, Texas, 1986.
- [53] G. L. Turin. , "Introduction to Spread-Spectrum Antimultipath Techniques and their Application to Urban Digital Radio". *Proceedings of the IEEE*, 68(3):328-353, March 1980.
- [54] A. L. Welti U. Grob and E. Zollinger. , "Microcellular Direct-Sequence Spread-Spectrum Radio System Using N-Path RAKE Receiver". *IEEE J. Select. Areas Commun.*, 8(5):772-779, June 1990.
- [55] W.T. Greer and B. Kean. , "Digital Phase-Locked Loops move into the Analog Territory". *Electron. Des.*, pages 95-100, March 1982.