

Quand tu partiras pour Ithaque, souhaite que le chemin soit long, riche en péripéties et en expériences. Ne crains ni les Lestrygons, ni les Cyclopes, ni la colère de Neptune. Tu ne verras rien de pareil sur ta route si tes pensées restent hautes, si ton corps et ton âme ne se laissent effleurer que par des émotions sans bassesse. Tu ne rencontreras ni les Lestrygons, ni les Cyclopes, ni le farouche Neptune, si tu ne les portes pas en toi-même, si ton cœur ne les dresse pas devant toi.

Souhaite que le chemin soit long, que nombreux soient les matins d'été, où (avec quelles délices !) tu pénétreras dans des ports vus pour la première fois. Fais escale à des comptoirs phéniciens, et acquiers de belles marchandises : nacre et corail, ambre et ébène, et mille sortes d'entêtants parfums. Acquiers le plus possible de ces entêtants parfums. Visite de nombreuses cités égyptiennes, et instruis-toi avidement auprès de leurs sages.

Garde sans cesse Ithaque présente à ton esprit. Ton but final est d'y parvenir, mais n'écourte pas ton voyage : mieux vaut qu'il dure de longues années, et que tu abordes enfin dans ton île aux jours de ta vieillesse, riche de tout ce que tu as gagné en chemin, sans attendre qu'Ithaque t'enrichisse.

Ithaque t'a donné le beau voyage : sans elle, tu ne te serais pas mis en route. Elle n'a plus rien d'autre à te donner. Même si tu la trouves pauvre, Ithaque ne t'a pas trompé. Sage comme tu l'es devenu à la suite de tant d'expériences, tu as enfin compris ce que signifient les Ithagues.

A mes parents...

Remerciements

Une thèse de doctorat ne saurait se résumer qu'à cet écrit où ne figurent que les traces d'un cheminement qui s'est étendu sur trois ans. Ma reconnaissance s'adresse à tous ceux qui, de près ou de loin, ont permis que ce travail se réalise dans les meilleures conditions. Je désire alors exprimer ma profonde gratitude:

A Jean Luc Danger et Lirida Naviner qui m'ont donné l'occasion de faire ma thèse dans d'excellentes conditions au sein de leur équipe.

A tous les partenaires de projet RNRT ASTURIES pour les discussions créatives et pour leur aide.

Au professeur Andreas Polydoros de l'Université d'Athènes, qui m'a fait découvrir le monde de recherche.

Aux professeurs Charalambos Charalambous et Jaques Palicot pour avoir accepté d'être Rap-
porteurs de ma thèse.

A madame Marylin Arndt et a monsieur Jean-Didier Legat qui ont accepté d'être examinateurs dans ce jury.

A monsieur Dominique Noguét qui a accepté de participer à ma soutenance comme membre invité.

A mes collègues du Département Comelec pour leur contribution à la bonne ambiance de travail qui a baigné le cours de ma thèse. Un grand merci à Daniel Cardoso: son soutien et son aide m'ont été indispensables.

Je remercie aussi ma famille, et en particulier mes parents, pour leur patience et la compréhension dont ils ont fait preuve tout au long de mes études.

Paris, le 17 Septembre 2005

Contents

List of Figures	9
List of Tables	15
Notations	17
Abbreviations	19
The Asturies project	21
Introduction	23
1 3G communication systems	25
1.1 Introduction	25
1.2 Spread spectrum communications	25
1.2.1 DS-CDMA for personal communications	27
1.2.2 The wireless propagation channel	27
1.3 CDMA detection	28
1.3.1 The conventional RAKE receiver	28
1.3.2 The advanced receiver schemes	31
1.4 3G communication systems	35
1.4.1 Differences between 3G and 2G air interfaces	36
1.4.2 UMTS	37
1.4.3 CDMA2000	42
1.4.4 3G and satellites	43
1.5 Conclusion	44
2 Algorithmic Reconfigurability: Applications	45
2.1 Introduction	45
2.2 An adaptive RAKE receiver with variable number of fingers and window-based channel estimation	46
2.2.1 Algorithm motivation	46
2.2.2 Problem formulation	47
2.2.3 The proposed receiver	49
2.2.4 Simulation environment	51

2.2.5	Simulation results	55
2.3	High data rates and WCDMA systems	59
2.4	Single-stage interference cancelation for high data rates	59
2.4.1	Algorithm motivation	59
2.4.2	Problem formulation	60
2.4.3	The proposed receiver	62
2.4.4	The Finger Configuration Algorithm	64
2.4.5	Numerical results	66
2.5	Multi-stage interference cancelation	70
2.5.1	Algorithm motivation	70
2.5.2	Problem formulation	71
2.5.3	The proposed receiver	72
2.5.4	The proposed scheme versus conventional receivers	74
2.5.5	Numerical results	77
2.6	Multi-stage interference cancelation with realistic channel estimation	81
2.6.1	Algorithm motivation	81
2.6.2	Problem formulation	82
2.6.3	The proposed receiver	84
2.6.4	Numerical results	86
2.7	Multi-stage interference cancelation for multi-user detection	88
2.7.1	Algorithm motivation	88
2.7.2	Problem formulation	89
2.7.3	The proposed receiver	91
2.7.4	Numerical results	94
2.8	CDMA2000 for a satellite environment	97
2.8.1	Algorithm motivation	97
2.8.2	Problem formulation and simulation environment	98
2.8.3	The proposed system	102
2.8.4	Numerical results	102
2.9	Conclusion	105
3	Hardware Reconfigurability: Applications	107
3.1	Introduction	107
3.2	DSP implementation	107
3.2.1	TigerSHARC	108
3.2.2	Time constraints	110
3.2.3	Optimized code	114
3.2.4	DSP performance	116
3.3	Hardware implementation	116
3.3.1	The iterative architecture	116
3.3.2	Implementation issues	121
3.3.3	Simulation results	125
3.3.4	FPGA implementation	130
3.3.5	FPGA performance	131

3.4	Conclusion	133
4	The iterative reconfigurability concept	135
4.1	Introduction	135
4.2	The need of a flexible radio	135
4.3	Definitions of radio flexibility	137
4.4	The two-layer reconfigurability concept	140
4.4.1	Layer 2: The algorithmic reconfigurability	140
4.4.2	Layer 1: The hardware reconfigurability	143
4.4.3	Layer 1+: The architectural reconfigurability	148
4.5	Conclusion	155
	Conclusions and Perspectives	157
A	Definitions and Comments	161
B	Linear Profile for the DSP implementation	165
C	Résumé	169
C.1	Introduction	169
C.2	Notion de Reconfigurabilité matérielle	171
C.2.1	Approche multiplexage	171
C.2.2	Approche pagination	171
C.2.3	Approche factorisation	172
C.2.4	Approche itération	172
C.3	Récepteur RAKE reconfigurable avec annulateur d'interférences	173
C.3.1	Formulation du problème	174
C.3.2	Algorithme proposé	175
C.3.3	Evaluation de la performance	176
C.3.4	Architecture reconfigurable	177
C.3.5	Implémentation	180
C.4	Un annulateur d'interférences basé sur une estimation de canal	183
C.4.1	Formulation du problème	183
C.4.2	Algorithme proposé	184
C.4.3	Evaluation de la performance	186
C.4.4	Architecture reconfigurable	186
C.5	Conclusion	188
	Bibliography	189
	List of publications	199

List of Figures

1.1	Basic spread spectrum technique.	26
1.2	RAKE receiver correlator.	29
1.3	Block diagram of a simple transmission scheme using a zero-forcing equalizer. . .	32
1.4	Block diagram of a simple transmission scheme employing an MMSE equalizer. .	32
1.5	Schematic of the SIC receiver for U users. The users' signals have been ranked, where user 1's signal was received at the highest power, while user U 's signal at the lowest power. In the order of ranking, the data estimates of each user are obtained and the received signal of each user is reconstructed and canceled from the received composite signal, r	34
1.6	Schematic of a single cancelation stage for user i in the PIC receiver for U users. The data estimates, $\hat{b}_1, \dots, \hat{b}_U$ of the other $(U - 1)$ users were obtained from the previous cancelation stage, and the received signal of each user other than the i -th one is reconstructed and canceled from the received signal, r	35
1.7	The channelization tree.	38
1.8	Relation between spreading and scrambling.	39
1.9	Uplink/Downlink dedicated physical channel structure.	40
1.10	Principles of FDD and TDD operation.	41
1.11	Relationship between the MC mode (3X) and IS-95 (1X) in spectrum usage. . . .	43
2.1	The Proposed Receiver- Smart Controller.	50
2.2	The transmitter chain.	52
2.3	The power envelope for different correlations.	53
2.4	The receiver chain.	54
2.5	RAKE receiver with different numbers of fingers for speeds equal to 3 Km/h and 50 Km/h, and perfect channel estimation.	55
2.6	RAKE receiver with different numbers of fingers for a speed equal to 3 Km/h, $E_b/N_0 = 5dB$ and imperfect estimation.	56
2.7	RAKE receiver with different numbers of fingers for a speed equal to 3 Km/h, $E_b/N_0 = 10dB$ and imperfect estimation.	56
2.8	RAKE receiver with different numbers of fingers for a speed equal to 50 Km/h, $E_b/N_0 = 5dB$ and imperfect estimation.	57
2.9	RAKE receiver with different numbers of fingers for a speed equal to 50 Km/h, $E_b/N_0 = 10dB$ and imperfect estimation.	57
2.10	The proposed algorithm.	62

2.11	The computational part of the proposed reconfigurable detector.	63
2.12	a) RAKE finger for the i -th channel path b) Interference “finger” for the cancelation of the term which arose from the i -th and j -th paths.	64
2.13	The block diagram of the finger configuration algorithm.	65
2.14	Performance comparison for a reconfigurable RAKE receiver with different computational power constraints (number of the available fingers).	67
2.15	BER performance for a spreading factor equal to 4 and $E_b/N_0 = 16\text{dB}$	67
2.16	BER performance for a spreading factor equal to 8 and $E_b/N_0 = 16\text{dB}$	68
2.17	BER performance for a spreading factor equal to 32 and $E_b/N_0 = 16\text{dB}$	68
2.18	Reconfigurability decision variable G	69
2.19	The proposed algorithm; A RAKE receiver with a multi-stage IC.	70
2.20	The general block diagram of the detection algorithm under consideration (combination of RAKE with the multistage IPI-IC).	72
2.21	The i -th stage of the detection algorithm.	73
2.22	The decision function $f_{dec}()$ with a threshold c	73
2.23	The interference generation process.	74
2.24	BER performance of the reconfigurable detector for different configurations versus E_b/N_o , for $N=4$ and $\mathbf{h} = [0\ 0\ 0]\text{dB}$, $\tau = [0\ 6\ 8]T_c$	78
2.25	BER performance of the reconfigurable detector for different configurations versus E_b/N_0 , for $N = 4$ and $\mathbf{h} = [0\ 0\ 0]\text{dB}$, $\tau = [0\ 1\ 2]T_c$	78
2.26	BER performance of the reconfigurable detector for different configurations versus the spreading factor, for $\text{SNR}=16\text{dB}$ and $\mathbf{h} = [0\ 0\ 0]\text{dB}$, $\tau = [0\ 6\ 8]T_c$	79
2.27	BER comparison over equal 3-path channel and $N = 2$	79
2.28	BER comparison over equal 3-path channel and $N = 4$	80
2.29	The intra (IPI_1)- and inter (IPI_2)- interference (from the data channel point of view).	81
2.30	The general structure of the proposed algorithm; A RAKE receiver with a pilot channel estimation and IC.	82
2.31	General structure of the multi-stage inter-path interference canceler with a realistic channel estimation.	84
2.32	Structure of the i -th cancelation stage.	85
2.33	The computational similarities of the essential operations of the proposed reconfigurable detector.	86
2.34	MSE performance of the proposed channel estimation scheme versus the classic estimation ($V=0$).	87
2.35	BER performance of the proposed reception scheme versus the conventional receiver ($V=0$).	87
2.36	The proposed reconfigurable PIC scheme for DS/CDMA downlink connections.	91
2.37	Structure of the proposed multi-stage PIC scheme for the i -th user.	92
2.38	Structure of the j -th MAI-PIC stage.	93
2.39	Structure of the j -th IPI-PIC stage.	93
2.40	The impact of SF on the performance of a conventional RAKE receiver. $E_b/N_0 = 16\text{dB}$, $V = 5$ stages and $U = 1$ users.	94

2.41	The impact of the decision function threshold on the PIC performance. $E_b/N_0 = 16\text{dB}$, $V = 5$ stages, $N = 8$ and $U = 5$ users. The thresholds are $c = 0.0, 0.3, 0.7$ and ∞ , respectively.	95
2.42	BER performance versus E_b/N_0 . $V = 5$ stages, $N = 8$, $U = 5$ users and the thresholds are $c = 0.0$ and 0.3	96
2.43	BER performance versus number of users. $E_b/N_0 = 16\text{dB}$, $V = 5$ stages, $N = 8$ and the thresholds are $c = 0.0$ and 0.3	96
2.44	BER performance versus number of canceled users. $E_b/N_0 = 16\text{dB}$, $V = 5$ stages, $N = 8$, $U = 7$ users and the thresholds are $c = 0.0$ and 0.3 , respectively.	97
2.45	Transmission chain for mode 1X.	99
2.46	Transmission chain for mode 3X.	99
2.47	Reception chain for mode 1X.	101
2.48	Reception chain for mode 3X.	101
2.49	The proposed reconfigurable CDMA2000 transceiver.	102
2.50	Comparison of DS-CDMA and MC-CDMA for a receiver speed equal to 3km/h	104
2.51	Comparison of DS-CDMA and MC-CDMA for a receiver speed equal to 50km/h	104
2.52	Comparison of DS-CDMA and MC-CDMA for a receiver speed equal to 130km/h	105
3.1	The block diagram of the DSP implementation; RAKE and IC correspond to software functions.	108
3.2	Top-level block diagram showing the major DSP subsystems and the data buses.	109
3.3	SIMD execution and subword parallel operations.	110
3.4	The processing time of the RAKE combination versus the number of the channel paths.	112
3.5	The processing time of the RAKE combination versus the DSP's frequency clock; $L = 4$, $M = 4$	112
3.6	The processing time of the IC versus the number of canceled interference terms; $L = 3$ and $M = 3$	113
3.7	The processing time of the IC versus the DSP clock frequency for an environment with $L = 2$ and $M = 2$	113
3.8	The efficiency of a DSP with a frequency 250 MHz in terms of the operated computations for different channel environments.	115
3.9	The FSM model of the iterative mapping approach for the detection scheme under consideration.	117
3.10	The corresponding computational core of the iterative approach.	117
3.11	Timing relationship of SRAM chip and SRAM symbol WR/RD operations.	118
3.12	The flow of data during the RAKE configuration mode.	119
3.13	The flow of data during the IC configuration mode.	119
3.14	The pipeline implementation of the reconfigurable RAKE/IC detector.	121
3.15	The integrator block for the time instant n	123
3.16	The complex multiplication between $a + bj$ and $x + yj$ in two processing cycles.	123
3.17	Modelsim time simulation for a channel environment with $L = 1$ and $N = 2$	126
3.18	Modelsim time simulation for a channel environment with $L = 3$ and $N = 8$	126
3.19	Modelsim time simulation for a channel environment with $L = 2$, $N = 4$, $V = 1$	126

3.20	Modelsim time simulation for a channel environment with $L = 2$, $N = 4$, $V = 2$.	126
3.21	Memory structure for a slot with 3 symbols, $L = 1$ and $N = 4$.	127
3.22	Generation of the chip memory addresses for the case of $L = 1$, $N = 4$ and $N_{Block} = 3$.	127
3.23	Memory structure for a slot with 3 symbols, $L = 3$, $\tau = [0 \ 1 \ 2]T_c$ and $N = 4$.	128
3.24	Generation of the chip memory addresses for the case of $L = 3$, $N = 4$ and $N_{Block} = 3$.	128
3.25	Interference structure for a slot with 3 symbols, $L = 2$, $\tau = [0 \ 2]T_c$ and $N = 4$.	129
3.26	Generation of the symbol memory addresses for the case of $L = 2$, $N = 4$ and $N_{Block} = 3$.	129
3.27	The Virtex architecture.	130
3.28	CLB frame organization.	131
3.29	The achieved computation power versus the number of paths.	132
4.1	Adaptivity, reconfigurability and modularity in the geometry domain.	138
4.2	The imagined Euclidean space of flexibility. The vector $[a_0, r_0, m_0, \dots]$ presents the degree/type of flexibility.	139
4.3	The general structure of a dynamic two-layer reconfigurable system.	141
4.4	Supervisor inputs and outputs.	142
4.5	(a) Von Neumann memory architecture, (b) Harvard memory architecture.	144
4.6	Typical FPGA architecture.	146
4.7	The behavior of the implementation cost as a function of production volume.	146
4.8	A hardware device added to an existing architecture for acceleration purposes.	148
4.9	DSP mapping approach.	150
4.10	The switching mapping approach.	151
4.11	The hardware paging mapping approach.	152
4.12	The factorization mapping approach.	153
4.13	The relation between factorization and iterative approaches.	153
4.14	The iterative mapping approach.	154
4.15	The FSM represents the iterative approach schematically.	154
B.1	Linear profiling for the RAKE function; Simulation parameters: $L=2$, $N=4$, $K=640$.	165
B.2	Linear profiling for the IC function (Part A); Simulation parameters: $L=2$, $N=4$, $K=640$.	166
B.3	Linear profiling for the IC function (Part B); Simulation parameters: $L=2$, $N=4$, $K=640$.	167
C.1	Exemple d'Adaptativité: Changement de pas de l'algorithme LMS.	170
C.2	Exemple de reconfigurabilité: Changement de code.	170
C.3	L'approche multiplexage.	172
C.4	L'approche pagination.	172
C.5	Plate-forme configurée pour 2 fonctions différentes.	173
C.6	Plate-forme avec opérateurs à grains fins pour la sérialisation.	173
C.7	Fonctions RAKE et annulateur d'interférences multi-étages.	176
C.8	Structure de l'étage i.	176

C.9	Fonction de décision $f_{dec}()$ avec pallier c .	177
C.10	Evaluation de la performance évaluation de l'annulateur d'interférences; SF=2.	178
C.11	Evaluation de la performance évaluation de l'annulateur d'interférences; SF=4.	178
C.12	Architecture reconfigurable.	179
C.13	Chronogramme des flux de donnée en réception et en traitement.	179
C.14	Configuration en mode Rake.	180
C.15	Configuration en mode IC.	180
C.16	L'efficacité de DSP en utilisant comme mesure le nombre de calculs effectuées pour différent nombre de trajets.	182
C.17	La comparaison de trois implémentations.	182
C.18	IC inter chemin avec estimation de canal.	185
C.19	Structure du i-ème étage d'annulation.	185
C.20	Performance au sens MSE du schéma d'estimation propose versus le schéma clas- sique (V=0).	187
C.21	Performance au sens BER du schéma d'estimation propose versus le schéma clas- sique (V=0).	187
C.22	de traitement de base pour le IC avec estimation de canal.	188

List of Tables

1.1	Main physical layer characteristics of UMTS.	38
1.2	FDD and TDD differences.	42
1.3	Major technical differences between CDMA2000 and UMTS.	43
2.1	The developed algorithms of this Chapter.	46
2.2	Number of operations per bit.	50
2.3	The ETSI channel model.	52
2.4	Our equivalent channel model.	52
2.5	Simulation Parameters.	54
2.6	The structure of an heuristic SPV, which is used by the adaptive detector with variable number of finger and window-based channel estimation.	58
2.7	Simulation parameters.	66
2.8	Channel Model 1	66
2.9	The complexity of matrix inversion.	76
2.10	The possible configurations of the parameterized computational term.	86
2.11	Interleaver parameters.	99
2.12	Simulation parameters.	103
3.1	Linear Profile for the RAKE combination (37.87 %).	114
3.2	Linear Profile for the IC scheme (47.74 %).	114
3.3	The required time (in μsec) for the processing of one UMTS slot.	115
3.4	The device utilization for the computational unit.	131
4.1	Comparison of the HW implementation alternatives.	148
4.2	Comparison of the HW mapping approaches.	155
C.1	Le temps nécessaire (en μsec) pour le traitement d'un "slot" UMTS	181
C.2	L'utilisation du FPGA Virtex 4 pour l'unité de calcul.	181
C.3	Les différentes Configurations.	188

Notations

x^*	denotes the conjugate value of the variable x
$ x $	denotes the absolute value of the variable x
N	is the spreading factor
L	is number of channel paths
$s(t)$	denotes the transmitted signal
$r(t)$	denotes the received signal
τ_l	is the delay of the l -th channel path
$h_l(t)$	is the channel coefficient of the l -th path
U	is the number of active users
$p(t)$	denotes the pulse shaping
$c_u(t)$	is the spreading code for the u -th user
M	is the number of RAKE fingers
T_b	is the symbol duration
T_c	is the chip duration
$b_u(t)$	is the data signal of the u -th user
$n(t)$	is the white gaussian noise
σ_l^2	is the variance of the l -th channel path
f_{dec}	denotes the decision function
K	is the number of symbols
\hat{b}	is the output of a RAKE combination
\tilde{b}	denotes the output of the decision function f_{dec}
V	is the number of IC stages
c	is the threshold for the decision function
R_f	is the Rician factor
A_i	is the average received power of the i -th user
N_p	is the moving average window of the channel estimation
Δt_c	is the coherence time of the channel
f_D	is the maximum Doppler frequency
\hat{h}_l	denotes the l -th estimated channel path
G	denotes the decision reconfigurability variable
\dot{V}	denotes the speed

Abbreviations

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BoD	Bandwidth on Demand
BPSK	Binary Phase-Shift Keying
BS	Base station
CC	Convolutional Code
CCTrCH	Coded Composite Transport Channel
CPICH	Common Pilot Channel
DCA	Dynamic Channel Allocation
DD	Decorrelation Detector
DPCH	Dedicated Physical Channel
DS-CDMA	Direct Sequence Code Division Multiple Access
DSP	Digital Signal Processor
DTX	Discontinuous Transmission
EDGE	Enhanced Data Rates for GSM Evolution
FDD	Frequency Division Duplex
FEC	Forward Error Correction
FH	Frequency Hopping
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
HW	Hardware
IC	Interference Cancelation
IF	Intermediate Frequency
ILP	Instruction-Level Parallelism
IMT	International Mobile Telephony
IPI	Inter-Path Interference
ITU	International Telecommunication Union
MAC	Multiply Accumulate
MAI	Multiple Access Interference
MC	Multi-Carrier
MF	Matched Filter
MLSD	Maximum Likelihood Sequence Detection
MMSE	Minimum Mean Square Error

MRC	Maximum Ration Combiner
MS	Mobile Station
MUD	Multiuser Detection
OFDM	Orthogonal Frequency-Division Multiplexing
OVSF	Orthogonal Variable Spreading Factor
PHY	Physical
PIC	Parallel Interference Cancellation
PN	Pseudo-Noise
QoS	Quality of Service
QPSK	Quaternary Phase-Shift Keying
RC	Radio Configuration
RNC	Radio Network Controller
SF	Spreading Factor
SDR	Software Defined Radio
SIC	Successive Interference Cancellation
SIMD	Single-Instruction Stream Multiple-Data Stream
SNR	Signal-to-Noise Ratio
SPV	Supervisor
SS	Spread Spectrum
SSMA	Spread Spectrum Multiple Access
SW	Software
TDD	Time Division Duplex
TDL	Time Delay line
TDMA	Time Division Multiple Access
TFCI	Transport Format Combination Indicator
TPC	Transmit Power Control
UMTS	Universal Mobile Telecommunications System
WARC	World Administrative Radio Conference
WCDMA	Wideband Code Division Multiple Access
ZF	Zero Forcing
3GPP	3 rd Generation Partnership Project
3G	Third Generation
2G	Second Generation

The Asturies project

This thesis has been done under the framework of the French RNRT project **ASTURIES** (Approche Système pour Terminaux mUltimodes multistandards ReconfIgurableS). The target of this project was the study and implementation of a reconfigurable terminal (Radio Frequency (RF) and Base Band (BB)) which can support different standards and technologies. The partners which have been involved in this project were: PHILIPS FRANCE, France Télécom R&D, STMicroelectronics, CEA-LETI, U. Bordeaux1, STEPMIND, ENST and LIS Grenoble.

Our research group SEN (Systèmes Electroniques Numériques) of ENST was responsible for the BB and more specifically for the reception block. We had to propose and implement a novel reconfigurable reception scheme for the downlink DS/CDMA communication links. However, the consideration from the project of operational environments with only one user, directed our research activities to high data rate communication links where conventional RAKE schemes are suboptimal and new equalization/reception schemes are necessary. This was the principal motivation of our developed algorithms. The proposed reconfigurable algorithms are based on two main observations. The first one is that the Interference Cancellation (IC) can be applied to the downlink high data rate connections and with the appropriate parameters, can suppress efficiently the interference in the output of the RAKE receiver. The second observation is that IC and RAKE have computational similarities and can be calculated iteratively. These two observations-computational properties can be considered for a reconfigurable design. The combination of RAKE and IC functions for different environments, computational constraints and optimization targets gave many reconfigurable algorithms which are presented analytically in the Chapter 2 of this thesis. Moreover, the focus of the project demonstrator on single user environments, has limited our implementation activities to an inter-path reconfigurable interference cancellation, which is the third proposed algorithm of Chapter 2. The dealing implementations correspond to TigerSHARC-DSP and Virtex (Xilinx)-FPGA devices, as the used demonstrator is a heterogeneous platform consisting of these two devices.

It is important to note that in this thesis, reconfigurability is limited to an intra-standard concept which is used in order to jointly optimize communication link and computational resources. This work differs slightly from classical literature works which consider reconfigurability as a means to support different standards, technologies and modes.

Introduction

Reconfigurability is a new research topic. In general it is the ability of a system to dynamically change its functionality. However, the type of change, its scope and its implementation, are subject to different approaches in the literature. The varying definitions of the reconfigurability concept tend to produce confusion.

For the algorithmic specialists, reconfigurability is a real-time change of algorithms in order to optimize some well defined quality of service (QoS) metrics, such as the bit error rate (BER) for the physical layer. On the other hand, for the hardware people, reconfigurability is the ability of a piece of hardware to dynamically change its implemented function. There is no right or wrong in these two points of view. The confusion is generated by the multi-layered nature of reconfigurability. Reconfigurability is a multiple concept, it combines algorithms and implementations, and thus a correct definition must include both of them.

In this thesis, we deal with the reconfigurability concept in the domain of DS-CDMA downlink reception. Motivated by the ASTURIES project, we propose many novel detection schemes which use reconfigurability, in order to jointly optimize performance and computational power. The developed algorithms are a combination of three main functions: RAKE demodulation, pilot channel estimation and interference cancelation (IC). The important characteristics of these functions are that they have computational similarities and that they can be calculated iteratively. An important contribution here was the development of algorithms which use the interference cancelation technique for the downlink. In general, the IC technique is a popular, non linear, multiuser detection scheme to suppress the multi-access interference (MAI) and it requires the codes of all the active users. Due to this required input, IC is used only for the uplink, as this information is known to the base station (BS). However, for the case of a high bit rate where a low spreading factor is used, MAI is negligible and inter-path interference is the most important factor of performance degradation. For this case, an IC can be used to mitigate IPI phenomena using as input information only the code of the user under consideration.

In order to implement the developed algorithms, a novel iterative reconfigurable architecture is proposed. The reconfigurability in this architecture is based on the computational similarities and the iterative nature of the three main functions, and it is accomplished without complicated architectural changes. The achieved optimizations of the hardware resources make this architecture suitable for implementations with strict computational constraints, as the terminal units.

Finally, the formalization of the developed reconfigurable algorithms and architectures, has lead to a new reconfigurability concept, which is called “iterative approach”. This concept is a new interplay of the general two-layer reconfigurability structure and requires algorithms with a high

degree of similarities and iterative computations. This approach is appropriate for the domain of communications and signal processing, which commonly deal with this type of algorithms. This new interplay has been compared to the other approaches of the literature and their advantages have been brought out.

These original contributions will be proposed throughout this thesis which is organized as following.

The first Chapter gives an overview to the DS-CDMA systems and their basic algorithmic schemes which are important for the following Chapter of this thesis. The most important DS-CDMA commercial standards with their major characteristics are also derived.

The second Chapter presents the development algorithms which show the algorithmic dimension of reconfigurability. Our contribution is the development of six different DS-CDMA detection schemes for the downlink case. More specifically, in contrast with the conventional solutions of the literature, the presented algorithms use the reconfigurability concept in order to jointly perform performance and computational power optimization.

The third Chapter is dedicated to the hardware dimension of reconfigurability. Different implementation issues are presented in order to support the algorithmic reconfigurability of the second Chapter. The hardware implementation which is based on the computational similarities and the iterative nature of the supporting algorithms, is the most efficient. In addition to benefits from the support of the algorithms under consideration, it minimizes the required hardware resources. The proposed architecture and implementation is an important contribution of this Chapter.

The fourth Chapter generalizes the conclusions of the previous two Chapters and proposes the iterative reconfigurability concept. Reconfigurability is a two-layer concept which consists of functional and implementation changes. The intermediate layer which is a function of the algorithmic structure and of the nature of the hardware elements, categorizes the different approaches of the literature. The proposed iterative reconfigurability approach is based on the computational similarities and the iterative nature of the supporting algorithms. This approach is an attractive solution for digital communications.

A conclusion and some perspectives are given at the end of this thesis.

Chapter 1

3G communication systems

1.1 Introduction

In this Chapter we give a brief introduction to third-generation (3G) communication systems and the technology on which they are based. We present the basic algorithms that we deal with in the following Chapters of this thesis. More specifically, we introduce the Spread Spectrum and CDMA communication systems in Section 1.2. Section 1.3 presents the basic detection schemes from the literature. Finally, a description of the most important commercial 3G communication systems is given in Section 1.4, followed by concluding remarks in Section 1.5.

1.2 Spread spectrum communications

The motivation for Spread Spectrum (SS) systems as a means of multiple-access communications is found in Claude E. Shannon's pioneering work in information theory [SHA48]. In particular, in 1948, Shannon derived the maximum channel capacity of a bandlimited communications system as

$$C = W \log_2 \left(1 + \frac{S}{N} \right) \quad (1.1)$$

In Eq. 1.1, C is the channel capacity (bits/second), W is the transmission bandwidth (in Hz), S is the received signal power (in watts), and N is the total noise power at the receiver. This contribution is important in that it provides a justification for increasing the transmission bandwidth in a communication system, as the capacity is directly related to the transmission bandwidth for a given Signal-to-Noise Ratio (SNR). This idea gave impetus to the use of SS as a means of increasing the available capacity for wireless systems [COO83], [PIC82].

SS may be defined as “a technique in which an auxiliary modulation waveform, independent of the information data, is employed to spread the signal energy over a bandwidth much greater than the signal information bandwidth”. The signal is “despread” at the receiver using a synchronized replica of the auxiliary waveform. Fig. 1.1 is an a general diagram to describe SS modulation. Multiplication of two unrelated signals produces a signal whose spectrum is the convolution of the spectra of the two component signals. Thus, if the digital data (binary) signal is relatively narrow-band compared to the spreading signal, the product signal will have nearly the spectrum

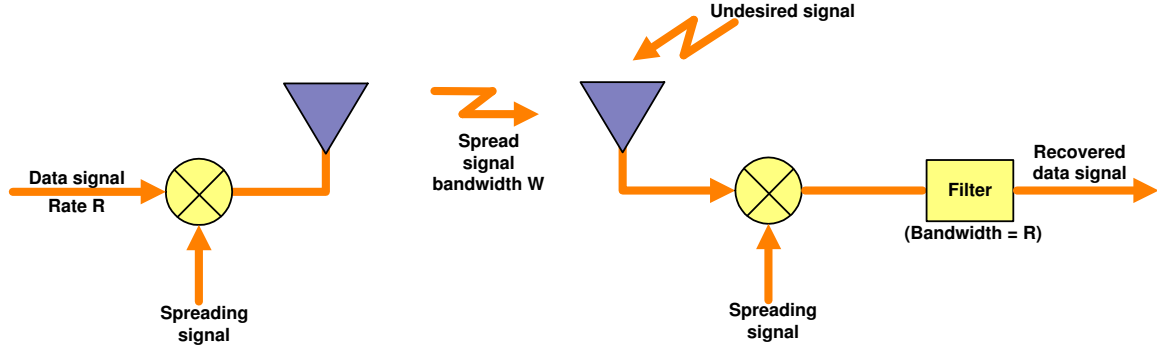


Figure 1.1: Basic spread spectrum technique.

of the wider (spreading) signal. At the demodulator, the received signal is multiplied by exactly the same spreading signal. If this spreading signal, locally generated at the receiver, is lined up (synchronized) with the received spread signal, the result is the original signal plus, possibly, some spurious higher-frequency components outside the band of the original signal, and hence easily filtered to reproduce the original data essentially undistorted. If there is any undesired signal at the receiver, on the other hand, the spreading signal will affect it just as it did the original signal at the transmitter. Thus, even if it is a narrow-band signal in the middle of the band of interest, it will be spread to the bandwidth of the spreading signal. The result is that the undesired (jamming) signal will have a bandwidth of at least W . More specifically, its average density, which is essentially uniform and can be treated as wideband noise, will be

$$N_0 = N/W \quad \text{Watts} \cdot \text{second} \quad (1.2)$$

If the data rate is R bits/second, the received energy per bit is

$$E_b = S/R \quad \text{Watts} \cdot \text{second} \quad (1.3)$$

Now it is generally recognized that, in digital communication systems, Bit Error Rate (BER) performance is a direct function of the dimensionless ratio E_b/N_0 , which for SS signals may thus be expressed as

$$\frac{E_b}{N_0} = \frac{S}{N} \frac{W}{R}, \quad (1.4)$$

and hence, the jamming power-to-signal power ratio is

$$\frac{N}{S} = \frac{W/R}{E_b/N_0} \quad (1.5)$$

This establishes that if E_b/N_0 is the minimum bit energy-to-noise density ratio needed to support a given BER, and if W/R is the ratio of spread bandwidth to the original data bandwidth, also called the processing gain, then N/S is the maximum tolerable jamming power-to-signal power ratio, also known as the jamming margin. It is obvious that for a fixed E_b/N_0 and jamming margin, the selection of the appropriate processing gain can guarantee the communication link.

There are two principal types of SS systems, Direct Sequence (DS) and Frequency Hopping (FH) [VIT95]. Direct-sequence signaling is accomplished by phase-modulating the data signal with a pseudo-noise (PN), i.e., pseudo-random, sequence of zeros and ones, which are called chips. The chip modulation is most commonly binary phase-shift keying (BPSK) for simplicity and is achieved by mod-2 adding the PN chip sequence. The number of PN chips per bit is a measure of the processing gain. Quaternary PSK (QPSK) chip modulation, while more complex, is sometimes used to prevent signal capture when a strong interferer drives the receiver into saturation. The second class of SS systems utilizes a frequency-hopping carrier. Here the spreading signal remains at a given frequency for each bit or even for several bits. Thus, locally it is no wider than the data signal, but when it hops to a new frequency, it may be anywhere within the ‘spreading’ bandwidth W . One fundamental difference between the two techniques is that direct-sequence PN spread signals can be coherently demodulated. With frequency-hopped signals, on the other hand, phase coherence is difficult to maintain when the signal frequency is hopped over a wide range; hence, this modulation is usually demodulated noncoherently.

1.2.1 DS-CDMA for personal communications

The enhancement in performance obtained from a DS spread spectrum signal through the processing gain can be used to enable many DS spread spectrum signals to occupy the same channel bandwidth provided that each signal has its own distinct PN sequence. Thus, it is possible to have several users transmit messages simultaneously over the same channel bandwidth. This type of digital communications in which each user (transmitter-receiver pair) has a distinct PN code for transmitting over a common channel bandwidth is called either Code division multiple access (CDMA) or Spread Spectrum Multiple Access (SSMA) [MAR98].

In the demodulation of each PN signal, the signal from the other simultaneous users of the channel appears as an additive interference. The level of interference varies, depending on the number of users at any given time. A major advantage of CDMA is that a large number of users can be accommodated if each transmits messages for a short period of time. In such a multiple access system, it is relatively easy either to add new users or to decrease the number of users without disrupting the system.

1.2.2 The wireless propagation channel

The signal transmitted or received by the mobile is subject to multiple reflections, diffractions, and attenuations of its energy which are caused by obstacles like buildings, hills or trees. The effect of these phenomena is to spread the signal in time, so that at the receiver we collect the superposition of several copies of the same transmitted signal arriving at different moments and having random phases and amplitudes. Thus, the multipath fading channel can be presented as a linear time-varying system having an impulse response $h(t, \tau)$ which is a wide-sense stationary random process in the t -variable.

Furthermore, the channel impulse response can be written at time t as a function of the parameter τ using the usual Tapped Delay Line (TDL) representation as:

$$h(t, \tau) = \sum_l^L \alpha_l(t) \cdot e^{j\phi_l(t)} \cdot \delta(t - \tau_l(t)), \quad (1.6)$$

where L is the number of resolvable paths, $\alpha_l(t)$, $\phi_l(t)$ are the amplitude and the phase of the l -th path, respectively, and τ_l is the path delay of the l -th path.

Using this equation, there are three quantities that have to be described by statistical methods: path amplitudes $\alpha_l(t)$, path phases $\phi_l(t)$ and delay times $\tau_l(t)$. The used statistical model depends on the propagation environment as well as the state of knowledge that we have for it [BRA99], [CHA99], [PRO95]. A successful channel model is the one which represents in the best way the propagation environment under consideration by incorporating only the available information and nothing more [DEB00]. For the statistical model used all along this thesis, we suppose that path phases are uniform distributed, path amplitudes are Rayleigh or Rice distributed, and path delays have a constant value. In the follow, we present the two path amplitudes' distributions [PAP02].

Rayleigh fading

The Rayleigh distribution

$$P_\alpha(\alpha) = \frac{\alpha}{\sigma^2} \cdot e^{-\frac{\alpha^2}{2\sigma^2}}, \quad \alpha \geq 0, \quad (1.7)$$

describes the amplitude fading for the case of a superposition of scattered components with approximately identical power. It is appropriate for the modeling of paths amplitudes in outdoor environment. In Eq. 1.7, $2\sigma^2$ is the variance of the Rayleigh random variable α .

Rice fading

The Rice distribution

$$P_\alpha(\alpha) = \frac{\alpha}{\sigma^2} \cdot I_0\left(\frac{\alpha \cdot \mu}{\sigma^2}\right) e^{-\frac{\alpha^2 + \mu^2}{2\sigma^2}}, \quad \alpha \geq 0, \quad (1.8)$$

represents a Rayleigh scattering process with a superimposed dominant component. It is appropriate for the modeling of the propagation channel in a satellite environment where a Line-of-Sight (LOS) signal exists. In Eq. 1.8, α is the Ricean random variable, $I_0()$ is a modified Bessel function of the first kind, and σ^2 and μ are the parameters of the distribution.

1.3 CDMA detection

1.3.1 The conventional RAKE receiver

In 1958, Price and Green proposed a method of resolving multipath problems using wideband pseudorandom sequences modulated onto a transmitter using other modulation methods (AM or FM) [PRI58]. The pseudorandom sequence has the property that time-shifted versions of itself are almost uncorrelated. Thus, a signal that propagates from transmitter to receiver over multiple paths (hence multiple different time delays) can be resolved into separately fading signals by cross-correlating the received signal with multiple time-shifted versions of the pseudorandom sequence. The receiver is called RAKE receiver since the block diagram looks like a garden rake. Fig. 1.2 shows a block diagram of a typical RAKE receiver.

The RAKE receiver is a simple implementation of the matched filter which accounts for the most important components of the multipath propagation channel. It consists of a bank of demodulators, called RAKE fingers, each tracking a path of the channel, and a multipath

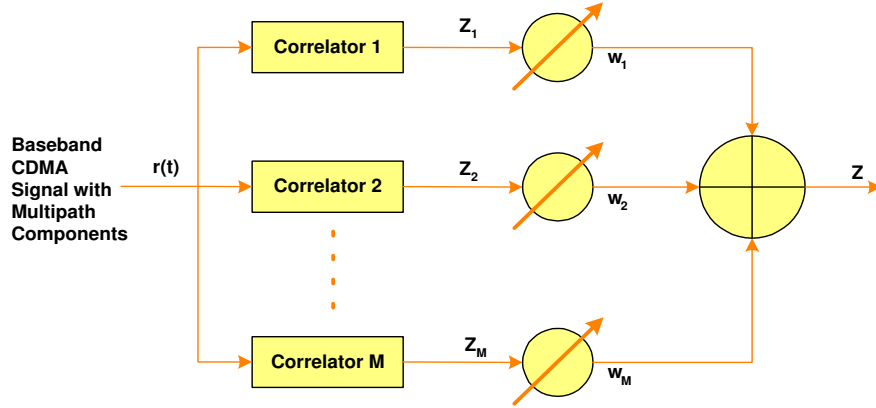


Figure 1.2: RAKE receiver correlator.

combiner which combines the outputs of these fingers (the multipath components) in proportion to their strengths. This combining is a form of diversity and can help reduce fading. Multipath components with relative delays of less than $Dt = 1/W$ cannot be resolved and, if present, contribute to fading; in such cases, Forward Error Correction (FEC) and power control schemes play a dominant role in mitigating the effects of fading.

The outputs of the M correlators are denoted as Z_1, Z_2, \dots, Z_M . The weights of the outputs are w_1, w_2, \dots, w_M respectively. The weighing coefficients are based on the power or the SNR from each correlator output [MAN04], [SIN02]. If the power or SNR is small from a particular correlator, it is assigned a small weighing factor. The composite signal is given by

$$Z = \sum_{m=1}^M w_m \cdot Z_m \quad (1.9)$$

As for the weighing coefficients, different approaches exist in the literature. Among them, the most important are the selection combiner, the equal-gain combiner and the Maximum-Ratio Combiner (MRC) [HAN00]. The selection combiner is the simplest of all the schemes. In its ideal application, it chooses the signal with the highest instantaneous SNR, so the output SNR is equal to that of the best incoming diversity component. The last two combiners require the estimation of some channel parameters and thus this type of detection can be considered as coherence detection. The equal-gain combiner requires that the receiver can estimate the phase offsets on each of the M received components. These estimated phases are the weighing coefficients of the equal-gain combination approach. According to equation 1.9, the equal-gain combination corresponds to $w_m = e^{-j\hat{\phi}_m}$, where $\hat{\phi}_m$ is the estimated phase offset of the m -th diversity component. If, in addition, the received signal power is estimated for each of the diversity components and it takes part in the previous combination, the resulting combination technique is called MRC. Based on the general equation 1.9, the MRC corresponds to $w_m = \hat{\alpha}_m e^{-j\hat{\phi}_m}$, where $\hat{\alpha}_m$ is the estimated signal power of the m -th diversity component. The MRC is the most efficient one when a reliable power estimation exists, and thus it is considered as the combination approach of the conventional RAKE receiver in the following Chapters of this thesis [ALO97].

In CDMA cellular/PCS systems (commercial IS- 95 system), the forward link (base station

(BS) to terminal (MS)) uses a three-finger RAKE receiver, and the reverse link (MS to BS) uses a four-finger RAKE receiver.

Channel estimation

The coherence detection introduced by the combination scheme of the conventional RAKE receiver requires the parameter estimation of the propagation channel. These parameters consist of the phase offset and the power amplitude for each diversity component, as well as the path delay.

Channel estimation is a huge research topic, and many approaches are proposed in the literature. In general, we can find all the classical algorithms of signal processing for parameter estimation, applied for channel estimation [HAY02], [MAM02]. The best trade-off between quality of estimation and implementation complexity is always the practical selection criterion of a channel estimation algorithm.

In this thesis, we suppose a simple data-aided channel estimation based on a pilot physical channel, which is transmitted in parallel with the data channel. In this case, the channel estimation is performed by a correlation of the received signal with different phases of the pilot code. Multipath arrivals at the receiver unit manifest themselves as correlation peaks that occur at different times. The time of each peak, relative to the first arrival, provides a measurement of the path delay. Moreover, the peak amplitude and phase provide an estimation of the path amplitude and phase, respectively. The complex amplitude estimation (phase and power envelope) needs to be averaged over a reasonably long period, while coherence time sets the upper limit for the averaging time (i.e., the channel should not change during the estimation). This process is necessary in order to average out the Gaussian noise and improves the estimation quality.

The required estimation speed for impulse response measurements depends on the mobile speed and the radio environment. The faster the mobile station is moving, the faster the measurements need to be performed in order to catch the best multipath components for the RAKE fingers. Furthermore, in a long delay spread environment, the scanning window needs to be wider.

The limitations of the RAKE receiver

DS-CDMA systems support a multitude of users within the same bandwidth by assigning different - typically unique - codes to different users for their communications, in order to be able to distinguish their signals from each other. When the transmitted signal is subjected to hostile wireless propagation environments, the signals of different users interfere with each other and hence CDMA systems are interference- limited due to this multiple access interference (MAI), generated by the users transmitting within the same bandwidth simultaneously. Moreover, the interference between the different paths of the channel, which is called Inter-Path interference (IPI), can further degrade the achieved performance for the cases where a low spreading factor is used.

The conventional RAKE receiver is optimized for detecting the signal of a user for a single-user environment. However, even its simple structure and implementation, it is inefficient for the environments with MAI and IPI interference, since the interference is treated as unstructured gaussian noise and the knowledge of the channel impulse response (CIR), or the spreading

sequences of the interferers, is not exploited. In order to mitigate MAI and IPI phenomena, a range of advanced detectors have been proposed in the literature, which will be reviewed in the forthcoming subsections.

1.3.2 The advanced receiver schemes

These schemes try to suppress MAI and IPI interference in order to improve performance and capacity. They are popular under the term Multi-User Detection (MUD), as for the detection of a user, they require information regarding the other (interfering) users [DUE95], [KOU00], [MOS96]. Because of the huge complexity of the optimal MUD based on a Maximum Likelihood Sequence Estimation (MLSE) criterion, many suboptimal MUD techniques have been developed in recent years. These suboptimal MUDs can be divided in two categories: linear MUDs (e.g. the Decorrelation Detector (DD), Minimum Mean Square Error (MMSE) detector) and nonlinear interference cancelers, such as the Successive Interference Canceler (SIC) and the Parallel Interference Canceler (PIC). Interference cancellation has a lower complexity than decorrelation and MMSE detection and can easily be implemented in hardware. In addition to this, in a synchronized CDMA system over a AWGN channel, nonlinear interference cancellation is superior to any type of linear MUD.

Maximum Likelihood Sequence Estimation (MLSE) Detector

In 1986, Verdu presented and analyzed probably the most widely recognized multi-user receiver so far, the MLSE [VER86]. Its implementation is achieved by placing a Viterbi decoder immediately after the bank of matched filters. Despite the fact that the performance of this receiver is optimum, its computational complexity, in terms of number of operations $O(2^U)$, grows exponentially with the number of users (U), making the implementation of such a receiver extremely difficult. Nevertheless, its performance serves as a benchmark for the comparison of other MUD receivers.

Decorrelation Detector (DD)

The DD applies the inverse of the correlation matrix to the conventional detector output in order to decouple the data [TSA96], [ZVO96]. The output is just the decoupled data plus a noise term. The DD completely eliminates the MAI. It is very similar to the zero-forcing equalizer which is used to completely eliminate ISI [VIT94]. It is shown to have many attractive properties. Foremost among these properties are:

- Provides substantial performance/capacity gains over the conventional detector, under most conditions.
- Does not need to estimate the received amplitudes. In contrast, detectors that require amplitude estimation are often quite sensitive to estimation error.
- Has computational complexity significantly lower than that of the maximum likelihood sequence detector. The per-bit complexity is linear in the number of users, excluding the costs of re-computation of the inverse mapping.

A disadvantage of this detector is that it causes noise enhancement (similar to the zero-forcing equalizer). The power associated with the noise term at the output of the DD is always greater than or equal to the power associated with the noise term at the output of the conventional detector for each bit. A more significant disadvantage of the DD is that the computations needed to invert the correlation matrix are difficult to perform in real-time. Despite these drawbacks, the DD generally provides significant improvements over the conventional detector. Fig. 1.3 shows the principle of the DD using a simple transmission scheme.

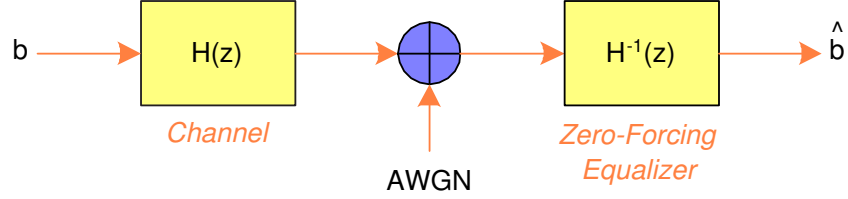


Figure 1.3: Block diagram of a simple transmission scheme using a zero-forcing equalizer.

Minimum Mean Square Error (MMSE) Detector

This detector is based on the MMSE criterion used for equalization against ISI [VIT94]. It performs a linear transformation that minimizes the MSE of the bit estimate $E[(b - \hat{b})^T(b - \hat{b})]$, where b and \hat{b} represent the transmitted and estimated bit respectively [MIL00], [XIE90].

Therefore, the MMSE detector calculates the inverse of the correlation matrix, at the same time taking into account the noise present in the system. The less noise exists, the more accurate will be the inverse of the correlation matrix, without increasing the asymptotic efficiency. Due to the fact that this detector considers and makes up for both noise and MAI, it has better BER performance than the DD. However, as the background noise tends to zero, its performance approaches that of the DD. On the other hand, as the system noise increases, the transformation approaches the identity matrix scaled by $N_0/2$, and thus reduces to the conventional matched filter receiver. Overall, MMSE imposes the same computational disadvantages as the DD, such as matrix inversion. In addition, it requires knowledge regarding the received powers of the interfering users, in its optimal application [KLE96, KLE97]. Fig. 1.4 shows the principle of the MMSE detector using a simple transmission scheme. In this simple system, $H(z)$ is the channel transfer function and $F(z)$ is the transfer function of the equalizer.

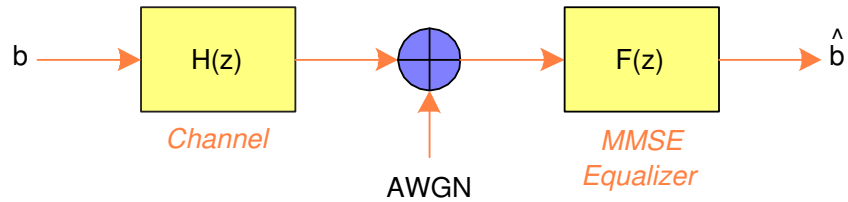


Figure 1.4: Block diagram of a simple transmission scheme employing an MMSE equalizer.

Interference Cancellation (IC) Detector

Another important group of detectors can be classified as subtractive interference cancellation detectors. These detectors have linear complexity with respect to the number of users U , they don't require matrix inversions and thus are appropriate for low-cost hardware implementations [RAJ02]. The basic principle is the creation, at the receiver, of separate estimates of the MAI contributed by each user, in order to subtract out some or all of the MAI seen by each user. However, this MAI regeneration requires the knowledge of the spreading codes for all the active users, and thus IC approach can be used only for the uplink. Such detectors are often implemented with multiple stages, where the expectation is that the decisions will improve at the output of successive stages.

These detectors are similar to feedback equalizers used to combat ISI. In feedback equalization, decisions on previously detected symbols are fed back in order to cancel part of the ISI. Thus, a number of these types of multi-user detectors are also referred to as decision-feedback detectors.

The bit decisions used to estimate the MAI can be hard or soft. The soft-decision approach uses soft data estimates for the joint estimation of the data and amplitudes, and is easier to implement. The hard-decision approach feeds back a bit decision and is nonlinear; it requires reliable estimates of the received amplitudes in order to generate estimates of the MAI. If reliable amplitude estimation is possible, hard-decision subtractive interference cancellation detectors generally outperform their soft-decision counterparts. However, the need for a reliable amplitude estimation is a significant liability of the hard-decision techniques: imperfect amplitude estimation may significantly reduce or even reverse the performance gains available [GRA95]. IC is presented in the literature with two basic approaches.

Subtractive Interference Canceller (SIC)

The SIC detector [HUI98], [PAT93], [YOO93] takes a serial approach to canceling interference. Each stage of this detector regenerates and cancels out one additional direct-sequence user from the received signal, so that the remaining users see less MAI in the next stage.

A simplified diagram of the first stage of this detector is shown in Fig. 1.5, where a hard-decision approach is assumed. The first stage is preceded by an operation which ranks the signals in descending order of received powers (not shown). The first stage implements the following steps:

1. Detects the strongest signal, r_1 , with the conventional detector
2. Makes a data decision on r_1
3. Regenerates an estimate of the received signal for user one, $\hat{r}_1(t)$ using:
 - Data decision from step 2
 - Knowledge of its PN sequence
 - Estimates of its timing and amplitudes (and phases)
4. Cancels (subtracts out) $\hat{r}_1(t)$ from the total received signal $r(t)$, yielding a partially cleaned version of the received signal $r^{(1)}(t)$

Assuming that the estimation of $r_1(t)$ in step 3 above was accurate, the outputs of the first stage are:

1. A data decision on the strongest user
2. A modified received signal without the MAI caused by the strongest user

This process can be repeated in a multistage structure: the i -th stage takes as its input the “partially cleaned” received signal output by the previous stage, $r^{(i-1)}(t)$, and outputs one additional data decision (for signal b_i) and a “cleaner” received signal, $r^{(i)}(t)$.

The reasons for canceling the signals in descending order of signal strength are straightforward. First, it is easiest to achieve acquisition and demodulation on the strongest users (best chance for a correct data decision). Second, the removal of the strongest users gives the most benefit for the remaining users. The result of this algorithm is that the strongest user will not benefit from any MAI reduction; the weakest users, however, will potentially see a huge reduction in their MAI.

The SIC detector requires only a minimal amount of additional hardware and has the potential to provide significant improvement over the conventional detector. It does, however, pose a couple of implementation difficulties. First, one additional bit delay is required per stage of cancelation. Thus, a trade-off must be made between the number of users that are canceled and the amount of delay that can be tolerated. Second, there is a need to reorder the signals whenever the power profile changes. Here, too, a trade-off must be made between the precision of the power ordering and the acceptable processing complexity.

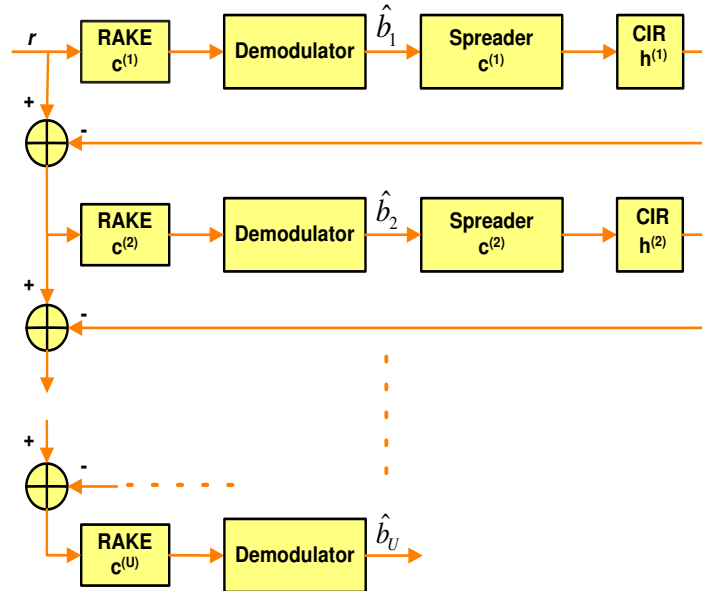


Figure 1.5: Schematic of the SIC receiver for U users. The users’ signals have been ranked, where user 1’s signal was received at the highest power, while user U ’s signal at the lowest power. In the order of ranking, the data estimates of each user are obtained and the received signal of each user is reconstructed and canceled from the received composite signal, r .

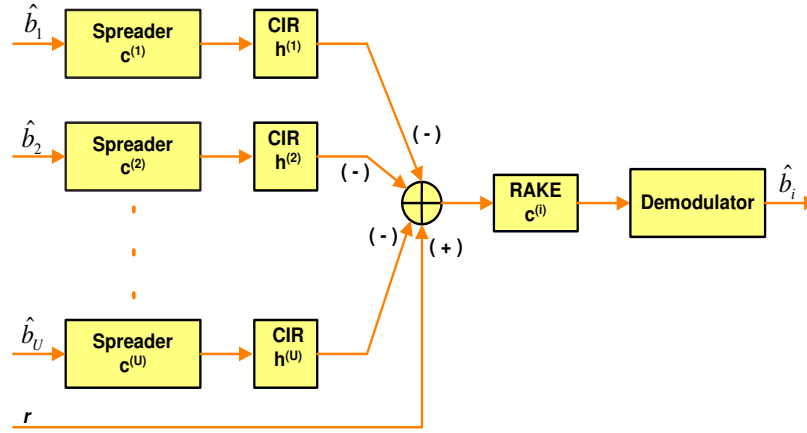


Figure 1.6: Schematic of a single cancellation stage for user i in the PIC receiver for U users. The data estimates, $\hat{b}_1, \dots, \hat{b}_U$ of the other $(U - 1)$ users were obtained from the previous cancellation stage, and the received signal of each user other than the i -th one is reconstructed and canceled from the received signal, r .

Parallel Interference Canceler (PIC)

Canceling all users simultaneously is the alternative to the successive approach [DIV98], [VAR90]. As illustrated in Fig. 1.6, all U users create replicas of their interference contribution to the other $U - 1$ users' signals. Then these replicas are subtracted simultaneously from the $U - 1$ users' signals. The data estimates from the output of the first stage can be fed into a second stage to be used as interference replica estimates, thus giving better data estimates at the output of the second stage. As the number of stages (V) increases the data estimates become better but the number of operations performed becomes greater.

Assuming that the original interference estimates are correct, this scheme can offer full interference reduction. Regarding the delay, the parallel process ensures low delay for the detection of all users. However, since each user has to cancel its signal $U - 1$ times, it means that the scheme requires a large number of regenerators/cancellations and thus has a high complexity. In order to jointly optimize performance, complexity and time latency, hybrid IC schemes have been proposed in the literature. These schemes consist of different combinations of SIC and PIC in order to obtain only the advantages from each IC approach [KOU98], [LI94].

1.4 3G communication systems

Work to develop third-generation mobile systems started when the World Administrative Radio Conference (WARC) of the ITU (International Telecommunications Union), at its 1992 meeting, identified the frequencies around 2 GHz that were available for use by future 3G mobile systems, both terrestrial and satellite. Within the ITU, these 3G systems are called International Mobile Telephony 2000 (IMT-2000). Within the IMT-2000 framework, several different air interfaces are defined, based on either CDMA or TDMA technology [HOL02]. The original target of the third-generation proposal was a single common global IMT-2000 air interface. Third-generation systems are closer to this target than were second-generation (2G) systems : the same air interface

-Wideband CDMA (WCDMA)- is to be used in Europe and Asia, including Japan and Korea, using the frequency bands that WARC-92 allocated for the third-generation IMT-2000 system at around 2 GHz. In North America, however, that spectrum has already been auctioned for operators using second-generation systems, and no new spectrum is available for IMT-2000. Thus, third-generation services and also WCDMA, there must be implemented within the existing bands.

In addition to WCDMA, the other air interfaces that can be used to provide third-generation services are Enhanced Data Rates for GSM Evolution (EDGE) and CDMA2000. EDGE can provide third-generation services with bit rates up to 500 kbps within a GSM carrier spacing of 200 KHz. It includes advanced features that are not part of GSM to improve spectrum efficiency and to support new services. Finally, CDMA2000 can be used as an upgrade solution for the existing IS-95 operators.

1.4.1 Differences between 3G and 2G air interfaces

To understand the differences between second- and third-generation systems, we need to look at the new requirements of the third-generation systems, which are listed below :

- Bit rates up to 2 Mbps.
- Variable bit rate to offer bandwidth on demand.
- Multiplexing of services with different quality requirements on a single connection, e.g. speech, video and packet data.
- Delay requirements from delay-sensitive real-time traffic to flexible best-effort packet data.
- Quality requirements from 10% Frame Error Rate (FER) to 10^{-6} BER.
- Coexistence of second- and third-generation systems and inter-system handovers for coverage enhancements and load balancing.
- Support of asymmetric uplink and downlink traffic, e.g. web browsing causes more loading to downlink than to uplink.
- High spectrum efficiency.
- Coexistence of Frequency Division Duplex (FDD) and Time Division Duplex (TDD) modes.

The differences in the air interface reflect the new requirements of the third-generation systems. For example, a larger bandwidth of 5 MHz is needed to support higher bit rates. Transmit diversity is included in 3G systems to improve the downlink capacity, to support the asymmetric capacity requirements between downlink and uplink. Transmit diversity is not supported by the second-generation standards. The mixture of different bit rates, services and quality requirements in third-generation systems requires advanced radio resource management algorithms to guarantee quality of service and to maximize system throughput. Also, efficient support of non-real-time packet data is important for new services.

1.4.2 UMTS

UMTS is the European proposition of the third-generation wireless communication systems [3GPPa, 3GPPb, 3GPPc]. It uses a WCDMA radio technology and thus the terms “UMTS” and “WCDMA” are often used interchangeably. It builds on GSM, which is currently the most widely used wireless technology in the world.

Main parameters in WCDMA air interference

WCDMA is a wideband DS-CDMA system, i.e. user information bits are spread over a wide bandwidth by multiplying the user data with quasi-random bits (chips) derived from CDMA spreading codes. In order to support very high bit rates (up to 2 Mbps), the use of a variable spreading factor and multi-code connections is supported [MAN02].

The chip rate of 3.84 Mcps used leads to a carrier bandwidth of approximately 5 MHz. DS-CDMA systems with a bandwidth of about 1 MHz, such as IS-95, are commonly referred to as narrowband CDMA systems. The inherently wide carrier bandwidth of WCDMA supports high user data rates and also has certain performance benefits, such as increased multipath diversity. According to its operating licence, the network operator can deploy multiple such 5 MHz carriers to increase capacity, possibly in the form of hierarchical cell layers.

WCDMA supports highly variable user data rates, in other words the concept of obtaining Bandwidth on Demand (BoD) is well supported. Each user is allocated frames of 10 ms duration, during which the user data rate is kept constant. However, the data capacity among the users can change from frame to frame.

WCDMA supports two basic modes of operation: FDD and TDD. In the FDD mode, separate 5 MHz carrier frequencies are used for the uplink and downlink respectively, whereas in TDD mode only one 5MHz carrier is time-shared between uplink and downlink. The TDD mode is based heavily in FDD mode concepts and was added in order to leverage the basic WCDMA system also for unpaired spectrum allocations of the ITU for the IMT-2000 systems.

WCDMA supports the operation of asynchronous BSs, so that, unlike in the synchronous IS-95 system, there is no need for a global time reference, such as a Global Positioning System (GPS). Deployment of indoor and micro BSs is easier when no GPS signal needs to be received.

WCDMA employs coherence detection on uplink and downlink based on the use of pilot symbols or a common pilot. While already used on the downlink in IS-95, the use of coherence detection on the uplink is new for public CDMA systems, and will result in an overall increase of coverage and capacity on the uplink.

The WCDMA air interface has been crafted in such a way that advanced CDMA receiver concepts, such as multiuser detection and smart adaptive antennas, can be deployed by the network operator as a system option to increase capacity and/or coverage. In most second-generation systems, no provision has been made for such receiver concepts and as a result they are either not applicable or can be applied only under severe constraints, with limited increases in performance.

WCDMA is designed to be deployed in conjunction with GSM. Therefore, handovers between GSM and WCDMA are supported in order to be able to leverage the GSM coverage for the introduction of WCDMA.

Chip rate	3.84 Mcps
Carrier spacing	4.4 to 5 MHz with a 200 KHz raster
Frame length	10 ms
Frame structure	15 time slots per frame
Modulation	QPSK
Spreading	OVSF
Scrambling	Long/Short codes
Channel Coding	Convolutional (rate 1/2 to 1/3) or Turbo codes for $BER < 10^{-3}$
Pulse shaping	Root-Raised Cosine (RRC) with roll-off factor 0.22

Table 1.1: Main physical layer characteristics of UMTS.

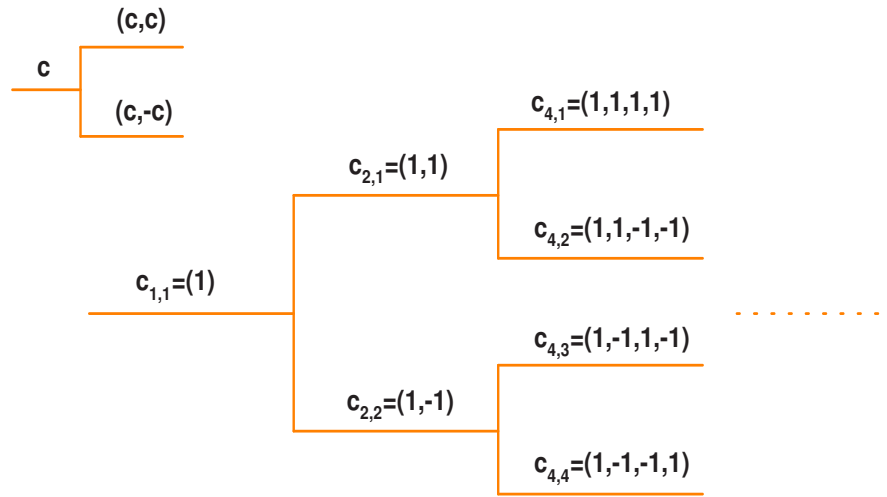


Figure 1.7: The channelization tree.

Physical Layer

In order to handle the network complexity, WCDMA uses a layered architecture, like other wireless systems. In this paragraph we present the most important characteristics of the physical layer (PHY) for a FDD connection, which is used in the simulation parts of this thesis. Table 1.1 summarizes the main physical layer characteristics of UMTS.

Spreading

Transmissions from a single source are separated by channelization codes, i.e. downlink connections within one sector and the dedicated physical channel in the uplink from one terminal. The spreading/channelization codes are based on the orthogonal variable spreading factor (OVSF) technique, which was originally proposed in [ADA97].

The use of OVSF codes allows the spreading factor to be changed and orthogonality between different spreading codes, of different lengths, to be maintained. The codes are picked from the code tree, which is illustrated in Fig. 1.7. In case the connection uses a variable spreading factor, the proper use of the code tree also allows despreading according to the smallest spreading factor.

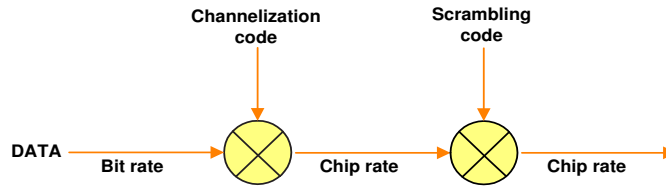


Figure 1.8: Relation between spreading and scrambling.

There are certain restrictions as to which of the channelization codes can be used for a transmission from a single source. Another physical channel may use a certain code in the tree if no other physical channel to be transmitted using the same code tree is using a code that is on an underlying branch, i.e. using a higher spreading factor code generated from the intended spreading code to be used. Neither can a smaller spreading factor code on the path to the root of the tree be used. The downlink orthogonal codes within each BS are managed by the radio network controller (RNC) in the network.

Scrambling

In addition to spreading, another part of the process in the transmitter is the scrambling operation. This is needed to separate terminals or BSs from each other. Scrambling is used on top of spreading, so it does not change the signal bandwidth but only makes the signals from different sources separable from each other. With the use of scrambling, it would not matter if the actual spreading were done with an identical code for several transmitters. Fig. 1.8 shows the relation of the chip rate in the channel to spreading and scrambling. As the chip rate is already achieved in the spreading by the channelization codes, the symbol rate is not affected by the scrambling.

For the uplink, there are two types of scrambling codes : the short and the long ones. The first code type is a 256-chip-long Extended S(2) code family, and used when a multiuser detection or interference cancellation scheme is applied in the reception. The long scrambling codes are Gold codes, which are 10 ms in length (38400 chips), to cover a WCDMA frame. The long scrambling code is truncated by a duration of 38,400 chips from the beginning of the Gold sequence, with a repetition period of 2^{24} chips, and thus there are 2^{24} chip-long scrambling codes.

The downlink scrambling uses long codes, the same Gold codes as in the uplink. The complex-valued scrambling code is formed from a single code by simply having a delay between the I and Q branches. The scrambling code in the downlink is generated by truncating the 38,400 chips from the beginning of the Gold sequence with the repetition period of 2^{18} , and its shifted version by 131,072 chips. The 8,192 scrambling codes are grouped into 512 scrambling code groups, where each group comprises 1 primary scrambling code with 15 corresponding secondary scrambling codes. The primary code is first used, and then the secondary scrambling codes are used to cover any shortage in the channelization code set associated with the primary scrambling code.

Physical Channels

Physical channels map a logical connection to the PHY. They are specified by the carrier frequency, codes (channelization code and scrambling code) and phase, and they can be classified as dedicated or common channels.

One radio frame of a physical channel has a frame length of 10 msec and comprises 15 slots.

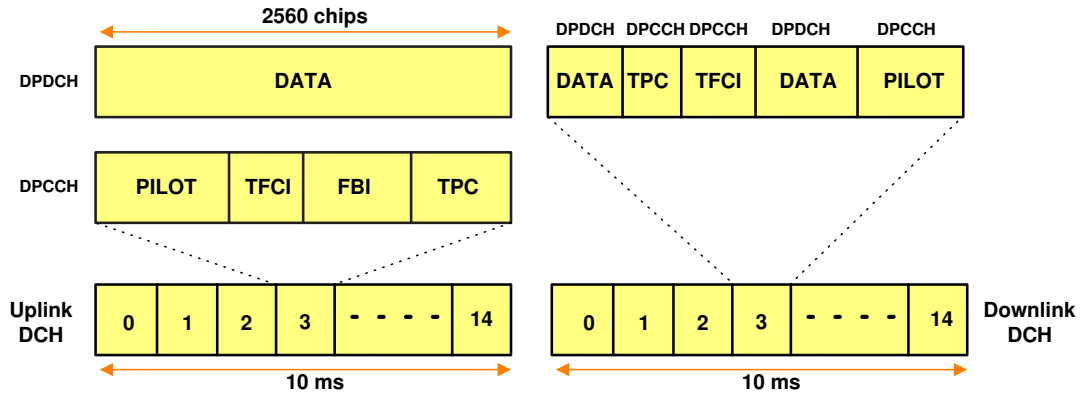


Figure 1.9: Uplink/Downlink dedicated physical channel structure.

The number of channel-coded information bits, which each physical channel conveys, differs according to the type of physical channel and the spreading factor (SF). The features of the major physical channels which are used in this thesis are described below :

- Dedicated Physical Channel (DPCH)

The Dedicated Channel (DCH) information is carried by the Uplink/Downlink Dedicated Physical Channels (DPCHs). The DPCH consists of two types of channels - Dedicated Physical Data Channel (DPDCH) and Dedicated Physical Control Channel (DPCCH). In uplink the DPDCH and DPCCH are I/Q multiplexed with each frame, since in downlink DPCHs are time-multiplexed with PHY related control information (such as pilot bits). Fig. 1.9 presents the frame structure of the uplink and downlink respectively. The DPCH spreading factor can range from 512 down to 4. Uplink I/Q multiplexing is used to ensure continuous transmission in order to reduce audible interference. Downlink time multiplexing is used to save the orthogonal codes. Since the downlink common channels are transmitted all the time, Discontinuous Transmission (DTX) is not used for the downlink.

The downlink DPCH bit rate can change from frame to frame, and lower data rate transmission will be handled by DTX. When the total transmitted bit rate in one downlink Coded Composite Transport Channel (CCTrCH) exceeds the maximum bit rate for a downlink physical channel, then multicode transmission can be used. For multicode operation, several parallel downlink DPCHs are transmitted for one CCTrCH using the same spreading factor [HOL02]. In this case, the PHY control information will be transmitted only over the first downlink DPCH.

- Common Pilot Channel (CPICH)

The common pilot channel is an unmodulated code channel, which is scrambled with the cell-specific primary scrambling code. The function of the CPICH is to aid the channel estimation at the terminal for the dedicated channel and to provide the channel estimation reference for the common channels, when they are not associated with the dedicated channels or not involved in the adaptive antenna techniques.

UMTS has two types of common pilot channel, primary and secondary. The difference is that the Primary CPICH is always under the primary scrambling code with a fixed

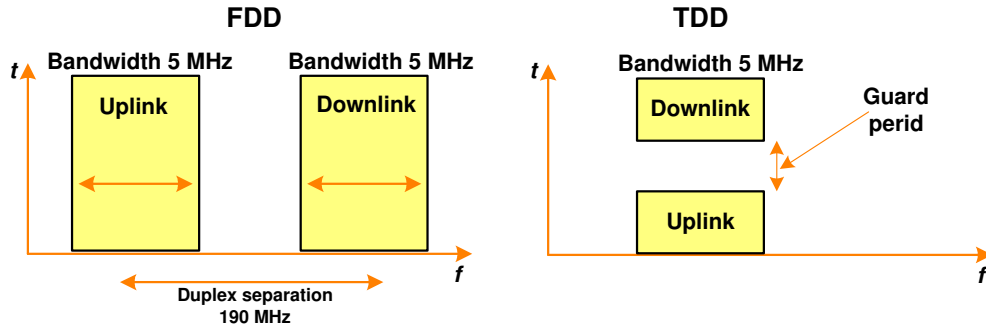


Figure 1.10: Principles of FDD and TDD operation.

channelization code allocation and there is only one such channel for a cell or sector. The Secondary CPICH may have any channelization code of length 256 and may be under a secondary scrambling code as well. The typical area of Secondary CPICH usage would be in operations with narrow antenna beams, intended for service provision at specific ‘hot spots’ or places with high traffic density.

An important application for the primary common pilot channel is in the measurements for the handover and cell selection/reselection. The use of CPICH reception level at the terminal, for handover measurements, has the consequence that by adjusting the CPICH power level the cell load can be balanced among different cells. Reducing the CPICH power level causes part of the terminals to hand over to other cells, while increasing it invites more terminals to hand over to this cell, as well as to make their initial access to the network in that cell.

UMTS-TDD Mode

The imminent arrival of the third-generation cellular system has resulted in the creation of many different usage scenarios, from video conference to Internet access in addition to conventional voice traffic. The TDD mode of UMTS potentially provides the flexibility and adaptability to support the different requirements of latency, asymmetric and variable rate traffic in a single terminal architecture and hence offers many advantages to service providers, operators, manufacturers and users.

The TDD mode uses a combined time-division and code-division multiple access (TD/CDMA) scheme that adds a CDMA component to a TDMA system. The different user signals are separated in both time and code domain. The TDD system can be implemented on an unpaired band, while the FDD system always requires a pair of bands. Fig. 1.10 schematically presents the principles of the two UMTS operation modes. In TDD operation, uplink and downlink are divided in the time domain. It is possible to change the duplex switching point and move capacity from uplink to downlink, or vice-versa, depending on the capacity requirements for both links.

Although much commonality now exists between the TDD and FDD modes, since their alignment within Third-Generation Partnership Program (3GPP) [3GPPc], several important key differences remain. Table 1.2 summarizes the basic differences between the two operational

	FDD	TDD
Multiple-access method	Direct-Sequence CDMA	TDMA/CDMA
Spreading factor	2-512 Mcps	1-16 Mcps
Burst types	No burst defined	Traffic, Random access, Synchronization
Intrafrequency handover	Soft	Hard
Multirate concept	Multicode and variable spreading factor	Multislot, multicode, and variable spreading factor
Power control for dedicated channels	Closed loop with rate of 1500 Hz	Uplink: open loop with rate of 100 or 200 Hz. Downlink: closed loop with rate of equal or less than 800 Hz
Channel allocation	No DCA required	Slow and fast dynamic channel allocation (DCA)
Capacity allocation between uplink and downlink	5 MHz for uplink/downlink	5 MHz carrier divided between uplink and downlink. Downlink / uplink capacity can be adjusted between 2-14 out of 15 slots

Table 1.2: FDD and TDD differences.

modes. Among them, the use of very low spreading factors ($SF = 2..16$) has an important impact in the designed reception schemes. It is obvious that the use of a conventional RAKE receiver, which has an efficient performance for a high spreading factor environment, can not mitigate the high interference embedded in the TDD mode. Advanced detection schemes are necessary for this operational mode [KOU00b].

However, an important advantage of TDD over FDD is that, since the same frequency channel is used, reciprocity exists between the link channel characteristics. This fact can be used to implement a number of important functions in an open-loop fashion, including power control, signal pre-emphasis and shaping, and diversity transmission (as compared to diversity reception) to respond to unfriendly urban mobile channel conditions [ESM93], [HAN02a]. All these functions will help further reduce the complexity of the portable mobile unit, resulting in less costly devices.

1.4.3 CDMA2000

CDMA2000 is the American proposition for 3G communication systems [RAO99], [3GPP2]. It is also a wideband, spread spectrum radio interface that uses CDMA technology to meet the needs of the next generation of wireless communication systems. Its physical layer retains backwards compatibility, not only to leverage IS-95 equipment development, but also to provide a smooth upgrade path for cellular operators. In this way, CDMA2000 systems could be gradually phased into existing IS-95 networks, without disrupting service. As a result, many mechanisms such as reverse link power control and soft handoff remain essentially the same from the PHY standpoint. However, their new characteristics can support the requirements embedded in 3G technology.

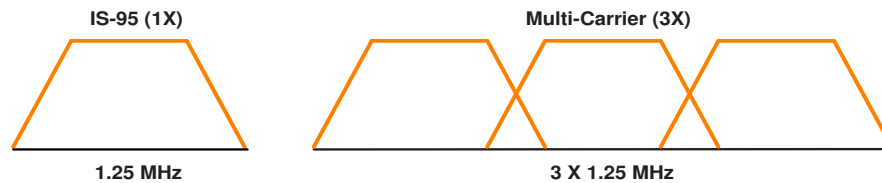


Figure 1.11: Relationship between the MC mode (3X) and IS-95 (1X) in spectrum usage.

	CDMA2000	UMTS
Core network	ANSI-41	GSM Mobile Application Part (MAP)
Chip rate	3.6864/1.2288 Mcps	3.84 Mcps
Synchronized BS	yes	no
Frame length	20 ms	10 ms
Multicarrier spreading option	yes	no
Voice coder	Enhanced Variable Rate Coder (EVRC)	new

Table 1.3: Major technical differences between CDMA2000 and UMTS.

The CDMA2000 physical layer classifies different modes of operation into radio configurations (RCs), for both the forward and reverse links. For instance, Radio Configurations 1 and 2 (RC1 and RC2) are the Rate Set 1 and Rate Set 2 modes of operation respectively, in IS-95. However, radio configurations greater than 2 define new modes of operation in CDMA2000.

In addition, the CDMA2000 radio configurations encompass two modes of operation: 1X and 3X. 1X refers to the mode that is bandwidth-compatible with IS-95, i.e., its bandwidth is 1.25 MHz. 3X refers to the multicarrier option, which involves the use of 3 1X carriers to increase the data rate to the mobile user on the forward link. The data rates on the reverse link, in the multicarrier version, increase data rates via direct spreading up to three times the 1X chip rate of 1.2288 MHz. More recently, modes that involve 3X forward link and 1X reverse link have been adopted to allow for asymmetric high-speed data services. Fig. 1.11 presents the two operated modes of CDMA2000. Table 1.3 lists the major technical differences between the two most important wideband CDMA 3G proposals.

1.4.4 3G and satellites

For the first time, the satellite is seen as an integral part of a global cellular communication network [GAU99]. For a 3G communication system, the part of the general architecture which is dealing with the satellite component is defined as satellite 3G (S-3G). Due to their geographical position, satellites provide a global coverage and can contribute towards a true roaming. More specifically, they can provide services in the rural areas where terrestrial infrastructure is not installed, or where it can not be installed, as in the middle of the oceans.

However, satellite services can only represent a subset of those provided by terrestrial systems due to their technological and physical constraints. The satellite environment involves large

propagation delays and low SNRs, and thus can only be used for some specific services. For example, due to their point-to-multipoint characteristics in the forward link, satellites provide an efficient solution for broadcast/multicast services which do not have very strict real-time constraints and are currently not supported by the terrestrial 3G [KAR04].

The S-3G component definition has been made with particular attention to the ongoing terrestrial 3G standardization activities performed in the 3GPP, in order to maximize commonality [BOU02]. From the terminal viewpoint, a commonality between terrestrial and satellite networks allows the design of a single device enabling connection to both of them. From the manufacturer point of view, commonalities between terrestrial and satellite components will allow an economy of scale.

1.5 Conclusion

This Chapter introduced third-generation communication systems and their basic theory. We have presented the different existing algorithms for the receiver detection, which are the subject of the reconfigurability concept. Finally, a brief description of the most important commercial 3G systems, which are used all along the application sections of this thesis, was also given.

Chapter 2

Algorithmic Reconfigurability: Applications

2.1 Introduction

This Chapter presents some novel algorithms which have been developed during this thesis under the framework of the French RNRT project **ASTURIES**. The proposed algorithms are based on three main functions which have computational similarities and can be calculated iteratively. These functions are the RAKE demodulation, the pilot channel estimation and the Interference Cancellation (IC). Their combination for different environments and computational constraints gave some novel algorithms for the reception operation in DS-CDMA downlink systems. All the proposed algorithms use reconfigurability to jointly optimize performance and computational power, and they have significant improvements over the conventional algorithmic scheme of the literature. In order to achieve this target, the proposed algorithms fit the number of iterations of each principal function to the respective operational parameters. Thus the number of RAKE fingers is always equal to the number of channel paths, the number of IC suppressions is equal to the interference terms under consideration, and so on.

The first proposed detector is a combination of a RAKE receiver and a pilot channel estimator. In contrast with the classical schemes, this detector can dynamically change the number of RAKE fingers and the size of the channel estimation window according to the channel behavior. The second proposed detector is a combination of a RAKE and a single-stage IC. Also in contrast with the classical static schemes, this detector can dynamically divide its constrained computational resources between the RAKE and the IC, in order to achieve the best possible performance. In order to study the performance of this algorithm when there are not strict computational constraints, the third algorithm supposes a RAKE receiver with a multi-stage IC scheme. In this case the achieved performance is similar to the conventional linear equalizers but with a lower complexity. The fourth algorithm involves the problem of channel estimation and the previous multi-stage IC scheme is generalized in order to suppress also the interference in the pilot channel. The fifth algorithm tries to further generalize the problem and thus considers also the MAI. The proposed receiver suppresses IPI and MAI in a multi-stage fashion with lower computational requirements than the conventional PIC schemes. Finally, differently from the first five algorithms, which regard reconfigurability from the receiver point of view, the

last application example regards reconfigurability as a global design characteristic. Here, reconfigurability involves changes not only in the receiver but also in the transmitter. Particularly, the last application example concerns the two transmission modes of the CDMA2000 standard and introduces a reconfigurable radio which can dynamically change its mode according to the system parameters. This change has an impact on the receiver and the transmitter. Table 2.1 summarizes the algorithm developed in this Chapter.

Algorithm	Operational Environment
RAKE+Channel Estimation	Multipath+Pilot+Single user
RAKE+Single-stage IPI-IC	Multipath+Single user+Low SF
RAKE+Multi-stage IC	Multipath+Single user+Low SF
RAKE+Multi-stage IC+Channel estimation	Multipath+Pilot+Single user+Low SF
RAKE+Multi-stage IPI-IC+Multi-stage MAI-IC	Multipath+Multi-user+Low SF
Mode 1X+3X	Satellite+CDMA2000

Table 2.1: The developed algorithms of this Chapter.

This Chapter is organized as follows: Section 2.2 presents the adaptive RAKE receiver with a variable number of fingers and window-based channel estimation. Section 2.3 introduces the necessity for equalization techniques in high data rate DS/CDMA communication links. Section 2.4 presents the reconfigurable single-stage IC scheme for high data rates. The multi-stage IC for high data rates is presented in Section 2.5. The application of the multi-stage IC in the pilot channel is described in Section 2.6, and the optimized multi-stage IC for multi-user detection, in Section 2.7. The reconfigurable CDMA2000 transceiver is introduced in Section 2.8, followed by concluding remarks in Section 2.9.

2.2 An adaptive RAKE receiver with variable number of fingers and window-based channel estimation

2.2.1 Algorithm motivation

The propagation channel is an external system parameter which changes dynamically. In our first application example, we do not consider changes in its statistical behavior and we study only its quantifiable changes. More specifically, we focus on the number of paths and the coherence time of the channel [PRO95]. The number of paths is not constant and depends on the physical environment. Thus, it can range from one path, considering a LOS environment without physical obstacles, to N paths for a multi-path out/in door environment. Traditional RAKE approaches deal with the worst operational case. They suppose a constant number of fingers, M , to be implemented in the system ($M = 3$ fingers in IS-95), a number which is efficient to face off the majority of the operational cases. However, this approach has a poor performance for the case of a channel with $L > M$ paths where the last $L - M$ paths have an important power, due to diversity components loss. In this case, the selective RAKE captures only the best subset of

multipaths out of all resolvable paths, according to their SNRs. On the other hand, in the case of a channel with $L < M$ paths, $L - M$ RAKE fingers have to be switched off. This finger deactivation corresponds to a useless consumption of the available computational power.

Moreover, the coherence time of a channel, which is defined as the maximum time interval where the channel can be considered as constant, is also a variable parameter and depends on the receiver velocity. Traditional channel estimation approaches, which are based on a pilot sequence, use a constant moving average window in order to average out the noise from the estimated channel coefficients. For the case of a window whose size is outside of the allowable space, which depends on the coherence time of the channel, a channel estimation error or a poor estimation performance are produced.

A RAKE receiver and a window-based channel estimator which can dynamically adapt the number of fingers and the window size, respectively, to the propagation environment, can jointly optimize performance and computational power. This detection scheme is presented and analyzed in the following subsections.

2.2.2 Problem formulation

In this subsection we describe the baseband downlink model of a CDMA communication system in order to formulate the problem under consideration. The downlink model is based on a single user DS-CDMA system over a multipath Rayleigh fading channel with pilot-aided channel estimation. The transmission process can be expressed as follows:

$$s(t) = A_0 c_0(t) + A_1 b(t) c_1(t), \quad (2.1)$$

$$b(t) = \sum_{k=-\infty}^{\infty} b(k) p_t(kT_b, (k+1)T_b), \quad (2.2)$$

$$c_m(t) = \sum_{k=-\infty}^{\infty} c_m(k) p_t(kT_c, (k+1)T_c), \quad (2.3)$$

where $s(t)$ is the transmitted signal, $b(t)$ denotes the QPSK data signal, $c_m(t)$ denotes the signature sequence signals for the pilot ($m = 0$) and the data signal ($m = 1$), respectively, $p_t(t_1, t_2)$ is a unit rectangular pulse on $[t_1, t_2]$, $b(k) \in \{\pm 1 \pm 1j\}$ with equal probabilities and $c_m(k) \in \{\pm 1\}$. A_0 and A_1 are the transmitted amplitudes of the pilot and the user signal, respectively. T_b and T_c are the symbol period and the chip period, and $T_b/T_c = N$, where N is the spreading factor. We note that no data symbols are present in the pilot channel.

A frequency-selective Rayleigh fading channel with L resolvable paths is modeled as

$$\begin{aligned} h(t, \tau) &= \sum_{l=1}^L h_l(t) \delta(t - \tau_l) \\ &= \sum_{l=1}^L \alpha_l(t) e^{-j2\pi f(t) n T_b} \delta(t - \tau_l), \end{aligned} \quad (2.4)$$

where f is the Doppler frequency and α_l the power envelop of the l -th channel path. f and α_l

are time-variant values with the probability density function $P_f(f)$ and $P_\alpha(\alpha_l)$ respectively as follows

$$P_f(f) = \begin{cases} \frac{1}{\pi\sqrt{f_D^2 - f^2}} & , \text{if } -f_D \leq f \leq f_D \\ 0 & , \text{if } f < -f_D, f > f_D \end{cases}, \quad (2.5)$$

$$f_D = \frac{\dot{V} f_0}{c}, \quad (2.6)$$

$$P_{\alpha_l}(\alpha_l) = \begin{cases} \frac{\alpha_l}{\sigma_l^2} e^{-\frac{\alpha_l^2}{2\sigma_l^2}} & , \text{if } \alpha_l \geq 0, \\ 0 & , \text{if } \alpha_l < 0 \end{cases} \quad (2.7)$$

where f_D is the maximum Doppler frequency, which according to Eq. 2.6 is a function of the receiver velocity (\dot{V}), f_0 is the carrier frequency of the transmitted signal, $c = 3 \cdot 10^8 m/sec$ is the speed of light and $2\sigma_l^2$ is the variance of the l -th Rayleigh path. The reciprocal of the maximum Doppler frequency denotes the coherence time of the channel ($\Delta t_c = 1/f_D$). It is obvious that the coherence time of the channel, and thus the time interval where the channel can be considered constant, is a function of the receiver velocity (we suppose that the transmitter is stationary).

Then the received signal can be written as

$$r(t) = \sum_{l=1}^L h_l(t) [A_0 c_0(t - \tau_l) + A_1 b(t - \tau_l) c_1(t - \tau_l)] + n(t), \quad (2.8)$$

where $n(t)$ is white Gaussian noise with double-sided power spectral density $N_0/2$.

The RAKE coherence detection requires an estimation of the channel coefficients. This channel estimation is produced by a Data-Aided Channel Estimation (DA-CHEST) algorithm with a moving average window [MAM02], [QAR01], which has been introduced in Chapter 1. We consider a perfect time acquisition of the channel delays and we deal only with the power estimation [BOU01]. Thus, the received signal is passed through a TDL according to channel delays, and the delayed versions of the signal are despread by a pilot Walsh sequence. The result of this operation is an initial estimation of every path of the channel. In order to improve this estimation, the estimated coefficients are passed through a bank of low pass filters (moving average window) with size N_p , to average out the AWGN. Thus, the estimated complex channel fading coefficient for the j -th path and the k -th transmitted symbol is given by

$$\begin{aligned} \hat{h}_j(k) &= \frac{1}{NN_p A_0} \sum_{m=-\frac{N_p}{2}}^{\frac{N_p}{2}} \int_{(k+m-1)T_b + \tau_j}^{(k+m)T_b + \tau_j} r(t) c_0^*(t - \tau_j) dt \\ &= h_j(k) + S_j(k), \end{aligned} \quad (2.9)$$

where $h_j(k)$ is the channel coefficient (envelop and phase) for the j -th path and the k -th transmitted symbol, $\hat{h}_j(k)$ is its estimation, N_p is the window size for the channel estimation algorithm in terms of symbols, and $S_j(k)$ denotes the estimation error. This error is a function of the channel conditions but also of the parameter N_p

$$S_j(k) = f(h(t, \tau), A_0, E_b/N_0, N_p) \quad (2.10)$$

Demodulation is similar to the channel estimation operation, except despreading is done with the data spreading code. The RAKE receiver is the conventional demodulation method. Assuming MRC with M available demodulation fingers ($M \leq L$), the decision variable of the k -th data symbol is given by

$$\begin{aligned} \hat{b}(k) &= \frac{1}{NA_1} \sum_{m=1}^M \hat{h}_m^*(k) \int_{(k-1)T_b+\tau_m}^{kT_b+\tau_m} r(t) c_1^*(t - \tau_m) dt \\ &= D(k) + I(k), \end{aligned} \quad (2.11)$$

where $D(k)$ is the produced diversity signal component and $I(k)$ is the total interference, which is a function of the channel conditions, the channel estimation error and the number of available RAKE fingers. This logic dependence can be formulated as

$$I(k) = f(h(t, \tau), A_1, E_b/N_0, S(k), M) \quad (2.12)$$

From the above equation set, we can see that the quality of the channel estimation using pilot symbols is a function of the shifting window. For the case where N_p has a value outside a value set which is defined by the coherence time of the channel, either an estimation error or a poor estimation is produced. This set is a function of the coherence time of the channel and of the SNR, and can be represented as $\Delta N_p = [N_{pmin} \ N_{pmax}]$, where N_{pmin} and N_{pmax} are the minimum and the maximum window size, respectively, which can yield the optimal estimation quality and $\Delta N_p = f(\Delta t_c, E_b/N_0)$. The channel estimation performance has also an influence over the RAKE performance, which uses the estimated channel coefficients in the MRC operation. Finally, the number of RAKE fingers is an important parameter of its performance and computational power. For a RAKE receiver with a number of fingers lower than the number of channel paths, a poor performance is achieved. Moreover, for a RAKE receiver with a number of fingers higher than the number of channel paths, an aimless use of the available computational power results.

2.2.3 The proposed receiver

The proposed adaptive detection scheme jointly optimizes performance and computational power. It consists of a conventional RAKE receiver with the ability to change its number of fingers in real-time, and a data-aided channel estimation, which can dynamically change the length of the moving window. Fig. 2.1 shows the proposed detection scheme. The demanding optimization is the use of the minimum possible values for the fingers and the length of the moving average window, in order to minimize the computational requirements and achieve the best possible performance. This optimization can be formulated by the parameter set (M_{opt}, N_{pmin}) , where $M_{opt} \leq L$ is the minimum number of fingers which gives the optimal performance.

In general, the power consumption of an algorithm is a function of its arithmetic operations and particularly of the number of multiplications. The RAKE, the considered algorithm of this Section, is a bank of parallel correlators. Each finger (correlator) effectuates a number of multiplications and thus consumes an important part of the total power. As far as the estimation

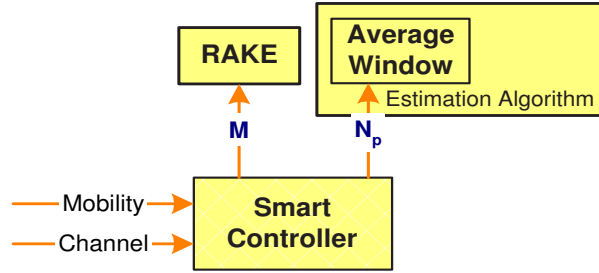


Figure 2.1: The Proposed Receiver- Smart Controller.

algorithm is concerned, it uses a moving average window and it is based on a succession of additions. Thus, the length of the average window is a less critical parameter than the number of fingers, for the power consumption of the system. If we consider a RAKE receiver with M fingers and an estimation window equal to N_p , the total number of operations per bit, for a system with a spreading factor equal to N , is presented analytically in Table 2.2.

Operation	\times (Real Multiplications)	$+$ (Additions)	\div (Divisions)
Despreading	$2MN$	-	-
Descrambling	$4MN$	$2MN$	-
Integration	-	$2M(N - 1)$	-
Weighting	$4M$	$2M$	-
MRC	-	$2(M - 1)$	-
Estimation Window	-	$2M(N_p - 1)$	M

Table 2.2: Number of operations per bit.

From Table 2.2, we can see that the number of operations, and thus the power consumption of the system, is a function of the parameters M , N_p and N . The spreading factor N is a transmitted parameter and thus the receiver can not change it. However, the parameters M and N_p are local parameters and can be changed.

The proposed receiver adapts in real-time the above parameter set in order to use the minimum possible values which give the optimal performance (M_{opt}, N_{pmin}). In the case where there are two candidate parameter sets $(M_1, N_{p1}), (M_2, N_{p2})$, where $M_1 > M_2$ and $N_{p1} < N_{p2}$, which give the same optimal performance, the set with the minimum number of fingers is the appropriate one, because M is the most critical parameter for the power consumption. In order to design this receiver, we propose a supervisor (SPV) which measures in real-time the SNR [BEA00], [SHI01] and the mobility (Doppler) [MOT99] and decides the appropriate parameter combination (M, N_p) which gives the optimal performance minimizing the power consumption.

With the simulation model presented in the next paragraph, we study the tradeoffs of different parameter settings. Simulation results can be used in order to construct heuristics to decide what to do in which case. The final goal is to construct a SPV which, by combining real-time

measurements of the current SNR and mobility with these heuristics, can make the appropriate decision at run-time. More specifically, an heuristic SPV can be viewed as a look-up table (LUT) which maps the appropriate configuration (M, N_p) to each real-time condition. This approach is easy and fast, but the construction of this table requires a high number of simulations and studies, and thus it is limited to the most probable operational cases.

2.2.4 Simulation environment

The following simulation study deals with CDMA2000 1X, which is an evolution of the second generation American standard IS-95. We consider the forward link of an isolated cell and we study the performance of one Forward Supplemental Channel (F-SCH) with data rate 153.6 kbps using the Radio Configuration 4 (RC-4) for a single-user environment. In the simulation, we assume that only the pilot and traffic channels are simultaneously transmitted, ignoring all other physical channels for the sake of simplicity.

Fig. 2.2 shows a block diagram of the transmitter chain. For the F-SCH channel, the binary generator generates 3072 equi-probable bits for a 20 ms frame with data rate 153.6 kbps. According to the RC-4, a convolutional code (CC) is used for this rate with constraint length $K = 9$ and code rate $R = 1/2$.

The output of the CC is interleaved by a block interleaver, whose even output symbol position (i is even) is given by:

$$A_i = 2^\nu \left[\frac{i}{2} \bmod J \right] + BRO_\nu(\lfloor \frac{i}{2} / J \rfloor), \quad (2.13)$$

and odd output symbol position (i is odd) is given by:

$$A_i = 2^\nu \left[\left(K - \frac{i+1}{2} \right) \bmod J \right] + BRO_\nu(\lfloor (K - \frac{i+1}{2}) / J \rfloor), \quad (2.14)$$

where K is the block size, $i = 0, \dots, K-1$, $\lfloor x \rfloor$ indicates the largest integer less than or equal to x , $BRO_\nu(y)$ indicates the bit-reversed ν -bit value of y , ν and J are the interleaver parameters.

QPSK is used as modulation scheme and the I-, Q- branches are spread with the same orthogonal Walsh Code $c_{8,i}$, where $1 \leq i \leq 8$, and transformed into a number of chips. The next operation is the complex scrambling, where the resultant signal is multiplied by a complex valued PN sequence. This sequence is taken from two independent M -sequences with period 2^{15} and rate 1.2288 Mcps.

The Forward Pilot Channel (F-PICH) is continuously broadcast throughout the cell in order to provide timing and phase information and is common for every user of the cell. It is unmodulated and contains only 0 bits, which is equivalent to constant amplitude. For the separation of the pilot channel from the F-SCH, the pilot sequence is spread by the orthogonal Walsh code $c_{64,1}$ and in the next step is complex scrambled as the traffic channel.

The power level of the pilot channel is a very important aspect because the system capacity and the performance of the estimation algorithm are a function of it [SHI02]. In this study, we consider a pilot power as 20% of the total transmitted power, according to the TIA/EIA/98. To simplify our simulation, perfect closed-loop power control is considered. The final signal (pilot+traffic channel) is impulse-shaped with a root raised cosine filter with rolloff factor 0.22. Finally both branches will be modulated by a carrier frequency, f_0 , of approximately 2 GHz.

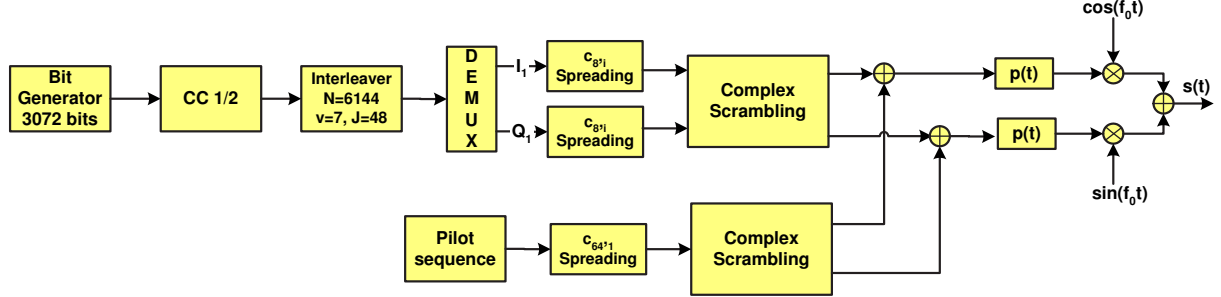


Figure 2.2: The transmitter chain.

Delay (ns)	Average Power (dB)
0	0
310	-1
710	-9
1090	-10
1730	-15
2510	-20

Table 2.3: The ETSI channel model.

Delay (chips)	Average Power (dB)
0	2.83
2	-10
3	-15
4	-20

Table 2.4: Our equivalent channel model.

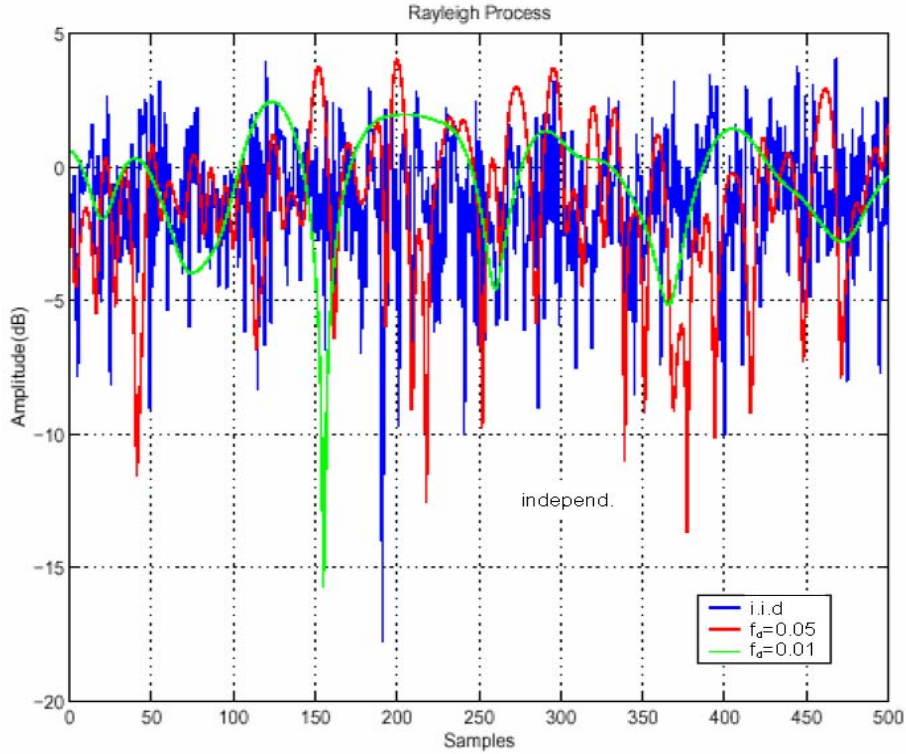


Figure 2.3: The power envelope for different correlations.

The propagation channel which is used in our study is a frequency selective channel ETSI [ETSI98]. This channel is characterized by the existence of multipath components and it can be represented with a TDL. The proposed model has six multipath components, so six taps with fixed delays. Every tap follows the Rayleigh distribution. This channel model is classic for a simulation system of the 3G European standard (UMTS) [HOL02]. In this standard, the chip rate is 3.84 Mcps, thus the six paths correspond to six different chip periods. In the CDMA2000, the chip rate is smaller, so the first three paths correspond to the same chip. In order to simplify our simulation system and to avoid an oversampling process for separating paths which arrive in the same chip period, we assume that the first three paths are equivalent to one path with average power equal to the sum of them. Tables 2.3 and 2.4 show the tap delay model proposed in the ETSI specifications and our equivalent model for the CDMA2000 standard, respectively. Moreover, Fig. 2.3 presents the envelope of one channel path for different values of the Normalized Doppler spectrum ($f_d = f_D/T_b$). In this figure, we observe that as f_d decreases (thus the speed decreases), the process becomes more correlated. According to the channel models, we either generate six independent processes, one for each tap of the TDL 1, or four processes for the case of the TDL 2.

Fig. 2.4 shows the receiver chain for the considered simulation environment. It consists of the inverse operations in order to detect the transmission signal. Thus a conventional RAKE receiver and a data-aided channel estimation with a moving window are supposed for the demodulation. A classic Viterbi algorithm with soft decisions is supposed for the decoding process. Table 2.5 summarizes the simulation parameters assumed in this application.

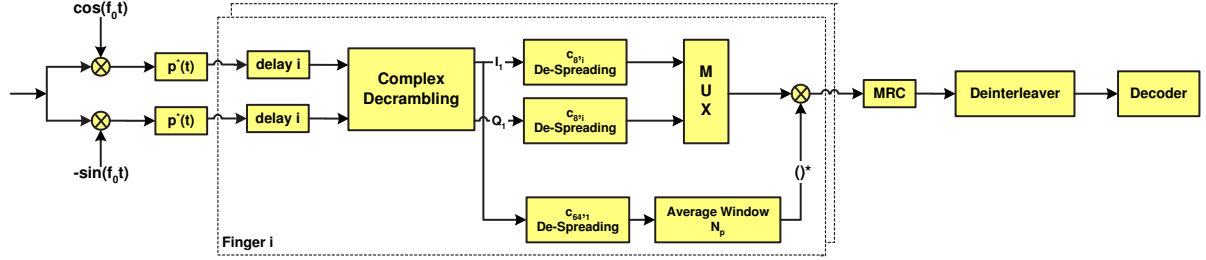


Figure 2.4: The receiver chain.

Chip Rate	1.2288 Mcps
Data Rate	153.6 kbps
FEC	Convolutional Code ($R = 1/2$, $K = 9$)
Interleaver	$K = 6144$, $\nu = 7$, $J = 48$
Spreading Factor	$N_{traffic} = 8$, $N_{pilot} = 64$
Pilot Power	20%
Channel	$\mathbf{h} = [2.83 \ -10 \ -15 \ -20]\text{dB}$ $\tau = [0 \ 2 \ 3 \ 4]T_c$
Velocity	3 Km/h, 50 Km/h
Detection	RAKE
Channel Estimation	Data-aided with moving average window
Decoding	Viterbi with soft decision

Table 2.5: Simulation Parameters.

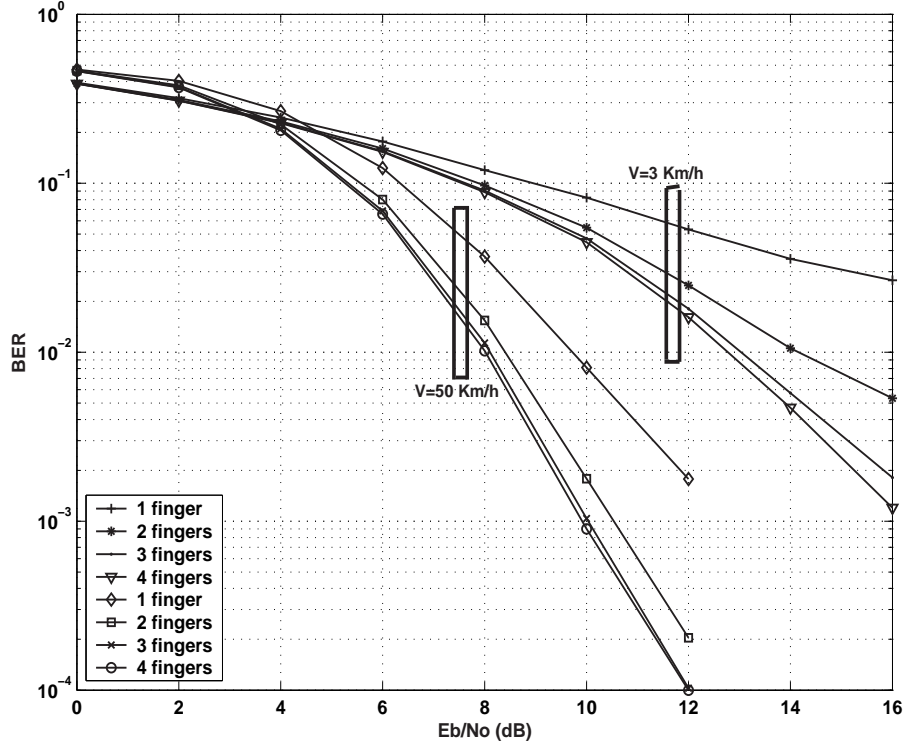


Figure 2.5: RAKE receiver with different numbers of fingers for speeds equal to 3 Km/h and 50 Km/h, and perfect channel estimation.

2.2.5 Simulation results

Computer simulations were carried out in order to show that the real-time adaptivity of the two receiver parameters under consideration can improve the performance and the power consumption of the system. Fig. 2.5 shows the performance by means of bit error rate (BER) of a conventional RAKE receiver with perfect channel estimation for the two considered speeds. This figure is used as a reference for the imperfect channel estimation results presented next.

Mobile speed equal to 3 Km/h

The first considered speed ($\dot{V} = 3$ Km/h) corresponds to a pedestrian user. In this case the propagation channel has a big coherence time and is very correlated. Therefore, the CC 1/2, which is designed for an independent gaussian channel, does not have a good performance.

We studied the optimal parameter combination for two characteristic E_b/N_0 levels. Fig.'s 2.6 and 2.7 present the performance (BER) for different lengths of the moving average window and numbers of RAKE fingers, for a E_b/N_0 equal to 5dB and 10dB, respectively. According to our previous discussion, the proposed detection scheme achieves the optimal performance using the minimum possible values of the parameters M , N_p . Thus, the appropriate combination is ($M_{opt} = 2, N_{pmin} = 101$ symbols) for $E_b/N_0 = 5$ dB and ($M_{opt} = 3, N_{pmin} = 51$ symbols) for $E_b/N_0 = 10$ dB.

The first important conclusion is that the minimum number of fingers and estimation window length necessary depend on the level of the SNR. As E_b/N_0 is increased, the addition of one more

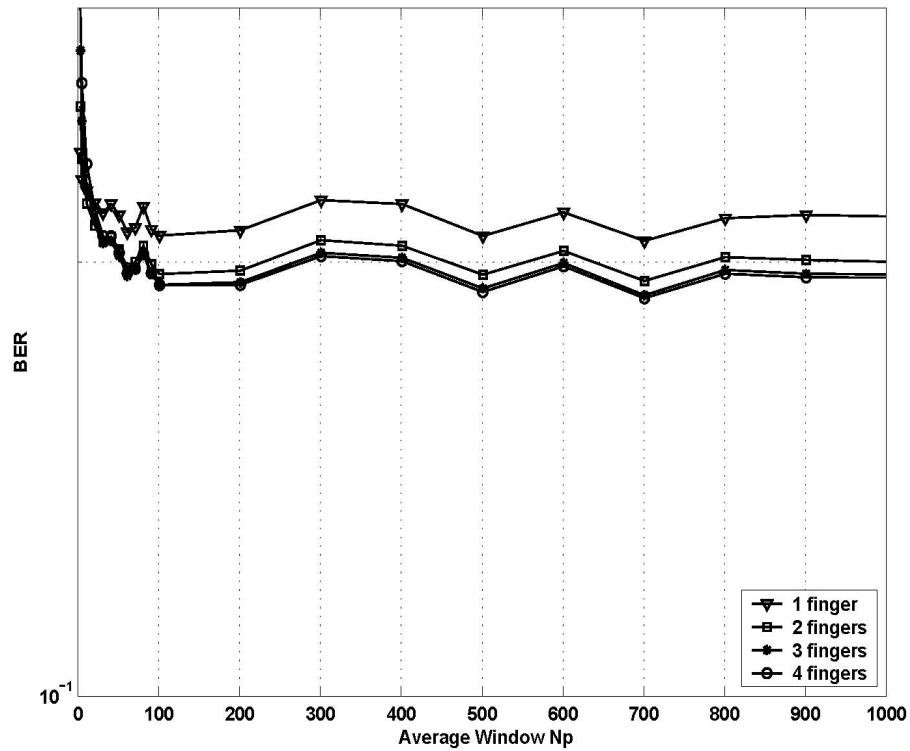


Figure 2.6: RAKE receiver with different numbers of fingers for a speed equal to 3 Km/h, $E_b/N_0 = 5dB$ and imperfect estimation.

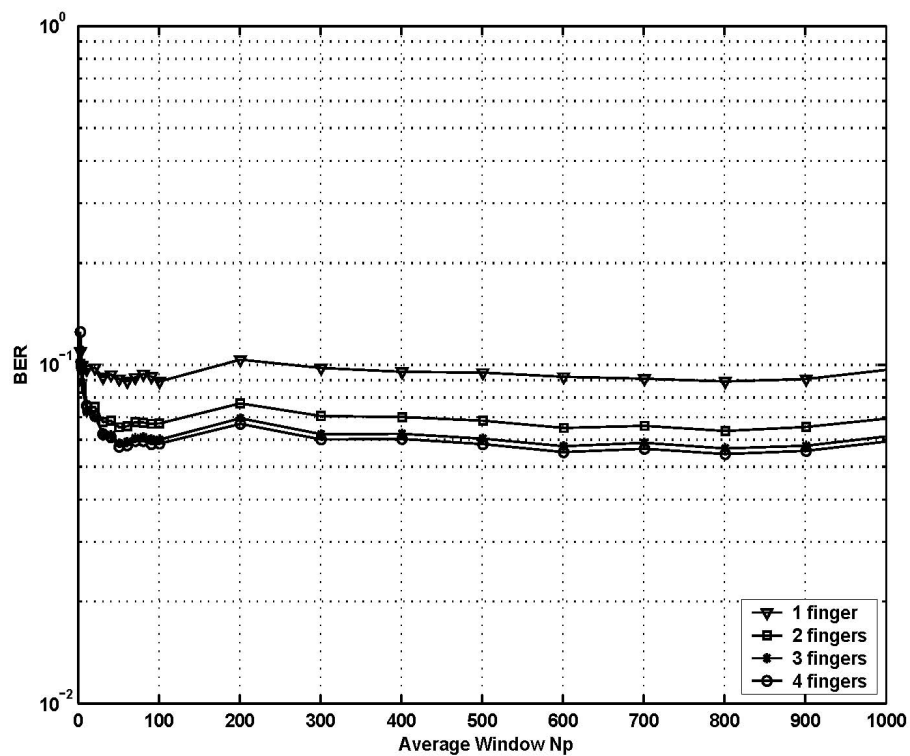


Figure 2.7: RAKE receiver with different numbers of fingers for a speed equal to 3 Km/h, $E_b/N_0 = 10dB$ and imperfect estimation.

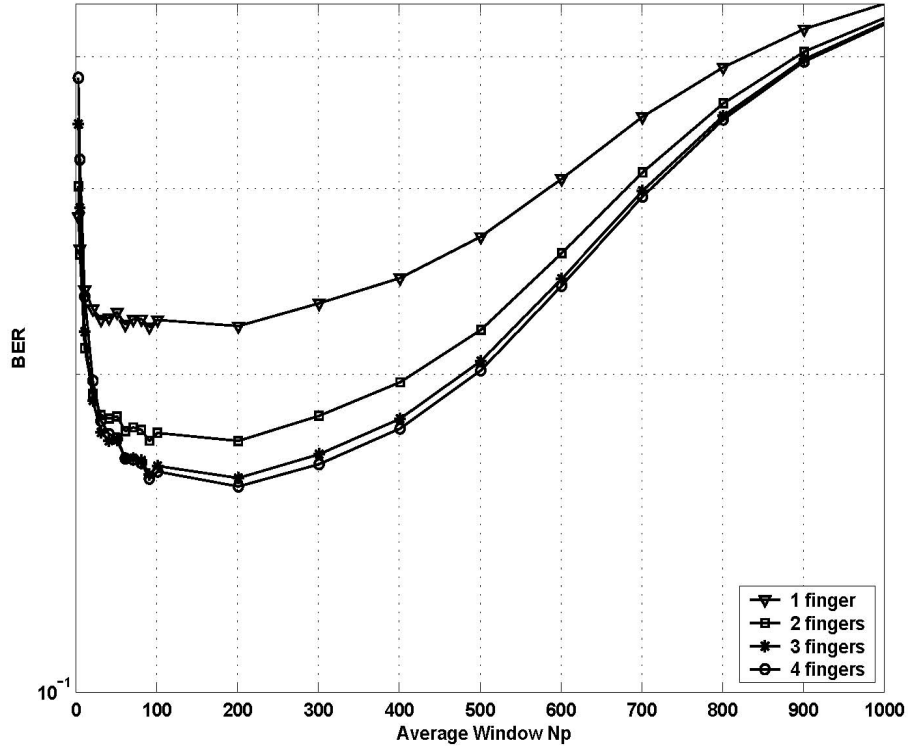


Figure 2.8: RAKE receiver with different numbers of fingers for a speed equal to 50 Km/h, $E_b/N_0 = 5dB$ and imperfect estimation.

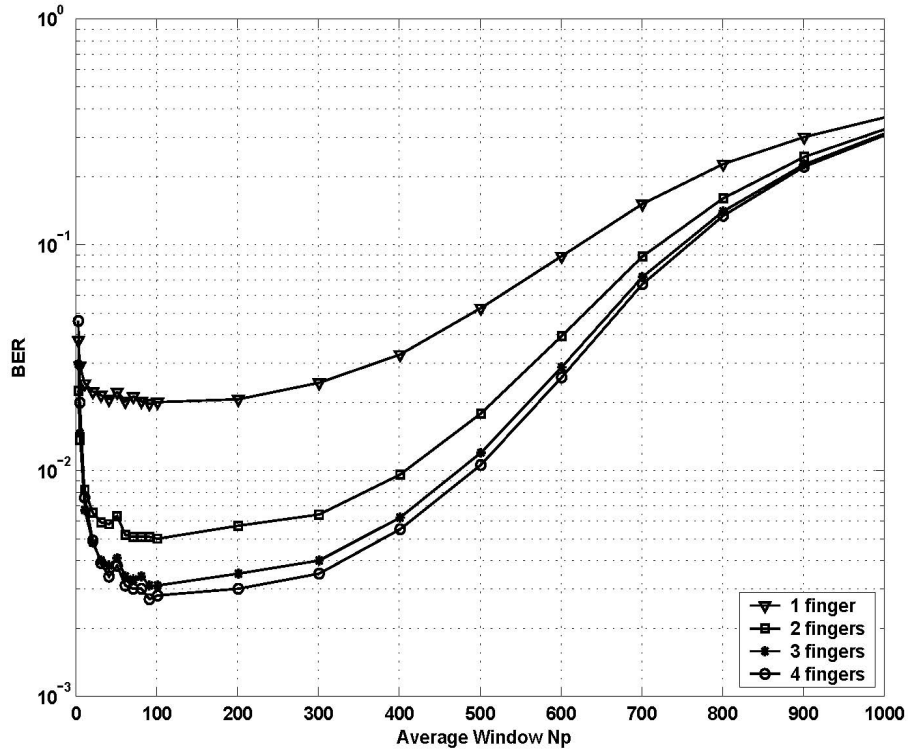


Figure 2.9: RAKE receiver with different numbers of fingers for a speed equal to 50 Km/h, $E_b/N_0 = 10dB$ and imperfect estimation.

Channel Model	E_b/N_0 (dB)	\dot{V} (Km/h)	M (fingers)	N_p (symbols)
...
Model ETSI	5 dB	3 Km/h	2	101
Model ETSI	10 dB	3 Km/h	3	51
Model ETSI	5 dB	50 Km/h	2	201
Model ETSI	10 dB	50 Km/h	3	101
...

Table 2.6: The structure of an heuristic SPV, which is used by the adaptive detector with variable number of finger and window-based channel estimation.

path in the RAKE combination can improve performance. For the ideal length of the estimation window, we can say that as E_b/N_0 is increased, its value is reduced. This window is used to average out the white Gaussian noise. For a high SNR, this type of degradation is lower and thus the length of the average window can be reduced. Moreover, we can see that as the estimation window is increased, the BER rests about the same. For this speed the coherence time of the channel is almost equal to the frame duration, thus the increase of the estimation window has no influence on the system performance ($N_{pmax} \geq \Delta_{frame}$), where Δ_{frame} is the frame duration.

Mobile speed equal to 50 Km/h

As can be seen from the reference figure with the perfect channel estimation, the increase of speed improved the performance of the RAKE receiver. This is expected, because for high speeds, the channel is less correlated, so the interleaver randomizes the received symbols in a better way, which enables the CC 1/2 to have a better performance.

Fig.'s 2.8 and 2.9 show the performance of the RAKE receiver with imperfect channel estimation for the two considered SNR's. From these curves it emerges that the appropriate parameter combination for our adaptive low-power receiver is ($M_{opt} = 2, N_{pmin} = 201$ symbols) and ($M_{opt} = 3, N_{pmin} = 101$ symbols) for $E_b/N_0 = 5$ dB and $E_b/N_0 = 10$ dB, respectively. The speed increase changed the values of the optimal parameter set, thus the second input of the smart controller is the measure of mobility. Our observations about the influence of the E_b/N_0 over the optimal values of the parameters M and N_p are similar to the case of $\dot{V} = 3$ Km/h. The increase of E_b/N_0 increases the appropriate number of RAKE fingers and reduces the average window length.

Moreover, the influence of the terminal velocity is more important on the optimal length of the average estimation window. According to the presented results, for a higher velocity, the length of the average window has to be increased in order to give the best possible estimation performance. However, when the estimation length overcomes the upper boundary of the allowable values ($N_p > N_{pmax}$) the BER starts to be increased. For this simulation example, the upper boundary is lower than the one of the previous case, because the speed increases and thus the corresponding coherence time is reduced.

Table 2.6 shows the structure of the LUT corresponding to the heuristic SPV. It is a two-dimension table, where each column represents an external system real parameter (channel model, E_b/N_0 , speed) or a local receiver parameter (M , N_p). SPV has the task of providing the appropriate table line which fits the table entities with the operational environment. This provided table line includes the optimal values for the adaptive parameters of interest.

2.3 High data rates and WCDMA systems

High data rate connections can be supported in WCDMA systems by employing low SF's while maintaining fixed bandwidth. If extremely low SF's, e.g. 2 or 4 are applied, it is likely that there is only one high data rate user present in the system [HOL99], [HOO99]. The other possible users employ significantly higher SF's, and thus are received at lower power. Therefore, for the high data rate user, MAI can be considered negligible. In this case, the IPI (which is caused by the multi-path components and due to the un-ideal autocorrelation properties of the spreading codes) becomes the basic reason of performance degradation for the conventional RAKE receiver. In this case, an equalization scheme is necessary in the output of the RAKE receiver, in order to improve the achieved performance. The following reconfigurable algorithmic schemes deal with the high data rate communication links and try to jointly improve performance and computational power, under different constraints and environments.

2.4 Single-stage interference cancelation for high data rates

2.4.1 Algorithm motivation

A basic goal of the reconfigurability concept is the optimal use of the available computational resources. The developed algorithmic scheme of this Section presents this important aspect.

According to our previous application example, the propagation channel is a dynamic system parameter. The number of channel paths is variable and depends on the physical operational environment. Traditional approaches use a constant number M of RAKE fingers, in order to resolve the possible multi-path effects. In this application example, we focus on the case of a channel with a number of strong paths lower than the number of available fingers ($L < M$). A channel path is characterized as strong when its participation to the RAKE combination improves the previously achieved performance. In this case, a number of fingers equal to $M - L$ has to be switched off and thus a part of the available computational power is not used. The proposed algorithm, thanks to the reconfigurability concept, tries to use the available computational resources for another operation which can improve the RAKE performance. The functional similarities between the RAKE demodulation and the suppression process, which is performed by an IC algorithm, permit the use of the available computational power for the IC process. More specifically, $M - L$ RAKE fingers can be used in order to suppress $M - L$ terms of the interference embedded in the RAKE output, and thus the available computational power is used in an optimal way.

2.4.2 Problem formulation

We consider the forward link of an isolated cell with one active user. In this environment, the basic reason for performance degradation is the IPI arising from the existence of several transmission paths. In the case of a small spreading factor, which corresponds to a high data rate, the IPI is very important and significantly reduces the system performance.

Transmitter Model

We consider a single base station, which generates a QPSK sequence at a rate of $1/T_b$ symbols per second. The data signal, $b(t)$, and the spreading signal, $c(t)$, are defined as

$$b(t) = \sum_{k=-\infty}^{\infty} b(k)p_t(kT_b, (k+1)T_b), \quad (2.15)$$

$$c(t) = \sum_{k=-\infty}^{\infty} c(k)p_t(kT_c, (k+1)T_c), \quad (2.16)$$

where $b(k)$ and $c(k)$ denote the k -th QPSK symbol and k -th chip with normalized energies, respectively; $p_t(t_1, t_2)$ being a unit rectangular pulse on $[t_1, t_2)$. In practical applications, $p_t(\cdot)$ has a bandlimited waveform, such as a raised cosine Nyquist pulse. However, for analysis and simulation simplicity, we will assume that it is a rectangular pulse. The integer $T_b/T_c = N$ is the processing gain, where T_b and T_c are the symbol and chip duration, respectively.

The base-band transmitted signal can be expressed as

$$s(t) = \sqrt{P_T}b(t)c(t), \quad (2.17)$$

where P_T is the transmitted power.

Channel Model

The propagation channel which is used in our study is a frequency-selective channel. This channel is characterized by the existence of multipath components and it can be represented using the truncated TDL structure [PRO95]. The complex low-pass equivalent impulse response of the channel is given by

$$h(t, \tau) = \sum_{l=1}^L h_l(t)\delta(t - \tau_l), \quad (2.18)$$

where L is the number of resolved paths, h_l and τ_l are, respectively, the complex fading factor and propagation delay for the l -th path. The complex path gain, $h_l(t)$, is treated as an uncorrelated complex Gaussian random variable with zero mean and variance σ_l^2 . Note that the amplitude of the channel coefficients is Rayleigh distributed. Also, without loss of generality, we suppose that $\tau_l = lT_c$ and that the channel is invariant over the symbol duration. Variations due to path loss and shadowing are assumed to be eliminated by power control.

Receiver model

The received signal, which is the sum of all the signals arriving from each path, in addition to the thermal noise, can be written as

$$r(t) = \sqrt{P_T} \sum_{l=1}^L h_l(t) b(t - \tau_l) c(t - \tau_l) + n(t), \quad (2.19)$$

where $n(t)$ is the additive white Gaussian noise (AWGN), having two-sided power spectral density $N_0/2$.

Based on the TDL model, a RAKE receiver, which is a filter matched to the multipath channel and the transmission process, can be used to detect the received signal. The tap weights of the RAKE are assumed to be estimated perfectly, to simplify derivations. For the case of QPSK, the output of a RAKE receiver for the k -th symbol, with MRC and $M - 1$ ($M \leq L$) fingers, which are locked on to the $M - 1$ strongest multipath components, can be expressed as a single decision variable

$$\begin{aligned} \hat{b}(k) &= \sum_{m=1}^{M-1} h_m^*(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} r(t) c^*(t - \tau_m) dt \\ &= D + S + W, \end{aligned} \quad (2.20)$$

where D , S and W are the desired signal component, the self-interference and the noise term, respectively. Moreover, each component of the decision random variable can be further written as [MAR98]

$$D = \sqrt{P_T} N b(k) \sum_{m=1}^{M-1} |h_m(k)|^2, \quad (2.21)$$

$$S = \sqrt{P_T} \sum_{m=1}^{M-1} \sum_{\substack{l=1 \\ l \neq m}}^L h_m^*(k) h_l(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} b(t - \tau_l) c(t - \tau_l) c^*(t - \tau_m) dt, \quad (2.22)$$

$$W = \sum_{m=1}^{M-1} h_m^*(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} n(t) c^*(t - \tau_m) dt, \quad (2.23)$$

The addition of one more finger in the RAKE combination with the $M - 1$ fingers, which is locked on to the M -th stronger channel path, changes the power of the above terms as

$$\Delta E\{DD^*\} = P_T N^2 \sigma_M^2 \left(2 \sum_{m=1}^{M-1} \sigma_m^2 + 1 \right), \quad (2.24)$$

$$\Delta E\{SS^*\} = P_T N \sigma_M^2 \sum_{\substack{m=1 \\ m \neq M}}^L \sigma_m^2, \quad (2.25)$$

$$\Delta E\{WW^*\} = \frac{NN_0}{2} \sigma_M^2, \quad (2.26)$$

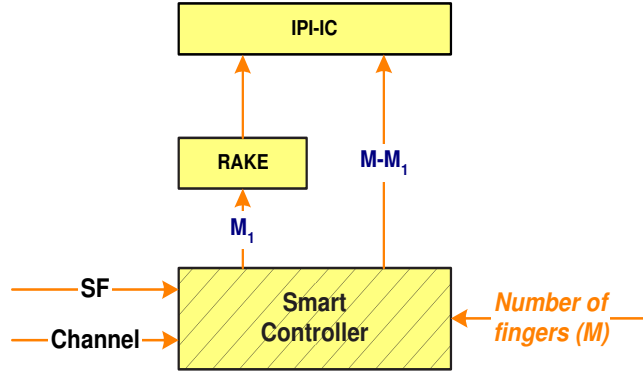


Figure 2.10: The proposed algorithm.

From the above equations, we can see that using more fingers in the RAKE combination increases not only the power of the useful component in the decision random variable, but also that of the interference and noise components. In some cases, these increases can be equivalent and thus the addition of more fingers is not appropriate. Reconfigurability makes it possible to use the computational power freed of these extra fingers for a different functionality, which can improve the system performance. In our proposition, we can set it up as a simple one-stage IC. This IC uses a hard decision on the output of the RAKE receiver to obtain an initial estimation of the transmitted signal and, based on this estimation, creates replicas of the contributions of the interference terms. Afterwards, the estimated interference terms are removed from the initial output of the RAKE receiver, and in this manner we improve the system performance. Ideally, if we can correctly estimate the terms of the interference component S and we have the appropriate number of fingers, the decision statistics will be simply equal to $D + W$.

2.4.3 The proposed receiver

In this application example, we propose a new reconfigurable receiver scheme over a frequency-selective fading channel with a single user and for a high data rate link. Its structure is a combination of the processing required for the conventional RAKE and the one-stage inter-path IC. It consists of a bank of reconfigurable fingers which can be configured either to demodulate the channel paths or to recreate replicas of the interference. Besides the classic operation of demodulation, each finger can be configured to represent a strong interference term of the interference set S , in order to be used by the one-stage IC. Fig.'s 2.10 and 2.11 show the block diagram of this receiver. The operations of a RAKE finger which demodulates a distinct multipath component and the generation process of a replica of one interference term are very close: the only difference is the input signals. Fig. 2.12 shows the two different configurations of a finger. Thus reconfigurability is possible without complicated architectural changes. We note that the first finger is not reconfigurable because the RAKE must have at least one finger.

Moreover, for the function of the IC, the output of the RAKE combination is hard-decided and in the sequel is spread to create an estimation of the transmitted signal. In general, the spreading code values are binary real numbers of (1, -1). As a result, there is no multiplication associated with correlators because they multiply the samples of the signal by either 1 or -1,

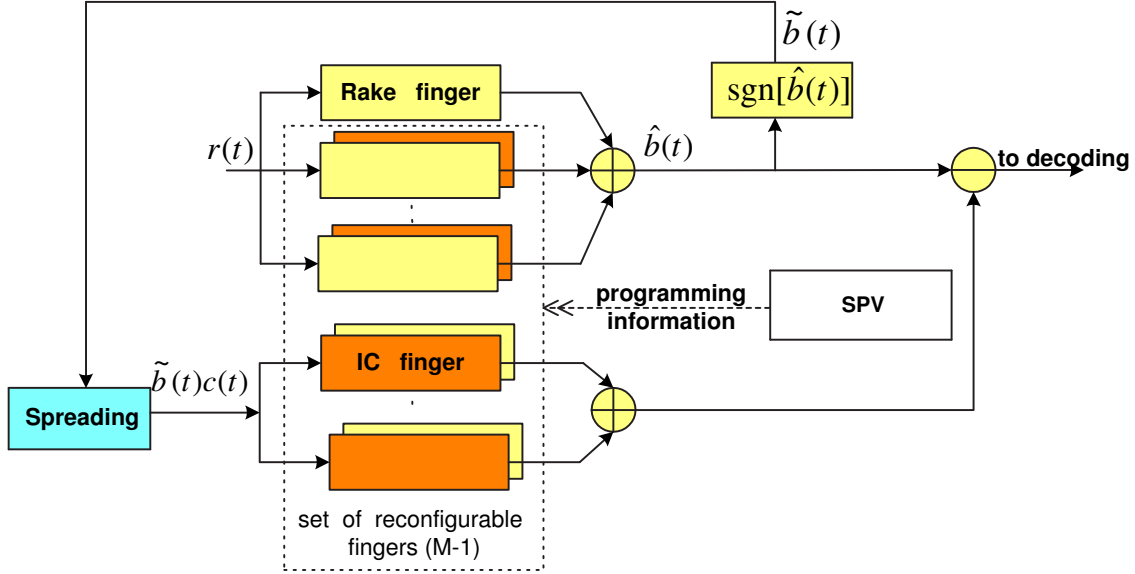


Figure 2.11: The computational part of the proposed reconfigurable detector.

which can be implemented by a control signal that switches between addition and subtraction. Thus, the proposed receiver has a complexity equivalent to that of a conventional RAKE receiver.

The heart of this receiver, as in every reconfigurable scheme, is the SPV, which decides the optimal division of the available reconfigurable fingers between the two functionalities, in real-time. In contrast with the existing controller approaches, it does not need either real-time measurements or complicated BER estimations. The input of this controller is the estimated average power of the multipath components by the channel estimation algorithm, and the output is the programming commands for the configuration of each finger. It uses a finger configuration strategy which, according to the channel estimation and via some simple algebraic comparisons, decides the configuration which optimizes the system performance. This new run-time configuration strategy is presented in subsection 2.4.4.

If we suppose that SPV has decided that M_1 fingers will be used for the RAKE combination and $M - M_1$ fingers for the IC suppression, the operations of the proposed reconfigurable detector can be formulated as

$$\hat{b}(t) = \sum_{k=1}^{\infty} \sum_{m=1}^{M_1} h_m^*(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} r(t) c^*(t - \tau_m) dt, \quad (2.27)$$

$$\tilde{b}(t) = \text{sgn}(\hat{b}(t)), \quad (2.28)$$

$$\tilde{b}'(t) = \hat{b}(t) - \hat{S}^{(M-M_1)}(\hat{b}(t)), \quad (2.29)$$

where $\text{sgn}(x)$ denotes the sign of x , and $\hat{S}^{(J)}(Y(t))$ is the regeneration of the J most important interference terms, based on the estimated transmitted signal $Y(t)$.

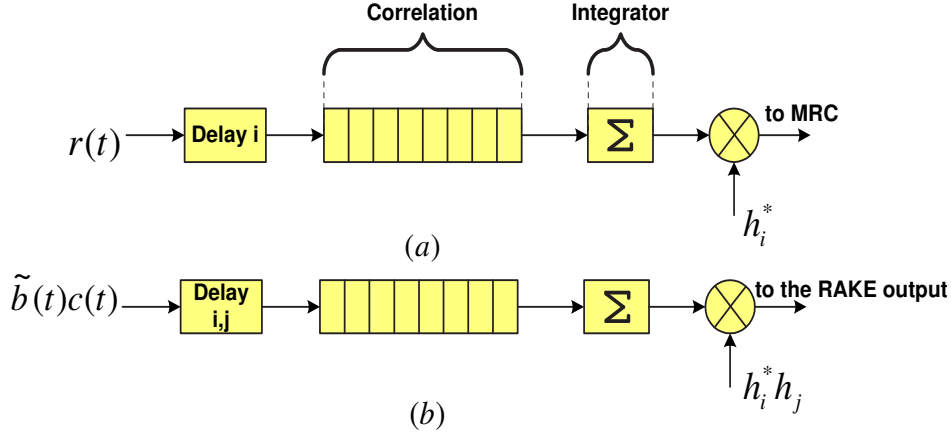


Figure 2.12: a) RAKE finger for the i -th channel path b) Interference "finger" for the cancelation of the term which arose from the i -th and j -th paths.

2.4.4 The Finger Configuration Algorithm

The finger configuration algorithm takes as inputs the average power of the distinct multipath components (\mathbf{h}) and the number of available fingers (M). Fig. 2.13 shows the block diagram of the finger configuration algorithm.

In the initial step, the first available finger which is not reconfigurable is assigned to the strongest multipath component. The powers of the interference terms which have been created from this allocation are calculated and stored as a set $\{F\}$.

$$F = \{P_T N \sigma_1^2 \sigma_2^2, \dots, P_T N \sigma_1^2 \sigma_L^2\} \quad (2.30)$$

In the next step, the algorithm calculates, by Eq.'s 2.24 and 2.25, the change in the power of the useful and interference components, by the assignment of the second available finger to the next stronger channel path. We suppose that the noise power is negligible. In this step, the reconfigurability decision variable G , which is defined as

$$G = \frac{\Delta E\{DD^*\} - \Delta E\{SS^*\}}{\max\{F\}}, \quad (2.31)$$

is calculated and compared to a threshold T_n , and a configuration is selected for the finger. If G is equal to or bigger than T_n , the considered finger is used in the RAKE combination and the interference set ($\{F\}$) is updated by the new interference terms which have been created from this new allocation. On the other hand, if G is smaller than T_n , the considered finger is used by the IC to mitigate the strongest (maximum) interference term of ($\{F\}$). The algorithm supposes that there is a perfect estimation of the transmitted signal and thus the interference term can be perfectly mitigated. This step is repeated until either the processing of all the channel paths or the configuration of all the available fingers has been completed.

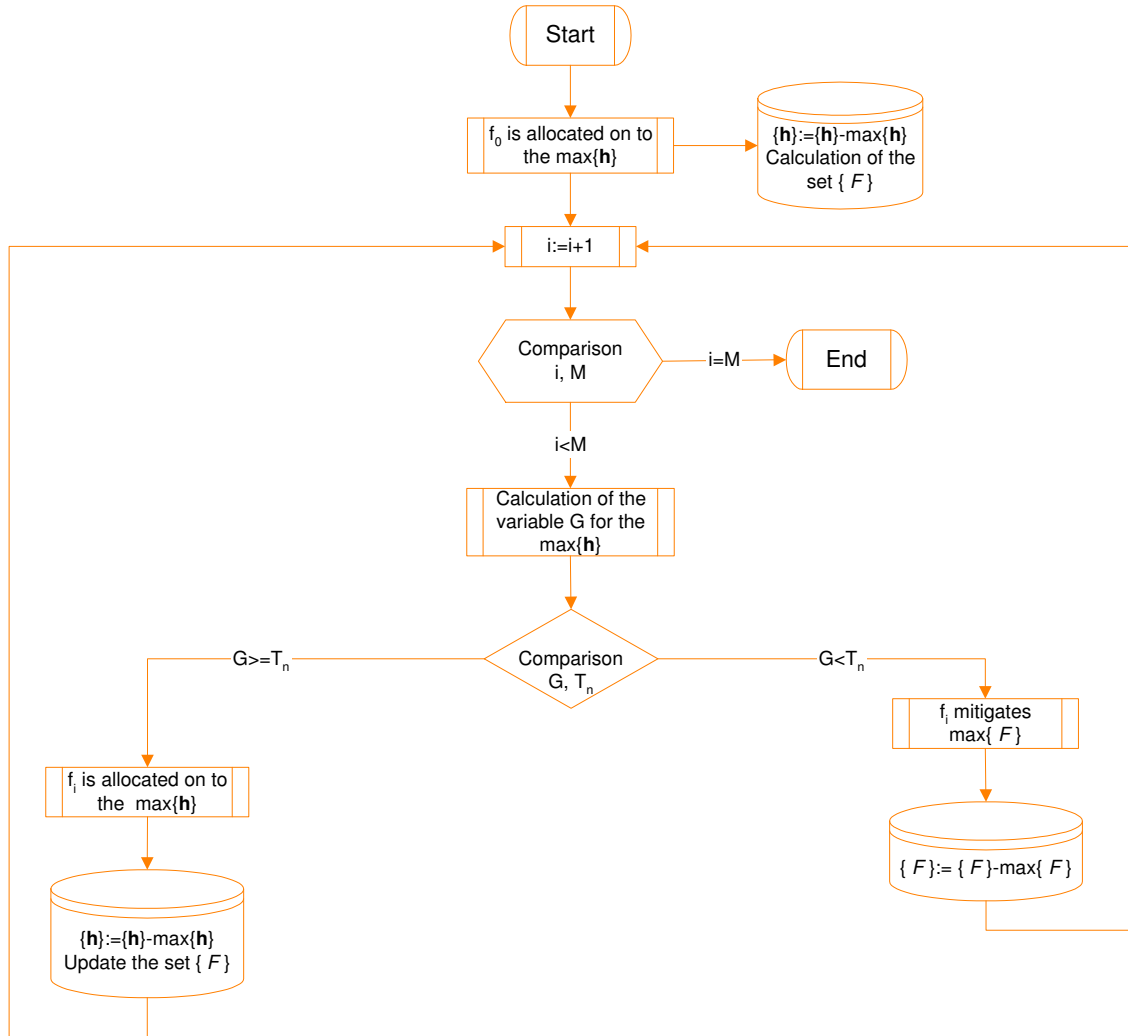


Figure 2.13: The block diagram of the finger configuration algorithm.

Modulation	QPSK
Chip Rate	3.84 Mcps
Frame length	10 ms
Spreading Codes	Walsh-Hadamard
Scrambling Codes	PN with period $2^{18} - 1$
Coding	-
Channel Estimation	Perfect

Table 2.7: Simulation parameters.

Delay (chips)	Average Power (dB)	Statistic
0	0	Rayleigh
1	-1	Rayleigh
2	-15	Rayleigh
3	-20	Rayleigh

Table 2.8: Channel Model 1

2.4.5 Numerical results

Computer simulations were carried out to evaluate the performance of the proposed reconfigurable detector and its associated finger configuration algorithm. The simulation environment is based on the UMTS-FDD specifications for the downlink case. We study the performance of the reconfigurable RAKE receiver for different channel models and computational power constraints. Table 2.7 summarizes the basic simulation parameters.

Firstly, we evaluate the performance of the proposed reconfigurable detector for an uncorrelated channel with four Rayleigh-fading paths. The channel statistics are presented in Table 2.8. The used spreading factor is $N = 4$, which for the uncoded case corresponds to a high bit rate equal to 1920 Kbps.

Fig. 2.14 shows the performance of the reconfigurable RAKE, in comparison with that of the conventional RAKE receiver, for different constraints of computational power. For a computational power equal to two fingers, a RAKE receiver with two fingers is the appropriate combination. This is to be expected, because the first two paths have a strong average power. For a computational power equal to three fingers, two fingers in the RAKE algorithm and one finger in the interference cancelation outperform the conventional RAKE with three fingers. Finally, for a computational power equal to four fingers, the use of two fingers in the RAKE algorithm and the other two fingers to mitigate two strong interference terms is the combination which gives the best possible system performance.

The efficiency of the proposed finger configuration algorithm is also evaluated by numerous computer simulations. For the following group of results, the radio channel has three paths of Rayleigh fading. The first two paths have the same average power, equal to 0dB, and the third one a variable average power ranging between $[0 - 25]$ dB. The available computational power for the proposed reconfigurable detection scheme is equal to three fingers. Thus, the first two are locked on to the strong multipath components of 0dB and the configuration algorithm is applied to the third one.

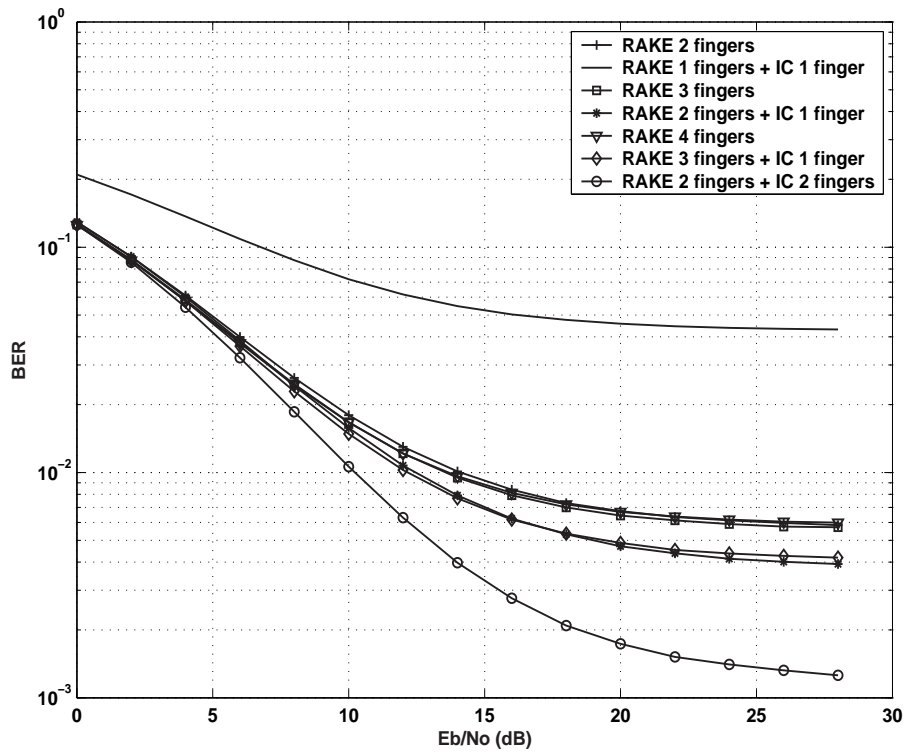


Figure 2.14: Performance comparison for a reconfigurable RAKE receiver with different computational power constraints (number of the available fingers).

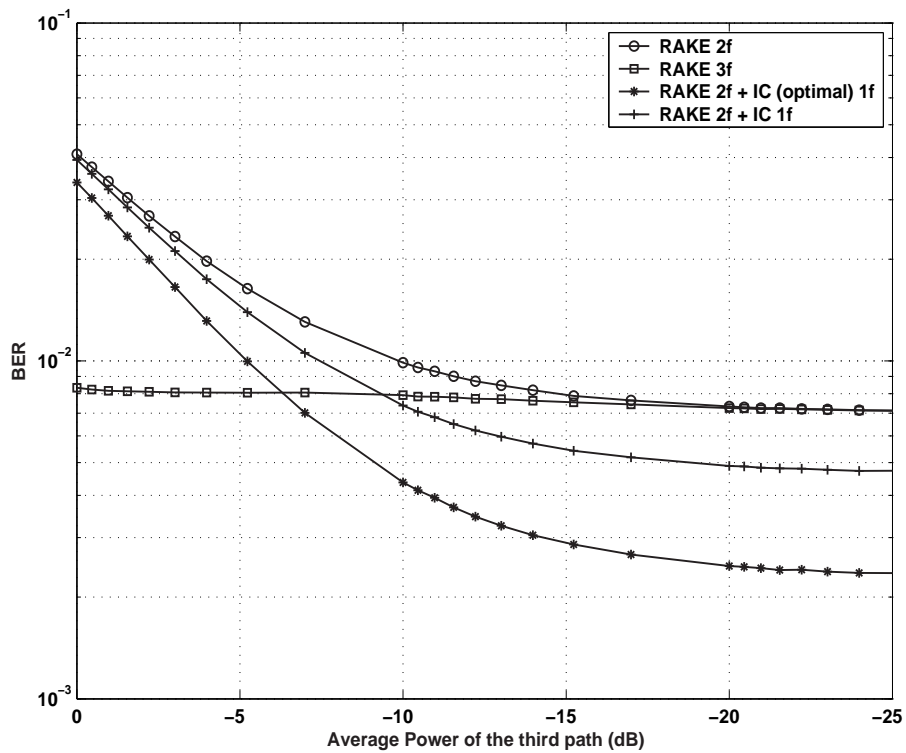


Figure 2.15: BER performance for a spreading factor equal to 4 and $E_b/N_0 = 16$ dB.

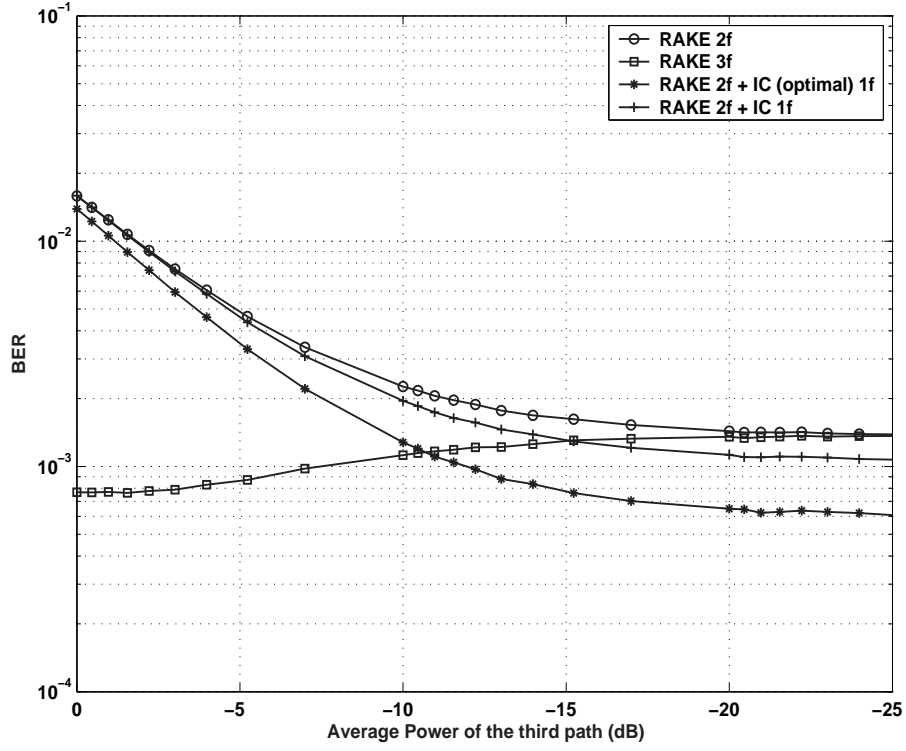


Figure 2.16: BER performance for a spreading factor equal to 8 and $E_b/N_0 = 16\text{dB}$.

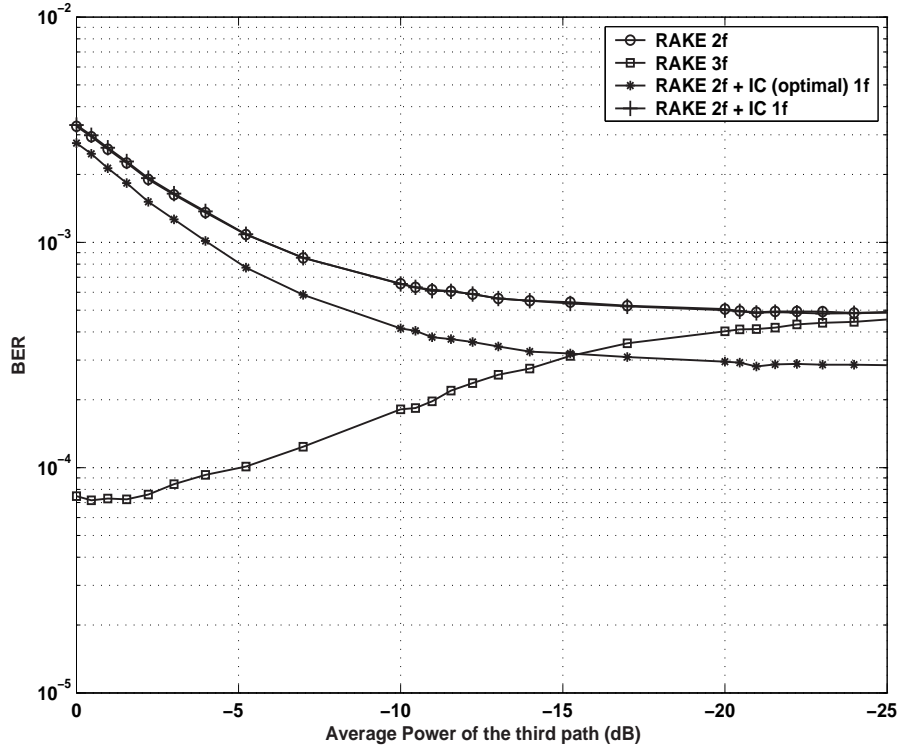
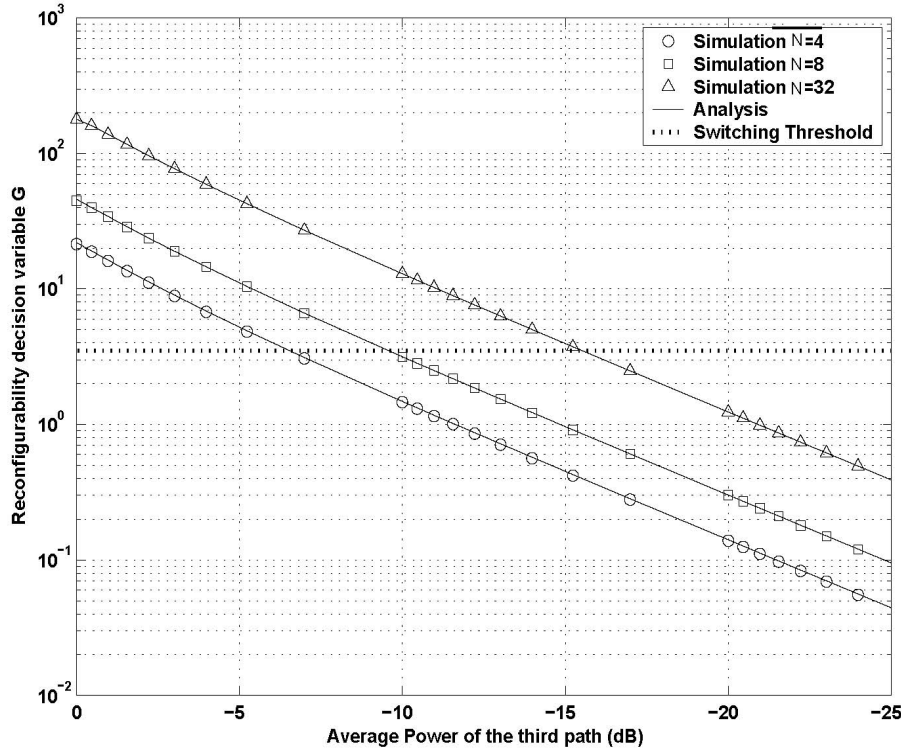


Figure 2.17: BER performance for a spreading factor equal to 32 and $E_b/N_0 = 16\text{dB}$.

Figure 2.18: Reconfigurability decision variable G .

In Fig.'s 2.15, 2.16 and 2.17 we present the system performance by means of BER for the two possible configurations of the third finger and for a spreading factor equal to 4, 8 and 32, respectively. For comparison purposes, simulation results for a conventional RAKE receiver with two fingers, and a receiver with two RAKE fingers and one IC finger using a perfect estimation of the transmitted signal, are also given. As can be seen from the graphs, there is a discrepancy between the optimal and the real IC performances. The basic reason for this difference is that the output of the RAKE combination is an erroneous estimation of the transmitted signal. However, as the spreading factor increases, the difference between them is reduced. This is to be expected because a larger spreading factor, for the same level of E_b/N_0 , improves the system performance (BER) and thus the output of the RAKE combination is a better estimation of the transmitted signal. Moreover, we can say that as the spreading factor increases, the gain from using the IC is reduced. The increase of the spreading factor reduces the relative IPI power and thus the IC scheme becomes less efficient. If we consider as the reconfigurability switching point, the intersection of the curves which correspond to a receiver with three RAKE fingers and a receiver with two RAKE fingers and one optimal IC finger, the resulting average power of the third path is -6.25dB, -10dB and -15dB for the three spreading factors, respectively. We selected this point and not the intersection with the real IC, because our proposed finger configuration strategy supposes a perfect mitigation of the interference terms.

Fig. 2.18 shows the reconfigurability decision variable G as a function of the mean average power of the third channel path, for the three spreading factors. A close observation of this figure indicates that the proposed analysis gives G estimates close to the true G 's as predicted by our simulations. Moreover, we can see that the switching average powers estimated by the

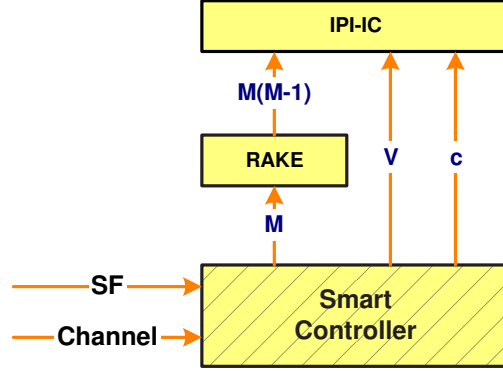


Figure 2.19: The proposed algorithm; A RAKE receiver with a multi-stage IC.

previous figures correspond to the same value of G . This value, which is equal to 3.5, is the reconfigurability switching threshold (T_n) defined in our analysis. This threshold allocates the available fingers in the case of a perfect estimation of the transmitted signal and consists of a useful lower-bound in the real case.

2.5 Multi-stage interference cancelation

2.5.1 Algorithm motivation

The use of a low spreading factor introduces an important IPI degradation which limits the performance of the RAKE receiver. In order to improve the achieved performance and reach close to the performance boundaries corresponding to a RAKE with high spreading factors, an equalization scheme is necessary in the output of the RAKE receiver. The conventional chip equalizers of the literature require complicated matrix inversions [NOG04], [NOG04b], [PRA01] and thus they are not suitable for terminal implementation, where the computational power is a limiting parameter.

The previous algorithm has shown that the partial and single-stage application of an IC algorithm can significantly improve the achieved BER performance. In order to study the limits of this algorithm and its total equalization power, in this Section we present the previous algorithm without computational constraints. More specifically, we deal with a multi-stage IC scheme which is applied in the output of a RAKE receiver in order to suppress all the generated IPI terms in a multi-stage structure.

We note that the multi-stage IC algorithm is a popular non-linear MUD technique with computational requirements lower than the conventional linear MUD algorithms [PAP00], as presented in Chapter 1. However, it is applicable only in the uplink, as it needs the knowledge of the codes of the other interfering users. For the case of high data rates, which correspond to single-user connections, this IC technique is also applicable in the downlink, as only one user-code is needed. The proposed equalization technique, with the appropriate parametrization, can give a performance similar to the conventional linear chip equalizers, but with lower computational requirements. Fig. 2.19 introduces the proposed reconfigurable detector.

2.5.2 Problem formulation

We consider a downlink scenario with only one active user using a QPSK modulation. The equivalent baseband received signal for a multi-path Rayleigh channel can be written as

$$r(t) = \sum_{k=1}^K \sum_{l=1}^L h_l(k) b(k) c(t - kT_b - \tau_l) + n(t), \quad (2.32)$$

where $r(t)$ is the received signal, K is the length of the observation window in symbols, L denotes the number of resolvable paths, $h_l(k)$ is the complex fading factor, $b(k)$ denotes the k -th QPSK transmitted symbol, τ_l is the l -th propagation delay, T_b is the symbol duration and $n(t)$ is the white Gaussian noise.

In Eq. 2.32 the normalized signature waveform, $c(t)$, is

$$c(t) = \sum_{n=1}^N c(n) p(t - nT_c), \quad (2.33)$$

where $T_b/T_c = N$ is the spreading factor, T_c is the chip duration, $c(n) \in \{+1, -1\}$ is the n -th element of the spreading sequence, and $p(t)$ is a rectangular chip pulse with duration $[0, T_c]$.

We assume a conventional RAKE receiver which is equipped with M correlators (fingers), each tuned to one of the channel paths. In each finger, the same user's signals arriving from other paths cause the self-interference. With maximal ratio combining and perfect channel estimation, the final decision statistic of the RAKE receiver is given by:

$$\begin{aligned} \hat{b}(k) &= \sum_{m=1}^M h_m^*(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} r(t) c(t - \tau_m) dt = \\ &= D + S + W, \end{aligned} \quad (2.34)$$

where D , S , and W are the desired signal component, the self-interference and the noise term, respectively, in the output of the correlator. Furthermore, each component can be written as

$$D = b(k) T_b \sum_{m=1}^M |h_m(k)|^2, \quad (2.35)$$

$$\begin{aligned} S &= \sum_{m=1}^M \sum_{\substack{l=1 \\ l \neq m}}^L h_m^*(k) h_l(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} b(t - \tau_l) c(t - \tau_l) c^*(t - \tau_m) dt = \\ &= \sum_{m=1}^M \sum_{\substack{l=1 \\ l \neq m}}^L h_m^*(k) h_l(k) [b(k - 1 - \delta_{l,m}) R(\tau_{l,m} - \delta_{l,m} T_b) + b(k - \delta_{l,q}) \hat{R}(\tau_{l,m} - \delta_{l,m} T_b)], \end{aligned} \quad (2.36)$$

$$W = \sum_{m=1}^M h_m^*(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} n(t) c^*(t - \tau_m) dt, \quad (2.37)$$

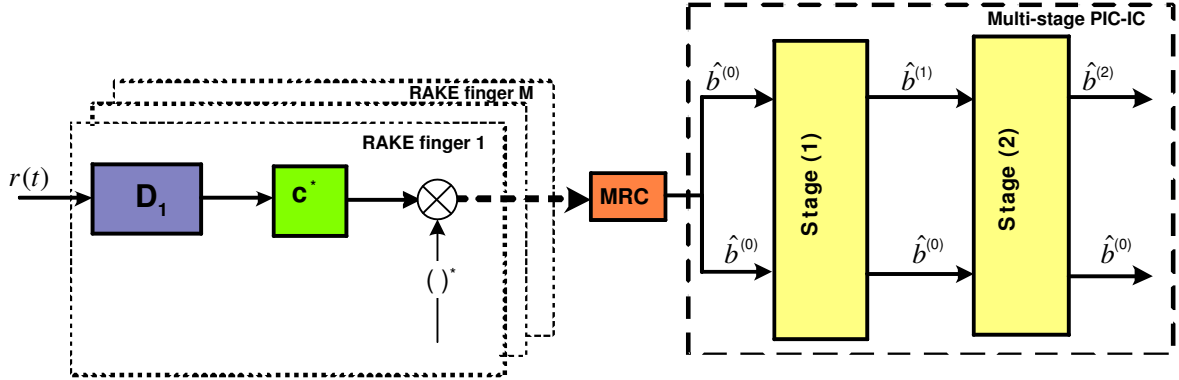


Figure 2.20: The general block diagram of the detection algorithm under consideration (combination of RAKE with the multistage IPI-IC).

where $\tau_{l,m} = \tau_l - \tau_m$, $\delta_{l,m} = \lfloor \tau_{l,m}/T_b \rfloor$, $R(\tau) = \int_0^\tau c(t-\tau)c^*(t)dt$ and $\hat{R}(\tau) = \int_\tau^{T_b} c(t-\tau)c^*(t)dt$ for $0 \leq \tau \leq T_b$ are the continuous-time partial cross-correlations of Pursley.

The term S is the mathematical representation of the self-interference and is generated by the impossibility to construct perfect orthogonal spreading codes for all the time shifting. It consists of $M(L-1)$ terms which are generated from the correlation of each finger with the other $L-1$ paths of the channel. This property can be formulated by the equation

$$R'(\tau) = R(\tau) + \hat{R}(\tau) = \begin{cases} 1 & , \text{if } \tau = 0 \\ -\frac{1}{N} & , \text{if } \tau \neq 0 \end{cases} \quad (2.38)$$

For the high bit rate communications, where N is very low, the IPI-degraded influence in the performance of the conventional RAKE is more important. An equalization scheme in the output of the RAKE receiver can improve its performance.

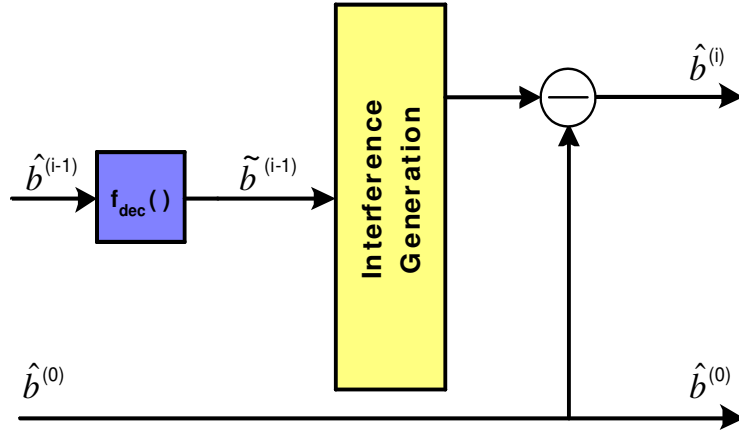
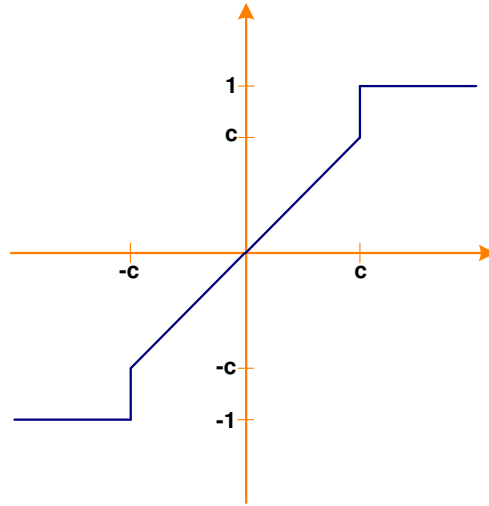
2.5.3 The proposed receiver

The general structure of the proposed multi-stage interference canceler is shown in Fig. 2.20. The basic idea is to reproduce the interference terms and then subtract them, in order to generate “cleaner” data estimations. The performing of this process in a successively way, would result in a significant improvement in the system performance.

In the initial stage, the receiver, operating in a conventional mode, demodulates and despreads the received signal as a simple RAKE receiver. The initial output of the correlator, $\hat{b}^{(0)}(k)$, is used in each stage of the multi-stage canceler. The total mitigation of its interference is the aim of the following cascaded processing.

The structure of the i -th interference stage is shown in Fig. 2.21. The first principal operation is the estimation of the transmitted signal. In order to achieve this, it performs a decision upon the correlation output from the $(i-1)$ -th stage, $\hat{b}^{(i-1)}(k)$. This can be expressed as

$$\tilde{b}^{(i-1)}(k) = f_{dec}(\hat{b}^{(i-1)}(k)) \quad (2.39)$$

Figure 2.21: The i -th stage of the detection algorithm.Figure 2.22: The decision function $f_{dec}()$ with a threshold c .

The decision function used in the proposed IC may be hard or soft. Hard decision can completely cancel IPI interference when the hard decisions are correct, but the interference could actually double from error propagation of incorrect hard decisions. Soft techniques such as linear decision have no error propagation but are not very efficient. In our proposition we consider a hybrid decision function

$$f_{dec}(r) = \begin{cases} 1 & , \text{ if } r > c \\ r & , \text{ if } r \in [-c, c] \\ -1 & , \text{ if } r < -c \end{cases} \quad (2.40)$$

Fig. 2.22 shows this function. It is applied for the real and the imaginary part of each QPSK symbol. The idea behind this hybrid function is that, when the signal is strong (outside the interval $[-c, c]$, where c is the decision threshold with $0 \leq c \leq 1$), a hard decision is made, but when the signal is weak, a soft-linear decision is made, to avoid propagation errors [ZHA03].

In the sequel, $\tilde{b}^{(i-1)}(k)$ and channel estimation are used in order to reproduce replicas of the

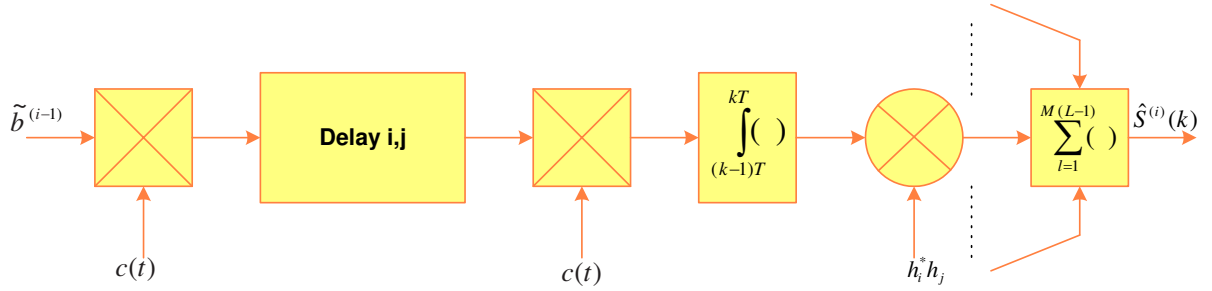


Figure 2.23: The interference generation process.

interference terms present in the initial correlation output. Fig. 2.23 presents the generation interference process. Assuming that $\hat{S}^{(i)}(k)$ denotes the generated interference terms, the new correlation output is written as

$$\hat{b}^{(i)}(k) = \hat{b}^{(0)}(k) - \hat{S}^{(i)}(k) \quad (2.41)$$

It is therefore clear that one must have good data and channel estimates in order for this iterative scheme to work well. When good estimates are not available, a propagation error is generated which degrades the performance.

An important structural block of the proposed reconfigurable receiver is the SPV, which decides at run time the configuration which jointly optimizes performance and computational power. According to the propagation conditions (multipath, SNR) and the used transmission parameters (SF), SPV defines the appropriate algorithmic response (RAKE or RAKE+IC) and adjusts its parameter set. For the general case of a sufficient available computational power, the SPV provides a number of fingers equal to L and a number of generators equal to $L(L-1)$. Moreover, for the IC processing, it defines the required threshold of the hybrid decision function and the number of IC stages. This is the optimal parameter selection, which gives the best possible performance with the minimum number of stages. In our study, we suppose that this type of information is given by computer simulations. For the cases where the available computational power is not sufficient for an optimal RAKE+IC configuration (L fingers, $L(L-1)$ generators), the SPV is more complicated and must have the auto-intelligence to divide the available computational power between these two different functionalities. The idea of the finger configuration algorithm introduced to Section 2.4.4 can be used for this advanced SPV. However, in our study, we suppose that the available computational power is enough to support an optimal configuration corresponding to each operational environment.

2.5.4 The proposed scheme versus conventional receivers

In order to compare the proposed equalization scheme with the conventional advanced receivers of the literature, a matrix representation is required. Thus, at the transmitter the following signal

$$\mathbf{s} = \mathbf{C}_b \mathbf{b}, \quad (2.42)$$

is sent, where \mathbf{C}_b depicts the spreading matrix with $\dim(\mathbf{C}_b) = [N \times 1]$, \mathbf{b} encloses the data with $\dim(\mathbf{b}) = [K \times 1]$ and N is the spreading factor. The downlink channel is characterized by its impulse response \mathbf{h} with $\dim(\mathbf{h}) = L$, where L are the channel samples at chip interval T_c . For simplicity, the channel is assumed to be time-invariant during the transmission of the data symbol sequence \mathbf{b} . Nevertheless, the following considerations could easily be extended to the case of time-varying channel by updating \mathbf{h} during the transmission. The combined channel impulse response is defined by the convolution

$$\mathbf{q} = \mathbf{C}_b * \mathbf{h}, \quad (2.43)$$

where \mathbf{q} is a vector of dimension $[N + L - 1 \times 1]$. In order to simplify the matrix representation of the received signal, we define a new matrix \mathbf{A} , called system matrix. It has a dimension $\dim(\mathbf{A}) = [KN + L - 1 \times K]$ and its elements are defined by the equation

$$A_{N(k-1)+l,k} = \begin{cases} q_l & , \text{for } k = 1 \dots K, l = 1 \dots N + L - 1 \\ 0 & , \text{else} \end{cases} \quad (2.44)$$

Thus the received signal can be written as

$$\mathbf{r} = \mathbf{A}\mathbf{b} + \mathbf{n}, \quad (2.45)$$

where \mathbf{n} is the Additive White Gaussian Noise column vector, of length $KN + L - 1$. for example, for $K = 3$, $N = 4$ and $L = 4$, the system matrix has the following form:

$$\mathbf{A} = \begin{pmatrix} q_1 & 0 & 0 \\ q_2 & 0 & 0 \\ q_3 & 0 & 0 \\ q_4 & 0 & 0 \\ q_5 & q_1 & 0 \\ q_6 & q_2 & 0 \\ q_7 & q_3 & 0 \\ 0 & q_4 & 0 \\ 0 & q_5 & q_1 \\ 0 & q_6 & q_2 \\ 0 & q_7 & q_3 \\ 0 & 0 & q_4 \\ 0 & 0 & q_5 \\ 0 & 0 & q_6 \\ 0 & 0 & q_7 \end{pmatrix} \quad (2.46)$$

In the following, three different linear detection algorithms are discussed: the Matched Filter (MF) algorithm and two linear joint detection algorithms working on data blocks, which are the Zero-Forcing (ZF) and the Minimum Mean-Square-Error (MMSE) algorithm. Both have been described in detail in [KLE96]. The MF is the simplest algorithm and computes estimates of the transmitted symbols $\hat{\mathbf{b}}$ by the linear operation

$$\hat{\mathbf{b}}_{\text{MF}} = \mathbf{A}^H \mathbf{b} \quad (2.47)$$

Technique	CMULs
Direct	$(NK + L - 1)^3$
Block-Toeplitz	$5/4(NK + L - 1)^2$
Polyphase	$(NK + 1)^2$
Gauss-Seidel	$n_{iter}(NK + L - 1)^2$

Table 2.9: The complexity of matrix inversion.

The RAKE receiver is a special, modified implementation of the MF detection algorithm, as only a certain number of the strongest paths are taken into account for reception. The MF algorithm suffers from MAI and IPI, which degrade the performance. For the following considerations, the special case of AWGN, resulting in a noise covariance matrix $\mathbf{R}_n = \sigma^2 \mathbf{I}$, is assumed. The Zero-Forcing algorithm minimizes the squared Euclidean distance $|\mathbf{r} - \mathbf{A}\mathbf{b}|^2$, and the ZF data estimation is computed with

$$\hat{\mathbf{b}}_{\text{ZF}} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{r} \quad (2.48)$$

This results in totally suppressed MAI and ISI. However, the drawback is an increased noise variance, which is especially distinct for low SNR environments. The MMSE algorithm minimizes the mean squared norm of the estimation error $\hat{\mathbf{b}} - \mathbf{b}$ with respect to $\hat{\mathbf{b}}$. The MMSE algorithm leads to the following data estimation expression, assuming AWGN and uncorrelated data with covariance matrix $\mathbf{R}_b = \mathbf{I}$,

$$\hat{\mathbf{b}}_{\text{MMSE}} = (\mathbf{A}^H \mathbf{A} + \sigma^2 \mathbf{I})^{-1} \mathbf{A}^H \mathbf{r} \quad (2.49)$$

This algorithm considers noise parts in the decision variables and thus can overcome the performance of the ZF scheme. From Eq. 2.48 and 2.49, we know that in the case of $\sigma \rightarrow 0$, the performance of the MMSE approaches that of the ZF. Furthermore, with Eq. 2.47 we can see that for $\sigma \rightarrow \infty$, the MMSE performance approaches that of the MF.

From the above equations, it is clear that the advanced conventional receivers have a high computational complexity, due to the matrix inversions involved. This complexity depends on the size of the system matrix. The matrix inversion in the linear equalizers is the focus of complexity reduction in many studies. Table 2.9 summarizes the achieved complexity in the number of complex multiplications required for four different techniques of matrix inversion [MAI01]. From this table, we can see that significant reductions in algorithm complexity are possible using the Block-Toeplitz, polyphase or iterative approaches (Gauss-Seidel), relatively to direct implementation, with only minimal performance loss. In spite of this complexity reduction, the block linear equalizers have a high implementation complexity. This complexity is more critical when the terminal has a velocity (channel change), and thus the equalizer vector must be computed in real-time and with a well-defined time constraint. On the other hand, the proposed reconfigurable detector does not require matrix inversions and has a total complexity which is linear with the number of symbols and channel paths $O(KL)$.

2.5.5 Numerical results

Algorithmic efficiency is measured by the BER performance of the proposed reconfigurable detector and is obtained through computer simulations. The simulation environment is based on the downlink specifications of the FDD-UMTS standard, and the used radio channel has three independent Rayleigh fading paths. In the first group of simulation results, a simple hard decision has been used in order to produce the transmitted signal estimations. In the second simulation group, simulation results with a hybrid decision function show the achieved performance optimization.

Hard decision

Firstly, a hard decision for the estimation of the transmitted signal is used by the IC generators. Fig. 2.24 shows the impact of the SNR and the number of IC stages in the achieved BER performance. The used spreading factor is $N = 4$ and the three-path radio channel is characterized by the vectors $\mathbf{h} = [0 \ 0 \ 0]\text{dB}$ and $\tau = [0 \ 6 \ 8]T_c$. From the results shown, it appears that the IC processing is suitable at the high SNRs. This is to be expected, because the output of the RAKE receiver at the high SNRs is a more reliable estimation of the transmitted signal and thus the cancelation scheme performs better. Also, we can see that for $E_b/N_0 \leq 8\text{dB}$, the IC with a single stage is the most appropriate configuration, since for $E_b/N_0 > 8\text{dB}$ the use of one more stage can further improve the performance. However, the use of one more stage for the IC processing (3 stages) does not improve the previously achieved performance and thus the maximum number of IC stages is limited to $V = 2$. The conclusion of this simulation figure is that the SNR is a significant configuration parameter of the proposed detection scheme and, as the SNR increases, the IC processing becomes more important.

In Fig. 2.25 we consider a different TDL of the channel, which is $\tau = [0 \ 1 \ 2]T_c$. We can see that, in this case, the cancelation scheme is less efficient. At the low SNR's, we can arrive at the best possible performance without using the IC and, at the high SNR's, an IC without iterations is the appropriate configuration. By Fig.'s 2.24 and 2.25, it is indicated that the TDL of the channel is a real parameter which has an impact on the configuration of the proposed reconfigurable detector. In general, as the delay line is increased (worst channel), the IC becomes more efficient in optimizing the RAKE performance.

Since the first two system parameters under consideration are determined by the external environment, the third one is chosen by the transmitter. This parameter is the spreading factor which defines the user service. Fig. 2.26 depicts the BER performance for different configurations of the IC, as a function of the spreading factor. Here the desired user's SNR is 16 dB and the other parameters are exactly as in the first simulation, that is, $\mathbf{h} = [0 \ 0 \ 0]\text{dB}$ and $\tau = [0 \ 6 \ 8]T_c$. It can be seen that the performance with a large spreading factor is close to the optimal performance without IPI. Thus for a spreading factor $N \geq 64$, the cancelation scheme must be switched off. However, when the spreading factor is reduced, the IC can significantly improve the performance. As for the optimal number of IC stages, we can say that for a spreading factor $8 \leq N \leq 32$, a simple use of the IC (a single stage) is enough, since for $N = 4$, one more stage can further improve the system performance. Thus, as the spreading factor increases, the IC processing is less efficient.

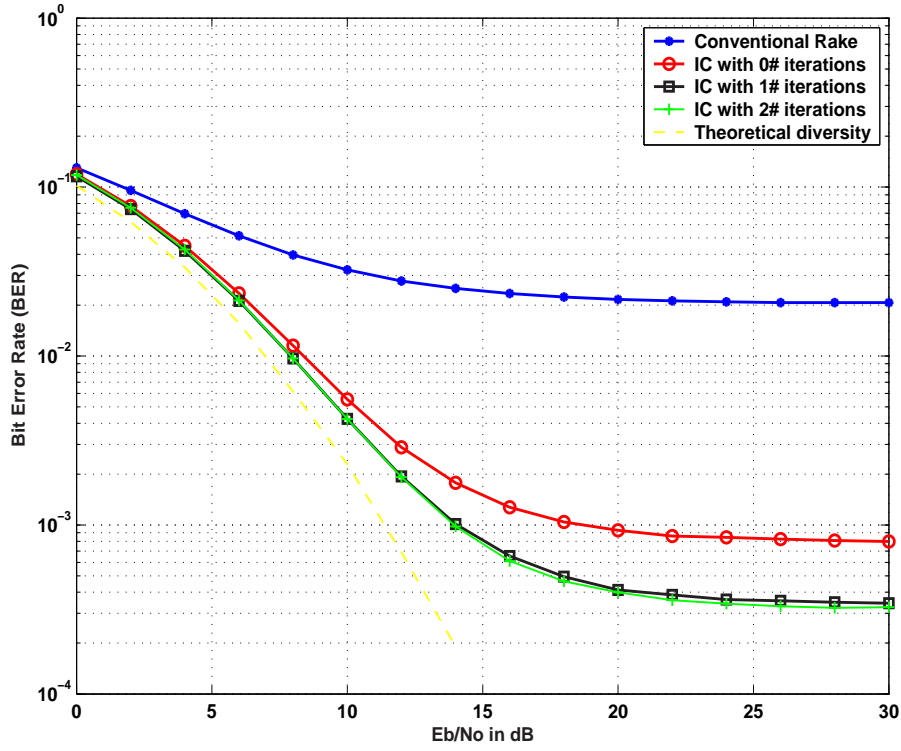


Figure 2.24: BER performance of the reconfigurable detector for different configurations versus E_b/N_0 , for $N=4$ and $\mathbf{h} = [0 \ 0 \ 0]$ dB, $\tau = [0 \ 6 \ 8]T_c$.

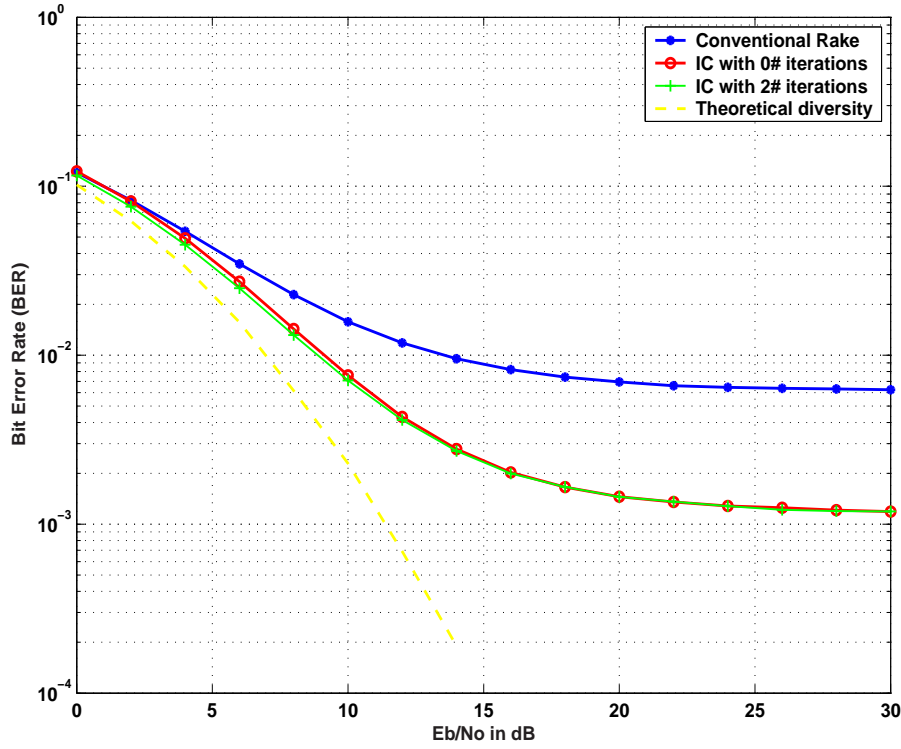


Figure 2.25: BER performance of the reconfigurable detector for different configurations versus E_b/N_0 , for $N = 4$ and $\mathbf{h} = [0 \ 0 \ 0]$ dB, $\tau = [0 \ 1 \ 2]T_c$.

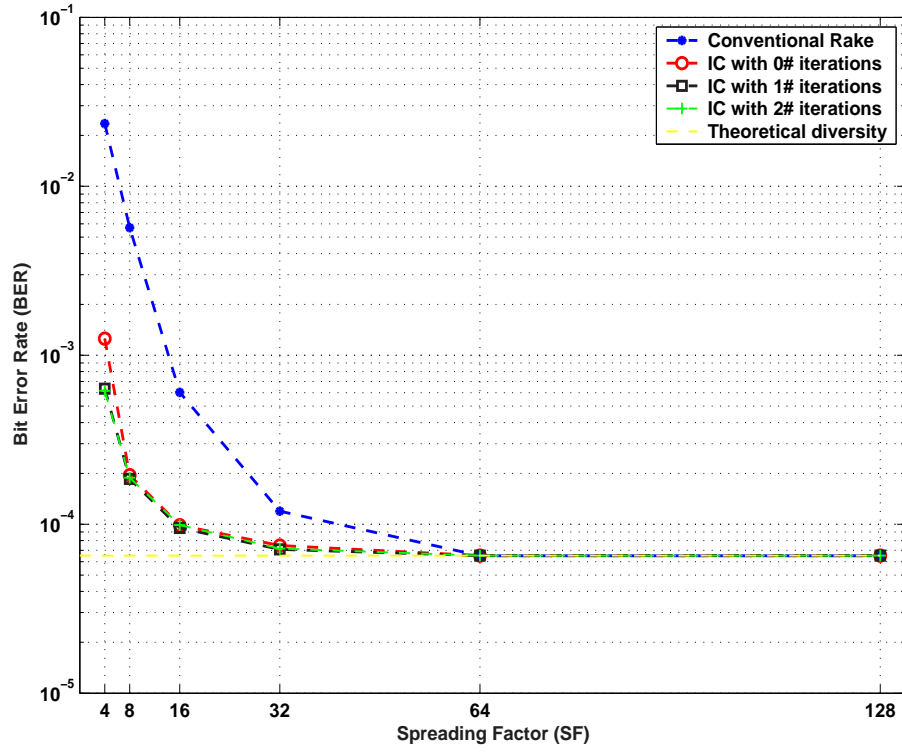


Figure 2.26: BER performance of the reconfigurable detector for different configurations versus the spreading factor, for SNR=16dB and $\mathbf{h} = [0 \ 0 \ 0]$ dB, $\tau = [0 \ 6 \ 8]T_c$.

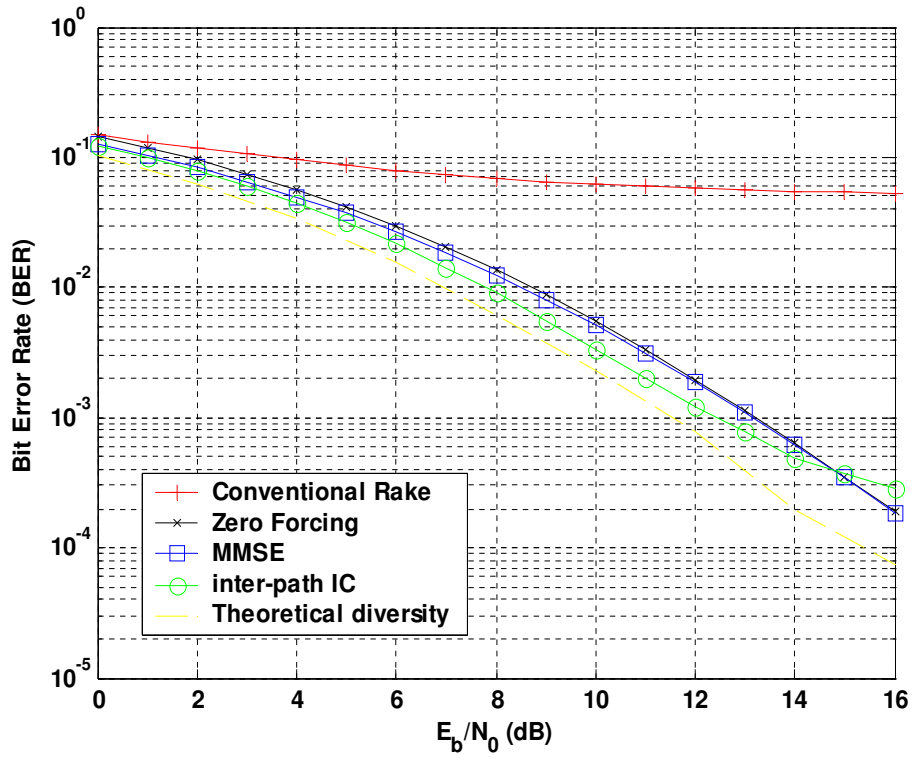


Figure 2.27: BER comparison over equal 3-path channel and $N = 2$.

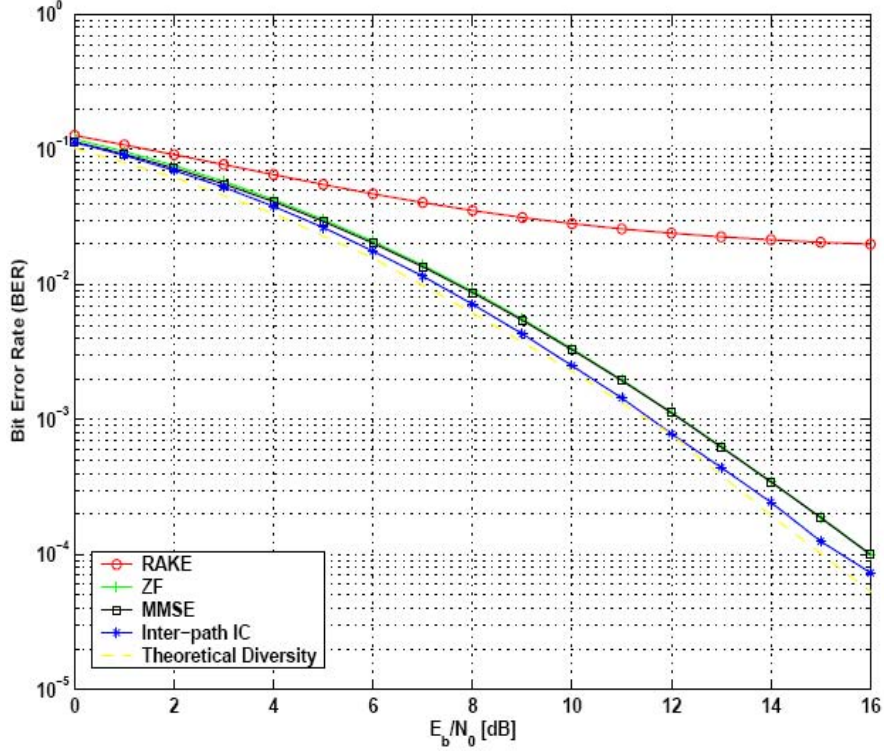


Figure 2.28: BER comparison over equal 3-path channel and $N = 4$.

Hybrid decision

In order to further improve the achieved performance of the proposed detection scheme, a hybrid decision function is considered for the IC generation process. The used function is more robust to propagation errors and thus our detection method has a performance similar to that of the complex conventional detection schemes of the literature.

Fig.'s 2.27 and 2.28 show the performance of the proposed detection scheme for a spreading factor equal to 2 and 4, respectively. The used radio channel has also three independent Rayleigh-fading paths, with average power $\mathbf{h} = [0 \ 0 \ 0]$ dB and delays $\tau = [0 \ 3 \ 6]T_c$. In order to show the efficiency of the proposed detector, the performance of the basic conventional MUD detectors (ZF, MMSE) is also presented. From the presented curves, we can see that for the case of $N = 2$, the proposed receiver outperforms the conventional complex detectors. The achieved gain is about 1dB for the high SNRs. For a spreading factor $N = 4$, the performance of the proposed detection scheme is similar to that of the conventional equalizers. These simulation results evaluate the efficiency of the proposed receiver, and they introduce it as a suitable detection solution for applications where the computational power is a critical parameter.

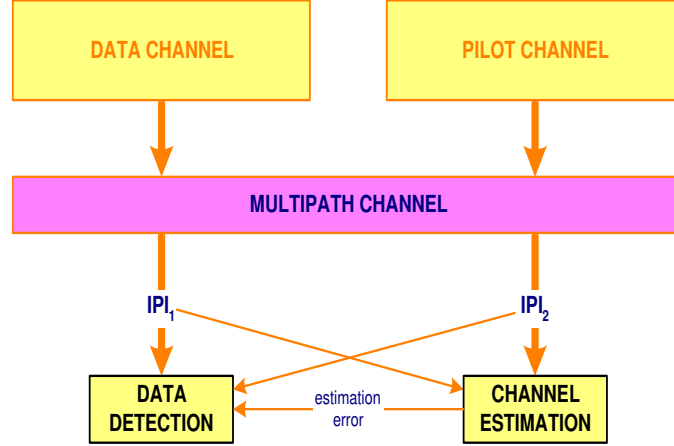


Figure 2.29: The intra (IPI_1)- and inter (IPI_2)- interference (from the data channel point of view).

2.6 Multi-stage interference cancelation with realistic channel estimation

2.6.1 Algorithm motivation

In the previous application example, the involved multi-stage IC scheme is used for a high data rate connection, which is represented by a single-user environment with only one logic channel.

However, in parallel with the data-dedicated channel, there is a group of common channels which are used for the system control. Between them, the pilot channel which is used for the channel estimation has a close connection with the detection of the dedicated channel. In the case of a high data rate service, the spreading factor used for the pilot channel is also low, and thus its detection suffers from IPI phenomena. This degradation influences the quality of the channel estimations and thus also the performance of the coherence RAKE detector. Moreover, it has a structure similar to the IPI introduced in the dedicated channel. Thus it can be considered for a multi-stage interference cancelation. The proposed receiver of this Section tries to remove the interfering signals from both dedicated and pilot channels. Each stage is divided in two parts, where in the first one, a cleaner channel estimation is provided, and in the second one, a cleaner symbol detection.

Furthermore, the existence of two parallel channels over the multi-path environment introduces an interference between them. This type of interference is more important for the dedicated channel, if we suppose that the pilot channel is transmitted in a higher power in order to facilitate the channel estimation. This interference has also a well-defined structure similar to intra-channel IPI interference, and thus can be introduced to a multi-stage IC scheme. Fig. 2.29 graphically presents the degradation impact of the two different IPI types.

The algorithmic reconfigurability concept is introduced in order to provide in real-time the configuration which jointly optimizes performance and computational resources. Thus, like in the previous example, the proposed detector is equipped with the appropriate algorithmic response and parameter set. The IC scheme is also used for the case of a low SF (and $L > 1$) in order

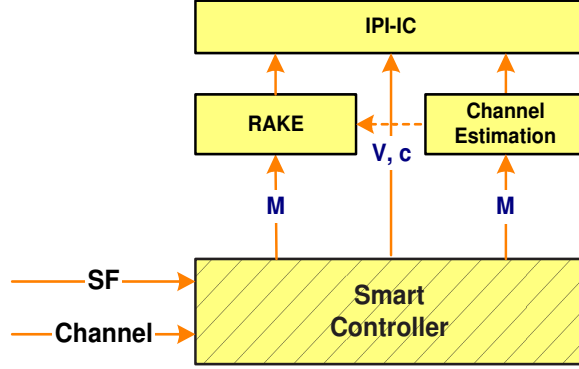


Figure 2.30: The general structure of the proposed algorithm; A RAKE receiver with a pilot channel estimation and IC.

to improve the initial channel estimation and RAKE detection. The basic difference is that the operational environment consists of one more logical channel, and the IC processing under consideration is more complicated. Fig. 2.30 presents the general structure of the proposed reconfigurable detector.

2.6.2 Problem formulation

In this section, we describe the baseband downlink model of the CDMA communication system considered throughout this application example. The downlink model is based on a single-user DS-CDMA system over a multipath Rayleigh fading channel, with pilot-aided channel estimation. The transmitted process can be expressed as

$$s(t) = A_0 c_0(t) + A_1 b(t) c_1(t), \quad (2.50)$$

$$b(t) = \sum_{k=-\infty}^{\infty} b(k) p_t(kT_b, (k+1)T_b), \quad (2.51)$$

$$c_m(t) = \sum_{k=-\infty}^{\infty} c_m(k) p_t(kT_c, (k+1)T_c), \quad (2.52)$$

where $b(t)$ denotes the QPSK data signal, $c_m(t)$ denotes the signature sequence signals for the pilot ($m = 0$) and the data signal ($m = 1$), respectively, $p_t(t_1, t_2)$ is a unit rectangular pulse on $[t_1, t_2)$, $b(k) \in \{\pm 1 \pm 1j\}$ with equal probabilities, and $c_m(k) \in \{\pm 1\}$. A_0 and A_1 are the transmitted amplitudes of the pilot and user signal, respectively. T_b and T_c are the symbol period and the chip period, and $T_b/T_c = N$, where N is the spreading factor. We note that no data symbols are present in the pilot channel.

A frequency-selective Rayleigh fading channel with L resolvable paths is modeled as

$$h(t, \tau) = \sum_{l=1}^L h_l(t) \delta(t - \tau_l), \quad (2.53)$$

where h_l and τ_l are the channel gain and the propagation delay of the l -th path. h_l follows a zero-mean complex Gaussian distribution with average power σ_l^2 and it is unchanged over one symbol duration. Then the received signal can be written as

$$r(t) = \sum_{l=1}^L h_l(t) [A_0 c_0(t - \tau_l) + A_1 b(t - \tau_l) c_1(t - \tau_l)] + n(t), \quad (2.54)$$

where $n(t)$ is white Gaussian noise with double-sided power spectral density $N_0/2$.

In the uncoded data transmission, it is assumed that the delays τ_l are perfectly recovered. In order to estimate the channel coefficients which are needed for the coherence demodulation, the received signal is despread by the spreading code of the pilot channel. Thus, the estimated complex channel fading coefficient for the j -th path and the k -th transmitted symbol, is given by

$$\begin{aligned} \hat{h}_j(k) &= \frac{1}{NA_0} \int_{(k-1)T_b + \tau_j}^{kT_b + \tau_j} r(t) c_0^*(t - \tau_j) dt \\ &= h_j(k) + S_j(k) + C_j(k) + \mu_j(k), \end{aligned} \quad (2.55)$$

where $h_j(k)$ is the channel coefficient for the j -th path and the k -th transmitted symbol, $S_j(k)$ and $C_j(k)$ are the interferences from the other paths of the data and pilot channel, respectively, and $\mu_j(k)$ is the noise term. Furthermore, each interference component can be written as

$$S_j(k) = \frac{A_1}{NA_0} \sum_{\substack{l=1 \\ l \neq j}}^L h_l(k) \int_{(k-1)T_b + \tau_j}^{kT_b + \tau_j} b(t - \tau_l) c_1(t - \tau_l) c_0^*(t - \tau_j) dt, \quad (2.56)$$

$$C_j(k) = \frac{1}{N} \sum_{\substack{l=1 \\ l \neq j}}^L h_l(k) \int_{(k-1)T_b + \tau_j}^{kT_b + \tau_j} c_0(t - \tau_l) c_0^*(t - \tau_j) dt, \quad (2.57)$$

$$\mu_j(k) = \frac{1}{NA_0} \int_{(k-1)T_b + \tau_j}^{kT_b + \tau_j} n(t) c_0^*(t - \tau_j) dt \quad (2.58)$$

Demodulation is similar to the channel estimation operation, except despreading is done with the data spreading code and a multiplication with the estimated channel coefficients is required. Assuming a RAKE combination with MRC, and M available demodulation fingers locked on to the M stronger channel paths, the decision variable of the k -th data symbol is given by

$$\begin{aligned} \hat{b}(k) &= \frac{1}{NA_1} \sum_{m=1}^M \hat{h}_m^*(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} r(t) c_1^*(t - \tau_m) dt \\ &= D(k) + I(k) + F(k) + \eta(k), \end{aligned} \quad (2.59)$$

where $D(k)$ is the produced diversity signal component, $I(k)$ and $F(k)$ are the self-interferences from the other paths of the data and pilot channel, respectively, and $\eta(k)$ is the noise component. We can write these parts of the correlator output as

$$D(k) = b(k) \sum_{m=1}^M h_m(k) \hat{h}_m^*(k), \quad (2.60)$$

$$I(k) = \frac{A_0}{NA_1} \sum_{m=1}^M \sum_{\substack{l=1 \\ l \neq m}}^L \hat{h}_m^*(k) h_l(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} c_0(t - \tau_l) c_1^*(t - \tau_m) dt, \quad (2.61)$$

$$F(k) = \frac{1}{N} \sum_{m=1}^M \sum_{\substack{l=1 \\ l \neq m}}^L \hat{h}_m^*(k) h_l(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} b(t - \tau_l) c_1(t - \tau_l) c_1^*(t - \tau_m) dt, \quad (2.62)$$

$$\eta(k) = \frac{1}{NA_1} \sum_{m=1}^M \hat{h}_m^*(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} n(t) c_1^*(t - \tau_m) dt \quad (2.63)$$

From the first equation set (2.55-2.58), we can see that the quality of the channel estimation using pilot symbols is reduced by multipath interference effects, and thus an estimation error is introduced. The second equation set (2.59-2.63) shows that the signal detection suffers from both estimation error and IPI. This embedded IPI is a result not only of the multipath impact on the data channel (F), but also of the parallel existence of the pilot channel over the same multipath channel (I). For very high bit rates, the IPI and the generating channel estimation error can significantly reduce the performance of the conventional RAKE receiver.

2.6.3 The proposed receiver

The general structure of the proposed multi-stage interference canceler is shown in Fig. 2.31. The basic idea is to reproduce the interference terms and then subtract them, in order to generate “cleaner” channel and data estimations. The successive execution of this process, would result in significant improvements in the system performance.

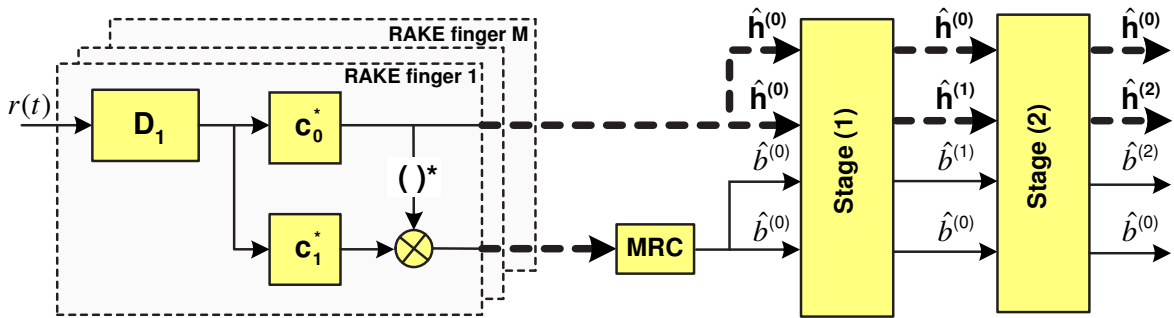
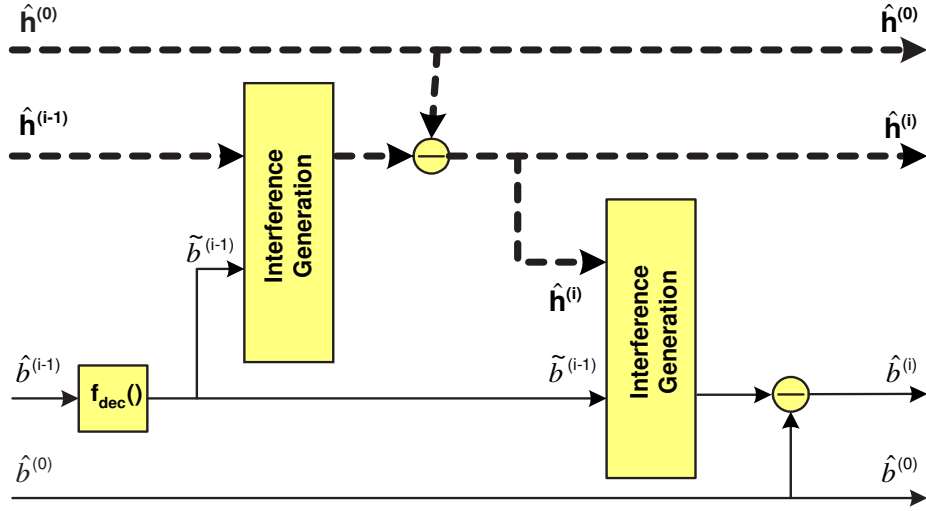


Figure 2.31: General structure of the multi-stage inter-path interference canceler with a realistic channel estimation.

In the initial stage, the receiver, operating in a conventional mode, demodulates and despreads the received signal as presented by Eq.'s 2.55 and 2.59. The initial outputs of the correlator, $\hat{b}^{(0)}(k)$, and of the channel estimator, $\hat{\mathbf{h}}^{(0)}(k) = [\hat{h}_1^{(0)}(k) \dots \hat{h}_M^{(0)}(k)]$, will be used in each stage of

Figure 2.32: Structure of the i -th cancellation stage.

the multi-stage canceler. The total mitigation of the interference embedded in these terms is the aim of the following cascaded processing.

The structure of the i -th interference stage is shown in Fig. 2.32. The first major operation is the estimation of the transmitted signal. In order to achieve this, the canceler performs a decision (similar to the previous application example) upon the correlation output from the $(i-1)$ -th stage, $\hat{b}^{(i-1)}(k)$. This can be expressed as

$$\tilde{b}^{(i-1)}(k) = f_{dec}(\hat{b}^{(i-1)}(k)) \quad (2.64)$$

This estimation is combined with the channel estimation output from the $(i-1)$ -th stage $\hat{\mathbf{h}}^{(i-1)}(k)$ for reproducing the interference terms involved in the initial channel estimation. If $\hat{\mathbf{S}}^{(i)}(k) = [\hat{S}_1^{(i)}(k) \dots \hat{S}_M^{(i)}(k)]$ and $\hat{\mathbf{C}}^{(i)}(k) = [\hat{C}_1^{(i)}(k) \dots \hat{C}_M^{(i)}(k)]$ are the generated replicas of the interference terms, the updated channel estimation can be written as

$$\hat{\mathbf{h}}^{(i)}(k) = \hat{\mathbf{h}}^{(0)}(k) - \hat{\mathbf{S}}^{(i)}(k) - \hat{\mathbf{C}}^{(i)}(k) \quad (2.65)$$

In the sequel, $\hat{\mathbf{h}}^{(i)}(k)$ and $\tilde{b}^{(i-1)}(k)$ are used to reproduce replicas of the interference terms presented in the initial correlation output. Assuming that $\hat{I}^{(i)}(k)$ and $\hat{F}^{(i)}(k)$, are these generated interference terms, the new correlation output is written as

$$\hat{b}^{(i)}(k) = \hat{b}^{(0)}(k) - \hat{I}^{(i)}(k) - \hat{F}^{(i)}(k) \quad (2.66)$$

It is therefore clear that one must have good data and channel estimates in order for this iterative scheme to work well. When good estimates are not available, a propagation error is generated which degrades the performance.

The SPV is also the central logic block of this receiver. Its task is similar to the SPV controller of the previous application example, except that, for this receiver, an IC stage is more complex. For this case also, we suppose that the available computational power is sufficient to

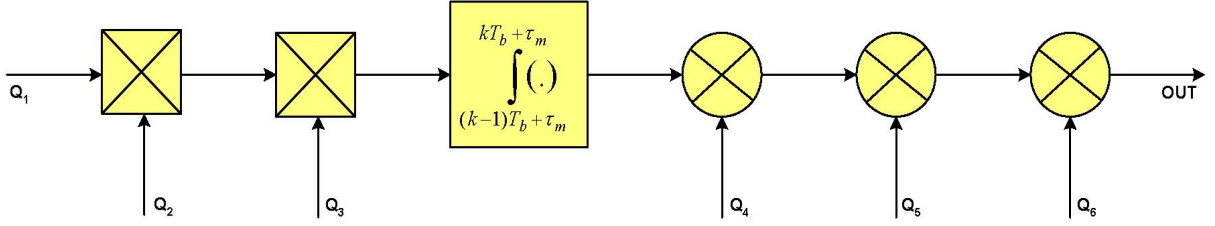


Figure 2.33: The computational similarities of the essential operations of the proposed reconfigurable detector.

Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	OUT
$\hat{b}(t - \tau_j)$	$c_1(t - \tau_j)$	$c_0^*(t - \tau_m)$	$\hat{h}_j(k)$	1	A_1/NA_0	$\hat{S}_{m_j}(k)$
1	$c_0(t - \tau_j)$	$c_0^*(t - \tau_m)$	$\hat{h}_j(k)$	1	$1/N$	$\hat{C}_{m_j}(k)$
$\hat{b}(t - \tau_j)$	$c_1(t - \tau_j)$	$c_1^*(t - \tau_m)$	$\hat{h}_j(k)$	$\hat{h}_m^*(k)$	$1/N$	$\hat{F}_{m_j}(k)$
1	$c_0(t - \tau_j)$	$c_1^*(t - \tau_m)$	$\hat{h}_j(k)$	$\hat{h}_m^*(k)$	A_0/NA_1	$\hat{I}_{m_j}(k)$
$r(t)$	1	$c_1^*(t - \tau_m)$	$\hat{h}_m^*(k)$	1	$1/NA_1$	m -th RAKE finger
$r(t)$	1	$c_0^*(t - \tau_m)$	1	1	$1/NA_0$	$\hat{h}_m(k)$

Table 2.10: The possible configurations of the parameterized computational term.

support all the required algorithmic operations for each operational environment. Thus the SPV provides the necessary number of RAKE fingers and IC generators, which correspond to the strong channel paths, and decides the optimal values for the IC parameters.

The computational similarities

An important characteristic of this proposed receiver is that its essential operations have computational similarities and are of iterative nature. Thus, this receiver can also be considered for an iterative reconfigurable approach. In order to explain this computational property of the proposed detection scheme, Fig. 2.33 presents a parameterized computational term which, according to its inputs (Q_1, \dots, Q_6), can present a RAKE, a channel estimation or an interference generation process. Table 2.10 summarizes the possible configurations of this parameterized computational term.

2.6.4 Numerical results

The simulation environment is based on the FDD-UMTS standard specifications for the downlink case. The radio channel has two independent Rayleigh fading multipaths with average power $\mathbf{h} = [0 \ 0]$ dB and delay $\tau = [0 \ 2]T_c$, and the considered spreading factor is equal to four. The power ratio between the pilot and data channels is equal to 6.5 dB (A_0/A_1). Finally, a RAKE receiver with $M = 2$ fingers is used and, for the decision function of each IC stage, a hard decision is applied.

Mean square error (MSE) comparisons of the conventional estimation ($V = 0$) and our method, for different numbers of stages, are shown in Fig. 2.34. We can see that the estimation of the channel is improved as the number of stages increases. However, after a number of

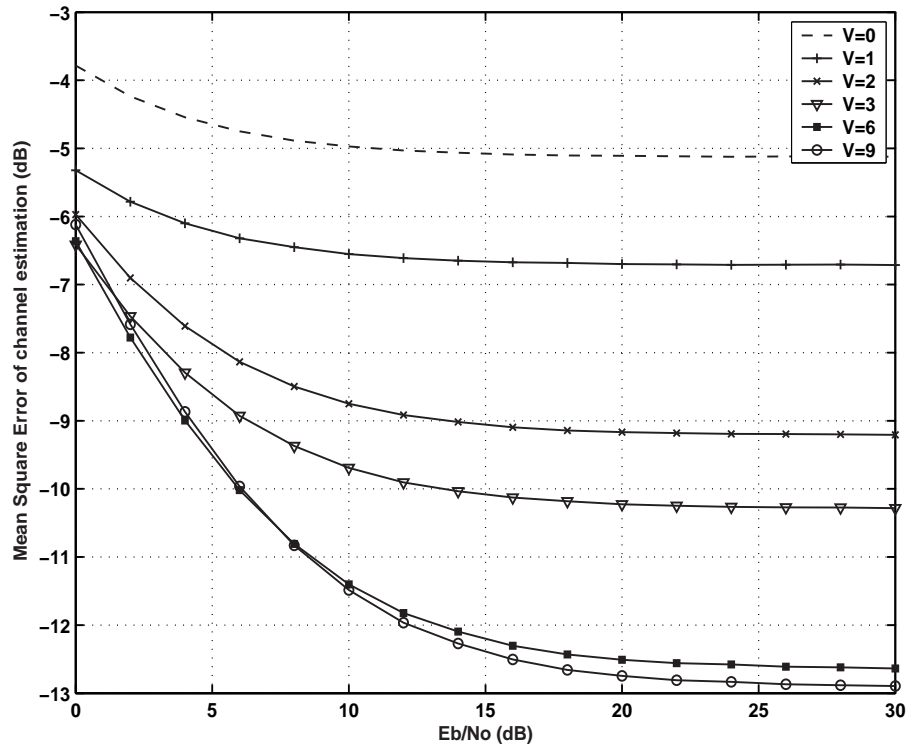


Figure 2.34: MSE performance of the proposed channel estimation scheme versus the classic estimation ($V=0$).

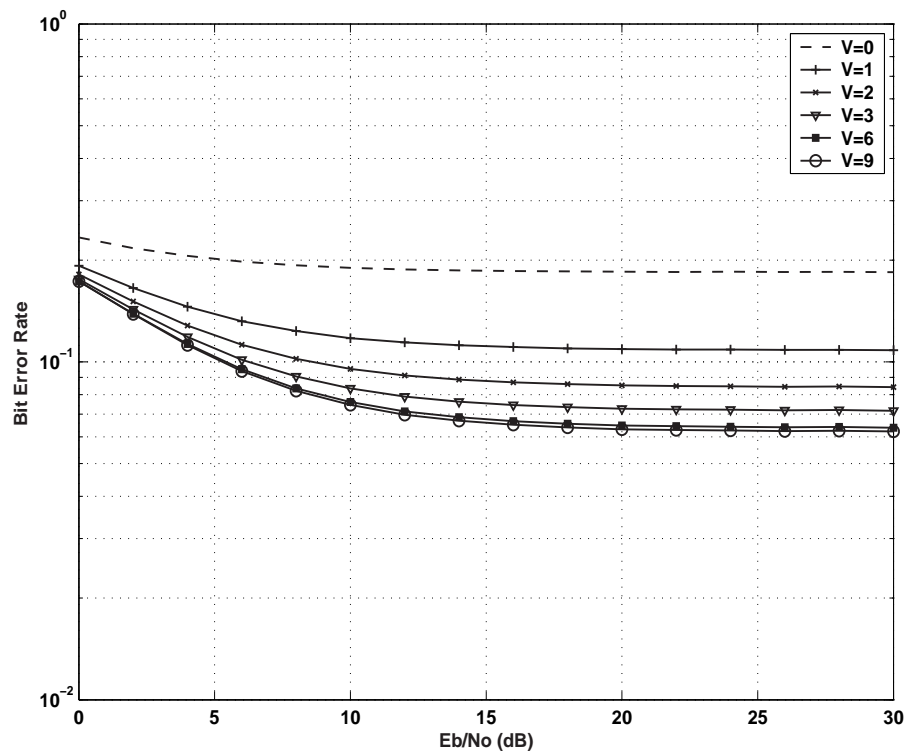


Figure 2.35: BER performance of the proposed reception scheme versus the conventional receiver ($V=0$).

iterations ($V > 6$), the method converges to a value equal to -13dB . A close observation of this figure indicates that for the low SNR's, the MSE becomes worse after the convergent number of iterations. The basic reason is that, in the low SNR's, the regenerating interference terms are unreliable, and thus the subtraction from the initial signals results in error propagation and degrades the performance.

In Fig. 2.35, BER performance of the proposed scheme, for different numbers of stages, is given. The curves confirm the above main observations and verify the accuracy of our method. We can also see that the error propagation observed in the MSE curves does not translate to BER performance degradation.

2.7 Multi-stage interference cancelation for multi-user detection

2.7.1 Algorithm motivation

PIC has the potential to approach the single-user performance boundary when the interference estimate is reliable. Because of the need to regenerate the interfering signal at the receiver, all existing PIC schemes have been proposed for the uplink, as they require the knowledge of all users' codes and energies. As a result, PICs have thus far been assumed to be applicable only at the BS, and not at the mobile terminal, where only one information stream is to be decoded and the spreading codes of interfering users are unknown. However, in the last years, many algorithms have been proposed for the estimation of the codes of interfering users, and thus PIC is also applicable for the downlink [BUR03b], [MOU04]. In this application example, we suppose a perfect estimation of the downlink interfering codes and we focus on the PIC application when a low spreading factor is used. In this case, IPI is very severe and must be mitigated in order to optimize the performance. The PIC schemes of the literature use the correlation input to mitigate MAI phenomena [SUN02], or the correlation output, when IPI is not negligible [YOO93], to mitigate both MAI and IPI. However, for the second case, the structure of the generating MAI terms is more complex and the considered MAI includes more individual terms.

In this Section, we propose a PIC algorithmic scheme which combines performance efficiency and complexity minimization. It is a combination of the above two cancelation techniques, which gives the best possible performance by minimizing the computational requirements. It is appropriate for the future terminal implementations, where the computational constraints are more critical than the BS. More specifically, in order to minimize computational requirements and integrate IPI phenomena, the proposed multi-stage detector cancels MAI to the input of the correlator and IPI to its output. Thus, on the one hand it reduces the number of generated interference terms (MAI), and thus reduces the required complexity, and on the other hand it improves the performance due to IPI cancelation.

The algorithmic reconfigurability concept is introduced in order to provide the detector with the appropriate algorithmic response. For this application example, we can say that there are four different algorithmic responses. The first one is the use of a single RAKE receiver and is used for the case of a single user environment with high SFs (Algo1=RAKE). The second one is the use of an IC scheme in the output of the RAKE receiver and is used for the case of a single user environment with low SFs (and $L > 1$) (Algo2=RAKE+IC_IPI). The third algorithmic response consists of a bank of RAKE detectors with an IC in its output to suppress MAI, and is used

for MAI environments with high SFs (Algo3=RAKE+IC_MAI). Finally, the last algorithmic response is the application of the IC to suppress MAI and IPI interferences, and is used for MAI environments with low SFs (Algo4=RAKE+IC_MAI+IC_IPI). The adaptivity concept is also used in order to adjust the parameter set of each algorithm. Thus the number of RAKE fingers, IC generators and IC stages are adapted in order to correspond to the current instantaneous number of channel paths and users.

2.7.2 Problem formulation

We consider a downlink scenario where U users communicating simultaneously at the same rate, each employing QPSK modulation. Since downlink connection is considered, synchronous transmission of all signals through the same multipath channel is assumed. The equivalent baseband received signal can be written as

$$\begin{aligned} r(t) &= \sum_{u=1}^U r_u(t) + n(t) \\ &= \sum_{u=1}^U \sum_{k=1}^K \sum_{l=1}^L A_u h_l(k) b_u(k) c_u(t - kT_b - \tau_l) + n(t), \end{aligned} \quad (2.67)$$

where $r_u(t)$ is the received signal of the u -th user, K is the length of the observation window in symbols, L denotes the number of resolvable paths, A_u is the average received amplitude of the u -th user, $h_l(k)$ is the complex fading factor, $b_u(k)$ denotes the k -th QPSK transmitted symbol of user u , τ_l is the l -th propagation delay, T_b is the symbol duration and $n(t)$ is the white Gaussian noise.

In Eq. 2.67, the normalized signature waveform of user u , $c_u(t)$, is

$$c_u(t) = \sum_{n=1}^N c_u(n) p(t - nT_c), \quad (2.68)$$

where $T_b/T_c = N$ is the spreading factor, T_c is the chip duration, $c_u(n) \in \{+1, -1\}$ is the n -th element of the spreading sequence for user u , $p(t)$ is a rectangular chip pulse with duration $[0, T_c)$. If we assume that the user of interest is user- i , we can consider that the received signal consists of three components. Therefore, equation 2.67 can also be written as

$$\begin{aligned} r(t) &= \sum_{k=1}^K \sum_{l=1}^L A_i \alpha_l(k) b_i(k) s_i(t - kT_b - \tau_l) + \\ &+ \sum_{\substack{u=1 \\ u \neq i}}^U \sum_{k=1}^K \sum_{l=1}^L A_u \alpha_l(k) b_u(k) s_u(t - kT_b - \tau_l) + n(t) = \\ &= D_i + M_i + n(t), \end{aligned} \quad (2.69)$$

where D_i and M_i are the useful diversity component and the total multiuser interference, respectively, in the correlation input. Assuming coherent detection, perfect channel knowledge and

MRC with M fingers ($M \leq L$) are used, the decision variable of the k -th data symbol and for the i -th user is given by

$$\begin{aligned}\widehat{b}_i(k) &= \sum_{m=1}^M h_m^*(k) \int_{(k-1)T_b+\tau_m}^{kT_b+\tau_m} r(t) c_i^*(t - \tau_m) dt = \\ &= D'_i + M'_i + S_i + N_i,\end{aligned}\quad (2.70)$$

where D'_i , M'_i , S_i , and N_i are the desired signal component, the total multiuser interference, the self-interference and the noise term, respectively, in the output of the correlator. Furthermore, each component can be written as

$$D'_i = A_i b_i(k) T_b \sum_{m=1}^M |h_m(k)|^2, \quad (2.71)$$

$$\begin{aligned}M'_i &= \sum_{\substack{u=1 \\ u \neq i}}^U \sum_{m=1}^M \sum_{\substack{l=1 \\ l \neq m}}^L A_u h_m^*(k) h_l(k) [b_u(k - 1 - \delta_{l,m}) R^{(u,i)}(\tau_{l,m} - \delta_{l,m} T_b) + \\ &+ b_u(k - \delta_{l,m}) \widehat{R}^{(u,i)}(\tau_{l,m} - \delta_{l,m} T_b)],\end{aligned}\quad (2.72)$$

$$\begin{aligned}S_i &= \sum_{m=1}^M \sum_{\substack{l=1 \\ l \neq m}}^M A_i h_m^*(k) h_l(k) [b_i(k - 1 - \delta_{l,m}) R^{(i,i)}(\tau_{l,m} - \delta_{l,m} T_b) + \\ &+ b_i(k - \delta_{l,m}) \widehat{R}^{(i,i)}(\tau_{l,m} - \delta_{l,m} T_b)],\end{aligned}\quad (2.73)$$

$$N_i = \sum_{m=1}^M h_m^*(k) \int_{(k-1)T_b+\tau_m}^{kT_b+\tau_m} n(t) c_i^*(t - \tau_m) dt, \quad (2.74)$$

where $\tau_{l,m} = \tau_l - \tau_m$, $\delta_{l,m} = \lfloor \tau_{l,m}/T_b \rfloor$, $R^{(u,i)}(\tau) = \int_0^\tau c_u(t - \tau) c_i(t) dt$ and $\widehat{R}^{(u,i)}(\tau) = \int_\tau^{T_b} c_u(t - \tau) c_i(t) dt$ for $0 \leq \tau \leq T_b$, $l = 1, 2, \dots, L$ and $u = 1, 2, \dots, U$, are the continuous-time partial cross correlations [PUR77].

If we read the mathematical form of the previous terms analytically [KO01], it is clear that the multiuser interference represented by the M_i (Eq. 2.69) term has a simpler structure than M'_i (Eq. 2.72). Also, M'_i includes $UM(M-2)$ more interference terms than M_i . Thus a PIC scheme using as reference the received signal ($r(t)$), has a lower complexity and is more appropriate for terminal implementations, where the computational power is limited. However, when the spreading factor is low, the IPI phenomena are not negligible and can reduce its performance.

IPI can not be considered in the received signal (interference of the lower paths to the stronger path) because the desired diversity gain will be lost. Thus, IPI can be considered only in the correlation output, due to the cross-correlation between the channel paths and the fingers of the i -th RAKE scheme. The proposed algorithm, in the next subsection, is based on these important observations.

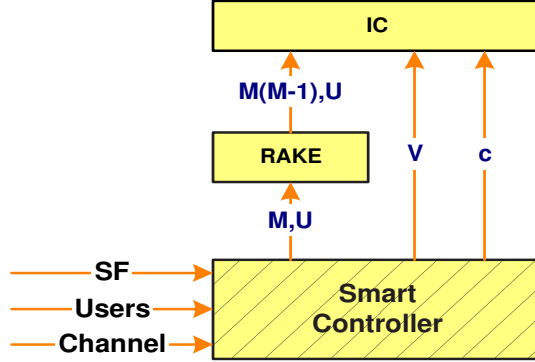


Figure 2.36: The proposed reconfigurable PIC scheme for DS/CDMA downlink connections.

2.7.3 The proposed receiver

Fig. 2.36 presents the general structure of the proposed reconfigurable detector. It consists of a processing block (RAKE+IC), which performs the required computations, and the SPV block, which supervises the radio and performs the necessary optimizations. The structure of the processing block is shown in Fig. 2.37. It is a PIC scheme with V stages. The basic difference for a conventional multi-stage PIC using the correlation input, is the consideration of the self-interference. In each stage of the proposed algorithm, after the MAI mitigation, the output of every correlator is fed into a parallel IPI canceler to produce “cleaner” signals. In particular, in the initial stage, the received signal is sampled and fed into a bank of RAKE receivers, one for each user. The RAKE receivers perform maximal ratio diversity combining for each user and dump soft decisions of the data symbols, $\hat{b}_u^{(0)}$, in the first stage of the proposed PIC scheme, for further processing.

In the first part of each stage, the PIC performs MAI mitigation. Fig. 2.38 schematically presents this process. The soft decisions from the $(j-1)$ -th stage, $\hat{b}_u^{(j-1)}$, are first passed through decision devices that produce an estimation of the transmitted signals. These decision devices use a well-defined decision function $f_{dec}()$, which has been presented in our previous discussion. The estimated transmitted signals are spread by the corresponding spreading codes and pass through the estimated channel. The resulting signals are combined appropriately and subtracted from $r(t)$. Ideally, if we can correctly estimate the transmitted signal and the channel, the resulting signal of this operation will be simply equal to $D_i + n(t)$. Finally, a bank of RAKE combiners is used to produce new “cleaner” decision variables. The operations of the j -th MAI-PIC for the i -th user can be formulated by the following equations

$$\tilde{b}_i^{(j-1)}(k) = f_{dec} \left(\hat{b}_i^{(j-1)}(k) \right), \quad (2.75)$$

$$r_i^{(j-1)}(t) = r(t) - \sum_{\substack{u=1 \\ u \neq i}}^U \sum_{k=1}^K \sum_{m=1}^M A_u h_m(k) \tilde{b}_u^{(j-1)}(k) c_u(t - kT_b - \tau_m), \quad (2.76)$$

$$\tilde{\hat{b}}_i^{(j-1)}(k) = \sum_{m=1}^M h_m^*(k) \int_{(k-1)T_b + \tau_m}^{kT_b + \tau_m} r_i^{(j-1)}(t) c_i^*(t - \tau_m) dt \quad (2.77)$$

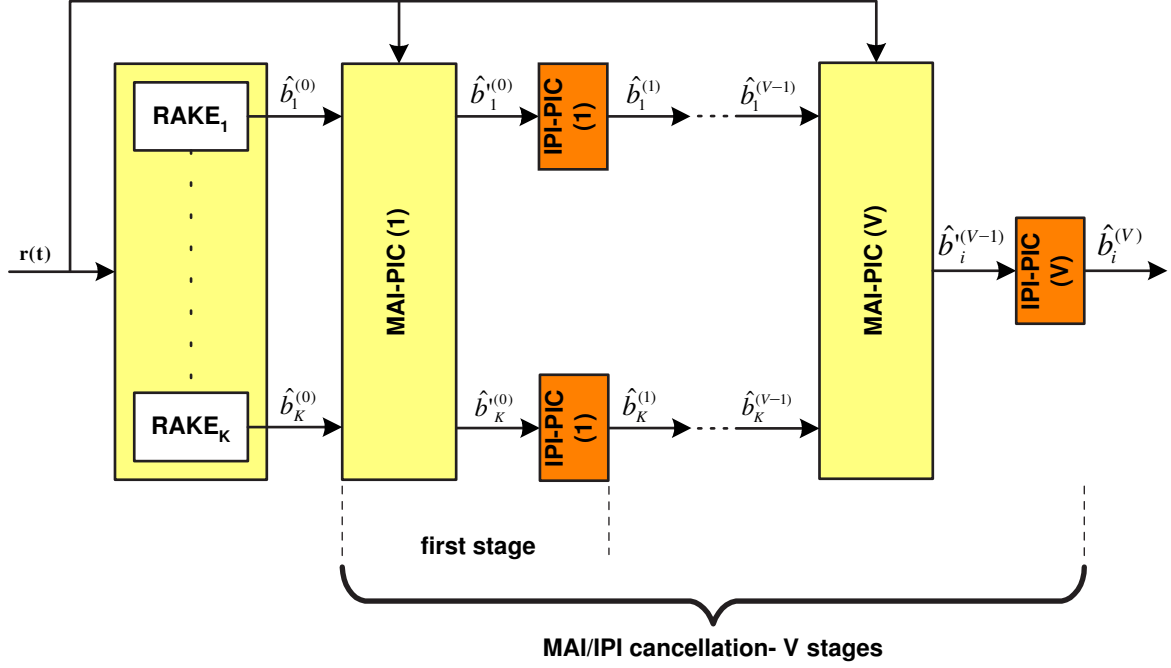


Figure 2.37: Structure of the proposed multi-stage PIC scheme for the i -th user.

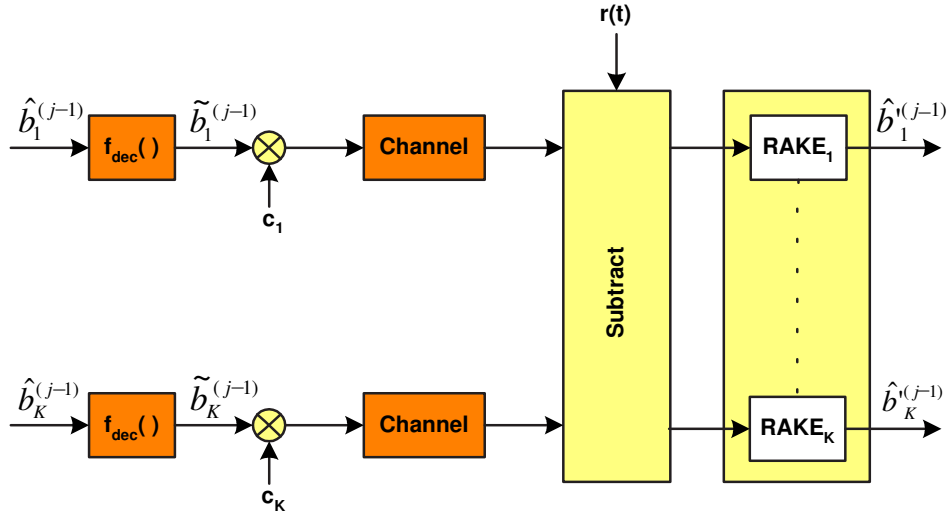
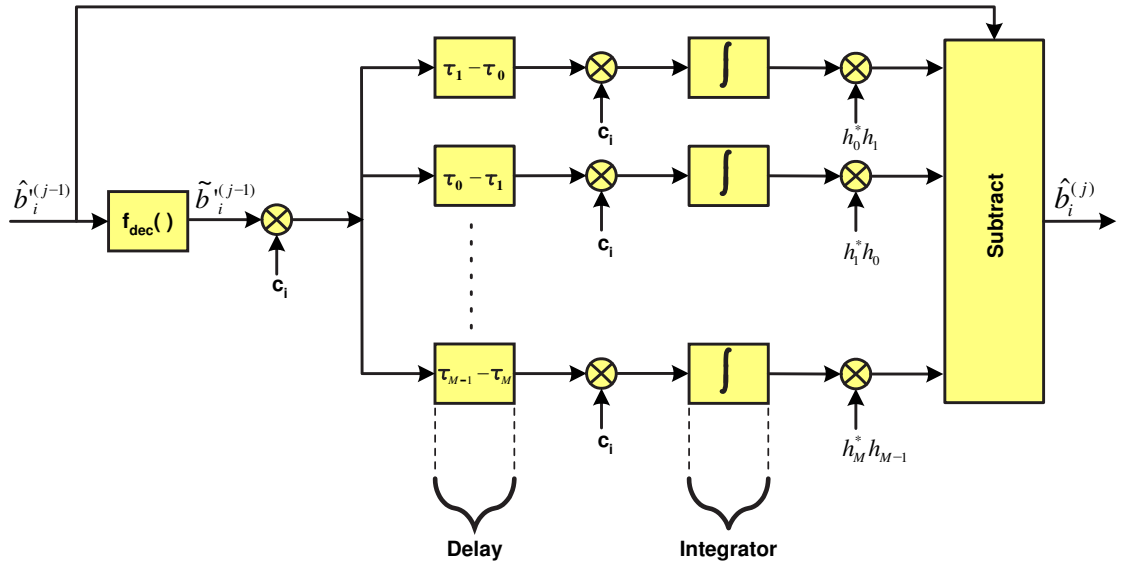
In the second part of each stage, the PIC performs IPI mitigation. Fig. 2.39 schematically presents this process. The output of each RAKE receiver subjected to IPI is used to reconstruct the terms of this self-interference. For the reconstruction, the interference generator simulates the operations of the transceiver chain. Thus, it spreads the estimated transmitted signal, creates the corresponding delays and channel degradations, and finally applies the matched filtering. To generate an estimation of the transmitted signal, the same decision device used for the MAI-PIC is used. Ideally, if we can correctly estimate IPI, the decision statistic will be simply equal to $W_i + N_i$. The following set of equations formulates the proposed IPI cancelation process,

$$\tilde{b}_i'^{(j-1)}(k) = f_{dec} \left(\hat{b}_i'^{(j-1)}(k) \right), \quad (2.78)$$

$$\hat{b}_i^{(j)}(k) = \tilde{b}_i'^{(j-1)}(k) - \hat{I}_i^{(j-1)}(k), \quad (2.79)$$

where $\hat{I}_i^{(j-1)}(k)$ denotes the reconstructed IPI for the k -th symbol, based on $\hat{b}_i'^{(j-1)}(k)$ decision.

In order to simplify the optimization task of the corresponding SPV, we suppose that the available computational power is sufficient to support the optimal configuration of the proposed detection scheme. The optimal configuration is the one which deals with all channel paths, users, interference terms, has the decision threshold which optimizes the performance and uses the minimum number of IC stages. As for the IC parameters, we suppose that their appropriate values (c, V) are provided to the SPV by a simulation study.

Figure 2.38: Structure of the j -th MAI-PIC stage.Figure 2.39: Structure of the j -th IPI-PIC stage.

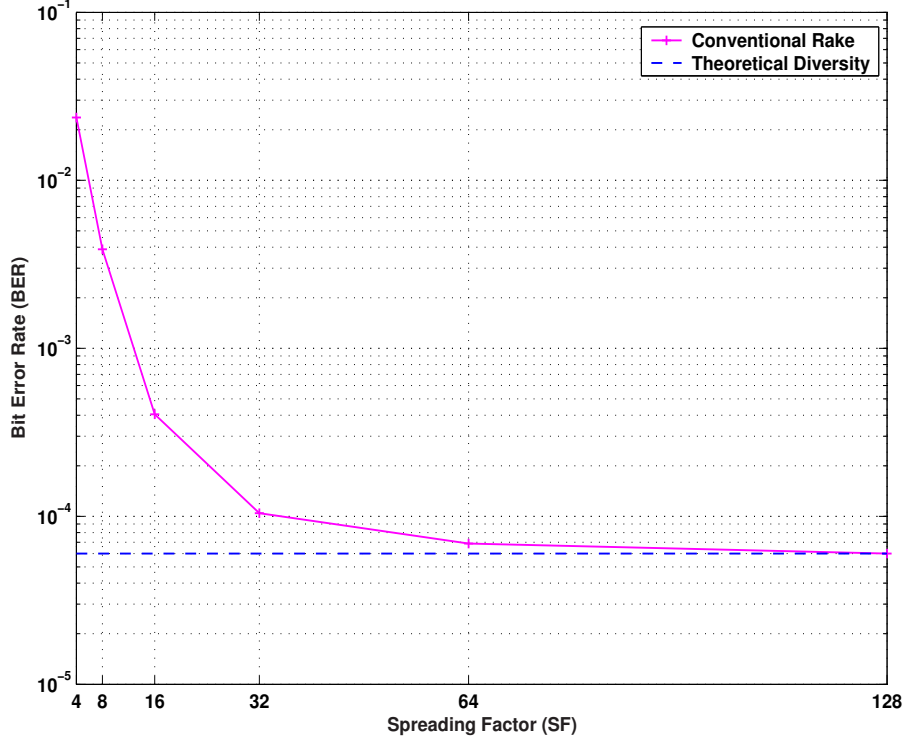


Figure 2.40: The impact of SF on the performance of a conventional RAKE receiver. $E_b/N_0 = 16\text{dB}$, $V = 5$ stages and $U = 1$ users.

2.7.4 Numerical results

Numerical results of the BER performance of the proposed multi-stage PIC are obtained through computer simulations. The simulation environment is based on the downlink specifications of the FDD-UMTS standard. The radio channel has three independent Rayleigh-fading paths with average power $\mathbf{h} = [0 \ 0 \ 0]\text{dB}$ and delays $\tau = [0 \ 3 \ 6]T_c$. As the focus of investigation was interference cancellation, perfect channel estimation and power control was assumed.

Fig. 2.40 shows the impact of the spreading factor on the BER performance. We consider a single-user environment with $E_b/N_0 = 16\text{dB}$, where the only reason for performance degradation is IPI. From this figure, we can see that when the SF is higher than 64, the IPI phenomena are negligible and do not impact the system performance. However, for a lower SF, the combination of spreading and scrambling codes gives a poor autocorrelation property and the performance is reduced. Thus, when $N \geq 64$, the proposed algorithm operates as a conventional PIC and IPI cancellation is switched off.

The impact of the decision function threshold for different numbers of stages (V) on the performance of the proposed PIC scheme is shown in Fig. 2.41. For this simulation, the number of users is $U = 5$ to account for a highly loaded system, the Signal-to-Noise Ratio (E_b/N_0) is equal to 16dB and N is equal to 8. As reference curves, the BER performance of a single-user environment and the theoretical diversity are drawn. From the results shown, it appears that the proposed algorithm outperforms the conventional PIC algorithm (without IPI). This is to be expected, because for a low spreading factor, the self-interference is very severe and its consideration can improve the achieved performance. As for the optimal decision function

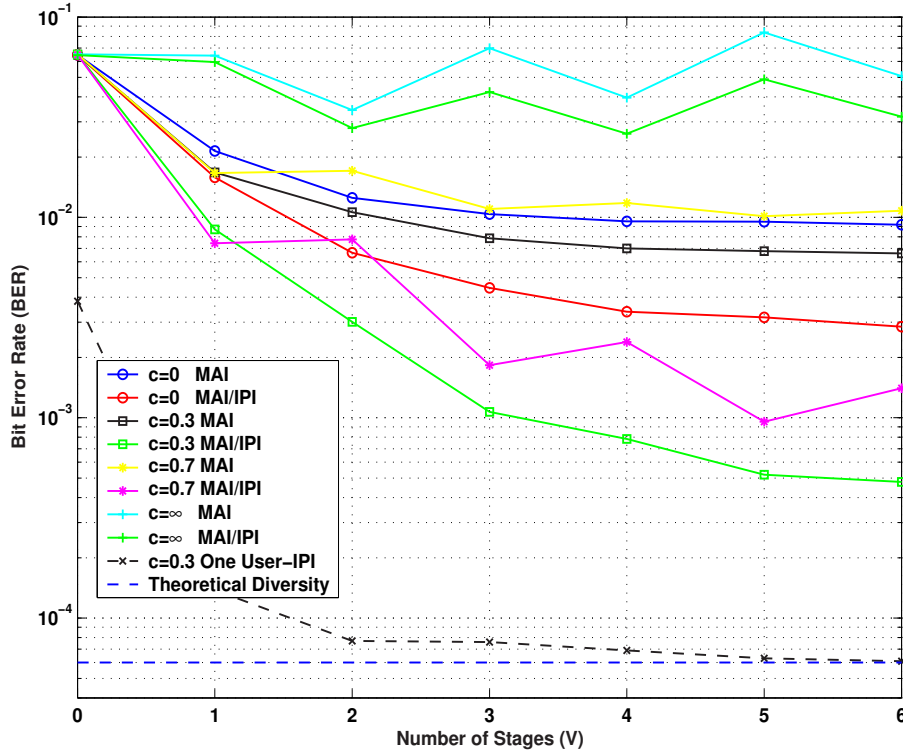


Figure 2.41: The impact of the decision function threshold on the PIC performance. $E_b/N_0 = 16\text{dB}$, $V = 5$ stages, $N = 8$ and $U = 5$ users. The thresholds are $c = 0.0$, 0.3 , 0.7 and ∞ , respectively.

threshold, we compared different possible values. In this figure we present the most representative ones, $c = 0$ (hard decision), $c = 0.3$, $c = 0.7$ and $c = \infty$ (linear- no threshold). The proposed PIC with $c = 0.3$ has the smallest distance to the single-user BER curve and it is the most appropriate one. Finally, we can say that as the number of stages V increases, the performance is improved. However, after a certain number of stages ($V = 4$) the algorithm converges. Moreover, for the linear and for the $c = 0.7$ case, the algorithm does not converge monotonically. This is known as divergence/convergence with ping-pong [RAS00].

Fig. 2.42 depicts the BER performance as a function of E_b/N_0 . Here the number of users is also $U = 5$, $N = 8$ and V is equal to 5 stages. We compare the performances of a conventional PIC (without IPI) and of the proposed scheme for a hard decision and the optimal threshold $c = 0.3$. As reference curves, the BER performance of a conventional RAKE receiver, the BER performance for a single-user environment, and the theoretical diversity are drawn. From the results, it is easily seen that the proposed improved PIC outperforms the conventional PIC for both thresholds. Moreover, the achieved gain for the case of the optimal threshold is higher than for the case of the hard decision. This is to be expected, because for a threshold equal to 0.3, the propagation errors from the MAI and IPI mitigation are limited. Finally, we can see that as E_b/N_0 increases, the BER for all PIC schemes saturates due to the MAI and the IPI.

Fig. 2.43 shows the BER versus the number of users in the system, with $E_b/N_0 = 16\text{dB}$, $V = 5$ stages and $N = 8$. From this figure, we can see that BER increases as the number of users increases, but PIC with IPI mitigation at the correlation output greatly improve performance.

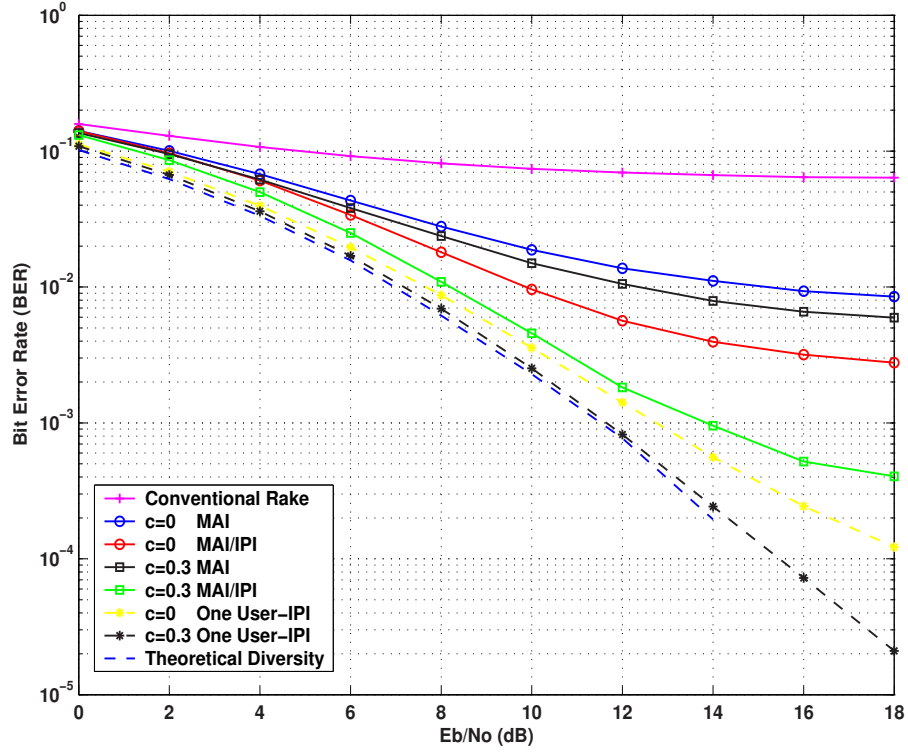


Figure 2.42: BER performance versus E_b/N_0 . $V = 5$ stages, $N = 8$, $U = 5$ users and the thresholds are $c = 0.0$ and 0.3 .

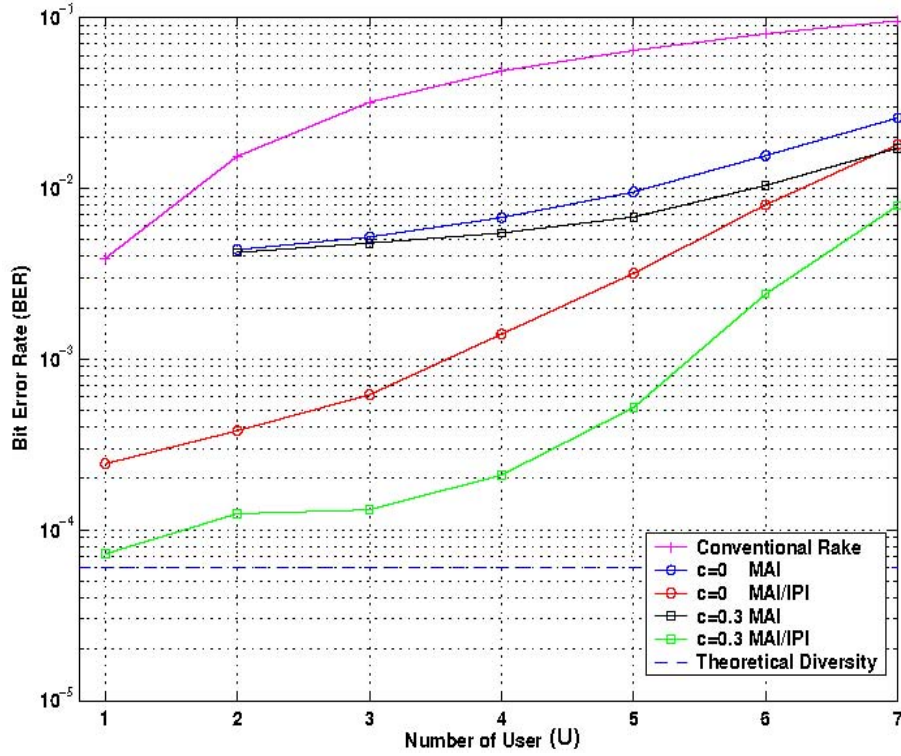


Figure 2.43: BER performance versus number of users. $E_b/N_0 = 16\text{dB}$, $V = 5$ stages, $N = 8$ and the thresholds are $c = 0.0$ and 0.3 .

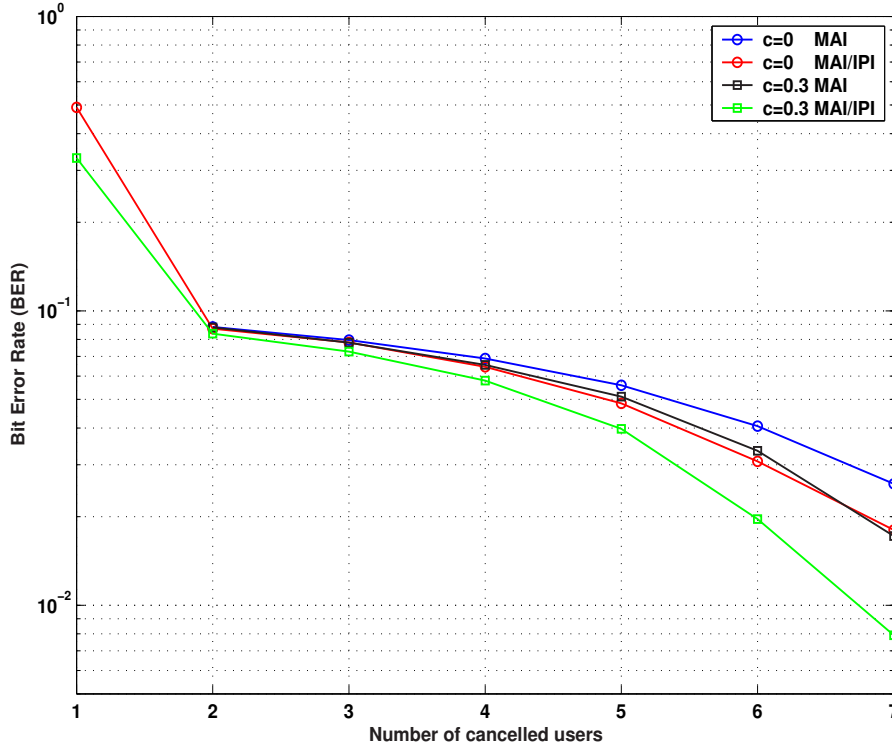


Figure 2.44: BER performance versus number of canceled users. $E_b/N_0 = 16\text{dB}$, $V = 5$ stages, $N = 8$, $U = 7$ users and the thresholds are $c = 0.0$ and 0.3 , respectively.

The impact of limiting the number of cancelations on the BER performance of the proposed PIC scheme is depicted in Fig. 2.44. Here the simulation parameters are $E_b/N_0 = 16\text{dB}$, $U = 7$ users, $V = 5$ stages and $N = 8$. Note that as the number of canceled users increases, the BER decreases and the gain of the proposed PIC becomes higher. Hence, there is a trade-off of complexity versus performance. Faster processing implies a higher number of allowable cancelations, which implies better BER performance.

2.8 CDMA2000 for a satellite environment

2.8.1 Algorithm motivation

Algorithmic reconfigurability does not concern only the design of auto-intelligent receivers which adapt their functionalities for achieving performance and computational optimizations. It can also support transmitters that adapt their transmission schemes according to real-time parameters, which are given by the receiver via a reciprocal logical channel.

In contrast with the previous application examples, which deal with the algorithmic reconfigurability concept from the receiver point of view, this application example concerns both transmitter and receiver. More specifically, we study the two communication modes of the CDMA2000 (1X and 3X) standard for a satellite downlink environment, and we propose a reconfigurable system which always uses the CDMA2000 mode, which optimizes jointly performance and computational complexity.

As we presented in the Chapter 1, CDMA2000 has two operation modes:

- The first mode is referred to as CDMA2000 1X. It has a chip rate of 1.2288 Mcps and a bandwidth of 1.25 MHz. The transmission technology for the forward and reverse links is single-carrier DS-CDMA.
- The second mode is referred to as CDMA2000 3X. For the downlink of this mode, the transmission technology is MC-CDMA with three consecutive 1.25 MHz carriers, where each carrier has a chip rate of 1.2288 Mcps. For the reverse link, we have a DS-CDMA transmission with chip rate of 3.6864 Mcps. This mode is added to the CDMA2000 standard, in order to increase the user's bandwidth to a level similar to that of the UMTS, which is the European standard for 3G. Thus, we can see this mode as a first step for the convergence of the two most important 3G standards.

In a traditional CDMA2000 system, the selection of the downlink transmission mode is a static decision, which is defined by the QoS of the service under consideration. More specifically, for a service with high QoS requirements, a multi-carrier transmission (CDMA200 3X) is used, otherwise a single-carrier transmission (CDMA2000 1X) is selected. In this application example, the transmission mode of the CDMA2000 standard is supposed as a dynamic parameter, which can be changed in real-time. According to the operational environment, the proposed reconfigurable CDMA2000 system always uses the transmission mode which jointly optimizes performance and computational requirements. We note that the 3X mode uses three different carriers and thus has a complexity (transmitter and receiver) three times higher than that of the 1X mode. Thus, the computational optimization refers to the use of the 1X mode, instead of the multi-carrier mode.

In order to simplify the operational environment and focus on the transmission mode, a simple LOS satellite environment is used in the following analysis. However, the operational environment can be generalized to include conventional terrestrial channel models.

2.8.2 Problem formulation and simulation environment

The transmitter

The DS-CDMA and MC-CDMA forward link systems are illustrated in Fig.'s 2.45 and 2.46, respectively. We consider the forward link of an isolated beam and we study the performance of one Forward Supplemental Channel (F-SCH) for a single-user environment. For the mode 1X (DS-CDMA), we study a service with a data rate of 153.6 kbps using the Radio Configuration 4 (RC-4), and for the mode 3X (MC-CDMA), a service with a data rate of 115.2 kbps using the Radio configuration 9 (RC-9) [GAR00]. These two simulated services have comparable data rates, use the same CC and their basic difference is the transmission technology. We note that a bit rate equal to 115.2 kbps is supported by the mode 1X, with a bit rate equal to 153.6 kbps by using the rate matching processing. Thus the mode comparison concerns the same bit rate of 115.2 kbps.

For the F-SCH channel of the DS-CDMA system, the binary generator generates 3072 equiprobable bits for a 20 ms frame with a data rate of 153.6 kbps. For the MC-system, the binary generator generates 2304 bits for a 20 ms frame with a data rate of 115.2 kbps. According to the RC-4 and RC-9, a CC is used for these rates with constraint length $\hat{K} = 9$ and code rate $R = 1/2$.

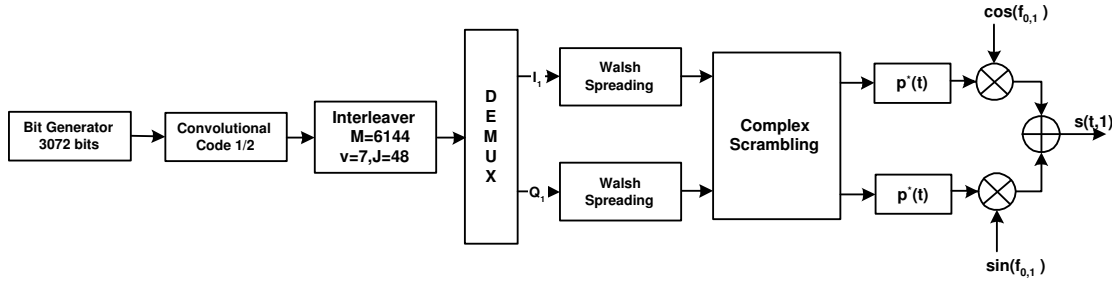


Figure 2.45: Transmission chain for mode 1X.

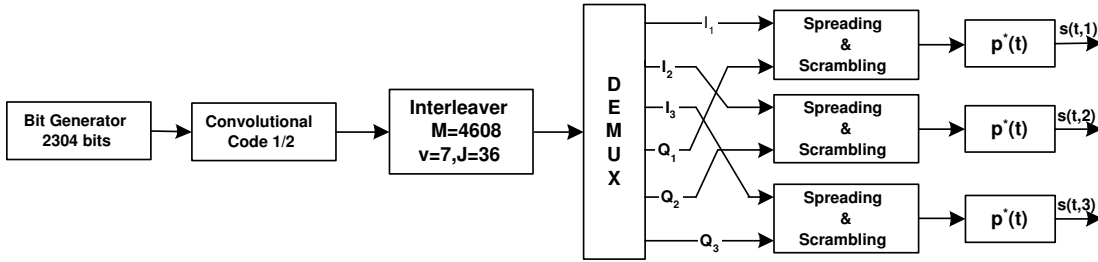


Figure 2.46: Transmission chain for mode 3X.

The output of the convolutional code is interleaved with a block interleaver, whose input symbols are written sequentially at addresses 0 to the block size minus one ($K - 1$). The even interleaved symbols (i is even) are read out in permuted order from address A_i , as follows:

$$A_i = 2^\nu \left[\frac{i}{2} \bmod J \right] + BRO_\nu \left(\left\lfloor \frac{i}{2} / J \right\rfloor \right) \quad (2.80)$$

The odd output symbol position (i is odd) is given by:

$$A_i = 2^\nu \left[\left(K - \frac{i+1}{2} \right) \bmod J \right] + BRO_\nu \left(\left\lfloor \left(K - \frac{i+1}{2} \right) / J \right\rfloor \right), \quad (2.81)$$

where K is the block size, $i = 0, \dots, K - 1$, $\lfloor x \rfloor$ indicates the largest integer less than or equal to x , $BRO_\nu(y)$ indicates the bit-reversed ν -bit value of y , ν and J are the interleaver parameters. Table 2.11 summarizes the interleaver configuration for the two considered data rates.

	Block size (K)	ν	J
153.6 kbps	6144	7	48
115.2 kbps	2304	7	36

Table 2.11: Interleaver parameters.

The interleaver output is demultiplexed by a symbol demultiplexer (Demux). The Demux function in both transmission schemes distributes input symbols sequentially from the top to the bottom output paths. The signal on each carrier is orthogonally spread by the appropriate Walsh code function, in such a manner as to maintain a fixed chip rate of 1.2288 Mcps per carrier. For

mode 1X and according to the RC-4, the Walsh Code is $c_{8,i}$, where $0 < i \leq 8$. For mode 3X and according to the RC-9, the Walsh Code is $c_{32,i}$, where $0 < i \leq 32$. The next operation is the complex scrambling, where the resultant signals are multiplied by a complex valued PN sequence. This sequence is taken from two independent M -sequences with period 2^{15} and a rate of 1.2288 Mcps. Finally, for the signal on each carrier there is baseband filtering and frequency modulation. The transmitted signal can be described as

$$\begin{aligned}
 s(t, i) = & \sum_{k=0}^{K-1} \sum_{n=0}^{N-1} [b^I(k, i)c(n)\alpha^I(kN + n) - b^Q(k, i)c(n)\alpha^Q(kN + n)]p(t - kT_c)\cos(f_{0,i}t) \\
 & + \sum_{k=0}^{K-1} \sum_{n=0}^{N-1} [b^I(k, i)c(n)\alpha^Q(kN + n) + b^Q(k, i)c(n)\alpha^I(kN + n)]p(t - kT_c)\sin(f_{0,i}t)
 \end{aligned} \tag{2.82}$$

where $b^I(k, i)$ and $b^Q(k, i)$ are, respectively, the real and imaginary parts of the k -th data symbol on the i -th carrier; $c(n)$ is the n -th element of the Walsh Code for the F-SCH traffic channel; $\alpha^I(k)$ and $\alpha^Q(k)$ are the real and imaginary parts, respectively, of the k -th scrambling code element; $p(t)$ is the impulse response of the root raised cosine pulse-shaping filter with rolloff factor 0.22, T_c is the chip interval, N is the spreading factor and K the symbol observation window.

The total transmitted signal for DS-CDMA is transmitted on one carrier, $s(t) = s(t, 1)$, and for MC-CDMA it is defined as a vector of three signals (one signal for each carrier), $s(t) = [s(t, 1) \ s(t, 2) \ s(t, 3)]$.

The channel model

The channel between the satellite and the terminal is characterized by the existence of a direct component. The direct component is received through a line-of-sight (LOS) path, hence it is subject to free-space attenuation, Faraday rotation and scintillation due to the ionosphere, and shadowing [KAR99]. For approximating these phenomena, we model the channel as a simple flat (non-selective) channel with a Ricean distribution [MEH99]. The Rice factor (the power ratio between LOS component and the diffuse component) typically ranges between 5 to 15dB.

The received signal can be expressed as:

$$r(t, i) = \alpha(t, i)e^{-j\phi(t, i)}s(t, i) + n(t, i) \tag{2.83}$$

where $r(t, i)$ is the received signal in the i -th carrier, $n(t, i)$ and $\alpha(t, i)e^{-j\phi(t, i)}$ are, respectively, the additive white gaussian noise and the complex fading factor of the i -th carrier.

In our simulation environment, we consider that the channel is constant during the symbol period. The coefficients $\alpha(t, i)$ follow the Ricean probability density function (PDF), which can be expressed as:

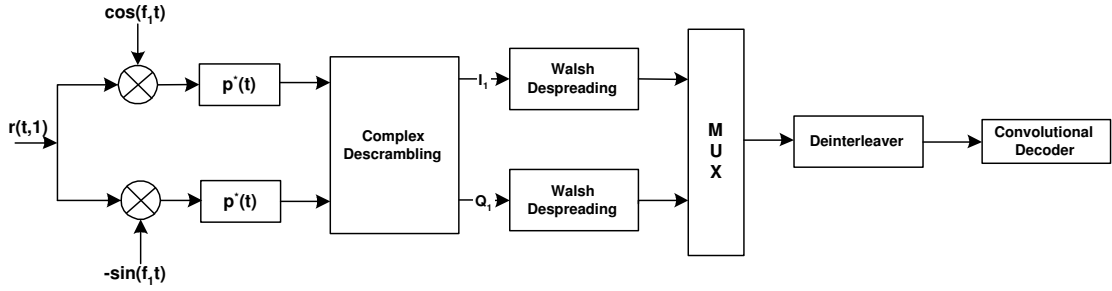


Figure 2.47: Reception chain for mode 1X.

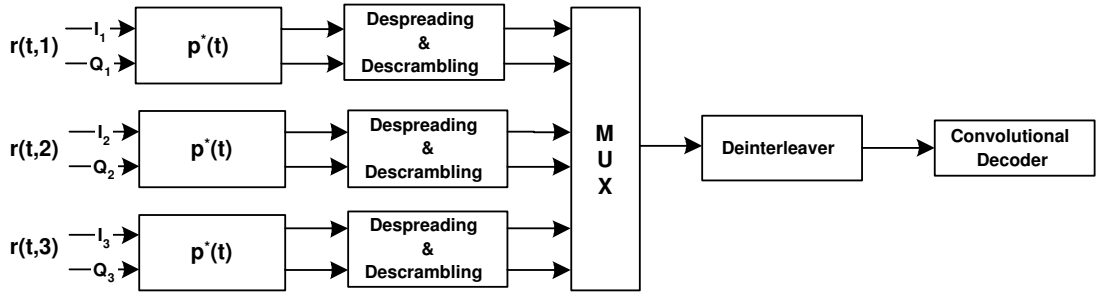


Figure 2.48: Reception chain for mode 3X.

$$P_{\alpha}(\alpha|p_{od}) = \frac{\alpha}{\bar{p}_{od}} e^{-\frac{\alpha^2 + \mu^2}{2\bar{p}_{od}}} I_0\left(\frac{\alpha\mu}{\bar{p}_{od}}\right), \quad (2.84)$$

$$R_f = \frac{\mu^2}{2\bar{p}_{od}}, \quad (2.85)$$

$$p_{od} = (R_f + 1)\bar{p}_{od}, \quad (2.86)$$

where $I_0()$ is the modified Bessel function of the first kind and zeroth order, μ is the peak value of the specular radio signal, \bar{p}_{od} is the average power of the scattered signal, R_f is the Rice factor which depends on the ratio of the signal power from the dominant signal path relative to that of the scattered signal, and p_{od} is the local mean power.

The receiver

Fig.'s 2.47 and 2.48 show the receiver chain for the two modes of CDMA2000. The basic difference from a terrestrial receiver is the absence of the RAKE receiver. In the satellite environment, there are no multipath effects, and so the RAKE receiver, which combines the energies of different paths, is not used.

For mode 1X, the receiver is a single finger receiver (MF) [PRO95] as in the classic narrowband systems, and for the mode 3X the receiver has three parallel single finger receivers, one for each carrier.

For the channel estimation, the standard supports a Common Pilot Channel (F-PICH), one for each carrier, which is continuously broadcast in order to provide coherence detection and

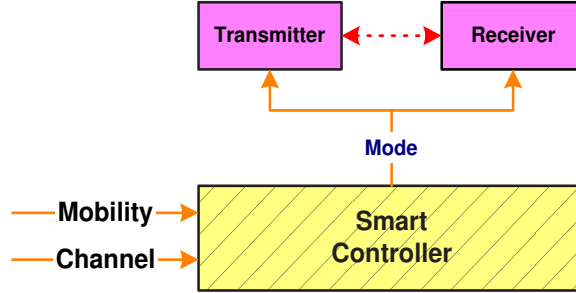


Figure 2.49: The proposed reconfigurable CDMA2000 transceiver.

synchronization [RAT00]. The common pilot is an all-zero sequence prior to Walsh spreading with Walsh code $c_{64,1}$, and it is shared by all physical channels and users. In this study, the estimation problem was not considered and a perfect channel estimation is assumed.

2.8.3 The proposed system

The proposed reconfigurable CDMA2000 system dynamically selects the transmission/reception mode which jointly optimizes performance and computational complexity. The operational environment is a satellite downlink channel, represented by a LOS Ricean fading path. In contrast with traditional CDMA2000 systems, which select the mode according to the QoS under consideration, the proposed system can change the mode at run-time, according to the real-time parameters of the operational environment. More specifically, the real-time parameters of interest are the Ricean factor, the Signal-to-Noise Ratio (E_b/N_0), and the velocity of the receiver. These parameters characterize the operational environment and have different impacts on the detection efficiency of each CDMA2000 mode. The knowledge of these parameters by the transmitter, as well as their related impacts on the detection performance of each mode, can be used for the real-time selection of the appropriate mode. The proposed intelligent transmitter selects the mode that gives the best possible performance. For the case where the two CDMA2000 modes have similar performances, it selects the 1X mode, which has transmitter and receiver complexities, three times lower than those of the multi-carrier mode. Fig. 2.49 presents the reconfigurable satellite transceiver.

It is obvious that the proposed reconfigurable scheme is based on a real-time knowledge of the receiver operational environment by the transmitter. This is possible by the use of a reciprocal logical channel (from the receiver to the transmitter), which continuously transmits the operational receiver conditions.

For the case of a satellite environment, the consideration of this reciprocal logical channel has a practical difficulty due to the high distance between transmitter and receiver. However, the satellite operational environment has been selected just for reasons of simplicity, and the considered ideas can be generalized for a conventional terrestrial multi-path environment.

2.8.4 Numerical results

Computer simulations were carried out to compare the performances of the two modes of CDMA2000 for a satellite environment. Two flat fading channels were constructed by using

	Mode 1X	Mode 3X
Trans. Technology	DS-CDMA	MC-CDMA
Data Rate	153.6 kbps	115.2 kbps
Processing Gain	8	32
Code	Convolutional Code 1/2	
Channel	Rice factor $R_f=5$, 15dB	
Estimation Channel	Perfect	

Table 2.12: Simulation parameters.

a Ricean distribution channel model. The first channel has a Rice factor $R_f = 5$ dB and corresponds to a satellite environment with a poor LOS signal. The second channel has a Rice factor $R_f = 15$ dB and corresponds to a satellite environment with a strong LOS signal.

For mode 1X, we simulated a service with a data rate of 153.6 kbps using the RC-4 and, for mode 3X, a service with a data rate of 115.2 kbps using the RC-9. Table 2.12 summarizes the simulation parameters.

In the following, we present the performances of the two modes for different mobilities, assuming a perfect channel estimation.

Fig. 2.50 presents the BER performances of the two modes, for the two considered channels and for a mobile speed equal to 3km/h, which corresponds to a pedestrian user. For this case, we can see that for the channel with $R_f = 5$ dB, the MC-CDMA has a gain of 2dB over the DS-CDMA, in the high SNR's. However, in the second channel ($R_f = 15$ dB), the two modes have similar performances.

The curves in Fig.'s 2.51 and 2.52 represent the performances (BER) of the two modes for mobile speeds 50km/h and 130km/h, respectively. From these curves, we can firstly see that the increase of speed improves the performance of the two receivers. This is to be expected, because for high speeds, the channel is less correlated, so the interleaver randomizes the received symbols in a better way, which enables the CC 1/2 to have a better performance. As far as the two modes are concerned, our observations are similar for the two speeds. The DS-CDMA has a better performance than MC-CDMA for both channels. In the first channel ($R_f = 5$ dB), the DS-CDMA performs better than MC-CDMA by about 1.5dB in the high SNR's (10dB). In the second channel ($R_f = 15$ dB), the two systems perform about the same.

According to our results, when the channel is characterized by a strong LOS signal, the two communication modes have the same performance. Moreover, we can say that MC-CDMA has a complexity three times greater than DS-CDMA due to the use of three different carriers. Thus, for the case of a strong LOS link, the mode 1X is the most appropriate. However, when the channel has a poor LOS signal, mode 1X is the appropriate solution for high speeds and mode 3X for small speeds. The proposed reconfigurable link supposes that there is a reciprocal channel which can inform the transmitter about the terminal velocity and the channel statistics. These informations are used by the SPV to select the communication mode.

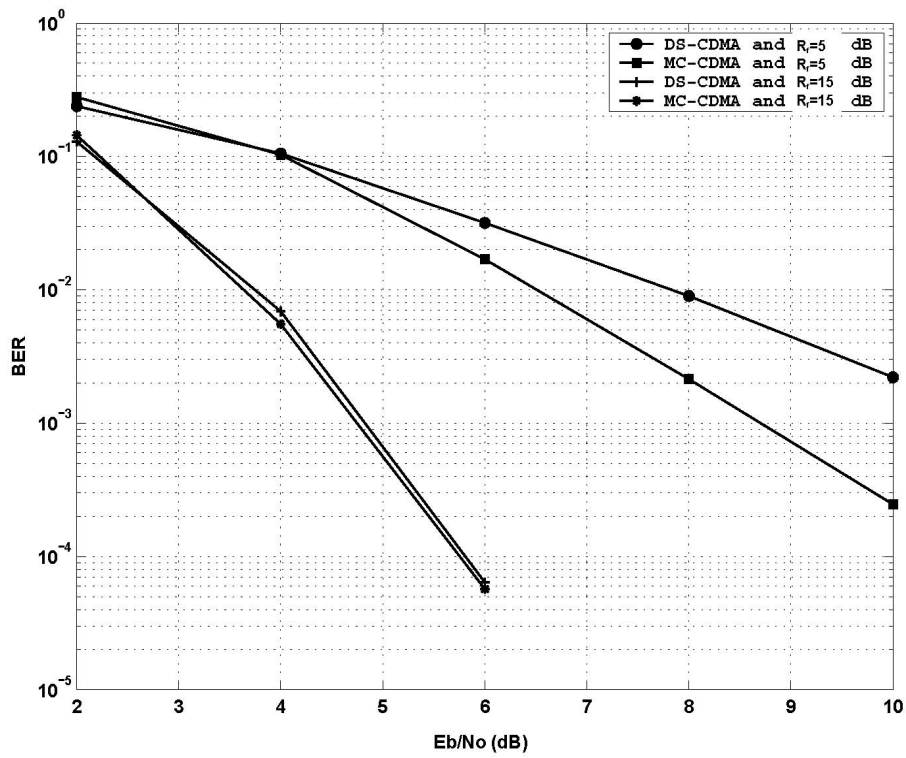


Figure 2.50: Comparison of DS-CDMA and MC-CDMA for a receiver speed equal to 3km/h.

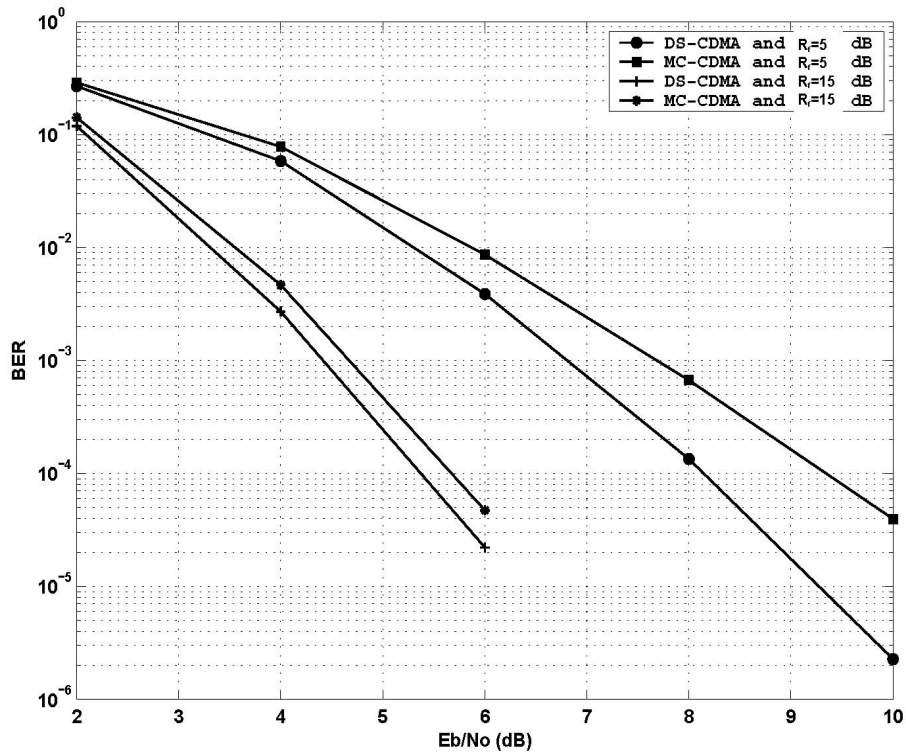


Figure 2.51: Comparison of DS-CDMA and MC-CDMA for a receiver speed equal to 50km/h.

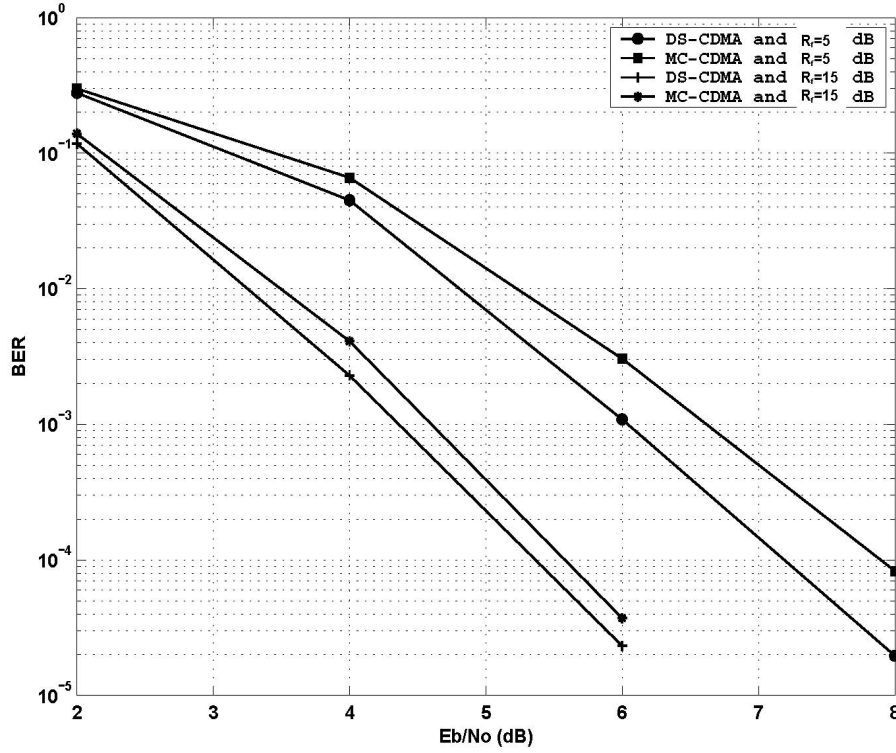


Figure 2.52: Comparison of DS-SS-CDMA and MC-SS-CDMA for a receiver speed equal to 130km/h.

2.9 Conclusion

This Chapter dealt with the algorithmic reconfigurability and presented some reconfigurable systems developed during this thesis. Thanks to algorithmic reconfigurability, each proposed system is continuously equipped with the appropriate configuration, which jointly optimizes performance and computational requirements. More specifically, this Chapter presented six algorithms which adjust their functionalities to the real-time conditions. The first algorithm was a combination of RAKE and Pilot Channel estimation, which can dynamically change the number of fingers and the size of the average estimation window. Simulation results have shown that this parameter change can significantly improve system performance. The second algorithm was a combination of RAKE and IPI-IC with a limited number of fingers. This algorithm is very efficient in an operational environment with low spreading factors and a small number of channel paths. The third algorithm was an expansion of the previous one, supposing a high number of fingers. In this case, we have shown that the proposed algorithm achieves the performances of the complicated linear equalizers, but with lower power requirements. In the fourth algorithm, we generalized the previous idea in an environment with pilot channel estimation. We have shown that the application of the IC process in both channel estimation and data detection outperforms classical detections. The fifth algorithm further generalized the operational environment and dealt with the MAI. The proposed scheme can significantly suppress the different types of interference (IPI and MAI) with lower complexity than the conventional PICs of the literature. Finally, the sixth algorithm has presented the reconfigurability concept as a general design characteristic which changes the transmitter and receiver behavior. The proposed satellite CDMA2000 system

switches between the two system modes according to the receiver speed and the quality of the communication link.

Chapter 3

Hardware Reconfigurability: Applications

3.1 Introduction

This Chapter focuses on the hardware implementation of the third developed algorithm. An efficient solution is presented, based on its computational similarities and its iterative nature. The proposed implementation is characterized by an optimization of the hardware resources used and thus it is suitable for terminal implementations with strict surface constraints. The selection of the third algorithm among the six developed ones was well suited to the demonstration of the **ASTURIES** project. However, the results can be generalized for the implementation of the other proposed algorithms.

Firstly, we try to implement the detection scheme under consideration in a commercial DSP. Despite its high degree of flexibility and computational parallelism, DSP seems to be inefficient to satisfy the time constraints imposed by the standards. Thus, a hardware implementation based on the computational similarities and the iterative calculation of the required computations is proposed. This implementation satisfies the time constraints and minimizes the area used.

This Chapter is organized as follows: The DSP implementation issues of the reconfigurable detector under consideration are presented in Section 3.2. Section 3.3 presents the implementation of this detector in a hardware device using iterative reconfigurability mapping, followed by concluding remarks in Section 3.4.

3.2 DSP implementation

Firstly, we implemented the third detection scheme of Chapter 2 in a commercial DSP. In this case the two functional configurations, the RAKE and the IC, are mapped to two different software functions which are stored in memory block. In each case the CPU executes the appropriate one with the correct value of parameters.

More specifically, the RAKE receiver and the IC are represented by software functions which have the number of paths (L) as basic parameter. We note that L tracked paths by the RAKE combination, generate $L(L - 1)$ interference terms. The real-time decision to switch on/off the IC scheme is implemented by a simple conditional (*if..then..else*) structure according to the SF

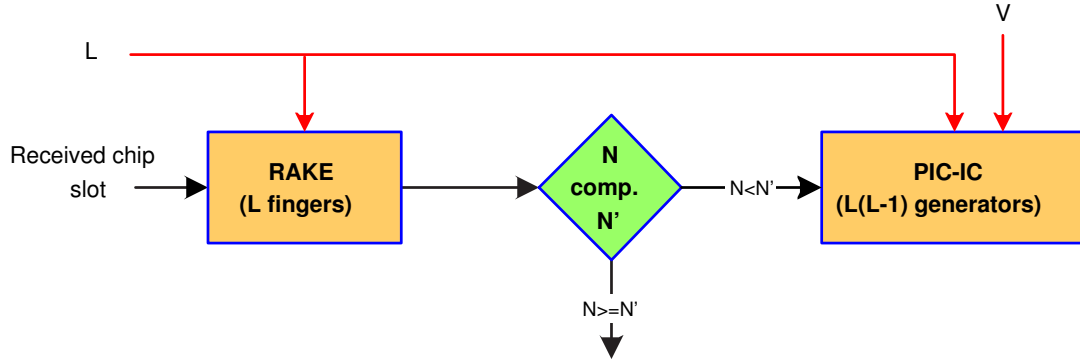


Figure 3.1: The block diagram of the DSP implementation; RAKE and IC correspond to software functions.

value. If the spreading factor N has a low value ($N < N'$), the IC scheme is applied in the output of the RAKE receiver, if not ($N \geq N'$) it is switched off. N' is the threshold SF which depends on the number of paths. Fig. 3.1 presents the block diagram of the DSP implementation.

3.2.1 TigerSHARC

The selected DSP is the ADSP-TS101S TigerSHARC from Analog Devices aimed at high-powered infrastructure projects such as 3G BSs and terminals [ANALa]. A detailed description of the DSP technology is presented in the next Chapter (4.4.2).

The first implementation of the TigerSHARC architecture is in a 0.25-micron technology, five-level metal process at 150-MHz core clock speed. It delivers 900 Mflops of single-precision floating-point performance, or 3.6 GOPS of 16-bit arithmetic performance. It sustains an internal data bandwidth of 7.2 Gbytes/sec.

This TigerSHARC implementation (the ADSP-TS001) has several mechanisms found in general-purpose computing. Some of the most significant aspects of this DSP are

- a register-based load-store architecture with a static superscalar dispatch mechanism in which Instruction-Level Parallelism (ILP) is determined prior to runtime under compiler and programmer control;
- highly parallel, short-vector-oriented memory architecture;
- support for multiple data types, including 32-bit IEEE single-precision floating point and 16-bit fixed point, with partial support for 8-bit fixed point;
- parallel arithmetic instructions for two floating-point multiply-accumulate (MAC) operations or for eight 16-bit MACs per cycle, with a Single-Instruction Stream Multiple-Data Stream (SIMD) execution mechanism;
- eight-stage, fully interruptible pipeline with a regular two-cycle delay on all arithmetic and load/store operations, and a 128-entry, four-way set-associative branch target buffer, or BTB; and

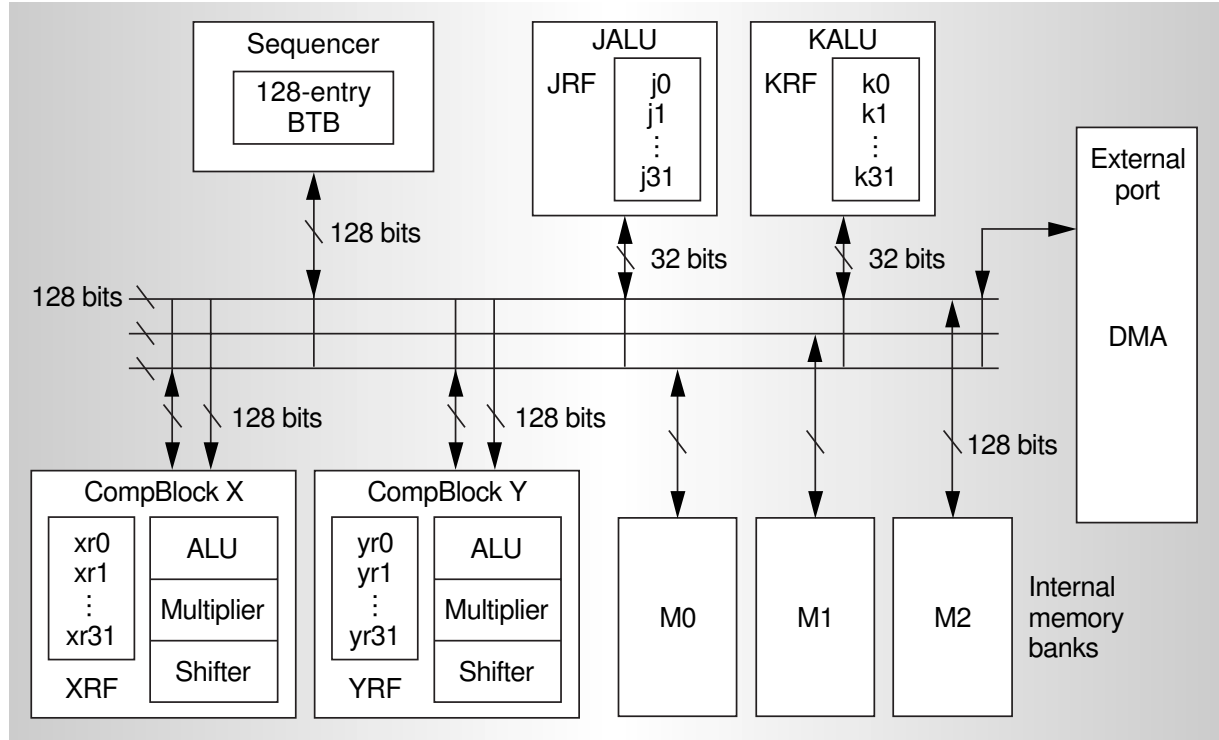


Figure 3.2: Top-level block diagram showing the major DSP subsystems and the data buses.

- 128 architecturally visible, fully interlocked registers in four orthogonal register files.

Architectural description

Fig. 3.2 shows a block diagram with the major components of the architecture as well as the primary data buses. Each of the two computation blocks on the figure's lower left (CompBlock X and Y, or CBX and CBY) consists of a 32-entry general-purpose register file (XRF and YRF), ALU, multiplier, and shifter. The two computation blocks constitute the primary data path of the processor [FRI00].

Each computation block has two 128-bit ports that connect to the three internal 128-bit buses. In the upper part of Fig. 3.2 there are two integer units (JALU and KALU, collectively called the IALU). They function as generalized addressing units; each one includes a 32-entry general-purpose register file. Although used primarily for addressing, the IALU also supports general integer arithmetic and pointer manipulation. One of four masters (JALU, KALU, sequencer, or external port) produces addresses, and one of five slaves (CBX, CBY, M0, M1, or M2) consumes them. Each data bus has an associated address bus, which for clarity is not shown here. This figure also shows three internal SRAM banks (M0, M1, and M2), each with a 128-bit connection to the bus system.

The sequencer appears in upper left of the figure, along with a 128-entry, four-way set associative branch target buffer. The sequencer, two IALUs, and the external port block are the four masters of the internal bus system and supply addresses and control to the memory banks. The three internal buses provide a direct path for instructions into the sequencer. They also

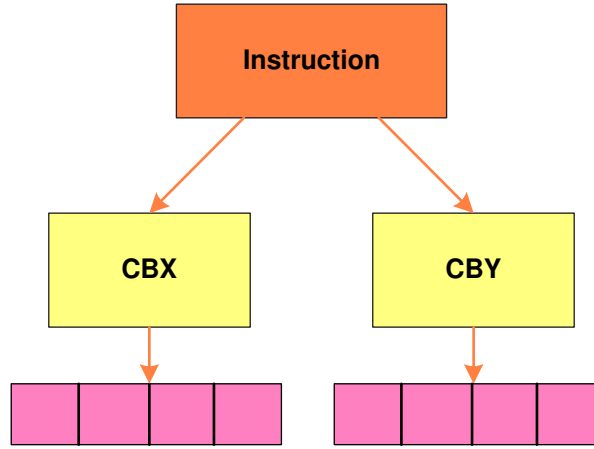


Figure 3.3: SIMD execution and subword parallel operations.

provide two independent paths that may connect each memory block with each computation block, where a path can carry up to four 32-bit words per cycle.

SIMD and subword parallelism

SIMD refers to the method by which one instruction operates on more than one data item. This DSP implements SIMD dispatch by optionally issuing one computational instruction to both CBX and CBY computational units. (All computational instructions are encoded with two bits that determine whether the target computational box is CBX, CBY, or both).

Subword parallelism is another distinct architectural technique. It increases parallelism at the data-element level by means of partitioning a processor's data path and performing more than one parallel computation on a single composite word. Subword parallelism is also sometimes referred to as multimedia extensions or packed operations.

The literature often uses SIMD to denote subword parallelism. However, although subword parallelism is a specialized form of SIMD, in this DSP it is very important to distinguish between SIMD execution and subword parallelism operations. The TigerSHARC architecture makes use of both techniques, but in different and quite distinct situations. Generally, computation at the 32-bit level is organized with SIMD execution only (that is, one 32-bit operation per computation block). However, computation at the 16- and 8-bit level is organized with SIMD execution and subword parallelism (four packed, 16-bit operations per computation block). Fig. 3.3 illustrates this conceptually, showing the two-way SIMD execution at the computation block level, and fourway subword parallelism at the subword level.

3.2.2 Time constraints

The detection scheme under consideration is a block based algorithm. This means that each sub-functionality of the algorithm has to be applied in the totality of the data block in order to continue the process. Therefore, the application of the IC scheme requires a RAKE decision for all the data symbols of the block under processing. Moreover, the application of the i -th IC stage requires the completion of the $(i - 1)$ -th IC process operations. A real-time application

generates time constraints and limits the available processing time. For the case of a block based application, the total time of processing has to be lower or equal to the block duration. Eq. 3.1 formulates this time constraint for the block based detection scheme under consideration.

$$T_{Processing} = T_{RAKE}(M) + V \cdot T_{IC}(M') \leq T_{BLOCK}, \quad (3.1)$$

where $T_{Processing}$ is the total processing time, T_{RAKE} is the processing time for the RAKE combination, T_{IC} is the processing time for one IC stage, M is the number of tracked paths, M' is the number of generated interference terms, V is the number of stages and T_{BLOCK} is the duration of a data block.

The parameter T_{BLOCK} is a constant parameter and depends on the system or standard specifications. On the other hand, the processing parameters, $T_{RAKE}(M)$ and $T_{IC}(M')$, depend on the programming techniques, the characteristics of the hardware device and the intelligence of the compiler. Also, they are a function of the parameters T_{BLOCK} , M and M' .

Fig. 3.4 shows the influence of the number of tracked paths (M) in the processing time of the RAKE combination. We can see that the processing time (T_{RAKE}) increases linearly with the number of tracked paths. If we consider the UMTS specifications in order to define the time constraint, $T_{BLOCK} = 666\mu\text{sec}$, we can see that the RAKE combination can process 8 paths at maximum. Moreover, the observed linearity is represented by two different angles, one for the odd number of paths and another one for the even number of paths. An odd number of paths corresponds to computational similarities, which can optimize the required processing time. Thus, for example, the processing time for 10 paths is lower than for 9 paths.

In the same figure, we present, also, the impact of the block size (T_{BLOCK}) over the required time of the RAKE combination. Therefore, the processing time is proportional to the block size. This conclusion can be formulated as

$$\frac{T_{BLOCK1}}{T_{BLOCK2}} = \frac{T_{RAKE1}}{T_{RAKE2}} = \beta, \quad (3.2)$$

where T_{BLOCK1} , T_{BLOCK2} are two block sizes and T_{RAKE1} , T_{RAKE2} are the corresponding processing times for the same number of channel paths, and β is a constant.

Fig. 3.5 shows the impact of the compiler intelligence on the processing time of the RAKE combination. We can see that a compilation which uses the interprocedural optimization [ANALb] generates a more efficient code and thus improves the processing time. When interprocedural optimization is used the compiler may be called again from the link phase to recompile the program using additional information obtained during previous compilations. For example interprocedural analysis identifies variables that only have one value and replaces them with constants that results in better optimization. In this figure, we can see also the influence of the DSP clock frequency on the processing time. It is obvious that an increase of the DSP frequency corresponds to a decrease of the required processing time. This can be formulated as

$$T_{RAKE} = N_{cycles} \cdot \frac{1}{f_{cl}}, \quad (3.3)$$

where N_{cycles} is the number of cycles which are required for the RAKE combination and f_{cl} is the DSP clock frequency.

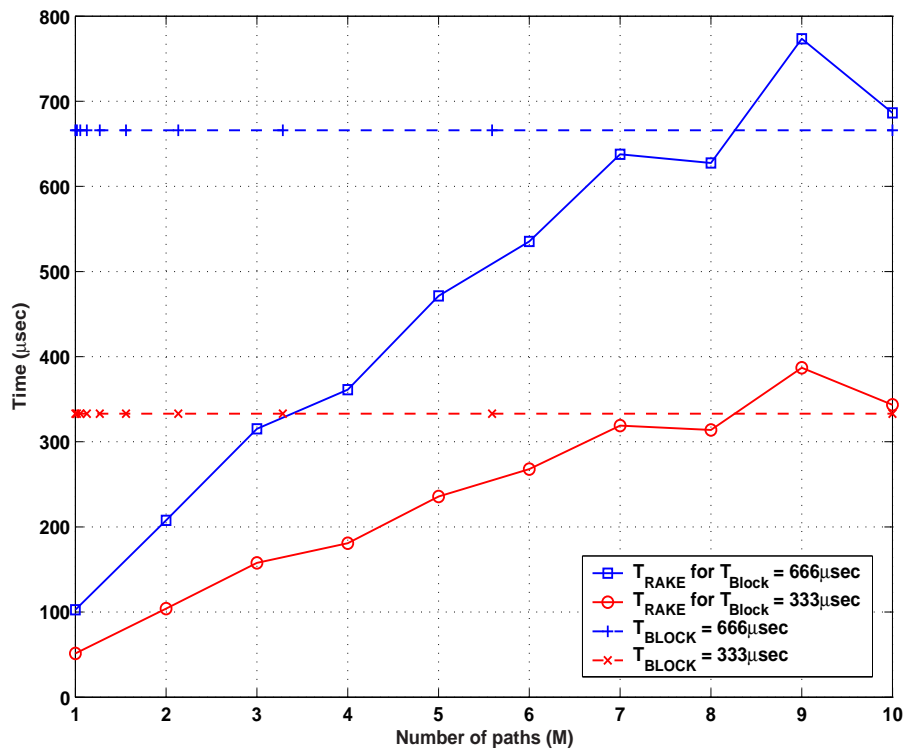


Figure 3.4: The processing time of the RAKE combination versus the number of the channel paths.

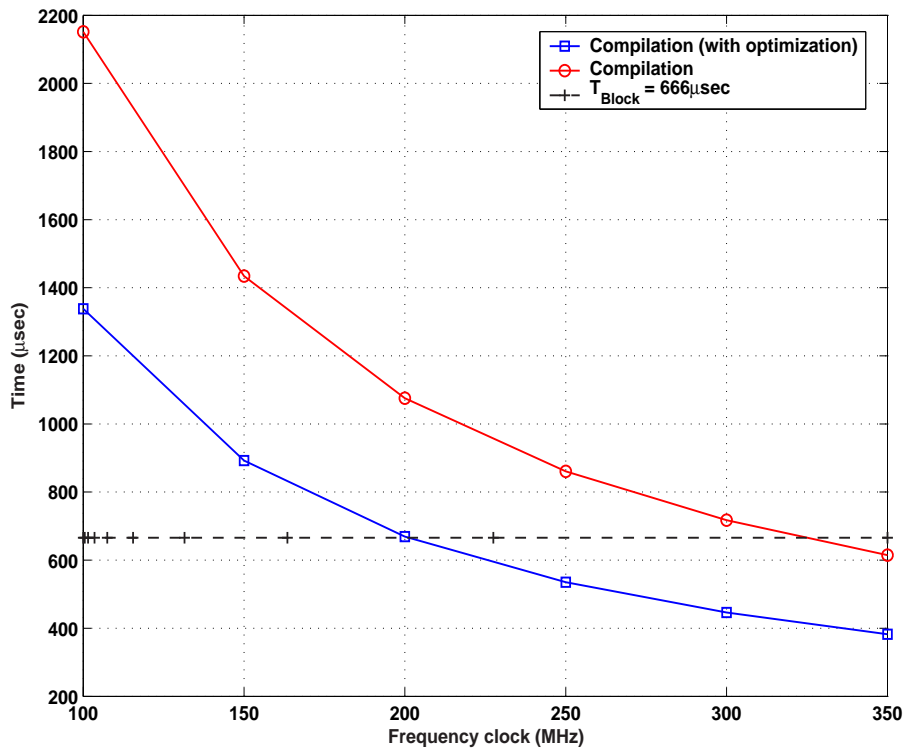


Figure 3.5: The processing time of the RAKE combination versus the DSP's frequency clock; $L = 4$, $M = 4$.

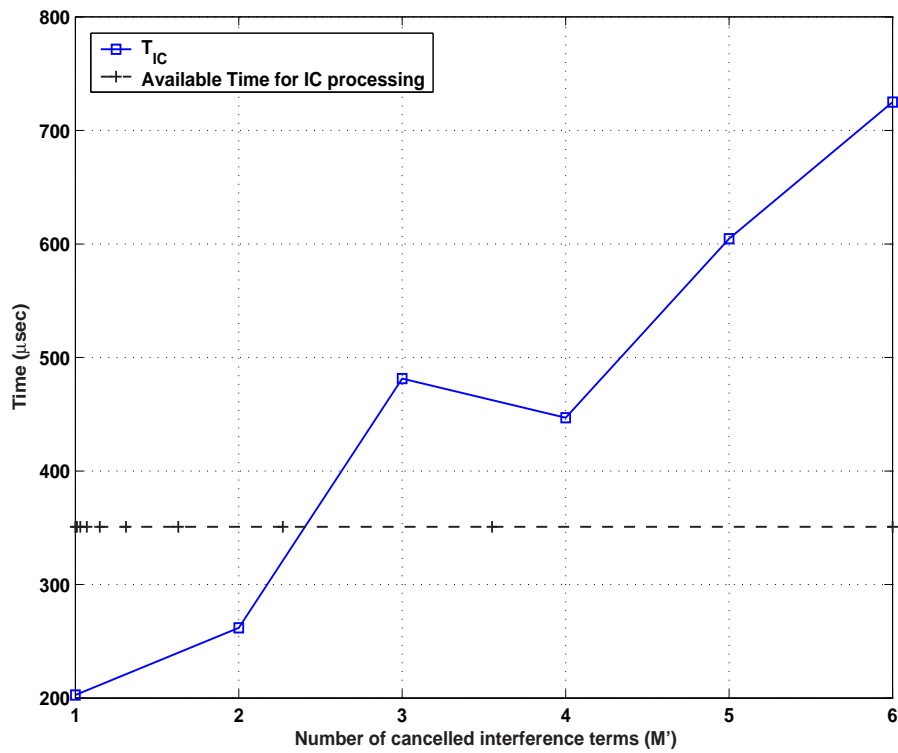


Figure 3.6: The processing time of the IC versus the number of canceled interference terms; $L = 3$ and $M = 3$.

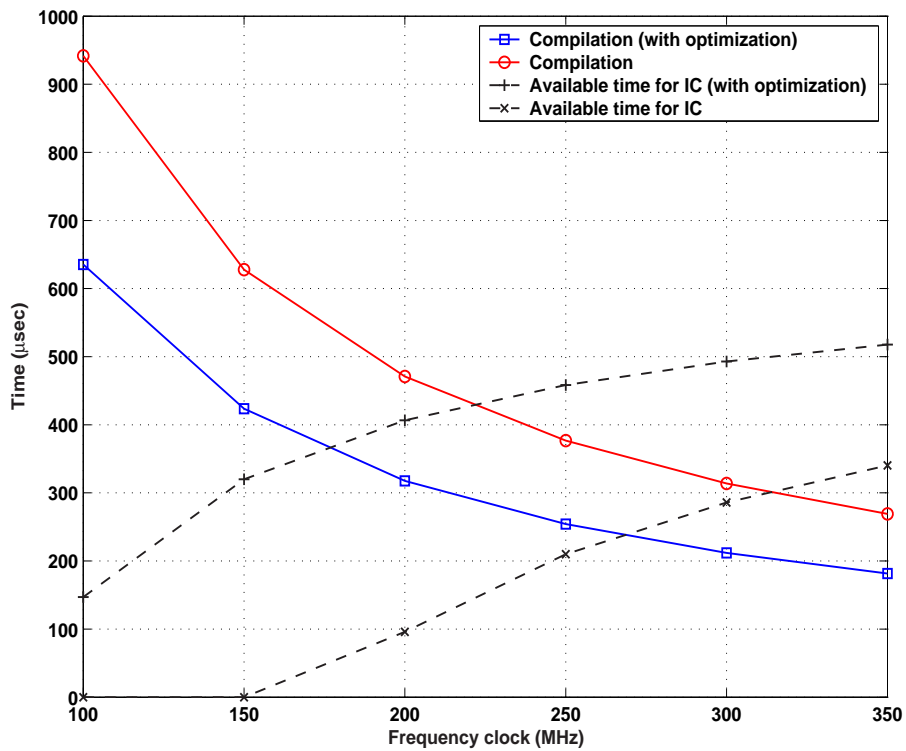


Figure 3.7: The processing time of the IC versus the DSP clock frequency for an environment with $L = 2$ and $M = 2$.

Operation	Percent
Despreading/Descrambling	30.37 %
Complex Multiplication/MRC	6.55 %

Table 3.1: Linear Profile for the RAKE combination (37.87 %).

Operation	Percent
Decision	5.84 %
Spreading/Scrambling	6.55 %
Despreading/Descrambling	26.45 %
Complex Multiplication/Subtraction	7 %

Table 3.2: Linear Profile for the IC scheme (47.74 %).

Fig.'s 3.6, 3.7 present the behavior of the IC developed function. We can see that the resulting remarks are similar to the RAKE combination. As for the available time for the IC processing which is defined as the time constraint of this function, it is calculated by the formula $T_{Available} = T_{BLOCK} - T_{RAKE}$.

3.2.3 Optimized code

In order to achieve the best possible performance and minimize the processing time, the programming method follows all the known optimization techniques [UEB97]. Thus,

- it avoids initialization commands for the used arrays by unrolling the loops
- data access from an array is implemented by increasing a pointer instead of using the array position (i.e $p \rightarrow x, *(p+n)$ Vs $x[n]$, where the pointer p points to the array x and n is the array position).
- it uses constants for the variables which do not change their values,
- it minimizes the use of conditional structures, which change the sequentially logic,
- it avoids the use of the dynamic memory allocation by using arrays with a size which corresponds to the worst operational case,
- it minimizes the number of loops,
- it uses some specific DSP commands which permit the execution of some arithmetic operations in parallel (ie. the command $mult_i2x16(A\ B, C\ D) \Rightarrow (A \cdot C, B \cdot D)$ allows two 16-bit multiplications in parallel).

Annex B presents the code with the linear profile for the two developed functions. For the linear profile, we selected the case of a channel environment with $L = 2$ paths which generates

$L(L - 1) = 2$ interference terms. The fact that the number of paths is equal to the number of interference terms allows a fair comparison between the developed functions. From the presented linear profile, we can see the computation requirements for each essential operation of the algorithms, and thus the operations which are more “time hungry”. Tables 3.1, 3.2 summarize the results of the linear profiles.

Number of paths (L)	RAKE	IPI-IC
1	103	-
2	208	458 (2 terms ; 1 stage, 2 terms ; 2 stage)
3	315	262 (2 terms ; 1 stage)
4	362	262 (2 terms ; 1 stage)
5	472	-
6	536	-
7	638	-
8	627	-

Table 3.3: The required time (in μsec) for the processing of one UMTS slot.

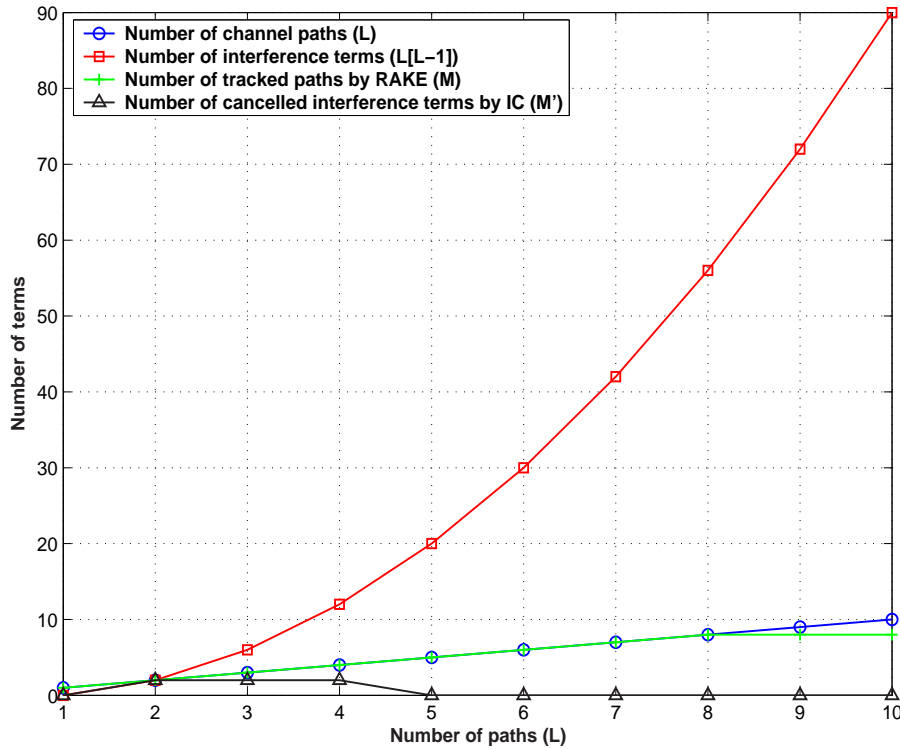


Figure 3.8: The efficiency of a DSP with a frequency 250 MHz in terms of the operated computations for different channel environments.

From these tables three remarks are in order. First, for the RAKE combination, the part of the code which implements the despreading/descrambling operation is the slowest. In contrast with the other RAKE operations, the despreading/descrambling is a chip based operation. This

means that for each symbol, it must be executed N times and thus, it has a complexity N times more than the symbol based Complex Multiplication/MRC operation. We note that the despreading/descrambling operation has been implemented as a complex multiplication of the received signal with the conjugate spreading/scrambling code. The implementation approach, which is based on the arithmetic nature of the spreading/scrambling code (sequence of ± 1), and uses a conditional structure, is more computationally complex. The conditional structure changes the sequential execution and the generated code jumps produce a high latency. Second, for the IC scheme, the remarks are similar to the RAKE processing. Thus, the chip based Despreading/Descrambling operation registers the highest time latency, which is N times more than that of the symbol based operations. Finally, the complexity of the IC function is higher than that of the RAKE combination. We can see that the difference in complexity is concentrated in the non common functional parts. More specifically, the decision device, which is used by the IC to produce symbols decisions, has a high degree of complexity (about 6%) which is not present in the RAKE combination.

3.2.4 DSP performance

The selected DSP is a TIGERSHARC (ADSP-TS101) at a frequency of 250 MHz which follows the architecture of the previous paragraph. However, despite its high degree of performance, flexibility, reprogrammability and parallelism, it cannot support the time constraints introduced by the standard. The processing time is equal to the duration of one slot which is $666 \mu\text{sec}$ for the UMTS specifications. Table 3.3, summarizes the required computational time for the RAKE combination for different channel environments. The available time is used by the IC processing. We can see that the RAKE can process 8 paths at maximum. The optimal IC cannot be performed due to time constraint limitations. Table (3.3) is translated in the Fig. 3.8, which uses the number of the performed computations as a measure of the DSP implementation efficiency. This figure presents the number of the required computations for each function and for each operational environment, as well as the number of operations which can be supported by the DSP implementation. We note that the priority in the using of the available computational power, is the tracking of as many channel paths as possible by the RAKE combination. Thus, as we can see from Fig. 3.8, the IC can be applied only for the cases ($2 \leq L \leq 4$), and it is limited to a partial cancelation of the two most critical interference terms with the possibility of one more stage for the case of $L = 2$. It is clear that DSP devices are inefficient for a reconfigurable implementation of this detector and a HW acceleration is necessary.

3.3 Hardware implementation

3.3.1 The iterative architecture

The functional similarities and the serial nature of the RAKE and IC processing can be used for the definition of a simple reconfigurable hardware architecture. This architecture consists of a simple computational unit which can perform either one RAKE demodulation or one IC suppression. Due to computational similarities, the change of functionality doesn't include complicated architectural changes and is simplified to switch on or off some operational blocks with a

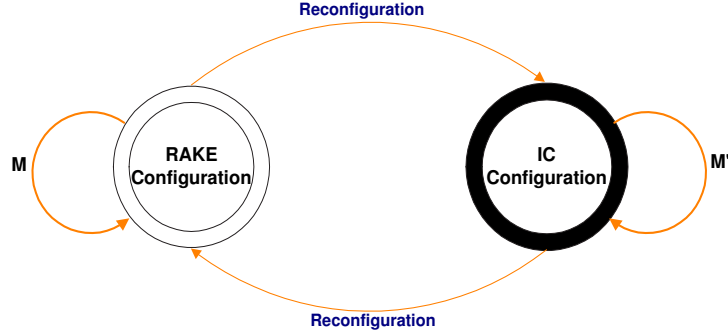


Figure 3.9: The FSM model of the iterative mapping approach for the detection scheme under consideration.

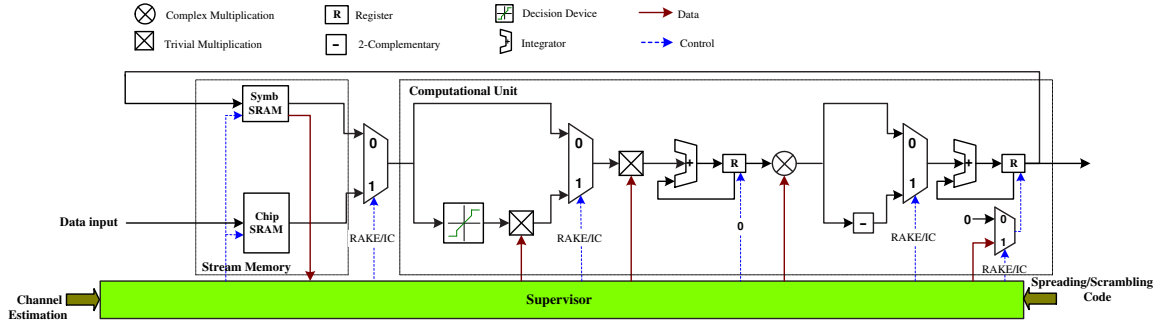


Figure 3.10: The corresponding computational core of the iterative approach.

very fine granularity. Moreover, we can say that the computational unit forms the least common denominator of the two algorithmic schemes under consideration. It contains almost all of the computational resources needed for a discrete iteration for both of them. The total algorithmic operation (RAKE or IC) requires a number of iterations which correspond to the dynamic parameters (number of paths or number of IPI terms).

The iterative mapping approach can be modeled as a FSM where the configurations are represented by the nodes of the graph. In this case the model consists of two discrete states which correspond to the two possible configurations of the basic unit. The FSM is presented in Fig. 3.9.

In order to distinguish the two different functionalities, some multiplexers are used in order to switch off the unused essential operators. A detailed block diagram of the proposed architecture is shown in Fig. 3.10. It contains three main units: stream memory, computational unit and Supervisor (SPV).

Stream memory

The multi-stage interference cancelation is a block algorithm (BLE) and works on data blocks. In the WCDMA the data transmission is organized in terms of time-slots. Thus, we can consider that the block is equal to one slot and its maximum available processing time is equal to its duration. The stream memory consists of two two-port SRAMs. The first one, SRAM chip, is

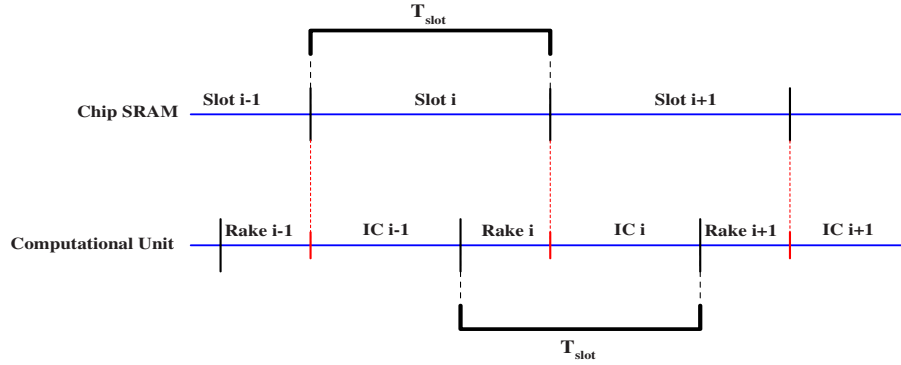


Figure 3.11: Timing relationship of SRAM chip and SRAM symbol WR/RD operations.

used to store the current transmitted slot and it is the data source for the RAKE processing. The second one, SRAM symbol, is the data source for the IC processing of the previous slot. Fig. 3.11 shows the relative timing and overlapping of the operations (SRAM chip WR/RD and SRAM symbol WR/RD) in order to avoid data hazards [HEN96].

The input slot arrives in the SRAM chip as I/Q sample pairs coming from pulse shaping filtering. Considering that read is higher than write access frequency, the necessary size for the SRAM chip can be calculated with the following formula

$$N_{SRAMchip} = \left\lceil \frac{T_{BLOCK} + T_{SP}}{T_c} \right\rceil 2RN_{bits}, \quad (3.4)$$

where $N_{SRAMchip}$ is the size of the SRAM chip memory in bits, T_{SP} is the delay spread, T_c is the chip duration, R is the oversampling ratio and N_{bits} is the word length for both I and Q samples.

As for the SRAM symbol, it is used as a data input and data buffering for the IC algorithm. More specifically, it stores the RAKE and the previous IC stage outputs and provides the necessary data (initial RAKE decision, last IC decision) for each symbol under processing. We note that each stage of the IC configuration requires the initial RAKE decision and the last IC output. The initial RAKE output is used as the reference “infected” signal, and the last IC output as the best available signal for estimating the transmitted signal. Thus, the necessary size of the SRAM symbol memory can be calculated as

$$N_{SRAMsymbol} = 2 \cdot \left\lceil \frac{T_{BLOCK}}{N \cdot T_c} \right\rceil 2N'_{bits}, \quad (3.5)$$

where $N_{SRAMsymbol}$ is the size of the SRAM symbol in bits, N is the spreading factor and N'_{bits} is the word length for both I and Q soft decisions of the decision device.

With 10-bit I/Q samples, $R = 4$, a slot duration 2/3 ms, a chip rate $f_c = 3.84$ Mcps, a delay spread equal to 2 ms, a $\min(N) = 2$ (where $\min(x)$ denotes the minimum value of the variable x), and a soft decision with $N'_{bits} = 4$ bits, a 207 Kb SRAM chip and a 20.48 Kb SRAM symbol would be required which are feasible considering hardware implementations.

The memory block and its complexity required is not introduced by the reconfigurability concept and thus it is not a complexity cost of this architecture. The need of memory is inde-

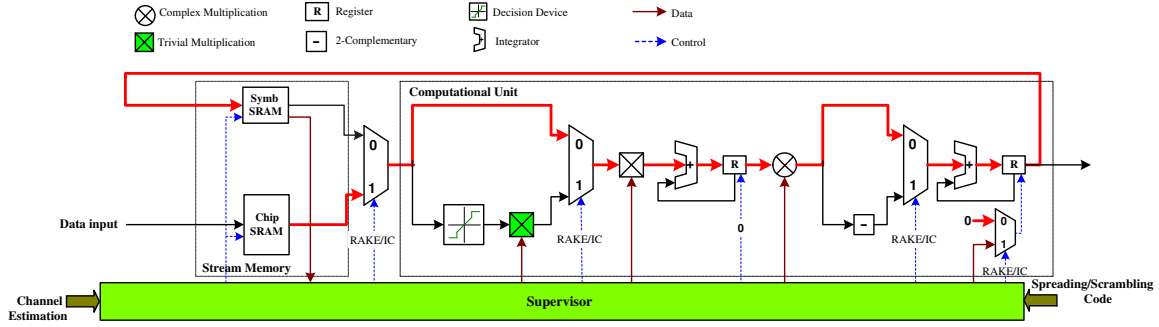


Figure 3.12: The flow of data during the RAKE configuration mode.

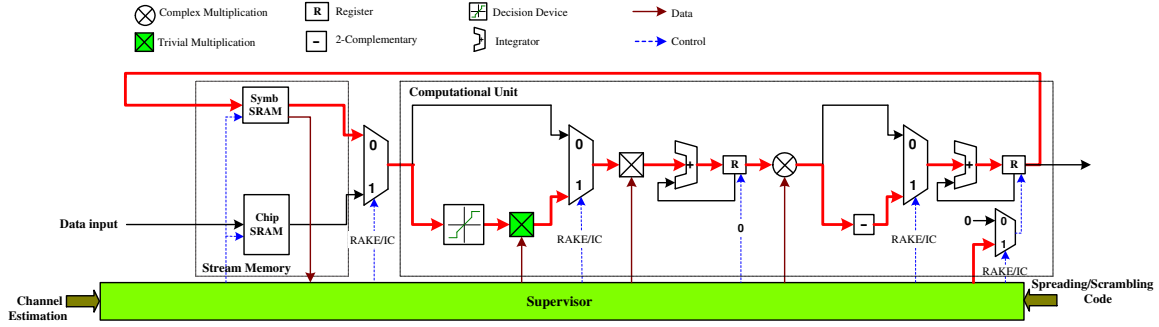


Figure 3.13: The flow of data during the IC configuration mode.

pendent of the reconfigurability and it is introduced due to the block nature of the interference cancelation. It has a constant and static complexity, which characterizes any implementation of the detector under consideration. In the following analysis, the block memory will be considered as a non critical parameter.

Computational unit

The proposed unit contains all of the computational and configuration resources needed for one channel demodulation and one IPI term suppression. Because its operation is time-multiplexed between the multipath tracking and the IPI suppression, its operation frequency has to be high enough, so that all the operations can be performed within a processing cycle defined by one time slot. Its identification is a key contribution of the application part of this thesis. It contains two chip multipliers, two integrators, one complex multiplication and three multiplexers to accomplish the reconfiguration.

In the RAKE mode, the data coming from the SRAM chip are chip samples. Depending on the channel estimation, the symbol and the path, one chip sample arrives at the computational unit at each clock cycle. The first chip multiplier is switched off and the second one performs the despreading process. In each clock cycle, it multiplies the input chip sample with the appropriate spreading and scrambling value. The resulting signals are integrated over a period corresponding

to the SF. Partial symbol integration results are stored in an integration register which has an initial value equal to zero and is initialized for each path and symbol. The output of the integrator is multiplied with the conjugate channel by the complex multiplier. Finally, a second integrator is used in order to perform the MRC. In this case, the integration period is equal to the number of tracked paths, the initial value of the register is also zero and the initialization is repeated for every symbol. The resulting symbol soft decisions are stored at the symbol SRAM, for further processing. Fig. 3.12 presents the data flow of the RAKE configuration.

In the interference cancelation mode and for the initial stage, the data coming from the SRAM symbol are the symbol soft decisions of the RAKE combination. In each clock cycle, one symbol is passed through the decision device to produce an estimation of the corresponding transmitted symbol. In our proposition, the decision function is a hybrid combination of hard and soft decision. The estimated symbol is fed into the first chip multiplier, which implements the spreading process. In the sequel, each chip produced is fed into the second multiplier which implements the despreading process. The spreading/scrambling codes used have different phases, which correspond to the generated IPI term. The resulting signals are integrated over a period corresponding to the SF. The output of the integrator is multiplied with the combination of two channel paths and its 2-complementary is taken. Finally, a second integrator is used to suppress the IPI term from the RAKE decision. The initial value of the integrator register is the RAKE decision of the processing symbol. The integration period is equal to the number of the considered IPI terms. The produced decision symbols are stored in the SRAM symbol. The next stages of the IC algorithm use the stored IC output of the previous stage as input to the computational unit. Fig. 3.13 presents the data flow of the IC configuration.

The supervisor

The SPV is an intelligent architectural block which controls and synchronizes the operations of the computational unit and the stream memory. It translates the TDL given by the channel estimation to memory addresses and provides the appropriate addresses for the tracking of a particular channel path or the reconstruction of a particular IPI term. Concerning the control of the computational unit, SPV loads the appropriate configuration, provides the required input signals at their different structural blocks and synchronizes their operations.

An important task of SPV is the optimization of the system performance under a well-defined computational power constraint. This constraint is the maximum number of cycles which can be used to process a symbol. This number can be calculated with

$$\begin{aligned}
 N_{C_{max}} &= \frac{f_{clk}}{f_{symb}} \\
 &= \alpha \cdot f_c \cdot \frac{N}{f_c} \\
 &= \alpha \cdot N,
 \end{aligned} \tag{3.6}$$

where f_c is the chip frequency, f_{clk} is the clock frequency, $f_{symb} = f_c/N$ is the symbol frequency, and α is defined as the ratio f_{clk}/f_c . The above constraint limits the number of the serial iterations of the computational unit. With a pipeline implementation of the computational unit,

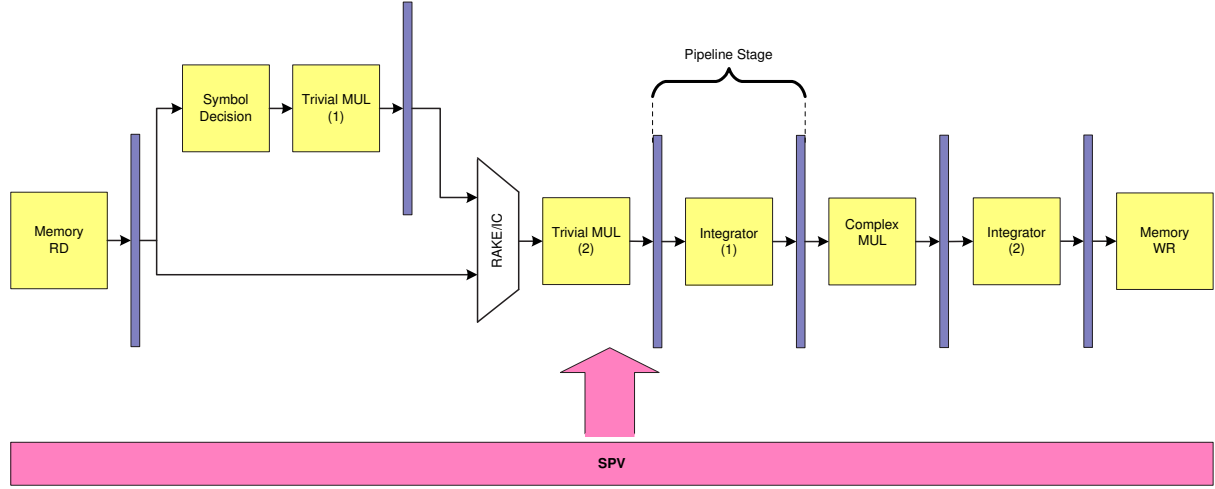


Figure 3.14: The pipeline implementation of the reconfigurable RAKE/IC detector.

the throughput is equal to 1 process per N cycles. Thus, if L is the number of resolvable channel paths, $M \leq L$ the number of tracked paths from the RAKE combination, the above constraint can be written as

$$\begin{aligned} (M + M'V) \cdot N &\leq N_{c_{max}} \Rightarrow \\ M + M'V &\leq \alpha, \end{aligned} \quad (3.7)$$

where $M' \leq M(M - 1)$ is the number of IPI terms under consideration and V the number of stages. At run time, the SPV selects these parameters with parameter M at the maximum. With $L = 3$ and $\alpha = 32$, we can use $M = 3$, $M' = 6$ and $V = 5$.

3.3.2 Implementation issues

The basic computational unit has been implemented by using a pipeline approach in order to maximize the throughput and thus the number of iterations, which is a critical parameter for the performance of the serial reconfigurable detector. In order to maximize the flexibility and minimize the switched off hardware, each essential block which corresponds to a pipeline stage has a resolution of one chip and performs one chip computation per cycle. Fig. 3.14 presents the general architecture of the pipeline implementation.

Chip multiplication

It implements the process of spreading/scrambling and despreading/descrambling. Due to the nature of the used scrambling and spreading codes, complex multiplication can be transformed to a simple sign change. Specifically, the scrambling codes are always sequences of 1's and -1's only (mapped to logic '0' and '1', respectively), and thus the complex multiplication in the correlations can be simplified to a simple sign change operation. This can be derived from the following formula

$$\begin{aligned}
Tr &= (S_I + jS_Q)(C_I + jC_Q) \\
&= (S_I C_I - S_Q C_Q) + j(S_Q C_I + S_I C_Q),
\end{aligned} \tag{3.8}$$

$$\begin{aligned}
C_I = 0, C_Q = 0 &\Rightarrow Tr = (S_I - S_Q) + j(S_Q + S_I) \\
C_I = 0, C_Q = 1 &\Rightarrow Tr = (S_I + S_Q) + j(S_Q - S_I) \\
C_I = 1, C_Q = 0 &\Rightarrow Tr = -(S_I + S_Q) - j(S_Q - S_I) \\
C_I = 1, C_Q = 1 &\Rightarrow Tr = -(S_I - S_Q) - j(S_Q + S_I),
\end{aligned} \tag{3.9}$$

where S_I , S_Q are the real and imaginary parts of the input samples, respectively, and C_I , C_Q are the real and imaginary parts of the corresponding code values, respectively. The code $C_I + jC_Q$ is the combined code obtained by $C_I = Walsh \otimes Re\{Scrambling\ Code\}$, $C_Q = Walsh \otimes Im\{Scrambling\ Code\}$. Note that in the despreading/descrambling process, the complex conjugate of the scrambling code is used.

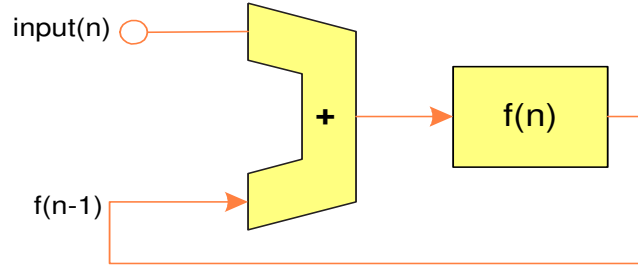
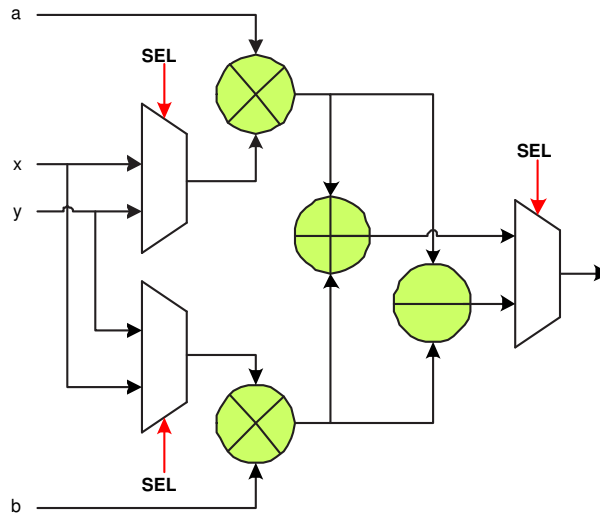
In the analysis of the DSP implementation, we have shown that this computational part is the most complicated as it is executed for each chip. In this implementation, the chip multiplication has been represented as a complex multiplication, because a conditional approach introduces code jumps and thus suffers from a high time latency. On the other hand, in the hardware implementation, the conditional approach is applicable and thus the chip multiplication does not require real multiplications. In this case, we can say that chip multiplication is transformed to trivial multiplication.

Integrator

It is an essential architectural element, it is consisting of an adder and a register. In each clock time, it adds the input data to the content of the register and stores the result in the same register. In the RAKE mode, it is used for two different processes. The first one is the chip integration, which transforms the chip sequence to symbols. In this case the initial value of the register is zero and it is forced to this value every N cycles, where an integration is carried out. The second integrator is used in the MRC process. In this case the initial value of the register is also zero and it is forced to this value every NM cycles, where a RAKE symbol detection is produced. Moreover, in the IC mode it is used for two different operations. Firstly, the transformation of the chip sequence to symbols as in the RAKE mode. Secondly, for the subtraction of the generated interference terms from the initial RAKE decision. In this case, the initial value of the register is the RAKE decision of the symbol being processed and it is updated for every NM' cycles, where an interference suppression has been carried out. Fig. 3.15 presents the integrator structure.

Complex Multiplication

It implements the multiplication of the symbol sequence with the conjugate channel coefficients (RAKE mode) or the combination of the channel coefficients which participates in the interference

Figure 3.15: The integrator block for the time instant n .Figure 3.16: The complex multiplication between $a + bj$ and $x + yj$ in two processing cycles.

structure (IC mode). It is a complex multiplication, which can be implemented by four real multiplications (two multiplications for the real part and two for the imaginary). In order to minimize the number of multipliers and to exploit the pipeline implementation technique of the computational unit, the complex multiplication can be achieved in two cycles and using only two multipliers at symbol rate. In the first cycle, the imaginary part is calculated and in the second one the real. Fig. 3.16 presents the process of complex multiplication. As we can see, three multiplexers are used in order to change between the computation of the real and imaginary part. In the first processing cycle, the selection signal ' SEL ' has the value '0' and thus the imaginary part is calculated. In the second processing cycle, it has the value '1', which corresponds to the computation of the real part. For the IC mode, the sign of the resulting multiplication (complementary-2) is inverse in order to prepare the generated interference terms for the subtraction in the next pipeline stage.

The possibility of implementing the complex multiplication in two cycles and thus optimizing the complex multiplication block is introduced from the specifications of the UMTS. The lowest possible SF is two (TDD operational mode) and thus in the worst case the input of the complex multiplier is constant for two cycles.

Write memory block

It is the last stage of the pipeline which has the task of generating memory addresses for storage in the memory of each symbol, which is produced from the previous integration pipeline stage. It generates memory addresses in a serial way starting by the address \$0. Thus, the n -th symbol of each data block will be stored in the position \$n. In the same time, it notifies the processing of the last data block symbol in order to inform the SPV about the completion of the current configuration.

SPV

The first task of the SPV block is the synchronization of the different functional blocks of the computational unit. It monitors the different pipeline stages and enables or disables the interface between them. For example, in the RAKE mode, the flip-flop which connects the output of the first integrator with the complex multiplication block is disabled till the processing of N chips. Moreover, it generates the appropriate initial signals of the integrator registers. Thus, for example, the register of the first integrator is reset to zero after the addition of N chips. We can say that SPV translates the dynamic system parameters (N, M, M') to signals which control the pipeline of the basic computational unit.

Afterwards, the serialization of the required computations and the design of the basic functional blocks in the level of one chip/symbol, requires the generation of one chip/symbol address in each processing cycle. SPV has the task of generating the appropriate memory address in each clock period. More specifically, the RAKE and IC algorithms are mapped to an address generation strategy. If we suppose that the first chip sample of the chip memory and the first symbol of the symbol memory are stored in the addresses \$0 and \$0, respectively, and the starting time instant $t_0 = 0$, the variables i, j, l are initialized to zero, the address generation by the SPV in the time instant $t_{i,j,l} = j + N(i + l)$ follows the formulas

$$\begin{aligned}
 \$Chip_addr_{i,j,l} &= (i \cdot N + j)R + delay(l) \\
 j &= \begin{cases} j + 1, & \text{if } j < N - 1 \\ 0, & \text{else} \end{cases} \\
 l &= \begin{cases} l + 1, & \text{if } j < M - 1 \\ 0, & \text{else} \end{cases} \\
 i &= i + 1, \quad \text{if } j = N - 1 \text{ and } l = M - 1,
 \end{aligned} \tag{3.10}$$

$$\begin{aligned}
\$Symb_addr_{i,j,l_1,l_2} &= \begin{cases} i - P - 1, & \text{if } j < |G| \text{ \& } T < 0 \\ i - P, & \text{if } j \geq |G| \text{ \& } T < 0 \\ i + P, & \text{if } j < |G| \text{ \& } T > 0 \\ i + P + 1, & \text{if } j \geq |G| \text{ \& } T > 0 \end{cases} \\
j &= \begin{cases} j + 1, & \text{if } j < N - 1 \\ 0, & \text{else} \end{cases} \\
l &= \begin{cases} l + 1, & \text{if } j < M' - 1 \\ 0, & \text{else} \end{cases} \\
i &= i + 1, \quad \text{if } j = N - 1 \text{ and } l = M' - 1,
\end{aligned} \tag{3.11}$$

where i, j, l are the symbol, chip and path, respectively, R is the sampling factor, $delay(l)$ is the delay of the l -th path in samples, $T = delay(l_1) - delay(l_2)$, $P = \lceil T \rceil / N$ and $G = T - N \cdot P$.

The generated address corresponds to a different sub-block of the stream memory depending on the configuration mode. Thus, in the RAKE mode, the generated address concerns the chip memory which stores the samples of the current data slot. For the first stage of the IC mode, the generated address concerns the sub-block, which stores the RAKE decisions. Finally for the i -th stage of the IC mode, the generated address concerns the sub-block which stores the IC decisions of the $(i - 1)$ -th stage.

The next task of the SPV is the control of the reconfiguration. In the initial stage the SPV configures the computational unit and the address generator mechanism to the RAKE mode. In this case the pipeline consists of five stages and the address generation follows the Eq. 3.10. When the last symbol of the processing block has been processed by the computational unit and is ready to be written in the symbol memory, the last stage of the pipeline sends a signal to the SPV to inform about the termination of the RAKE processing in the data block under consideration. In the case where the system parameters impose an interference cancelation, the SPV resets the pipeline synchronization and adds one more stage. Moreover, the address generation logic now follows the Eq. 3.11. The change of the configuration is implemented in one clock cycle and thus the reconfiguration time is negligible. When the first stage of the IC is terminated, the last stage of the pipeline informs the SPV about the end of the process. In the case of a multi-stage IC, the SPV does not change the configuration till the termination of the all IC stages. After the completion of the last IC stage, the SPV resets the system and loads the RAKE configuration in order to process the following data block.

The last task of SPV is the control of the writing process. According to the selected configuration, SPV uses the generated memory addresses from the W/R block to write the produced symbols in different memories. More specifically, in the RAKE mode, the initial symbol decisions are stored in the first symbol memory block. In the IC mode, the “cleaner” symbol decisions are stored in the second symbol memory block. We note that for the IC processing, the block RAKE decisions are required.

3.3.3 Simulation results

The proposed architecture has been fully described in VHDL and simulated in Modelsim for different parameter conditions in order to verify its functionality.

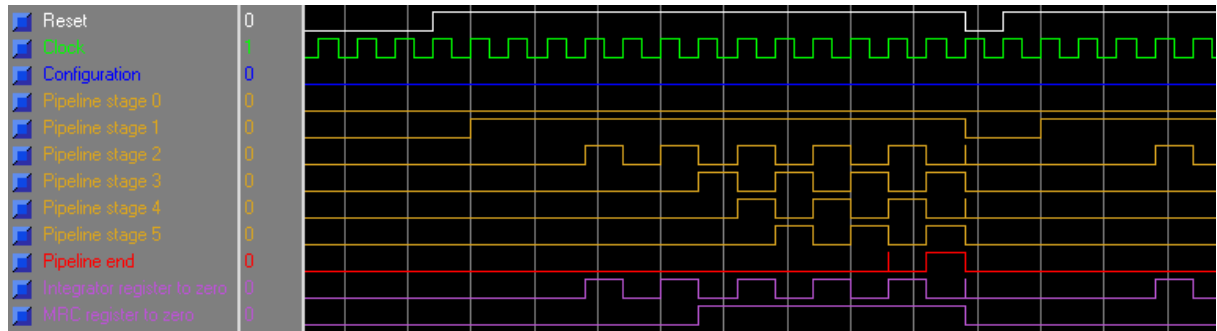


Figure 3.17: Modelsim time simulation for a channel environment with $L = 1$ and $N = 2$.

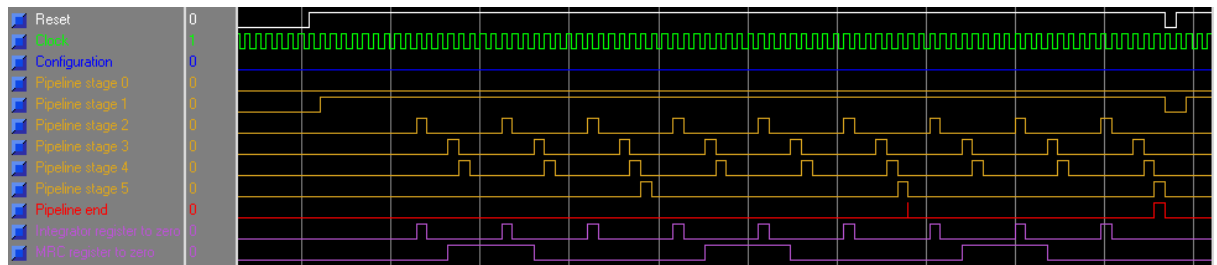


Figure 3.18: Modelsim time simulation for a channel environment with $L = 3$ and $N = 8$.

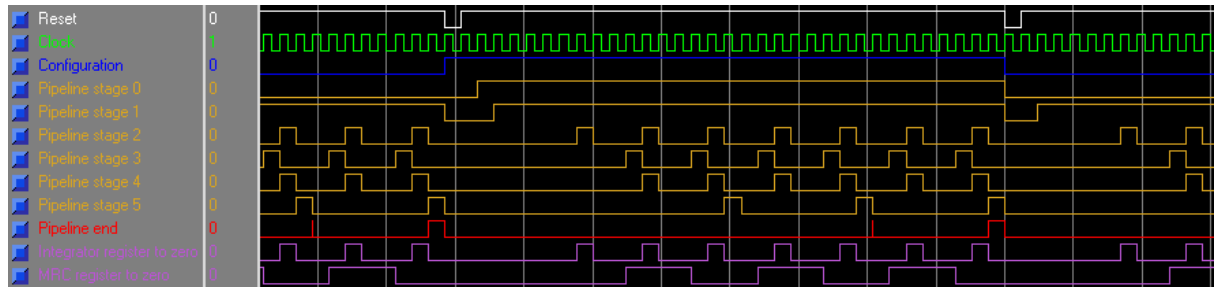


Figure 3.19: Modelsim time simulation for a channel environment with $L = 2$, $N = 4$, $V = 1$.

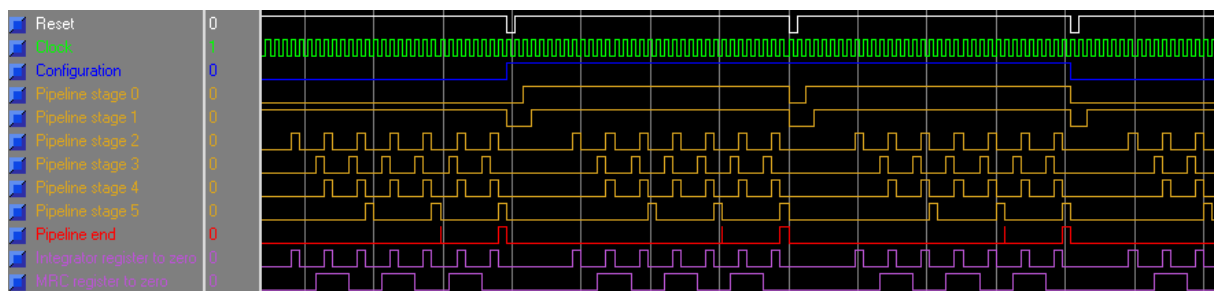


Figure 3.20: Modelsim time simulation for a channel environment with $L = 2$, $N = 4$, $V = 2$.

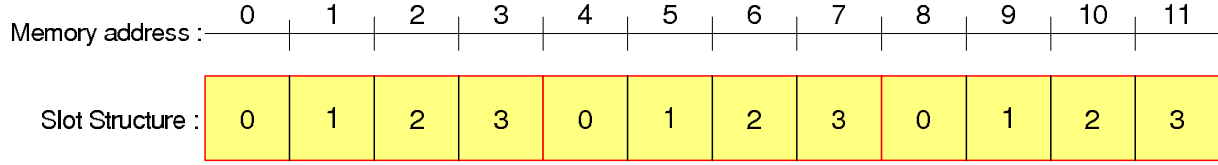
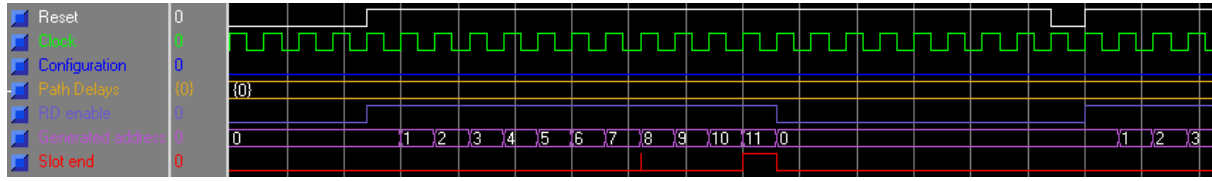
Figure 3.21: Memory structure for a slot with 3 symbols, $L = 1$ and $N = 4$.Figure 3.22: Generation of the chip memory addresses for the case of $L = 1$, $N = 4$ and $N_{Block} = 3$.

Fig. 3.17 shows the pipeline synchronization for the case of an environment with $L = 1$, $N = 2$ and $N_{Block} = 3$ symbols. In this case the IC mode is not used and thus the first stage of pipeline is always switched off. After the pipeline is filled, a symbol is produced every 2 cycles and thus the pipeline throughput is 2 cycles, which is equal to the spreading factor N . The integrator register is filled with zero every 2 cycles in order to implement the transformation of the chip sequence to symbols. On the other hand the MRC register content is always zero, as there is only one path in the channel. Moreover, we can see that each pipeline stage is enabled when the data are ready to go over to the next pipeline stage. At the end of the last block symbol, the pipeline is reset and the process of the next data block starts.

Fig. 3.18 shows the pipeline synchronization for the case of an environment with $L = 3$, $N = 8$ and $N_{Block} = 3$ symbols. We suppose also that for this environment the IC processing is not necessary and thus the first pipeline stage is always switched off. The throughput of the pipeline is equal to the spreading factor and thus it produces a symbol decision every 8 cycles. The integrator register is set to zero every 8 cycles in order to add the first chip for each symbol and path under consideration. Moreover, the MRC register is set to zero for the first path of each symbol in order to implement the path combination. After the processing of the last block symbol, the pipeline is reset and is ready to apply the RAKE combination to the next data block.

Fig. 3.19 presents the pipeline synchronization for the case of an environment with $L = 2$, $N = 8$ and $N_{Block} = 3$ symbols. For this case, IPI degrades the RAKE performance and thus an IC scheme is necessary. We suppose that the IC scheme has one stage $V = 1$. Firstly, the pipeline is configured to the RAKE mode and thus the first pipeline stage is switched off. The pipeline synchronization follows that of the previous examples and produces a symbol decision every $N = 4$ cycles. After the processing of the last block symbol, the pipeline is reset and configured to the IC mode. The configuration change is executed to one processing cycle and thus the reconfiguration time is negligible. After this cycle, the pipeline works as an IC generator/subtractor. The first pipeline stage is enabled and effectuates the symbol decision and the first trivial multiplication. For the selected propagation scenario of the $L = 2$ paths, the

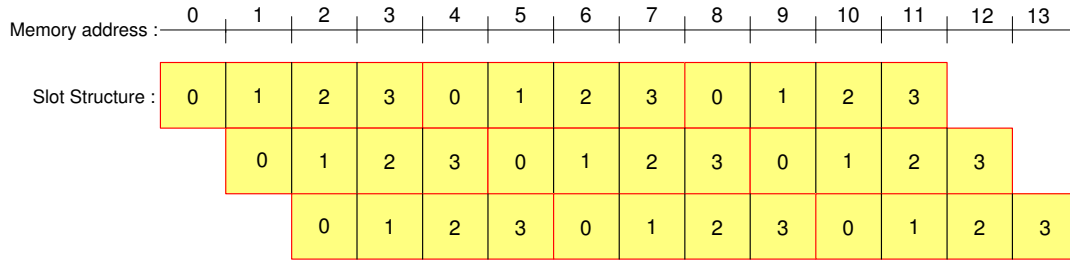


Figure 3.23: Memory structure for a slot with 3 symbols, $L = 3$, $\tau = [0 \ 1 \ 2]T_c$ and $N = 4$.

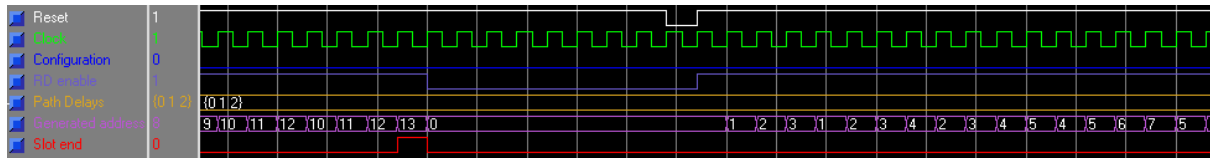


Figure 3.24: Generation of the chip memory addresses for the case of $L = 3$, $N = 4$ and $N_{Block} = 3$.

number of interference terms is also $L(L-1) = 2$. Thus for each symbol, two interference generation/subtractions are required. The integrator and MRC registers are set to zero periodically for each interference term under consideration. After the IC processing of the last block symbol, the pipeline is reset to zero and it is configured again to the RAKE mode in order to process the following data block.

Fig. 3.20 presents the pipeline synchronization for the same environment but we suppose that the IC scheme has one more processing stage, $V = 2$. Firstly, the pipeline is configured to the RAKE combination as in the previous case and after the processing of the last block symbol, it is reset and configured to the IC mode. The operations are similar with the case of one IC stage $V = 1$. When the pipeline finishes the first IC stage, it is reset to zero and is configured again to IC mode in order to implement the second IC stage. The processing and the pipeline synchronization are similar to the first IC stage. At the end of the second IC stage, the pipeline is reset and configured to RAKE mode in order to process the next data block. In general, at the end of the processing of a data block, the pipeline is reset and configured to the RAKE or IC mode depending on the decided algorithmic response.

An important part of the proposed architecture is the generation of the memory addresses. As we have presented in the previous paragraphs, the RAKE and IC algorithms have been transformed to a mechanism of serial generation addresses. In the following simulation examples, we try to show this generation mechanism for different slot structures. Firstly, we suppose an environment with $L = 1$, $N = 4$ and $N_{Block} = 3$ symbols. In this case there is no IPI interference and the received chips are stored in the chip memory as Fig. 3.21 shows. Address generation is simplified to a sequence of chip positions in succession. Fig. 3.22 shows the generation address mechanism. In each clock cycle the pointer is moved one chip to the right starting by the position \$0\$ which stores the first received chip.

Interference structure:

0	1	2	3	0	1	2	3	0	1	2	3				
		0	1	2	3	0	1	2	3	0	1	2	3		

Figure 3.25: Interference structure for a slot with 3 symbols, $L = 2$, $\tau = [0 \ 2]T_c$ and $N = 4$.

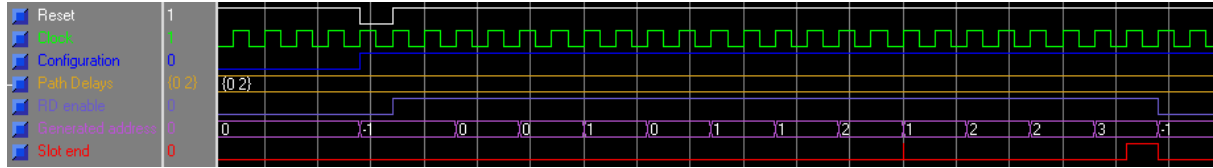


Figure 3.26: Generation of the symbol memory addresses for the case of $L = 2$, $N = 4$ and $N_{Block} = 3$.

Fig. 3.23 supposes a slot structure corresponding to the simulation environment $L = 3$, $N = 4$, $N_{Block} = 3$ symbols and $\tau = [0 \ 1 \ 2]T_c$. The chips are introduced to the pipeline in a serial way according to the mapping algorithm presented in the previous paragraph. Fig. 3.24 presents the address generation strategy. Thus, for the first symbol the address generator produces the addresses $\{0 \ 1 \ 2 \ 3 \ 1 \ 2 \ 3 \ 4 \ 2 \ 3 \ 4 \ 5\}$ and so on. The memory pointer is moved chip to chip, starting each time from the first chip of each path.

The symbol generation process is also an important task of the SPV. This generation is enabled during the IC operational mode in order to provide the pipeline processing with the appropriate symbol decisions in each clock cycle. Fig 3.25 presents the slot structure under consideration in order to show the symbol generation mechanism. This slot structure corresponds to a simulation environment with $L = 2$, $N = 4$, $N_{Block} = 3$ symbols and $\tau = [0 \ 2]T_c$. Fig. 3.26 presents the address generation strategy. In each clock cycle, the symbol generator produces a symbol address which corresponds to the chip of the interference term under consideration. When the generator provides a negative address, it is translated by the system as an introduction of a zero to the pipeline. More specifically, for the simulation case under consideration we can see that for the first symbol the interference introduced by the second path to the first path, is null for the first two chips (the second path has $\tau_2 = 2T_c$). The second path only interferes on the last two chips and this interference term is based on the first symbol. Thus the symbol generator generates a negative address for the first two chips and the address of the first symbol (0) for the last two. The second interference term of the first symbol is generated by the influence of the first channel path on the second one. In this case the interference for the first two chips involves the first symbol and for the last two the second symbol. Thus the generation strategy provides the address of the first symbol during the first two cycles and the address of the second symbol during the following two cycles. This generation strategy is repeated for each interference term.

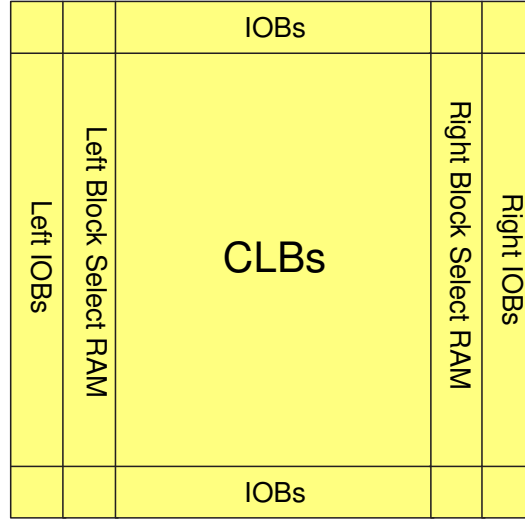


Figure 3.27: The Virtex architecture.

3.3.4 FPGA implementation

For the implementation of the proposed iterative reconfigurable architecture a Virtex FPGA from Xilinx has been selected.

Xilinx Virtex Series FPGA

Each Virtex device contains configurable logic blocks (CLBs), input-output blocks (IOBs), block RAMs, clock resources, programmable routing, and configuration circuitry [TOD05], [XIL]. These logic functions are configurable through the configuration bitstream. Configuration bitstreams that contain a mix of commands and data can be read and written through one of the configuration interfaces on the device. A simplified block diagram of a Virtex FPGA is shown in Fig. 3.27. Moreover a detailed description of the FPGA technology is presented in the next Chapter (4.4.2).

The Virtex configuration memory can be visualized as a rectangular array of bits. The bits are grouped into vertical frames that are one-bit wide and extend from the top of the array to the bottom. A frame is the atomic unit of configuration, meaning that it is the smallest portion of the configuration memory that can be written to or read from. Frames are grouped together into larger units called columns. In Virtex devices, there are several different types of columns, including one center column, two IOB columns, multiple block RAM columns and multiple CLB columns. As shown in Figure 3.28, each frame sits vertically, with IOBs on the top and the bottom. For each frame, the first 18 bits control the two IOBs on the top of the frame, then 18 bits are allocated for each CLB row, and another 18 bits control the two IOBs at the bottom of the frame. The frame then contains enough “pad” bits to make it an integral multiple of 32 bits.

The configuration for the Virtex device is done through the Frame Data Input Register (FDR). The FDR is essentially a shift register into which the data is loaded prior to transfer to configuration memory. More specifically, given the starting address of the consecutive frames to be configured the configuration data for each frame is loaded into the FDR and then transferred

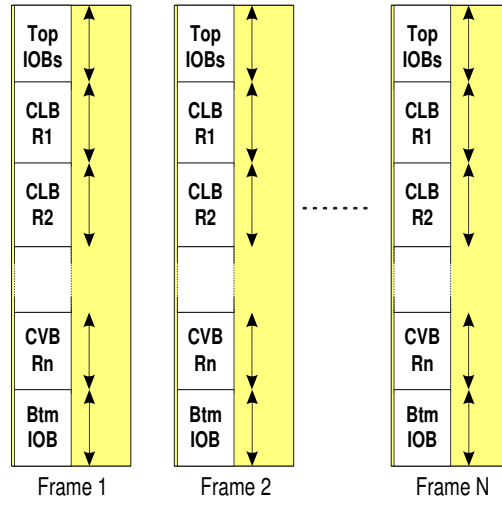


Figure 3.28: CLB frame organization.

Device Utilization	Percentage
Slices	501/19968 (2%)
Flip Flops	378/39936 (0%)
LUTs	885/39936 (2%)
IOBs	133/410 (3%)
GCLKs	1/32 (3%)
DSP48s	2/48 (4%)

Table 3.4: The device utilization for the computational unit.

to the frames in order. The FDR allows multiple frames to be configured with identical information, requiring only a few cycles for each additional frame, thus accelerating the configuration. However, if even one bit of the configuration data for the current frame is different from the previous frame the entire frame must be reloaded.

3.3.5 FPGA performance

Firstly, for the synthesis, a commercial FPGA from the Xilinx Virtex series, XC4V40 is selected. The XC4V40 features over one million gates partitioned into 39936 logic blocks, it has a low cost and offers a good alternative to ASICs and ASSPs [TOD05].

Table 3.4 summarizes the synthesis results of the computational unit, which is the most important part of the proposed architecture. As we can see from this table, the internal FPGA resources consumption, which correspond to the computational unit is very small.

The maximum path delay is translated to a clock speed of approximately 168.8 MHz which corresponds to a maximum number of iterations equal to 44 ($168.8/3.84$). In order to increase the number of iterations, the reconfigurable architecture is synthesized in $0.13\mu\text{m}$ CMOS technology (HCMOS9 STMicroelec). We note that in the commercial terminals, an ASIC implementation is always used, while the commercial FPGAs are used for prototype implementations. From the

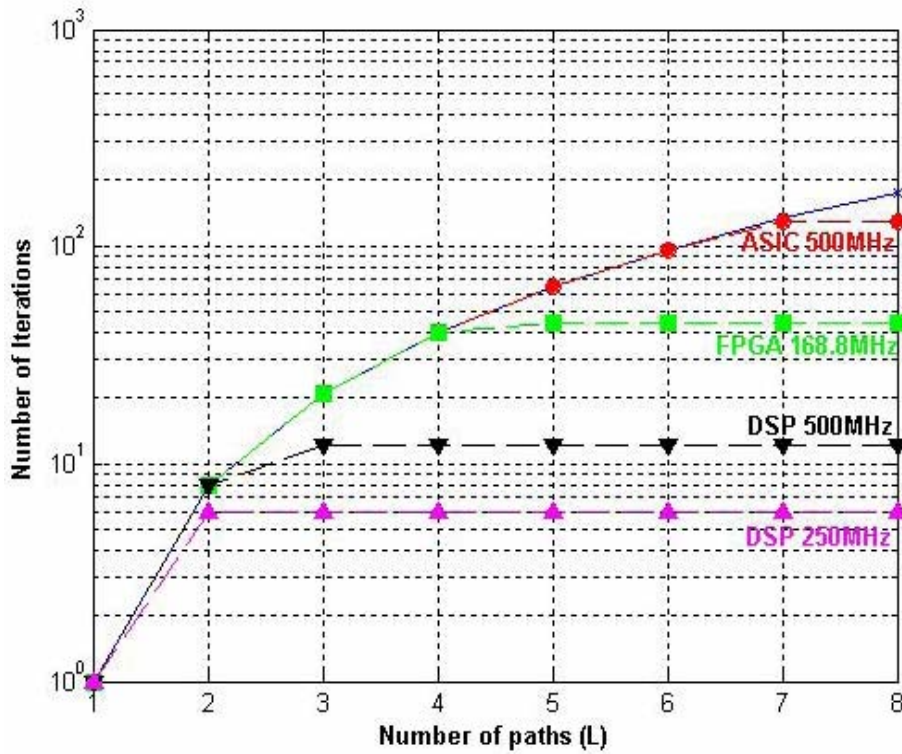


Figure 3.29: The achieved computation power versus the number of paths.

synthesis results, we found that the used area of the computational unit is equal to $0.07mm^2$, which corresponds to a very small and fine circuit. This property makes the proposed iterative architecture suitable for terminals implementations which are characterized by limited area constraints. Moreover, in this case, the critical path delay allowing operation at rates in excess of 500 MHz which corresponds to a number of iterations equal to 130 ($500/3.84$). This value can satisfy all the computation requirements of the optimal detection.

Fig. 3.29 compares the achieved performances of the three reconfigurable devices (DSPs, FPGAs and ASICs) in the terms of the performing computation according to the previous presented results. In this figure, we suppose that the optimal RAKE+IC detection needs $V = 3$ stages and the channel has six paths at maximum. As we can see, the DSPs have a very poor performance and they cannot support the required computation. The commercial FPGAs are efficient through to a channel with four paths. For the other cases, they are limited to a partial cancelation of the most critical terms. Finally, the ASIC implementation can support the required computation of the optimal detection in each operational case.

The proposed implementation minimizes the hardware area overhead due to serialization of the required computations. If we suppose a traditional implementation which deals with the worst operational case, the required area can be estimated by the equation (3.12), where F is the total used area, A is the required area for one processing (finger/generator), L_{max} is the maximum number of the channel paths and V is the number of the IC stages. Thus we have an optimization of the area surface by a factor K .

$$\begin{aligned}
F &= [L_{Max} + L_{Max}(L_{Max} - 1)V] \cdot A \\
&= K \cdot A
\end{aligned} \tag{3.12}$$

3.4 Conclusion

In this Chapter we presented two implementations of the proposed multi-stage equalization scheme of the previous Chapter. Firstly, we tried to implement this algorithm to commercial DSP's. Although their high degree of flexibility and parallelism, DSP's have a very poor performance and they are inefficient to support the required computational power in the time constraints of the application. DSPs are limited to a partial and single-stage IC processing, which is inefficient for the operational environments with a high number of channel paths. Thus a HW implementation is necessary. The proposed architecture is based on the computational similarities and the iterative calculation of the two principal functions (RAKE and IC), which compose the equalizer. More specifically, it consists of a processing unit, which is able to be configured either as a demodulator finger or as a generator/sub-tractor of one interference term without complicated architectural changes. This architecture has been simulated for different propagation environments in order to verify its functionality and it is synthesized in commercial FPGA's and ASIC's. The produced results have shown that this implementation approach satisfies the computational requirements and the time constraints by using a low area and flexible hardware resources. The achieved optimizations are suitable for terminal implementations.

Chapter 4

The iterative reconfigurability concept

4.1 Introduction

This Chapter formalizes the results of the previous two Chapters by defining the **iterative reconfigurability** concept. Reconfigurability can be viewed as a two-layer concept and a fair definition requires the specification of its algorithmic and hardware dimensions. The first layer is the algorithmic reconfigurability and refers to the ability of a system to change functions and algorithms in real time. The second layer is the hardware reconfigurability and refers to the ability of a system to change the implementations of the functions and algorithms in real time. An important notion of this layered approach is the intermediate layer, which defines how an algorithmic change will be mapped on the hardware layer. This intermediate layer, which is called architectural layer, is a function of two parameters: the first one comes from the algorithmic layer and is the computational structure of the supporting algorithms, and the second one comes from the hardware layer and is the nature of the reprogrammable devices.

Due to this layered approach, the different reconfigurable schemes of the literature can be regarded as different interplays between the two layers. A new interplay, which achieves some appropriate optimizations for mobile terminal systems, is proposed and analyzed. This new interplay is the iterative approach, based on the computational similarities and the iterative nature of the supporting algorithms.

This Chapter is organized as follows: Section 4.2 introduces the need for a flexible radio and Section 4.3 gives some definitions of the reconfigurability concept. The proposed two-layer reconfigurability approach and different interplay schemes are presented and analyzed in Section 4.4, followed by concluding remarks in Section 4.5.

4.2 The need of a flexible radio

The radio of the 2G telecommunication systems was designed for the worst operational case. This traditionally designed radio approach, which is popular with the term Hardware-Defined Radio (HDR), is based on a fixed in function hardware computational core, which corresponds to the worst operational scenario [BUR03]. The used implementation technology is Application-Specific Integrated Circuit (ASIC), which is based on a dedicated and fixed in function hardware for each application. It requires a long time to design, test and produce the dedicated circuit and, after the

development of the final system, it does not permit updates or future functional changes. Thus, the developed systems have a short life and are designed to be discarded and replaced. Moreover, systems based on the HDR approach are characterized by a high cost and poor performance, due to their continuous operation with the worst computational configuration. The operational environment of the wireless systems is a statistical parameter and changes dynamically. The operation of a system with the worst configuration introduces a useless power consumption for the favorable operational cases, and a poor performance for the operational cases which are not supported by its pre-loaded configuration.

As we move towards the 3rd Generation of Mobile Communications, the need for lower-cost network elements, that can continuously provide, a high Quality-of-Service (QoS) to the mobile user, becomes increasingly important. Thus, the real-time joint optimization of performance and computational power is introduced as an urgent open problem. The flexibility concept, which aims to move physical and network layer processing into a programmable environment, seems to be an attractive solution to this problem.

Flexibility can be envisioned as a desirable property (or set of properties), which enables the system to respond to various changes in requirements or specifications, present or future. These can be service or user requirements and their related attributes (data rates, QoS, latency constraints, etc.) [GAR00b], “environmental” conditions (e.g., system dynamics, channel changes, mobility, other user interference, other-system interference, etc.), or system conditions (operating band) [PAL00]. The user, the operator (or system) and the channel seem to be the three actors that affect the general operation, and each can affect it independently; thus, flexibility must be able to accommodate any such circumstances. In the 2G systems, flexibility is limited to a type of dynamic power control (in IS-95 the handset can dynamically adjust the transmitted power level needed to achieve a given link) and a change of the physical parameters during the initial call setup. For the 3G telecommunication systems, flexibility goes beyond of a local real-time parameter change and is enlarged to standard, network and technology changes [DEM04]. It gives to the system a high degree of possible real-time changes in order to always provide the necessary QoS to the mobile user. These changes can be simple parameter modifications [WEB95], replacements of the used algorithms [BAR00], [SES99], but also total changes of the used standards or technologies [DRE01]. As new hardware (HW) elements and tools are continuously improved, flexibility can be considered as a new radio design philosophy [LUN01].

Flexibility scenarios

In response to the demand for increasingly flexible radio systems from industry and government, as well as various user demands, the field has grown rapidly over the last twenty years or so (perhaps more in certain quarters), and has motivated numerous research projects [E2R], [WINDFLEX]. Because of the enormity of the subject matter, it is hard to draw solid boundaries that exclusively envelop the scientific topic, but it is clear that terms such as Software Radio (SR), Software Defined Radio (SDR) [MIT95, MIT99], reconfigurable radio, cognitive/intelligent/smart radio [MIT99b], [PAL03], etc., are at the center of this activity. Similar arguments would include work on flexible air-interface waveforms and/or generalized (and properly parameterized) descriptions and receptions thereof. Furthermore, an upward look (from the physical-layer “bottom” of the communication-model pyramid) reveals an ever-expanding role of research on networks that

include reconfigurable topologies, flexible medium-access mechanisms, inter-layer optimization issues, agile spectrum allocation and so on. In a sense, *ad hoc* radio networks [COR99] fit the concept, as they don't require any rigid or fixed infrastructure. Similarly, looking "down" at the platform/circuit level, we see intense activity on flexible platforms that are best suited for accommodating such flexibility. In other words, every component of the telecommunication and radio universe can be seen as currently participating in the radio-flexibility R&D work, making the field exciting as well as difficult to describe completely.

Among the many factors that motivate the field, the most obvious seems to be the need for multi-standard, multi-mode operation, in view of the proliferation of different, mutually-incompatible radio standards around the globe [BOG02], [SHE99]. The natural desire for having a single end device handling this multitude in a compatible way is then at the root of the push for flexibility. This would incorporate the desire for "legacy-proof" functionality, i.e., the ability to handle existing systems in a single unified terminal (or single infrastructure access point), regardless of whether this radio system is equipped with all the related information pre-stored in memory or whether this software-downloaded to a generically architected terminal. In a similar manner, "future-proof" systems would employ flexibility in order to accommodate yet-unknown systems and standards with a relative ease (say, by a mere re-setting of the values of a known set of parameters), although this is obviously a harder goal to achieve than legacy-proofness.

There are many possible ways to exploit the wide use of a single flexible baseband transceiver, either on the user side or on the network side. One scenario could be the idea of location-based flexibility for either multi-service ability or seamless roaming. A flexible user terminal can be capable of reconfiguring itself to whichever standard prevails (if there is more than one that can be received) or exists (if it is the only one), at each point in space and time, either to be able to receive the ever-available (but possibly different) service or to receive seamlessly the same service. Additionally, the network side can make use of the future-proof reconfiguration capabilities of its flexible BSs for "soft" infrastructure upgrading: each BS can be easily upgradeable to each current and future standard.

Another interesting scenario involves the combined reception of the same service via more than one standard, in the same terminal. This can be envisaged either in terms of "standard selection diversity", according to which a flexible terminal will be able to download the same service via different air-interface standards and always sequentially (in time) select the optimum signal (to be processed through the same flexible baseband chain); or in terms of service segmentation and standard multiplexing, meaning that a flexible terminal will be able to collect frames belonging to the same service via different standards, thus achieving throughput maximization for that service, or receive different services (via different standards) simultaneously. Finally, another flexibility scenario could involve the case of peer-to-peer communication whereby two flexible terminals could have the advantage of reconfiguring to a specific PHY (according to conditions and optimization criteria) and establish a peer-to-peer *ad hoc* connection.

4.3 Definitions of radio flexibility

Flexibility, adaptivity and reconfigurability (FAR) have been entertained extensively in the broad area of wireless (radio, mostly) systems and networks, occasionally with overlapping meanings,

visions and manifestations [POL03]. Instead of attempting to sort through the literature and reconcile competing visions, philosophies and terminologies, a set of terms and notions is proposed in the Appendix A, which can serve as the basis of the present discussion. The guiding principles in adopting this set have been consistency, completeness, and adherence to commonly-used language terms and concepts. Thus, the central notion of flexibility is defined as an “umbrella” concept, encompassing a set of independently occurring (design) features, such as adaptivity, reconfigurability, modularity, etc., such that the presence of a subset of those would suffice to attribute the qualifying term “flexible” to any particular system under consideration. The absence of all of these features would make the system inflexible.

If we wish to explain the flexibility features further, we may say that a system is adaptive if it can respond to environmental changes by properly altering the numerical value of a set of parameters [CHI99], [GOL97], [LET98]. It is reconfigurable if it can be rearranged, at a structural or architectural level, by a non-quantifiable change in its configuration [DAG03] (see Appendix A). Here, “non-quantifiable” means that it cannot be represented by a numerical change (for example, the structural change of going from a serial concatenated turbo code to a parallel turbo code cannot be represented by a change in a numerical quantity; similar, the Intermediate Frequency (IF) stage by digitization and software-controlled processing, as in pure software-defined radio, cannot be represented by a numerical change). Clearly, certain potential changes may fall in a gray area between definitions. For instance, changing the number of subcarriers in Orthogonal Frequency-Division Multiplexing (OFDM) may appear as an adaptive change (it is quantifiable), but because it has structural implications at the FFT and other levels, it may also be considered a structural reconfigurable change [DAG05]. Finally, a system is modular if it can divide its processing to separate essential modules with a well-defined connection. Software radio, for example, is meant to exploit reconfigurability and modularity to achieve flexibility. Fig. 4.1 presents the three flexibility features, using an example from the geometry domain. Thus, the change in the dimension of a square is defined as adaptivity. The transformation of a square into a circle is defined as reconfigurability and the division of a square into three smaller geometric schemes, is defined as modularity.

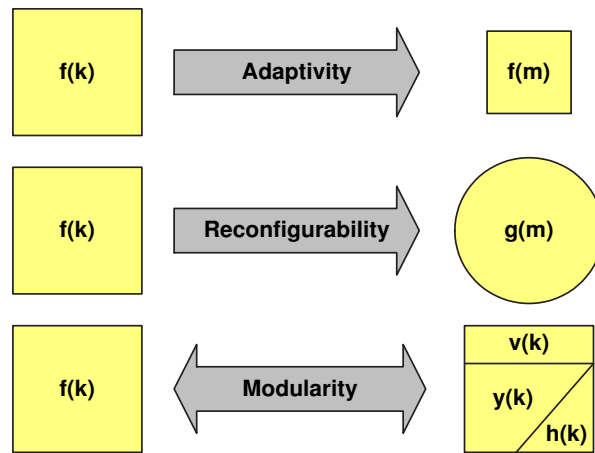


Figure 4.1: Adaptivity, reconfigurability and modularity in the geometry domain.

The flexibility features are termed “independent” in the sense that the occurrence of any

particular one does not predicate or force the occurrence of any other. For example, an adaptive system may or may not be reconfigurable; etc. In a sense, flexibility resembles a vector in an imagined Euclidean space where each such attribute represents an “orthogonal” coordinate of this space, defining the degree to which a system is adaptive, reconfigurable, modular, etc., and thus the degree to which it is flexible in that sense. Fig. 4.2 presents the imagined Euclidean space of flexibility.

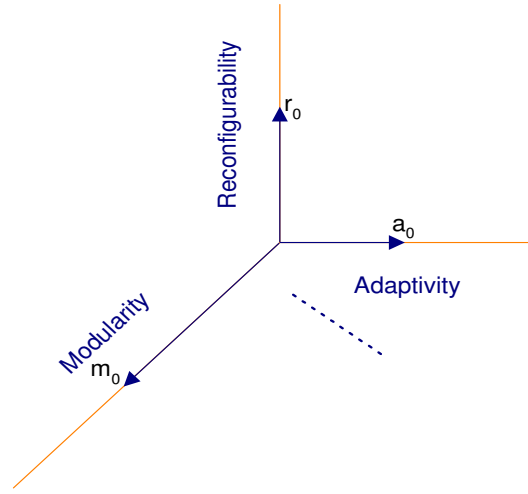


Figure 4.2: The imagined Euclidean space of flexibility. The vector $[a_0, r_0, m_0, \dots]$ presents the degree/type of flexibility.

Clearly, flexibility may include more such coordinates, and these should be defined if possible, under the caveat that they don’t overlap with previously defined features. They should add something “new” to the lot, and they should also be easily identifiable by inspection of the system. For example, it may be desirable to include concepts such as “ease of use”, or “seamlessly operating from the user’s standpoint” into the broader notion of flexibility. To the degree that these can be quantified and identified in a straightforward way, they are useful and should be included in the expanded definition.

A primitive example of flexibility, as we presented in the flexibility scenarios subsection, is the multiband operation of the mobile terminals. Although this kind of flexibility driven by the operator is not of great research interest from the physical-layer point of view. A more sophisticated version of such a flexible transceiver would be more the one that has the intelligence to autonomously identify the incumbent system configuration and also has the further ability to adjust its circumstances and select its appropriate mode of operation accordingly. The reconfigurability dimension of flexibility is the principal notion of this thesis.

4.4 The two-layer reconfigurability concept

In this Section, we deal with one attribute of the general flexibility concept, the reconfigurability, from the physical layer point of view. However, in practice it is difficult to deal with the reconfigurability without involve the other flexibility features (adaptivity, modularity). Using the imagined Euclidean space of flexibility, we can say that for the following study, we are placed very close to the reconfigurability axe but not exactly on it. The adaptivity or the modularity or both of them, are always presented.

In order to focus on the reconfigurability concept and simplify its study, we don't use reconfigurability to support multiple standards and technologies, but as a means to optimize, at run time, the communication link of a specific standard.

Reconfigurable systems do not just incorporate all possible point solutions for delivering high QoS under various scenarios, but possess the ability to make changes not only on the physical parameters, but also on the structural level, in order to meet their goals. Thus, the reconfigurability goal is to bring the classic design procedure of the PHY layer into the intelligence of the transceiver and initiate new system architectural approaches, capable of creating the tools for on-the-fly reconfiguration. In order to further explain the reconfigurability concept and facilitate its study, we regard reconfigurability as a two-layer concept. The first one, called algorithmic reconfigurability, refers to the ability of the system to dynamically change algorithmic schemes in order to support some well-defined optimization criteria and constraints. The second layer, called hardware reconfigurability, refers to the ability of a system to dynamically change the implemented functions in order to support the algorithmic change, also under a set of optimization criteria and constraints. The communication between these two layers is an important framework of the layered reconfigurability concept. The design of any reconfigurable system requires the definition of these two layers and the interplay between them. Fig. 4.3 schematically presents the structure of a two-layer reconfigurable system.

4.4.1 Layer 2: The algorithmic reconfigurability

In contrast with traditional inflexible design, reconfigurable system design requires a continuous monitoring of the operational environment and introduces new logical elements that can perform the various related optimization procedures. Based on inputs from higher layers (QoS on demand) and real-time measurements of the operational environment, this reconfigurability layer is responsible for generating optimized directives to the appropriate reconfigurable functional blocks, according to an appropriately designed inference algorithm. This algorithm can be regarded as a new logical block, called supervisor (SPV). Devising such optimization algorithms for the SPV has been a challenging task, not only because it must perform joint optimization of a number of different parameters at run time, according to various requirements/constraints, but also because it should be kept simple and efficient enough to be successfully implemented in cost-effective power-limited low-complexity wireless terminals.

SPV receives as inputs the algorithmic set, which includes the available algorithmic schemes, the optimization criteria and constraints, and the dynamic system parameters from the PHY layer. Its intelligent task is to analyze the input data and to decide the appropriate response for each functional block of the PHY layer. Fig. 4.4 presents the SPV block of the algorithmic

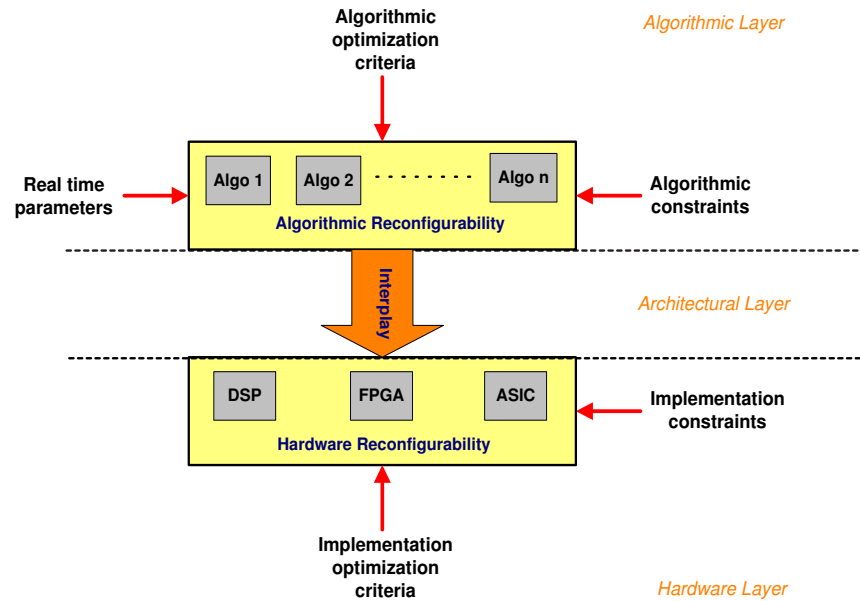


Figure 4.3: The general structure of a dynamic two-layer reconfigurable system.

reconfigurability which concentrates all the self-intelligence of a reconfigurable system.

The optimization criteria represent the reconfigurability goal. They are a combination of algorithmic and hardware ones [HAN02], [WON00]. The use of more optimization criteria provides a better solution, but in the same time increases the computational complexity of the SPV. The algorithmic optimization criteria can be

- Maximum capacity
- Maximum throughput
- Minimum computational power
- Minimum BER

On the other hand, the hardware optimization criteria can be

- Minimum area overhead
- Minimum power consumption

Furthermore, the real-time PHY parameters can be system parameters, which are defined by the transmitter, or external parameters like the channel environment and the interference. The external parameters are not known a priori and must be estimated by the reconfigurable system. The selection of the real-time parameters under monitoring depends on the algorithmic set.

Finally, the constraints can be algorithmic and hardware ones. The algorithmic constraints, in most of the cases, are time constraints which are introduced by each real-time application. For example, in a block-based communication link, the block duration is a time constraint for the detection schemes. On the other hand, hardware constraints are related with the available

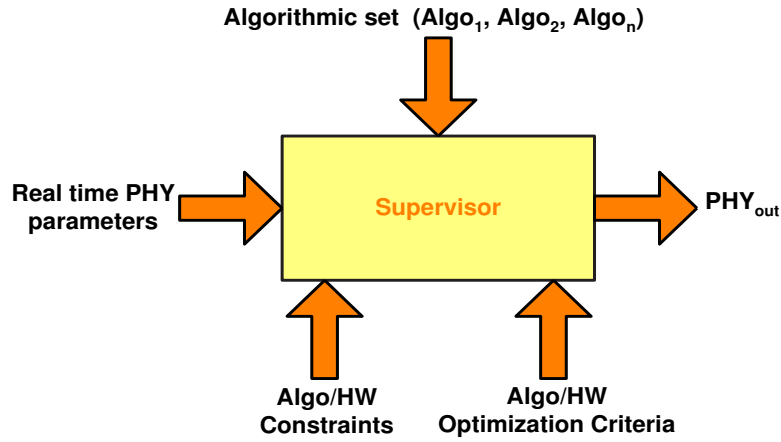


Figure 4.4: Supervisor inputs and outputs.

hardware resources and can be maximum computational power, maximum speed, maximum computational resources and so on. The constraints parameters must be chosen so as to satisfy the target application requirements as well as the implementation feasibility. The use of less constraints provides a better solution, but in the same time increases the computational complexity of the SPV. On the other hand, the use of more constraints provides a lower-performance solution but it significantly reduces the SPV computational complexity.

The most difficult aspect of the design of a SPV block is the definition of the algorithm which analyzes the input information and returns the optimal combination of the different functional blocks to the PHY layer. There are two different approaches for dealing with the definition of this algorithm. The first one supposes that the SPV can access static data from an internal Look-Up Table (LUT), which usually describes the performance of the different algorithmic schemes. Thus, the SPV logic is simplified to a search of a pre-existing data table which maps each operational condition to the optimal functional configuration [SMI01]. It is obvious that this method requires an a priori simulation study of the target system and the storage of a data table which in some cases can have a large dimension. The second approach provides the SPV with the necessary logic in order to analytically calculate the performance of each possible configuration response. This method can be very complicated for the cases where the analytical performance calculation requires a significant number of complex mathematical operations [HAN02].

To qualify this layer further, there are two different types of algorithmic reconfigurability. The first one, ordered reconfigurability, consists of a simple rearrangement of the basic functional blocks of the system diagram. There is no change in their self-functionality but only in the order of their logical connectivity, which corresponds to a different algorithmic operation. The ordered reconfigurability can be formulated as

$$F_i = h_i(P_1, P_2, \dots, P_K), \quad (4.1)$$

where $1 \leq i \leq n$, n is the size of the algorithmic set, h_i denotes the logical connectivity of the K basic functional blocks, which corresponds to the i -th algorithmic scheme F_i , P_j is the j -th basic functional block ($1 \leq j \leq K$).

The second type of algorithmic reconfigurability is the structural reconfigurability. This type

of change implies a new definition of some basic functional blocks of the system. In this case, we redefine not only their logical connectivity but also some of the basic blocks themselves. The structural reconfigurability can be formulated as

$$F_i = h'_i(P_1^i, P_2^i, \dots, P_K^i), \quad (4.2)$$

where $1 \leq i \leq n$, n is the size of the algorithmic set, h'_i denotes the logical connectivity of the K basic functional blocks, which corresponds to the i -th algorithmic scheme F_i , and P_j^i is the j -th basic functional block used for the i -th algorithmic scheme.

4.4.2 Layer 1: The hardware reconfigurability

This layer consists of reconfigurable modules called “processing modules”, that perform the selected functionality of the algorithmic layer. The processing modules have the ability to be modified, in order to perform different functions at different times. Each processing module has a different mechanism to be changed and thus the term modification characterizes any type of change, from a simple change in the data flow (e.g, DSP) to a total change of the implemented circuit (e.g, FPGA).

It is important to note that the hardware technology, based on CMOS, is continuously progressing (the number of transistors doubles every 2 years), according to Moore’s Law. This evolution will cease when the semi conductor physical limits are reached in the 2010s. Developers must often anticipate this evolution for long-term projects. For instance, the characteristics of the 90nm or 65 nm CMOS technology are taken into account for 2006 chip designs even if the current 2005 technology is 130nm.

For the processing modules there is a variety of possible solutions. The appropriate one is that which satisfies the optimization criteria and constraints, and which can support the demanding functionality of the algorithmic layer. In the following paragraphs, we give a brief description of the different circuit technologies.

DSP resources

Traditional General Purpose Processors (GPPs) use the classical Von Neumann memory architecture, as shown in fig. 4.5a. The single data bus causes a bottleneck in the system by only allowing either new instructions or data to be fetched from external memory and loaded into the CPU [HEN96]. DSP chip designers in the 1980s realized that GPP computing architectures could be improved upon to suit high-speed signal processing, by providing them the ability to load operands at the same time as instructions are fetched.

More specifically, DSP chips avoid the instruction and data contention by employing the Harvard architecture, as shown in Fig. 4.5b. By using two address buses and two data buses, each connected to its own piece of external memory, it is possible for new instructions to be fetched at the same time as new data. This allows for effective pipelining, where instructions for the next series of data can be loaded at the same time as operations are performed on the current set of data.

The other big change with the development of a DSP-centric architecture is the ability to perform the multiply and accumulate functions in a single clock cycle. GPPs without a dedi-

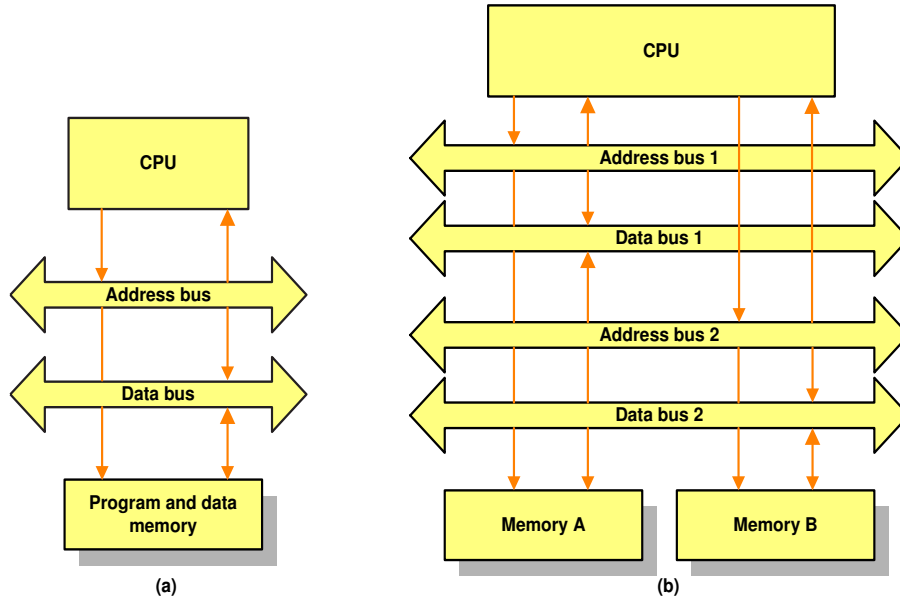


Figure 4.5: (a) Von Neumann memory architecture, (b) Harvard memory architecture.

cated multiplier require many shift and add operations to achieve the same result, consuming precious clock cycles. Many communications-related signal processing algorithms are both multiply and accumulate (MAC)-intensive and repetitive, where a relatively small set of instructions is performed over and over in tight loops.

Despite these significant improvements over GPP computational units, DSPs remained limited for performing the growing applications in the multimedia and communication areas, due to their serial processing of the required computations. Since the mid 1990s, the DSP chips take advantage of parallel architectures based on a set of parallel execution units, using the Single Instruction Multiple Data (SIMD) or the Very Long Instruction Word (VLIW) approaches. The first one to appear on the market was the TI C6200 in 1997 [TI]. The idea is to consider a large instruction word composed of a set of parallel instructions, each associated with a specific execution unit of the processor. Much work has also been done on compilers in order to take profit of the parallelism. It remains however preferable to use an optimized library, written in assembly code, in order to get the maximum performance. Many other techniques have been used to improve speed or reduce memory size. For instance, the subword parallelism concept allows the designer to get two 16-bit operations instead of one 32-bit operation which allows for doubling the speed or dividing the memory by two, if 16-bit data width is enough.

For 3G purposes, some coprocessors can also be added to the DSP cores, as a Viterbi or turbo channel decoder. An example of the DSP technology, the TigerSHARC from Analog Devices, has been used in the previous Chapter (3.2.1).

ASIC

ASIC technology refers to a full-custom or semi-custom integrated circuit designed for a specific application [BAN93], [SMI99]. The semi-custom design process uses existing cell libraries. Hence the hardware designer has to produce a routed standard cell netlist from a high-level circuit

model. The models are usually described with a hardware description language such as VHDL or Verilog [DOU02]. A synthesis tool is then used to quasi-automatically generate a netlist, which is then placed and routed using the standard cell characteristics. The final step is to send a tape containing the technology masks to the foundry. Compared to DSP solutions, applications developed in ASIC chips are much faster, cost-effective and less power-consuming. The main drawback is the development cost, time and the lack of flexibility compared to a software solution. To be more cost-effective and flexible, the ASICs can integrate RISC and/or DSP cores to make a System On a Chip (SOC). This is generally the case for mobile phone chip sets, where at least two processors are used in the specific Integrated Circuit (IC). To reduce the development cost and the time-to-market constraints, new tools and methods are continuously investigated. They are mostly based on the reuse of existing designed blocks and abstract models, which allows for rapid validation of the whole system. For instance, System C is a C++ package which allows the designers to describe and simulate concurrent and technology-independent models. It makes it easy to find errors, but also to help partitioning between hardware and software [BLA04].

FPGA resources

A typical FPGA device consists of an array of identical logic cells surrounded by configurable routing [HAU98]. The cell functions and the interconnects between the cells are programmable by memory points. The memory technology is generally SRAM (so volatile) but some FPGAs use anti-fuse (one-time programmable) or FLASH technology [BRO92]. SRAM cells are based on LUTs followed or not by a D-Flip Flop (DFF) to implement sequential logic. FPGAs also have embedded SRAM blocks to process data blocks locally. For DSP applications, most FPGAs have Multiply and Accumulate hardware blocks (DSP blocks). Other hardware blocks, such as RISC cores or rapid I/O interfaces, can be integrated in FPGA architectures. The design method is the same as for ASIC technology, except that the resources are limited by the number of logic cells, routing capacity and I/Os. Fig. 4.6 illustrates a typical FPGA device. In this figure we can see the restricted matrix and the peripheral pads which provide connection of the FPGA to other electronic devices.

Traditionally, FPGAs have been used for prototyping designs before implementation in fixed-function ASICs. Moreover, they are widely used as an alternative to dedicated ASIC designs, as their performance is comparable for many applications, with a development cost far less than dedicated silicon. However, for large volume production it is preferable to move towards an ASIC design, as the diagram of Fig. 4.7 shows. The threshold to move from commercial FPGA to ASIC design is generally between 1000 to 10000 pieces/year.

New methods of configuration are currently available on FPGA devices, which allow relatively fast download of configuration information. This is, however, slow in comparison to typical processing speeds of the configured logic. Partial reconfiguration is another important enhancement to FPGAs [MES03]. Sections of the FPGA logic can be reconfigured without interrupting any processing being carried out on other parts of the same FPGA. This is not particularly fast for commercial FPGAs but it could be considered for embedded custom FPGAs. An example of the FPGA technology, the Virtex 4 from Xilinx, has been used in the previous Chapter (3.3.4) of this thesis.

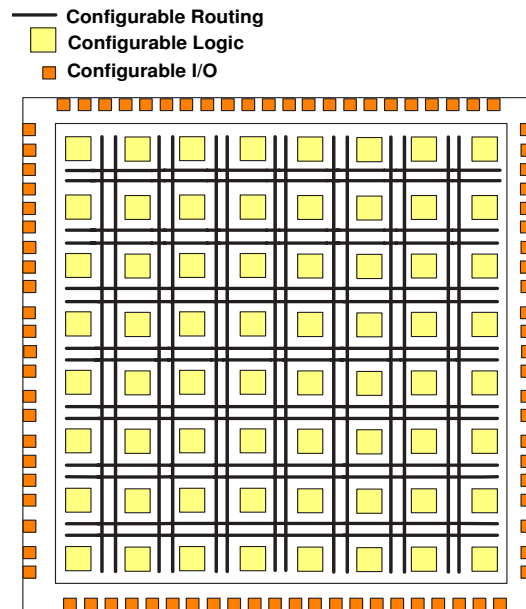


Figure 4.6: Typical FPGA architecture.

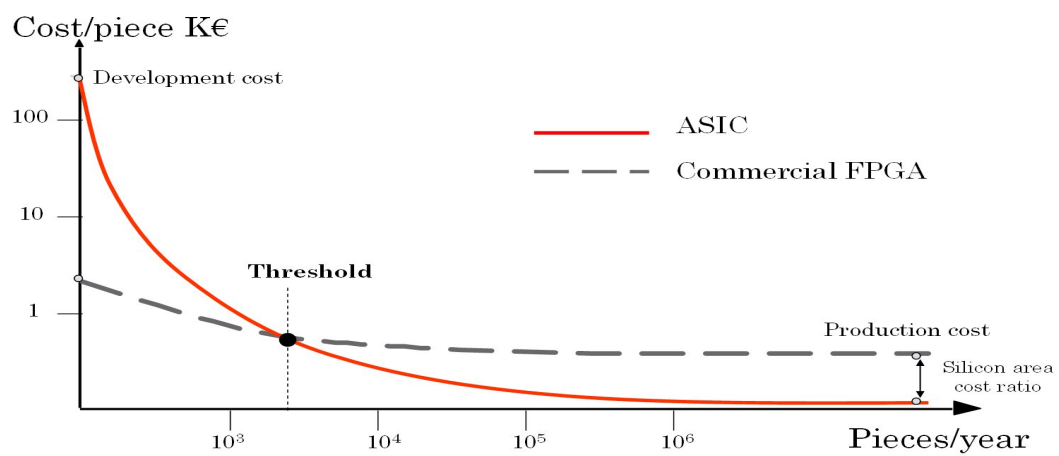


Figure 4.7: The behavior of the implementation cost as a function of production volume.

Configurable Computing Machines

A Configurable Computing Machine (CCM) uses coarse-grain custom FPGAs and is regarded as an extremely powerful processing engine with ASIC-like speeds, which can also be rapidly reconfigured. It is an optimized FPGA with application-specific capabilities [HAU97], [HEY02], [SRI03]. It attempts to customize the FPGA so that system flexibility is retained while taking advantage of the specific properties of communication-oriented cores. Due to the coarser granularity in its fundamental composition, which matches the applications (signal processing/communications) under consideration, its speed is higher than that of commercial FPGAs.

As for the structure of a CCM device, it has a static hardware for frequently used cores, such as multiplication, filtering, or other communication-oriented algorithms, which results in efficient radio designs. Other features that are typically used to enhance the CCM could be strategically-placed shift registers, circular buffers, a large number of I/O pins for good data throughput, and so on. These components and the connections between them are programmable to allow for reconfiguration.

CCMs can be designed as stand-alone hardware or as accelerators for GPPs or DSPs.

Reconfiguration Computing

In order to exploit the best of all technologies (cost, speed, power consumption, flexibility) it is worthwhile to gather processor/software and hardware. This is not a new concept. A common example in the Personal Computer (PC) industry is the use of graphic boards to accelerate graphical transformations.

An hybrid architecture can be any combination of the previous HW computational units. An interesting combination is the connection of a DSP (or GPP) with a commercial FPGA device. In the reconfigurable architectures area, this type of hardware device, is generally known under the term Reconfigurable Computing (RC) [ATH93], [HAU97a], [OLU94]. Fig. 4.8 presents the general structure of an RC system.

In RC devices, DSPs handle the main flow of the algorithm and control the FPGAs that handle computationally-intensive repetitive operations. In some cases there is a confusion between RC and parallel computation. RC systems and parallel computing systems have the same objective: to speed-up the execution of a given application. For parallel computations, acceleration is achieved through the exploitation of the parallelism in a program, and its mapping onto an architecture of various processors, but for RC it is achieved through the migration of the most computationally-intensive parts of the program to the hardware elements. It is obvious that, in order to increase the performance of the traditional RC approach, customized FPGAs and CCMs can be used instead of the commercial FPGAs.

Comparison

ASICs provide the most optimized hardware implementation of an algorithm, as they are based on a dedicated fixed in function circuit for the application under consideration. However they are not flexible. In order to provide the reconfigurability property, they require the multiplexing of the dedicated circuit for each supporting mode. Using a dedicated ASIC for each mode of the radio leads to a large form factor or very large silicon area.

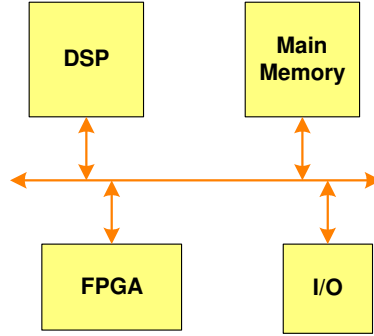


Figure 4.8: A hardware device added to an existing architecture for acceleration purposes.

	DSPs	ASICs	commercial FPGAs	RCs	CCMs
Performance	*	****	**	**	***
Power consumption	*	****	**	**	***
Area	*	****	**	**	***
Flexibility	****	*	***	***	***
Development Time	****	*	**	**	*

Table 4.1: Comparison of the HW implementation alternatives.

On the other hand, DSPs have excellent flexibility and programmability, but today's DSPs alone cannot handle very complex algorithms at the required speed with reasonable power consumption.

FPGAs use hardware reconfiguration, which allows implementation of complex high-speed algorithms. They can present the best trade-off between performance and flexibility if their structure is adapted to the applications. Existing fine-grain commercial FPGAs are not cost-effective. Compared to an ASIC, the same functionality needs between 20 to 50 times more silicon in an FPGA. However, they are well-suited for fast prototyping.

RC devices combine the high degree of flexibility of the DSP with the high performance of the traditional FPGAs. However, they can not overtake the performance of the commercial FPGAs.

Finally, CCMs try to give flexibility to the ASIC technology. Their combination with DSP devices (hybrid DSP/ASIC) provides a high degree of flexibility due to the embedded DSP devices and a high performance due to the customized FPGAs. However, they require a long time to be designed, tested and produced compared to the classical ASIC circuits. Table 4.1 summarizes the basic differences between the implementation alternatives.

4.4.3 Layer 1+: The architectural reconfigurability

The connection and interplay between algorithmic and hardware reconfigurability is a promising field for future reconfigurable transceivers. This intermediate layer refers to the mapping strategy of the algorithmic reconfigurability to the implementation level and is called "architectural reconfigurability". It represents the architectural changes that must be performed in order

to load, to the processing modules of the hardware layer, the new configuration that has been decided at the algorithmic layer [CHA01], [HAL98], [HAS03]. The architectural intermediate layer concentrates on the implementation intelligence and logic, and defines how an algorithmic reconfigurable decision will be mapped to the available implementation elements. It is a function of the nature of the processing modules (SW, HW), of their special characteristics (speed, size), but also of the computational structure of the supporting algorithms (iterations, computational overlaps). The architectural layer defines an important parameter of the reconfigurable system, which is the reconfiguration time. The reconfiguration time is the required time to perform a change in the implemented function. Different architectural methodologies correspond to different reconfiguration times. It is obvious that for real-time applications this parameter is very critical.

Moreover, the architectural reconfigurability defines the cost and the used computational resources of the processing modules. Different architectures correspond to different costs and complexities. The appropriate one is that which best uses the characteristics of the available processing elements, optimizes their hardware resources, maps the algorithmic reconfigurability and introduces a reconfiguration time which is tolerable by the application constraints.

In the following paragraphs, we present the major mapping strategies of the literature. Among them the first three approaches are a function of the nature of the reprogrammable devices. With the term nature, we mean the way of reprogrammability. The last two, are a function of the computational structure of the supporting algorithms.

Functional approach

It is a software-oriented mapping strategy which characterizes the DSP implementations [BAI95]. It is a trivial method which consists of programming a special function for each algorithm of the algorithmic set. In this case, reconfiguration is simplified to a conditional (*if ... then*) structure where each condition is a function of the real-time system parameters. Whenever a condition evaluates to a true value, the sequence of functions which corresponds to this operational environment is executed. Fig. 4.9 presents the software approach. All the supporting algorithms are stored in memory, in the form of a software function, and according to the well-defined dynamic selection conditions, in each time, the CPU executes the appropriate one. Despite its simplicity and implementation easyness, this reconfigurable mapping is used for a low number of applications due to the speed limitation of the current DSPs. The basic reason for this bottleneck is that a CPU processes the data in a serial or a poorly parallel way, with a speed which corresponds to the DSP cycle frequency. This processing requires also a high number of accesses to the external memory, and thus the achieved performance is very poor and inefficient to support the current high-rate applications.

Switching approach

It is the application of the functional approach to an ASIC-type device [DON99]. This approach supports ordered and structural nature algorithmic reconfiguration. As the previous approach, it is very general and has no limits in the supporting of the different algorithmic schemes. Each configuration mode is considered as a fixed-function hardware circuit. We can say that this method is generated from the functional approach by transforming the memory stored-functions

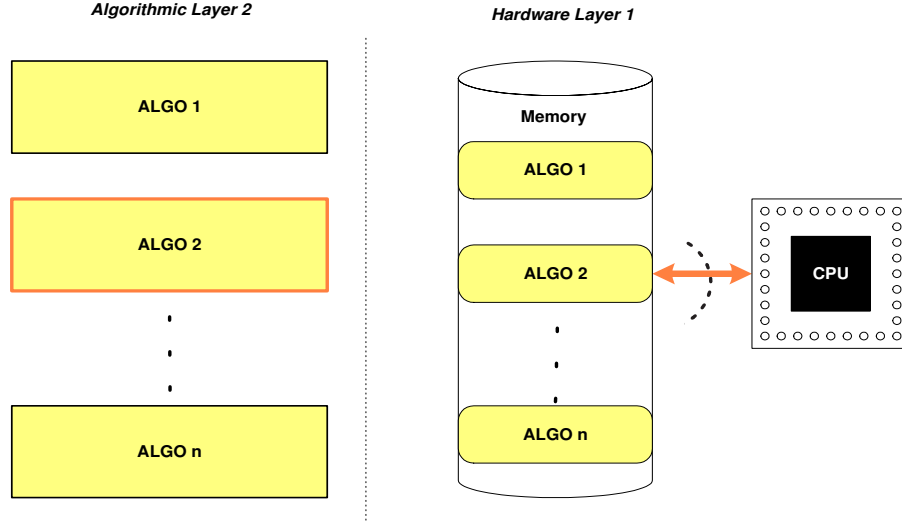


Figure 4.9: DSP mapping approach.

into separate pre-existing hardware functions, and thus the mapping strategy is simplified to a mere switching between them. Therefore, it is very simple to be implemented and the introduced reconfiguration time is negligible.

However, this approach requires a dedicated circuit for each case, which leads to a high power consumption and a large form factor, or very large silicon area. This result gives to the switching approach a poor degree of flexibility and limits the number of supported operational cases. Moreover, it is characterized by an inefficient use of the hardware resources, since in each operational case a large part of the hardware is switched off. Fig. 4.10 presents the switching approach schematically. We can see that each function of the algorithmic set is represented as a separate pre-implemented circuit in the HW device.

Hardware paging

This approach also supports ordered and structural algorithmic reconfigurability and is FPGA-oriented. The mapping strategy is based on the idea that only the function in use is on the hardware, while the other functions are stored in the memory. It enables a swapping of functions in and out of hardware, similar to what is done in personal computing software with the use of virtual memory [SRI00, SRI00a, SRI00b]. The mapping strategy and the variety of algorithmic schemes give to this approach a high degree of flexibility. Moreover, the absence of switching off hardware optimizes the use of the available hardware resources, as power consumption and material surface. However, this approach maximizes the critical parameters of the hardware reconfiguration. The time for a full reconfiguration and redefinition of the hardware resources is very critical and in some cases can not be supported by the commercial processing modules (FPGAs). Moreover, the reconfiguration logic which supervises and reprograms the processing modules is highly complex. Thus, this approach is characterized by a high degree of flexibility, but also by a high complexity of the reconfiguration parameters. Fig. 4.11 presents the hardware paging approach. We can see that the different algorithms of the algorithmic set are represented by HW configurations, which are stored in the memory. In each case, the configuration which

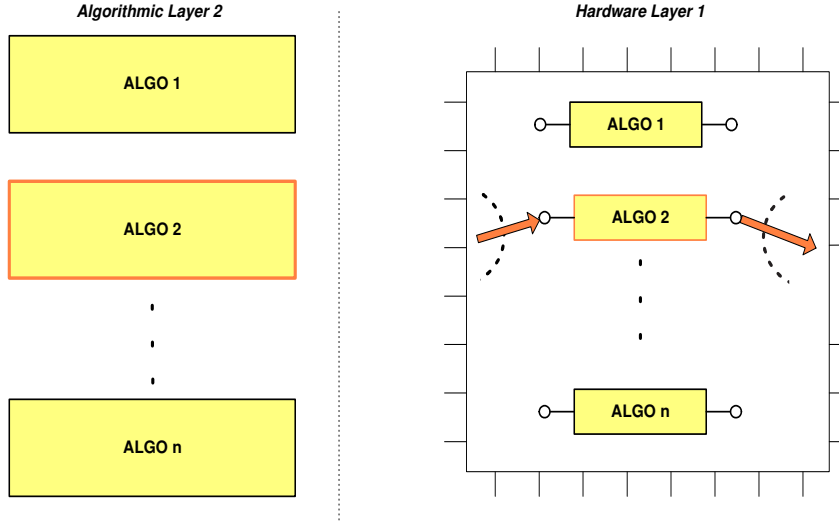


Figure 4.10: The switching mapping approach.

corresponds to algorithmic reconfigurability is loaded on the HW device.

Factorization approach

To overcome some of the practical implementation issues, this approach moved away from the ideal software radio, where a single hardware device is reconfigured to any algorithm. In contrast with the previous two approaches, this one deals only with the ordered algorithmic reconfiguration and is HW-oriented. This corresponds to supporting algorithms with computational and functional overlaps. Thus each algorithm can be seen as a different logic connection between the same essential functional blocks or a sub-group of them [BRA02], [GRA00]. This property can be formulated as

$$F_i = f_i(P_1, P_2, \dots, P_K), \quad (4.3)$$

where $1 \leq i \leq n$, n is the size of the algorithmic set, F_i denotes the i -th algorithmic function, P_i is the i -th basic functional block and K is the total number of them. The function f_i denotes a serial order of a subset which is generated from the set (P_1, P_2, \dots, P_K) .

The supporting hardware consists of a set of basic operators, where each one implements one basic functional block (P_i) for $i = 1 \dots K$, with a reprogrammable connection corresponding to the different logic connections under consideration. Thus, we can say that the factorization approach besides the reconfigurability concept is based also on the modularity concept. An algorithmic change corresponds to a logic change in the connection of the basic functional blocks, and the mapping strategy is simplified to select the appropriate routing between the basic hardware operators. In [GRA00], a simple computational core is proposed which can be used to realize different blocks within an OFDM transmitter and a QAM receiver. Specifically, the computational core can support seven different functionalities via a reprogrammable routing. These functionalities are real FIR, dual real FIR, complex FIR, real IIR, adaptive FIR, digital frequency synthesis and Discrete Fourier Transform.

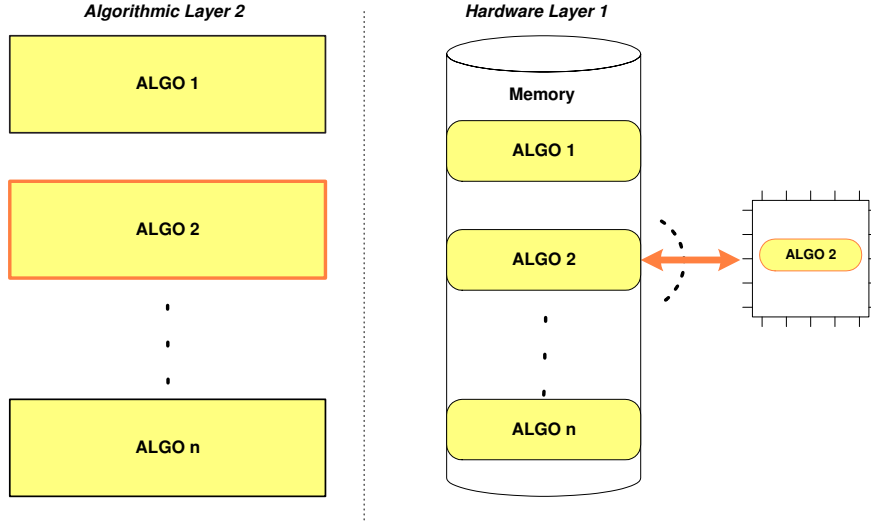


Figure 4.11: The hardware paging mapping approach.

This approach implies an appropriate algorithmic selection in order to satisfy the functional and computational similarity properties. Moreover, it gives a high degree of flexibility and an optimization of the hardware reconfiguration parameters (reconfiguration time and logic). The change of configuration is accomplished by proper utilization of some multiplexers and thus the time and the logic of the reconfiguration are negligible.

However, when the number of supporting algorithms (n) is increased, the number of non-common functional blocks in the logic chain is also increased (K). Thus, some basic operators of the supporting hardware device must be switched off and this corresponds to an inefficient utilization of the hardware resources. Fig. 4.12 presents the factorization mapping approach. Each one of the algorithms under consideration can be represented as a logic connection between the basic functionalities (P_1, P_2, \dots, P_K). These functionalities are implemented in the hardware layer with a reprogrammable connection and they consist of the basic operators. Thus, when the device works on the F_2 mode, for example the operators (P_3, P_4, \dots, P_K) are switched off.

Iterative approach

The proposed approach is a combination of the hardware paging and the factorization approach and it is also HW-oriented. It deals with the ordered algorithmic reconfiguration which in addition to the functional and computational overlap, satisfies one more property, that of computational serialization. This approach supports algorithmic schemes which, besides the functional similarity, can translate their total processing into serial processing. It is based on operators with a smaller granularity than those of the factorization approach, which corresponds to a single essential iteration of the supporting algorithmic schemes [HAR01, HAR02]. Fig. 4.13 shows the relation between the factorization and iterative approaches. We can see that the iterative operators have a finer grain than the factorization operators. The iterative approach can be formulated as :

$$F_i = N_i \cdot f'_i(P'_1, P'_2, \dots, P'_K), \quad (4.4)$$

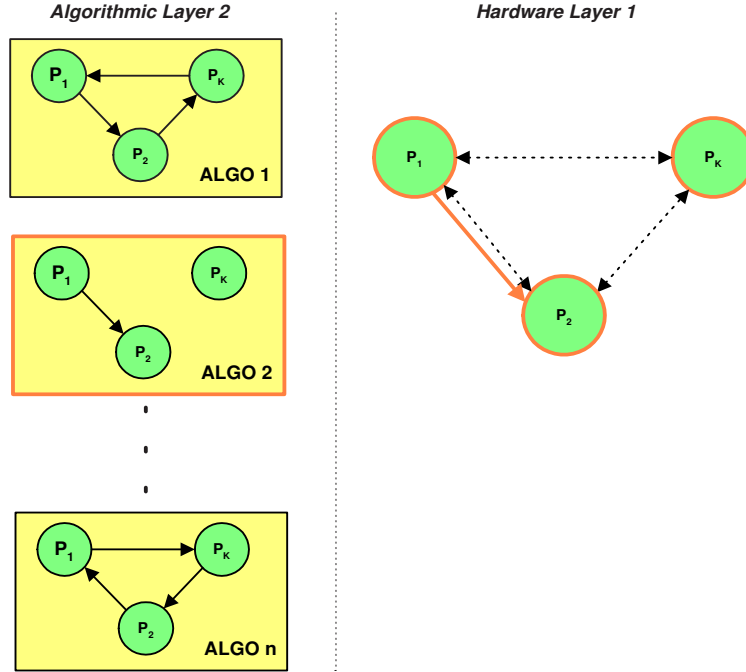


Figure 4.12: The factorization mapping approach.

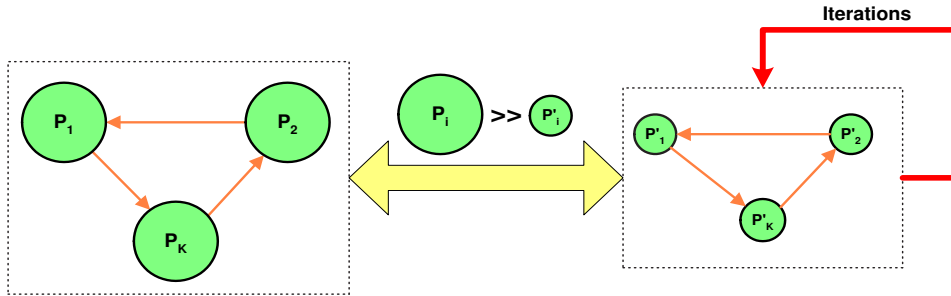


Figure 4.13: The relation between factorization and iterative approaches.

where $1 \leq i \leq n$, n is the size of the algorithmic set, F_i is the i -th algorithmic function, f'_i is a function similar to the one of the factorization approach, P'_i is the i -th basic function with a finer granularity than the P_i elements of the factorization approach, N_i is the required number of iterations of the subset under consideration and K is the total number of basic functional blocks. Thus the functional chain consists of “simpler” basic functional blocks which can represent one iteration of any algorithm of the algorithmic set, with the appropriate connection and order. These essential functional blocks $\{P'_1, P'_2, \dots, P'_K\}$ are implemented in the hardware device with a reprogrammable connection, and thus a change in the f'_i function is translated to a redefinition of this reprogrammable routing.

The iterative approach besides the reconfigurability and the modularity concepts, is based also on the adaptivity concept, as the number of iterations is adapted to the operational environment. The iterative approach combines the advantages of factorization and hardware paging. It gives a high degree of flexibility, small reconfiguration time and simple reconfiguration logic, thus it

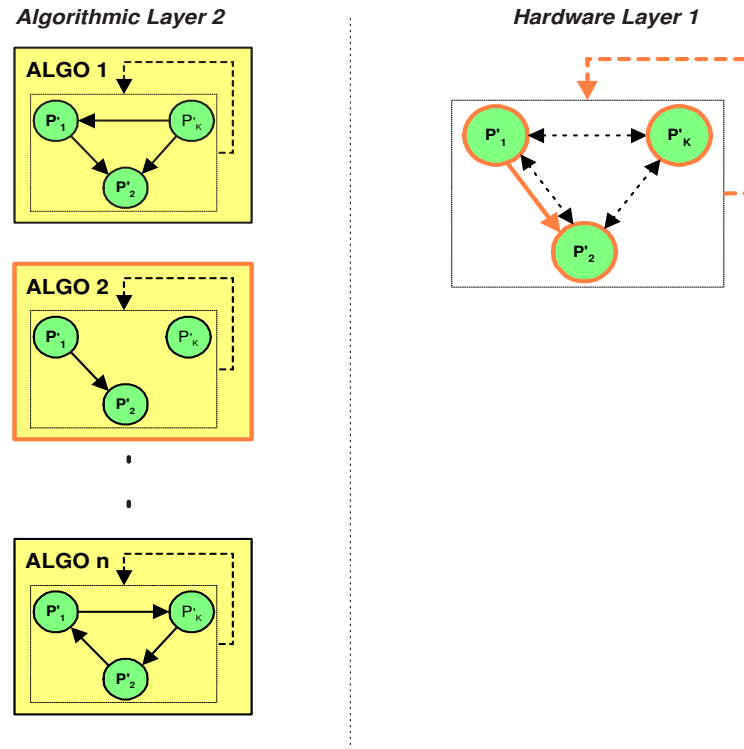


Figure 4.14: The iterative mapping approach.

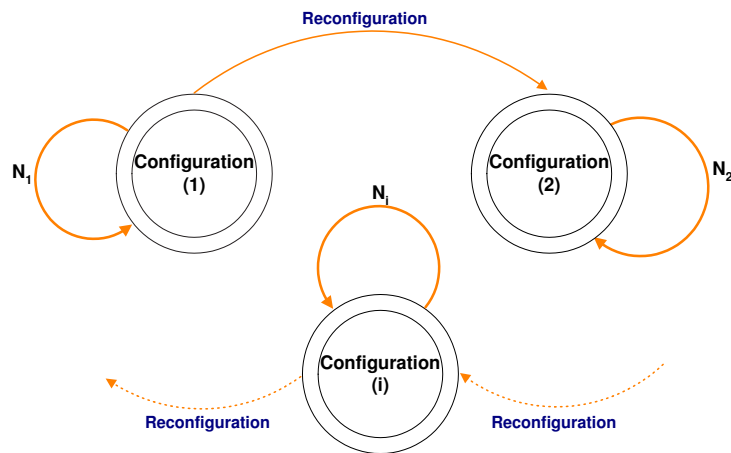


Figure 4.15: The FSM represents the iterative approach schematically.

	High Flexibility	Small Area	Fast Reconfigurability	Simple Logic
Switching	*	*	****	****
Hardware paging	****	****	*	*
Factorization	**	*	****	****
Iterative	**	***	****	****

Table 4.2: Comparison of the HW mapping approaches.

is a sub-group of the factorization approach. On the other hand, the simplification of the basic functional blocks and thus of the basic processing modules, due to the iterative nature of the total calculations, minimizes the switched-off part of hardware and approximates the hardware paging optimization. Fig. 4.14 presents the iterative mapping approach. Each one of the algorithms under consideration can be represented as an iterative use of a logic connection between the basic functionalities $(P'_1, P'_2, \dots, P'_K)$. These functionalities are implemented in the hardware layer with reprogrammable connections and they consist of the basic operators. In the selected example, when the device works on the F_2 mode, the operators $(P'_3, P'_4, \dots, P'_K)$ are switched off. The most important gain in comparison with the factorization approach, is that due to the finer granularity $(P'_i < P_i)$, a more efficient use of the hardware resources is achieved.

The limitation of this approach is that it only supports algorithms with some special and well-defined computational properties. These properties consist of the computational overlaps and iterative nature computations. However, in the domain of digital signal processing and communication, this type of algorithm is very common.

The iterative approach can be represented with the help of a Finite State Machine (FSM). Each state represents a configuration of the basic computational unit and each transition arc corresponds to a configuration change. The number of iterations for every configuration is represented by the arc which starts and finishes in the same state. Fig. 4.15 shows the FSM representation of the iterative approach. The i -th state-configuration repeats its computations N_i times.

Finally, Table 4.2 summarizes the advantages and disadvantages of the different HW mapping approaches.

4.5 Conclusion

This Chapter dealt with the iterative reconfigurability concept, which is the general notion of this thesis. Based on the two-layer reconfigurability structure, we presented the iterative reconfigurability concept as a new interplay between algorithmic and hardware reconfigurability layers. This interplay uses the iterative nature of the algorithms under consideration and their computational overlaps. It corresponds to a simple computational hardware core able to perform one single iteration for each algorithm without complicated architectural changes. A number of iterations is required to perform the totality of each algorithmic operation. The area minimization resulting from the implementation of only one iteration, and the simple logic for the algorithmic switching, makes the iterative reconfigurability concept suitable for implementations with strict

area and computational constraints, like in terminals. Finally, this approach provides a new parameter (the number of iterations) which allows the configuration supervisor to adjust the optimal trade-off between the different iterative algorithms.

Conclusions and Perspectives

The need for a continuously high quality of service to the mobile user has introduced the need for auto-intelligent terminals. These terminals have the ability, independently of the network, to adapt their functionality in real-time in order to optimize both performance and computational power. This functional change can be an algorithmic change or a change of the radio technology. In the literature, this type of auto-intelligence is represented by the terms “reconfigurability” and “adaptivity”. However, the different existing approaches to these terms, as well as their algorithmic and hardware point of view, generates confusion.

In this thesis we tried to clarify the reconfigurability concept and to categorize all the existing approaches based on a general reconfigurability framework. We regard reconfigurability as a layered concept which consists of two separated logical layers. The first layer is the algorithmic reconfigurability and represents the point of view of the algorithmic specialists. It refers to the ability of a reconfigurable system to dynamically change algorithms according to real time parameters to optimize well defined performance and complexity metrics. This layer introduces the need for an intelligent block which supervises the radio and makes the appropriate decisions.

The second layer is hardware reconfigurability and represents the point of view of the hardware specialists. In this layer, reconfigurability is translated as a piece of hardware which can be modified to perform different functions at different times, allowing the hardware to be tailored to the application at hand. This results in greatly increased speed and silicon efficiency, while maintaining a high degree of flexibility.

The interplay between these two layers and their logical connection, is the architectural sub-layer and is an important notion of the layered reconfigurability concept. It refers to a set of rules which assign the mapping of the algorithmic layer to the hardware layer. This sub-layer specifies the architecture of the reconfigurable system and defines how the two actions of reconfigurability (algorithmic and hardware) will be carried out.

An important contribution of this thesis in the global reconfigurability framework is the iterative mapping approach. This new interplay supposes algorithms with computational similarities and of iterative nature which are very common in the domain of digital communications. In this case, an algorithmic change which is decided in the first reconfigurability layer corresponds to a connection change between basic computational operators implemented in the hardware layer. This new approach provides a negligible reconfiguration time and a minimization of the required hardware resources, and thus it appears to be an attractive solution for future terminals. Moreover, the iterative approach introduces a new dynamic system parameter which is the number of the available iterations. This parameter can be appropriately adjusted by the configuration logic, in order to support the best trade-off between the different iterative algorithms.

The interest of the algorithmic reconfigurability has been shown by the development of reconfigurable detection schemes for DS-CDMA downlink connections. The proposed algorithms are based on essential functionalities which have computational overlaps repeated in an iterative way. These functionalities are the RAKE demodulation, the pilot-aided channel estimation and the interference cancelation. An important contribution of the proposed detectors was the application of the interference cancelation technique for the downlink environments. The proposed receivers are compared to conventional receivers and their advantages are brought out.

The developed DS-CDMA detection schemes have been a subject of study for the hardware reconfigurability layer, required by the global layered reconfigurability concept. A new architecture, which corresponds to our iterative mapping approach, has been proposed in order to support the functionality changes of the algorithmic reconfigurability. The proposed architecture is an important contribution for future DS-CDMA terminal implementations. It can support almost all the required functionalities of a DS-CDMA receiver, and due to the optimization of the hardware resources is appropriate for terminals where the computational requirements are more critical than these for the BSs. An important parameter of the proposed iterative approach is the maximum number of iterations which defines the available computational power.

Future research

The work achieved in this thesis brings a new formal framework to the iterative reconfigurability concept. A global approach is now available to design iterative reconfigurable systems. Our application of this new reconfigurable approach to the DS-CDMA downlink detection has given a set of new reconfigurable receivers appropriate for terminal implementations. In the future, we will use this proposed reconfigurability approach for the design of other reconfigurable systems. The idea of a RAKE receiver, with a flexible number of fingers, can support pre/post RAKE detections for the TDD mode. In this case the application of the iterative approach in the implementation of the two RAKE schemes, can minimize the cost at both the transmitter and the receiver and optimize the performance for each operational environment.

The same idea appears to be an interesting solution for ultra-wideband (UWB) communications, where the multipath channel includes a high number of paths. In this case, traditional architectures can not support this required high number of fingers. The proposed architecture seems to be suitable for this category of applications, as the number of fingers and IC generators are considered as dynamic input parameters. Another important domain for the application of our reconfigurability approach is in Multiple-Input Multiple-Output (MIMO) communications. In this case, system parameters, like the number of carriers, can be considered for a reconfigurable approach, which optimizes performance and computational power on-the-fly. In general, we can say that reconfigurable system design techniques can be profitably applied to any communication area.

As for our future research implementation activities, an important perspective is the increase in the number of iterations of the basic computational core implementing the different configurations and functionalities. Asynchronous logic, which allows computation independent of clock frequency, can increase the number of available iterations. Moreover, the addition of parallelism, which can be expressed by the co-existence of many reconfigurable fingers, can increase the number of iterations by a factor equal to the number of fingers. Finally, the decrease of the memory

requirements, which corresponds to an important optimization of the system complexity, can be considered also, as a principal future research direction. The processing of each slot using a moving window requires the memory storage of only some chips (=size of window) and thus the system does not need to store all the slot.

Appendix A

Definitions and Comments

This Appendix A introduces a set of definitions for certain useful notions, followed by some comments related to the system at hand.

System: “a regularly interacting or interdependent group of items forming a unified whole”. So a system implies a set of interacting items or *blocks* or *sub-systems* put together for a well-defined purpose.

Architecture: “(a) formation or construction as or as if as the result of conscious act; (b) a unifying or coherent form or structure; (c) a method or style of buildings; (d) the manner in which the components of a computer or computer system are organized and integrated.” We can thus distinguish the similarly-sounding concepts of “architecture” and “system” in two ways: we can say that (1) architecture is the thoughtful *process* of putting together the blocks of a system in order to serve a well-defined purpose, and (2) it may stand for the final *form* of that system, where the sub-blocks, their functionality, their interfaces and relationships with the other sub-blocks, etc., are well defined and justified. For our purposes, we use it in the sense of (2); namely, the description of the system arrangement and the rationale behind it.

We further assume that “architecture” implies the arrangement of *heterogeneous* or *dissimilar* elements or sub-blocks. For example, the architecture of a house includes not only the rooms and their functional dependence, but also its appearance (and aesthetic but non-functional concept), the landscape around it, etc. For our purposes, the receiver architecture includes the dissimilar but complementary parts of the RF/IF chain, the BaseBand (BB), and the higher-layer Network-and-Terminal-Connectivity (NTC) software.

Structure: “(a) something arranged in a defined pattern of organization; (b) organization of parts as dominated by the general character of the whole”.

We note that the first definition makes it sound very close to that of *architecture*. We may then borrow the second definition of “structure” to emphasize the *homogeneous* or *character-dominated* aspect of the arrangement. So we can be consistent by saying that *architecture* consists of heterogeneous or differently functions parts or subsystems, called *structures*, each of which now consists of similarly-functioning parts. As an example, we can give an error-correcting structure, which itself consists of similar and interconnected parts: for a turbo code structure, these parts are two

short convolutional codes, an interleaver and proper bit-output selection.

Design: “(a) a plan or protocol for carrying out or accomplishing something (as a scientific experiment); also: the process of preparing this; (b) a preliminary sketch or outline showing the main features of something to be executed”.

Part (a) it sounds very close to the first interpretation of *architecture*, namely the procedural plan; part (b) implies a sketchy structure or architecture plan. To be consistent, we can use it to mean either the process or the intermediate results towards the final layout of the structures and the architecture.

Configuration: “(a) relative arrangement of parts or elements; (b) functional arrangement; (c) something (as figure, contour, pattern, or apparatus) that results from a particular arrangement of parts or components”.

Note that the definition makes it virtually indistinguishable from the notions of “architecture” or “structure”. Thus, “reconfiguration” simply means the rearrangement of the parts, and “reconfiguration” is simply the ability to do so in some manner. It follows that it can be used to mean the ability to be either “reconstructed” or “re-architected”, depending on whether we are looking at the system macroscopically (inter-block-level) or microscopically (intra-block-level). This can happen, then at either level, and can be hardware-based, software-based, or else.

Reconfigurability: is the first in a list of *abilities*. Ability is either a process (such as the design process) or a final configuration (like the “architecture” or the “structure”). Here are some other abilities:

Flexibility: “the ability to be flexible: (a) capable of being flexed; (b) characterized by a ready capability to adapt to new, different, or changing requirements”.

Adaptivity: “showing or having a capacity for a tendency toward *adaptation*;

Adaptation: (a) adjustment to *environmental conditions*; (b) modification of an organism or its parts that makes it more fit for existence under the conditions of its environment”.

Adaptation is thus the process of adjustment due to an induced change, or some type of *dynamic re-adjustment*, which in our case is either a physical change (like the channel dynamics), or *ad hoc* network configuration change, and the like. We exclude service-requirements changes or system-requirements changes should be excluded from the “environment”, but they also comprise sources of changes themselves.

It follows that adaptation (if the ability is present) can manifest itself in various ways, but in order to distinguish it from “reconfigurability”, we may envision a condition of “continuity”, or “graduality”: adaptation implies small incremental changes, little at a time, in response to external stimulus or measurements, as opposed to “wholesale” changes at the structural or architectural level, which are more “discontinuous” in nature (a rearrangement of blocks, for example, is not done gradually in time).

Reprogrammability: “the ability to reconfigure by software commands”.

Modularity: “ability of being modular: (a) of, relating to, or based on a module; (b) constructed with standardized units for use together: (b) a usually packaged functional assembly of electronic components for use with other such assemblies”.

Appendix B

Linear Profile for the DSP implementation

```

#include "my_lib.h"
#include <stdlib.h>

/*****
 * RAKE receiver
 *****/
void rake(int M,          /* Number of symbols */
          int over,       /* Oversampling factor */
          int sf,         /* Spreading factor */
          int* seq,       /* Walsh Code */
          int2x16* scr,   /* Scrambling Code */
          int L,          /* Number of paths */
          int* del,       /* Delays of the paths */
          int2x16* channel_matrix, /* Power of the paths */
          int2x16* in,    /* Input Slot(samples) */
          int2x16* out)   /* Output of Rake combination (soft decisions)*/
/***** The conventional Rake receiver *****/
{
    int2x16 x[8];          /* Enough for all reasonable cases */
    if( L > 8 ) exit(1);

    int i,j,l;

    for(i=0; i<M; i++)     /* Processing of every chip of the slot */
    {
        const int isfj = i*sf;
        const int isfo = isfj * over;
        const int seqj = seq[0];

        int2x16 *pin, *psc;

        for( l=0, pin = in+isfo, psc=scr+isfj; l<L; l++)
        {
            const int dell = del[l];
            x[l]=mult_j2x16(compact_to_j2x16_from_j32(sum_j2x16(mult_j2x16(*pin+dell),*psc)),
                sum_j2x16(mult_j2x16(*pin+dell),compact_to_j2x16_from_j32(-expand_high_of_j2x16(*psc),
                expand_low_of_j2x16(*psc)))).compact_to_j2x16_from_j32(seqj,seqj));
        }

        for( j=1; j<sf; j++) /* chip--> symbols */
        {
            const int isfj = i*sf+j;
            const int isfo = isfj * over;
            const int seqj = seq[j];

            for( l=0, pin = in+isfo, psc=scr+isfj; l<L; l++)
            {
                const int dell = del[l];
                x[l]=add_j2x16(x[l],mult_j2x16(compact_to_j2x16_from_j32(sum_j2x16(mult_j2x16(*pin+dell),*psc)),
                    sum_j2x16(mult_j2x16(*pin+dell),compact_to_j2x16_from_j32(-expand_high_of_j2x16(*psc),expand_low_of_j2x16(*psc)))).
                    compact_to_j2x16_from_j32(seqj,seqj));
            }
        }

        int2x16 *pcm, *px;
        px = x;
        pcm = channel_matrix;

        out[i]=compact_to_j2x16_from_j32(sum_j2x16(mult_j2x16(*pcm,*px)),
            sum_j2x16(mult_j2x16(compact_to_j2x16_from_j32(-expand_high_of_j2x16(*pcm),
            expand_low_of_j2x16(*pcm)),*px)));

        for(l=1,pcm=channel_matrix+1,px=x+1; l<L; l++, pcm++, px++) /* MRC+Complex MUL */
        {
            out[i]=add_j2x16(compact_to_j2x16_from_j32(sum_j2x16(mult_j2x16(*pcm,*px)),
                sum_j2x16(mult_j2x16(compact_to_j2x16_from_j32(-expand_high_of_j2x16(*pcm),
                expand_low_of_j2x16(*pcm)),*px))),out[i]);
        }
    }
}

```

0.01%

0.94%

7.01%

23.36%

2.81%

3.74%

Figure B.1: Linear profiling for the RAKE function; Simulation parameters: $L=2$, $N=4$, $K=640$.

```

#include "my_lib.h"
#include <stdlib.h>

/***** Interference Cancellation *****/
void ic(int M,          /* Number of symbols */
        int over,       /* Oversampling factor */
        int sf,         /* Spreading factor */
        int* seq,       /* Walsh Code */
        int2x16* scr,   /* Scrambling Code */
        int L,         /* Number of paths */
        int* del,       /* Delays of the paths */
        int2x16* channel_matrix, /* Power of the paths */
        int2x16* in,    /* Input Slot(M symbols) */
        int2x16* out)   /* Output (M symbols) */
/*****/

{
    int i, j;
    int2x16 dec1, dec2;
    int inre, inim;
    int2x16 inde[2560], inde2[2560], gen_ic[1280], gen_ic2[1280];

    const int tmpre = sum_i2x16(mult_i2x16(channel_matrix[0], channel_matrix[1]));
    const int tmpim = sum_i2x16(mult_i2x16(channel_matrix[0], compact_to_i2x16_from_i32
        (-expand_high_of_i2x16(channel_matrix[1], expand_low_of_i2x16(channel_matrix[1]))));
    const int2x16 mul1 = compact_to_i2x16_from_i32(tmpre, tmpim);
    const int2x16 mul2 = compact_to_i2x16_from_i32(tmpim, tmpre);
    const int2x16 mul3 = compact_to_i2x16_from_i32(tmpre, -tmpim);
    const int2x16 mul4 = compact_to_i2x16_from_i32(-tmpim, tmpre);

    for (i=0; i<M; i++)
    {
        const int isf=i*sf;
        inre = expand_low_of_i2x16(in[i]);
        inim = expand_high_of_i2x16(in[i]);

        if (abs(inre) >= abs(inim) )
        {
            if ( inre >= 0 )
            {
                dec1=compact_to_i2x16_from_i32(1,0);
                dec2=compact_to_i2x16_from_i32(0,1);
            }
            else
            {
                dec1=compact_to_i2x16_from_i32(-1,0);
                dec2=compact_to_i2x16_from_i32(0,-1);
            }
        }
        else
        {
            if ( inim >= 0 )
            {
                dec1=compact_to_i2x16_from_i32(0,-1);
                dec2=compact_to_i2x16_from_i32(1,0);
            }
            else
            {
                dec1=compact_to_i2x16_from_i32(0,1);
                dec2=compact_to_i2x16_from_i32(-1,0);
            }
        }

        for (j=0; j<sf; j++)
        {
            const int seqj=seq[j];
            const int isfj=isf+j;

            int tmp2, tmp3;
            tmp2=sum_i2x16(mult_i2x16(dec1, scr[isfj]));
            tmp3=sum_i2x16(mult_i2x16(dec2, scr[isfj]));
            inde[isfj]=mult_i2x16(compact_to_i2x16_from_i32(tmp2, tmp3),
                compact_to_i2x16_from_i32(seqj, seqj));
            inde2[isfj+del[1]]=inde[isfj];
        }
    }
}

```

0.01%

0.01%

0.47%

0.47%

0.93%

2.80%

2.98%

0.06%

1.87%

0.93%

3.74%

Figure B.2: Linear profiling for the IC function (Part A); Simulation parameters: $L=2$, $N=4$, $K=640$.

```

for (j=0 ; j<del[1] ; j++)
{
    inde[M*sf+j]=compact_to_i2x16_from_i32(0,0);
    inde2[j]=compact_to_i2x16_from_i32(0,0);
}

for (i=0 ; i<M ; i++)
{
    const int isf=i*sf;
    const int seqj=seqj[0];
    const int2x16 scrj=scrj[isf];
    gen_ic[i]=mult_i2x16(compact_to_i2x16_from_i32(sum_i2x16(mult_i2x16
    (inde[isf+del[1]],scrj)),sum_i2x16(mult_i2x16(inde[isf+del[1]],
    compact_to_i2x16_from_i32(expand_high_of_i2x16( scrj ),expand_low_of_i2x16( scrj ))))),
    compact_to_i2x16_from_i32(seqj,seqj));

    gen_ic2[i]=mult_i2x16(compact_to_i2x16_from_i32(sum_i2x16(mult_i2x16
    (inde2[isf],scrj)),sum_i2x16(mult_i2x16(inde2[isf],
    compact_to_i2x16_from_i32(-expand_high_of_i2x16( scrj ),expand_low_of_i2x16( scrj ))))),
    compact_to_i2x16_from_i32(seqj,seqj));

    for (j=1 ; j<sf ; j++)
    {
        const int isfj=isf+j;
        const int seqj=seqj[j];
        const int2x16 scrj=scrj[isfj];

        gen_ic[i]=add_i2x16(gen_ic[i],mult_i2x16(compact_to_i2x16_from_i32(
        sum_i2x16(mult_i2x16(inde[isfj+del[1]],scrj)),sum_i2x16(mult_i2x16
        (inde[isfj+del[1]],compact_to_i2x16_from_i32(-expand_high_of_i2x16(scrj),
        expand_low_of_i2x16(scrj)))))),compact_to_i2x16_from_i32(seqj,seqj));
        gen_ic2[i]=add_i2x16(gen_ic2[i],mult_i2x16(compact_to_i2x16_from_i32(
        sum_i2x16(mult_i2x16(inde2[isfj],scrj)),sum_i2x16(mult_i2x16(inde2[isfj],
        compact_to_i2x16_from_i32(-expand_high_of_i2x16(scrj),
        expand_low_of_i2x16(scrj)))))),compact_to_i2x16_from_i32(seqj,seqj));

    }

    out[i]=sub_i2x16(in[i],compact_to_i2x16_from_i32(sum_i2x16(mult_i2x16(
    gen_ic[i],mul3)),sum_i2x16(mult_i2x16(gen_ic[i],mul2))));
    out[i]=sub_i2x16(out[i],compact_to_i2x16_from_i32(sum_i2x16(mult_i2x16(
    gen_ic2[i],mul1)),sum_i2x16(mult_i2x16(gen_ic2[i],mul4))));

}

}

```

Figure B.3: Linear profiling for the IC function (Part B); Simulation parameters: $L=2$, $N=4$, $K=640$.

Appendix C

Résumé

C.1 Introduction

Dans les systèmes de télécommunications mobiles de 2ème génération, mis à part le contrôle de puissance, les paramètres radio sont généralement fixés statiquement pour que le système puisse fonctionner en tenant compte d'une dégradation de la qualité de liaison. Par exemple les paramètres de modulation et le schéma de codage sont initialisés au début de communication pour tenir compte du "pire cas" et ne changent plus par la suite. Cette méthode statique de paramétrage propre au principe HDR (Hardware Defined Radio) est sous optimale tant du point de vue service pour garantir la "qualité", s'il y a modification du canal, que d'un point de vue électronique, où la consommation du récepteur portable est inutilement majorée. Les nouveaux standards et protocoles offrent une plus grande souplesse dans la gestion des paramètres et ainsi la possibilité d'améliorer la qualité de communication. De façon à s'adapter d'une façon statique ou dynamique à l'environnement, les terminaux mobiles incorporent de plus en plus de modes et de standards de communication. La conséquence pour les concepteurs électroniques est une augmentation importante de la complexité et de nouveaux défis tels que trouver des compromis pour:

- L'intégration d'un système multi-standards multi-modes flexible
- Un coût de production minimal
- Une arrivée rapide possible sur le marché
- Une consommation minimale.

La flexibilité recherchée des systèmes de télécommunication peut s'inspirer du principe de la radio logicielle "Software radio" dans laquelle, idéalement, le signal est traité numériquement tout de suite après la réception à l'antenne. Même s'il est utopique d'appliquer ce principe à la lettre, en particulier à cause des limitations des convertisseurs analogique/numérique, il est intéressant d'étudier la façon de gérer du mieux possible la flexibilité de la partie numérique en tenant compte des compromis. Certes un traitement numérique 100% logiciel est par essence flexible. Mais même en tenant compte des progrès technologiques de processeurs de traitement

du signal, “DSP”, il est nettement sous optimal en terme de consommation et de performances. Il faut encore nécessairement considérer un minimum de traitement matériel.

La notion de flexibilité peut être vue sous 2 aspects complémentaires: “l’adaptativité” et la “reconfigurabilité”. Un système est adaptatif si les valeurs numériques de ses paramètres peuvent changer à la volée. Par exemple le changement du nombre de pas dans l’algorithme des moindres carrés “LMS” illustré en figure C.1, correspond à la modification d’une valeur numérique et répond ainsi au critère d’adaptativité.

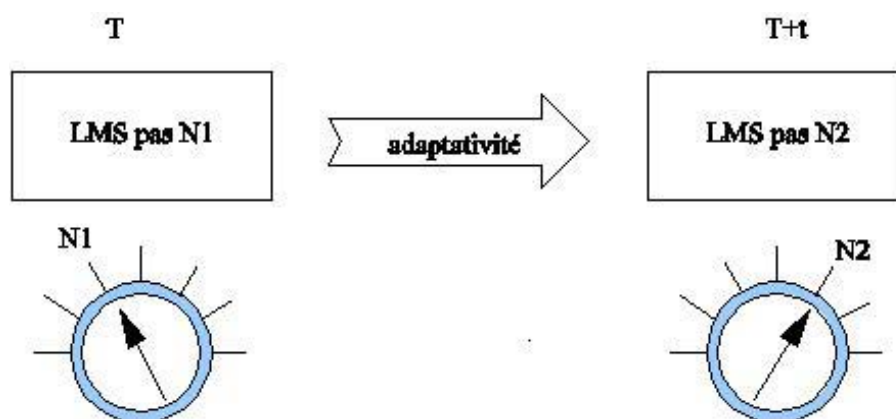


Figure C.1: Exemple d’Adaptativité: Changement de pas de l’algorithme LMS.

La reconfigurabilité sous entend la possibilité de modifier à la volée une fonction par un changement non quantifiable, c’est-à-dire sans représentation suffisante par une valeur numérique. Par exemple le passage d’un schéma de codage convolutionnel à un turbo-code, comme illustré en figure C.2, ne peut pas être représenté par un changement de valeur de paramètre, il s’agit donc de reconfigurabilité.



Figure C.2: Exemple de reconfigurabilité: Changement de code.

Une réalisation logicielle d’un système flexible est donc naturellement adaptable et reconfigurable. La recherche du meilleur compromis (coût, temps de développement, performances, consommation) nécessite cependant l’utilisation conjointe de logiciel et de structures matérielles optimisées, spécifiques aux structures “SoC” (System On a Chip). La question est de savoir si la reconfigurabilité matérielle a un sens dans le domaine des systèmes de communication multi-standards, la plupart des constructeurs peuvent opter pour l’utilisation des blocs matériels préconçus et correspondants à chaque standard. La reconfigurabilité consiste ici à commuter d’un bloc à un autre. Si cette solution a le mérite de gagner du temps de développement, elle est certainement sous optimale en surface de silicium et consommation. Une autre approche, que

nous allons développer, consiste à étudier les structures communes ou proches entre différents modes et standards de communication. Les paragraphes suivants traitent un point de vue sur la reconfigurabilité matérielle avec des exemples significatifs basés sur un opérateur en réseau “RAKE” d’un récepteur DS-CDMA.

C.2 Notion de Reconfigurabilité matérielle

La reconfigurabilité matérielle consiste en un circuit capable de réaliser des fonctions différentes à des temps différents. Un microprocesseur est un exemple de circuit reconfigurable en considérant que son jeu d’instructions est un ensemble de fonctions configurables. Son fonctionnement sériel et générique engendre cependant une diminution des performances globales et une augmentation de la consommation qui ne permet pas toujours de respecter les contraintes physiques de spécification. Sur ce dernier point un circuit électronique dédié de type ASIC (Application Specific Integrated Circuit) est recommandé mais est naturellement non configurable. Dans les systèmes sur puce SoC, on retrouve la complémentarité des processeurs et du matériel dédié qui agit comme “co-processeur” accélérant les calculs ou comme unité d’exécution spécifique d’un processeur ASIP (Application Specific Instruction set Processor). Un circuit matériel programmable FPGA (Field Programmable Gate Array) est potentiellement un circuit reconfigurable s’il est possible de le reprogrammer à la volée pour changer sa fonctionnalité. La finesse de son grain de calcul ainsi que son caractère générique en font un composant très coûteux à l’achat et très “consommant”. La reconfigurabilité matérielle ne peut donc avoir de sens qu’en utilisant une structure reconfigurable adaptée et optimisée au sein d’un ASIC de façon à optimiser les contraintes: performances, consommation et coût.

C.2.1 Approche multiplexage

Il s’agit simplement dans cette approche de commuter les fonctions matérielles préexistantes. Cette méthode tire profit des blocs préconçus et permet une conception rapide d’un ASIC pour disposer d’un système multi-standards. En revanche une grande partie du matériel peut ne pas être utilisée. Le surcoût de la surface engendré par la concaténation de blocs, laisse à penser qu’une structure configurable offre un moindre coût et consommation et une plus grande flexibilité. La figure C.3 illustre l’approche multiplexage.

C.2.2 Approche pagination

Dans cette approche le circuit électronique est de type FPGA et change de fonction par reconfiguration. Cette approche utilisée dans [SRI00a] est aussi connue sous le terme de matériel paginé. Elle permet de minimiser le matériel mais reste critique sur des structures FPGAs à “grain” fin identiques à celle du commerce, car le temps de reprogrammation du FPGA est souvent trop lent pour respecter les contraintes de performances de l’application. Pour tirer parti de cette approche, il est nécessaire de concevoir une structure matriciée de type FPGA dédié à une gamme d’applications, où le grain de calcul est plus “épais” mais est plus performant en terme de temps de reconfiguration et de consommation. La figure C.4 illustre l’approche pagination.

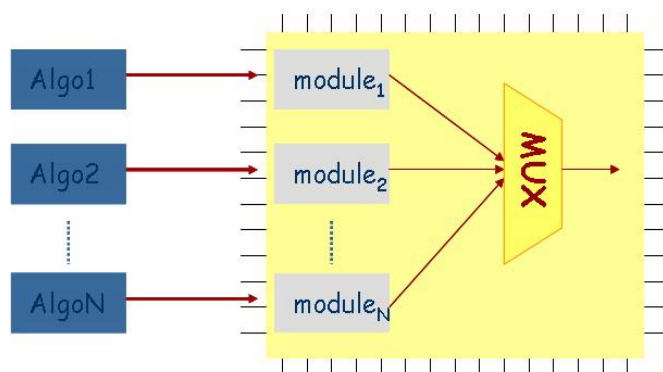


Figure C.3: L'approche multiplexage.

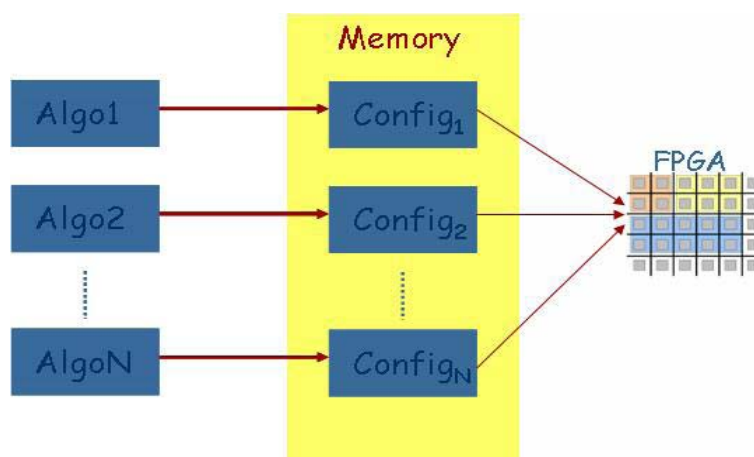


Figure C.4: L'approche pagination.

C.2.3 Approche factorisation

Cette approche tire parti des opérateurs communs à un groupe de fonctions à configurer. On considère un ensemble de fonction $S = (f_1, f_2, \dots, f_n)$ disposant d'un minimum d'opérateurs communs, ce qui peut s'exprimer par $f_i \cap f_j \neq \emptyset$ pour chaque i, j [GRA00]. En utilisant une même base matérielle constituée de tous les opérateurs, la reconfiguration consiste à changer de fonction en choisissant les bons opérateurs avec les connexions associées pour chaque fonction f_i . La figure C.5 illustre le fonctionnement de cette approche, où apparaît la plate-forme constituée de M opérateurs et un organe de supervision SPV chargé de générer les signaux de contrôle pour gérer les opérateurs et les connexions.

C.2.4 Approche itération

Une autre approche est basée sur l'approche "factorisation" et essaye de tirer parti des d'algorithmes se prêtant à la sérialisation des calculs en disposant d'opérateur ayant une "granularité" plus fine dans lequel une fonction peut boucler plusieurs fois. Les systèmes de communications utilisent de

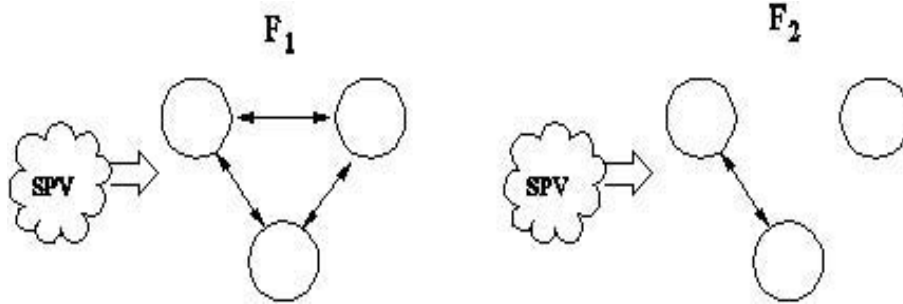


Figure C.5: Plate-forme configurée pour 2 fonctions différentes.

nombreuses fonctions facilement “sérialisables” ou de natures itératives, c’est le cas des décodeurs turbo-codes et LDPC (Low Density Parity Check Codes) mais aussi de l’égalisation comme il est étudié dans les chapitres suivants. La figure C.6 illustre l’approche proposée avec des opérateurs de bases plus petits et soumis à plusieurs itérations.

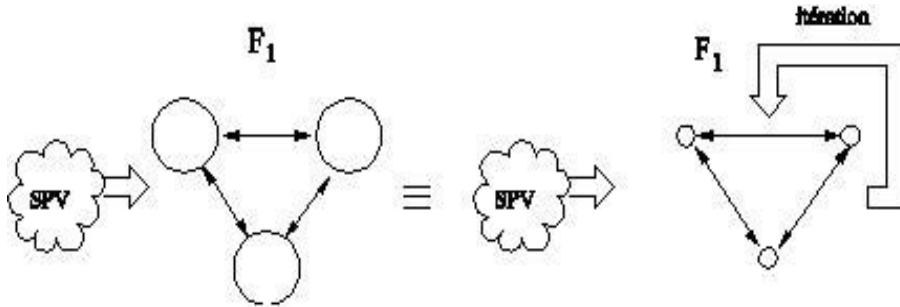


Figure C.6: Plate-forme avec opérateurs à grains fins pour la sérialisation.

L’approche proposée permet de diminuer la taille de la plate-forme matérielle et autorise un ajustement de la qualité de l’algorithme en jouant sur le nombre d’itérations. La consommation dynamique d’une structure sérielle de ce type est généralement supérieure à la structure “factorisation”. En revanche la consommation statique qui devient non négligeable dans les technologies CMOS $< 0,1\mu m$ est nettement inférieure du fait de la taille réduite du grain de calcul. Pour certaines fonctions, tous les opérateurs peuvent ne pas être utilisés, mais la faible granularité des opérateurs minimise l’utilisation sous optimale du matériel. Le nombre d’itérations peut aussi être limité par le temps imparti pour le traitement. Mais en considérant qu’une itération ne peut durer qu’un cycle d’horloge avec une architecture pipeline, le rapport entre la fréquence d’horloge ($>> 200Mmots/s$) et les débits binaires (quelque Mbits/s à quelques 10Mb/s) offre un nombre d’itération suffisant pour de nombreuses fonctions de télécommunications. Les exemples décrits par la suite illustrent l’intérêt de cette méthode.

C.3 Récepteur RAKE reconfigurable avec annulateur d’interférences

Cet exemple se place dans le cadre d’un récepteur DS-CDMA (Direct Sequence Code Division Multiple Access) utilisé dans les standards de téléphonie portables de 3ème génération. Cet

algorithme illustre un exemple de reconfigurabilité matérielle avec l'approche itération.

C.3.1 Formulation du problème

Dans le système 3G, il est possible de changer le débit de communication en jouant sur le facteur d'étalement propre au principe DS-CDMA. Pour fournir un service très haut débit, il faut un facteur d'étalement faible (2 ou 4). A ces débits, il y a une grande probabilité de n'avoir qu'un seul utilisateur ou un utilisateur non perturbé par d'autres utilisateurs ayant des facteurs d'étalement plus grand. Dans ce cas les interférences inter utilisateurs (MAI : Muti-user Interference) peuvent être négligeables. En revanche l'interférence entre symboles "IPT"(Inter Path Interference) liée à un canal avec de nombreux trajets, est souvent très importante et peut dégrader significativement les performances du système.

En considérant un modèle de transmission d'une station de base vers un terminal "downlink" dans un canal à évanouissement de Rayleigh multi-trajets, le signal en bande de base reçu peut s'écrire :

$$r(t) = \sum_{k=1}^K \sum_{l=1}^L h_l(k) b(k) s(t - kT_b - \tau_l) + n(t) \quad (C.1)$$

où $r(t)$ est le signal reçu, K est la longueur de la fenêtre d'observation en symboles, L est le nombre de trajets, $h_l(k)$ est un nombre complexe représentant le facteur d'évanouissement, $b(k) \in \{\pm 1 \pm 1j\}$ est le symbole QPSK transmis, $\tau_l \in [0 T_b]$ est le retard de propagation du trajet l , T_b est la durée du symbole et $n(t)$ est le bruit blanc additif gaussien. Dans l'équation C.1, le signal $s(t)$ est étalé avec le facteur d'étalement SF sur SF "chips" selon la formule C.2:

$$s(t) = \sum_{n=0}^{SF-1} c(n) h(t - nT_c) \quad (C.2)$$

T_c est la durée d'un chip et $c(n)$ est la valeur du chip n ; $h(t)$ est une impulsion chip de durée T_c .

Le récepteur en râteau RAKE est chargé de "désétalement" le signal $s(t)$ et ce sur les L chemins parcourus, puis de recombinaison le résultat des L chemins. En supposant une détection cohérente, une connaissance parfaite du canal et une recombinaison des «doigts» du Rake par l'algorithme MRC (Maximum Ratio Combining), le signal issu du récepteur RAKE peut s'exprimer par

$$\begin{aligned} \hat{b}(k) &= \sum_{l=1}^L h_l^*(k) \int_{(k-1)T_b + \tau_l}^{kT_b + \tau_l} r(t) c^*(t - \tau_l) dt \\ &= D + S + W, \end{aligned} \quad (C.3)$$

où D , S et W représentent respectivement le signal démodulé, une composante d'interférence et le bruit. Ils s'expriment par les équations suivantes:

$$D = \sqrt{P_T} N b(k) \sum_{l=1}^L |h_l(k)|^2, \quad (C.4)$$

$$S = \sqrt{P_T} \sum_{l=1}^L \sum_{\substack{q=1 \\ q \neq l}}^L h_l^*(k) h_q(k) \int_{(k-1)T_b+\tau_l}^{kT_b+\tau_l} b(t-\tau_q) c(t-\tau_q) c^*(t-\tau_l) dt, \quad (C.5)$$

$$W = \sum_{l=1}^L h_l^*(k) \int_{(k-1)T_b+\tau_l}^{kT_b+\tau_l} n(t) c^*(t-\tau_l) dt, \quad (C.6)$$

Le terme S représente un composant d'interférence IPI. Ce terme vient du fait qu'il est impossible de concevoir des codes d'étalement de spectre parfaitement orthogonaux pour tout les décalages de temps. Le "désétalement" qui correspond à une fonction d'auto-corrélation produit donc un résultat non nul quand le signal est décalé, comme il est exprimé dans l'équation C.7.

$$R(\tau) = \begin{cases} 1 & , \text{if } \tau = 0 \\ -\frac{1}{SF} & , \text{if } \tau \neq 0 \end{cases} \quad (C.7)$$

Pour les hauts débits, c'est-à-dire quand SF est faible, le récepteur RAKE est plus fortement dégradé par l'IPI. Un système d'annulation d'interférences peut être utilisé pour diminuer l'impact négatif de cette interférence.

C.3.2 Algorithme proposé

L'annulation d'interférence "IC" (Interference Cancellation) est basée sur la connaissance des générateurs de bruit que constituent les autres utilisateurs dans le cas du MAI ou des autres symboles dans le cas de l'IPI. Le principe est de reproduire l'interférence de façon à la soustraire par la suite, de façon à générer une information où le bruit d'interférence est diminué [DIV98], [HUI98]. Il est aussi possible de réitérer le processus en utilisant comme signal d'entrée, le signal déjà traité lors d'une première itération. Dans notre exemple, cette technique est appliquée pour supprimer l'IPI avec un seul utilisateur. La figure C.7 montre le découpage fonctionnel du récepteur:

Après le récepteur RAKE, plusieurs étages pour annuler l'interférence sont utilisés. La structure de l'étage i est illustrée en figure C.8.

La première opération d'un étage d'annulateur d'interférence consiste à estimer le signal en sortie de récepteur RAKE ou de l'étage $i-1$. Une fonction de décision est utilisée à cette fin. Elle peut être exprimée par:

$$\tilde{b}^{(i-1)}(k) = f_{dec}(\hat{b}^{(i-1)}(k)) \quad (C.8)$$

Puis le signal est re-modulé comme s'il s'agissait d'un émetteur, c'est à dire que l'opération d'étalement est effectuée et les retards, identiques à ceux du terme S de l'IPI, sont générés. Ensuite le signal est traité comme dans un récepteur RAKE de façon à régénérer l'IPI. En sortie de chaque étage l'interférence générée est retranchée au signal

$$\hat{b}^{(i)}(k) = \hat{b}^{(0)}(k) - \hat{S}^{(i)}(k) \quad (C.9)$$

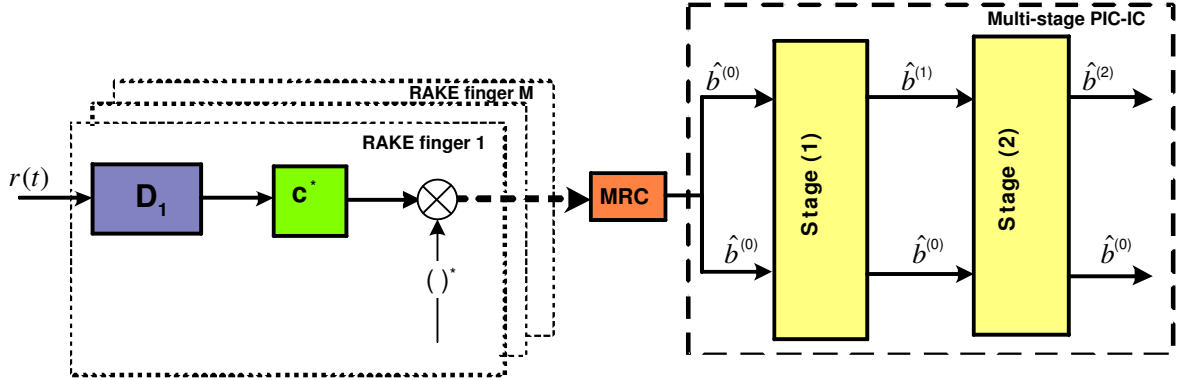


Figure C.7: Fonctions RAKE et annulateur d'interférences multi-étages.

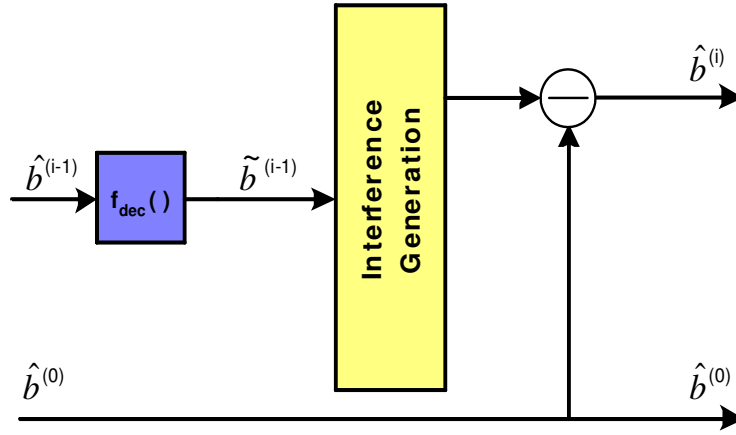


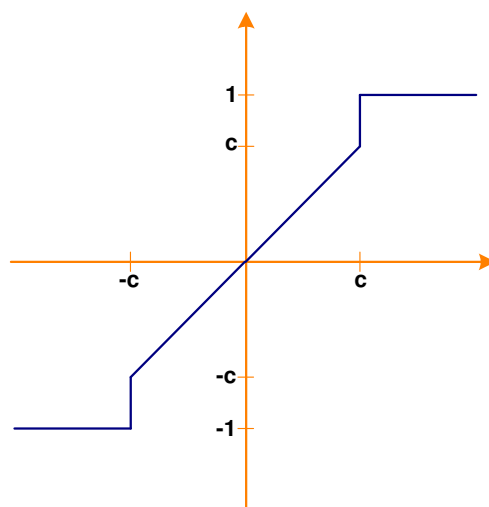
Figure C.8: Structure de l'étage i.

où $S^{(i)}(k)$ correspond à l'IPI reconstruite par l'étage i pour le symbole k . La fonction de décision peut être sur 1 bit (décision dure) ou plus d'un bit (décision souple). Il est préférable d'utiliser une fonction de décision mixte pour allier la rapidité de convergence de la décision dure, si celle-ci est correcte, avec la plus grande garantie de convergence de la décision souple [ZHA03]. La figure C.9 illustre la fonction de décision appliquée sur les parties réelles et imaginaires du signal. Quand le signal est fort, au dessus d'un certain pallier c , la décision dure est choisie, sinon une décision souple est utilisée pour éviter la propagation de l'erreur.

C.3.3 Evaluation de la performance

Des résultats de simulation en terme de taux d'erreur binaire "TEB" ont été obtenus en considérant un environnement et les paramètres suivant du standard UMTS:

- Canal à évanouissement de Rayleigh
- $L = 3$, puissance par trajets $P = [0 \ 0 \ 0]dB$ et retards $\tau = [0 \ 3 \ 6]T_c$.

Figure C.9: Fonction de décision $f_{dec}()$ avec pallier c .

Les figures C.10 et C.11 montrent respectivement le TEB obtenu pour un facteur d'étalement $SF=2$ et $SF=4$ et 5 étages pour l'annulation d'interférences. Dans ces conditions, l'algorithme proposé est meilleur que l'algorithme (BLE-MMSE) et (BLE-ZF) [KLE96] dont les résultats sont aussi représentés sur les figures C.10 et C.11.

C.3.4 Architecture reconfigurable

Cet algorithme peut être réalisé en tirant parti de la reconfigurabilité de type "itération" et dispose d'une structure de calcul composée d'opérateurs utilisés ou non suivant l'utilisation en mode RAKE ou en mode IC. Il y a autant d'itérations L en mode RAKE qu'il y a de doigts. En mode IC, le nombre d'itérations correspond au nombre d'étages de l'algorithme précédemment décrit, multiplié par le nombre d'interférences par symbole $= L * (L - 1)$ suivant l'équation C.5.

Un schéma détaillé de l'architecture est illustré en figure C.12. Elle est constituée de 3 blocs principaux : la mémoire donnée, l'unité de calcul et le superviseur de reconfiguration.

La mémoire donnée

L'algorithme d'annulation d'interférences travaille sur un bloc de donnée transmises durant un "slot", correspondant au quantum temporel d'un paquet de données des standards DS-CDMA. La mémoire donnée est constituée d'une RAM statique SRAM disposant de deux types d'accès : un accès pour recevoir les échantillons du signal et un autre pour effectuer le traitement RAKE + IC des données du slot précédent. Le temps de traitement correspond à un slot et les 2 accès se chevauchent, comme illustré dans la figure C.13, de façon à commencer le traitement le plus tôt possible et diminuer la taille mémoire.

L'unité de calcul

De façon à permettre la reconfigurabilité, cette unité dispose d'opérateurs communs aux fonctions de RAKE et IC ainsi que des opérateurs spécifiques à chaque fonction. Les figures C.14 et C.15 représentent respectivement le chemin des données pour les deux configurations.

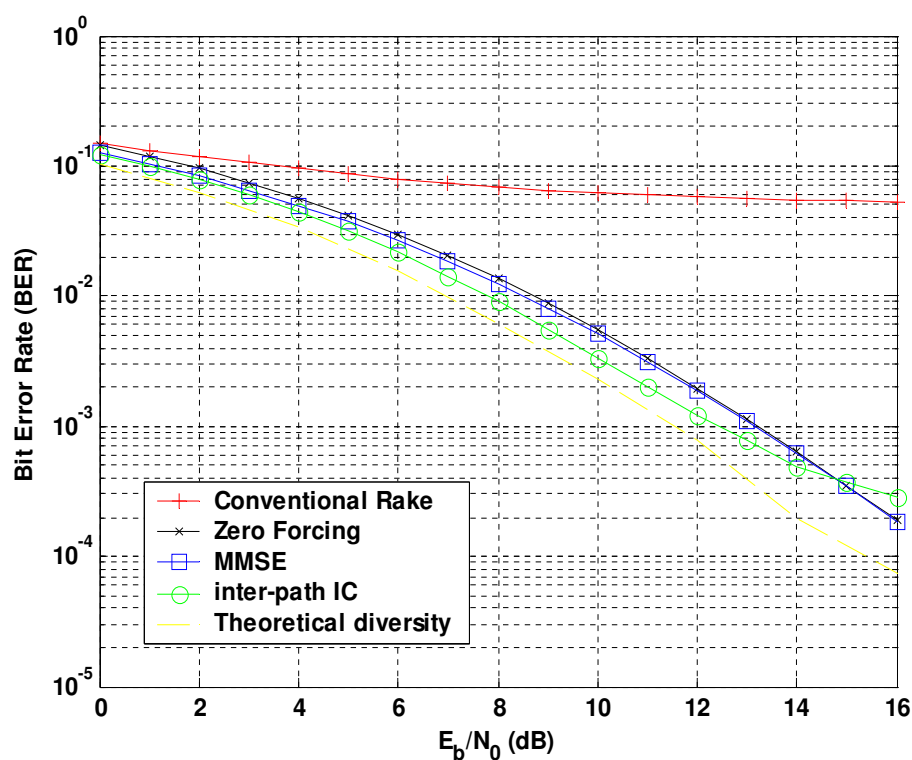


Figure C.10: Evaluation de la performance évaluation de l'annulateur d'interférences; SF=2.

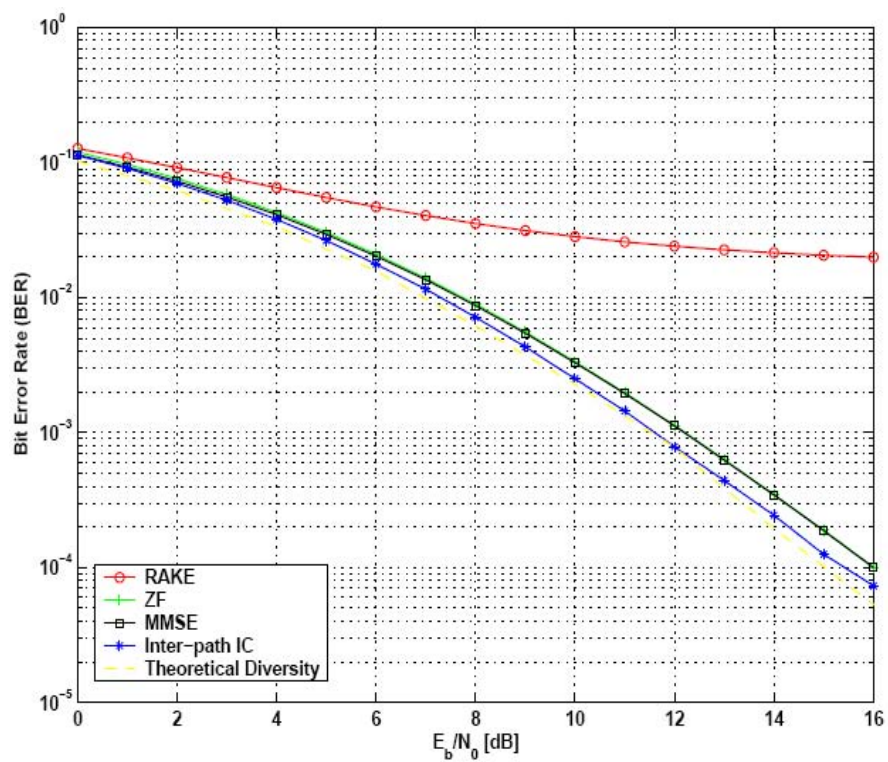


Figure C.11: Evaluation de la performance évaluation de l'annulateur d'interférences; SF=4.

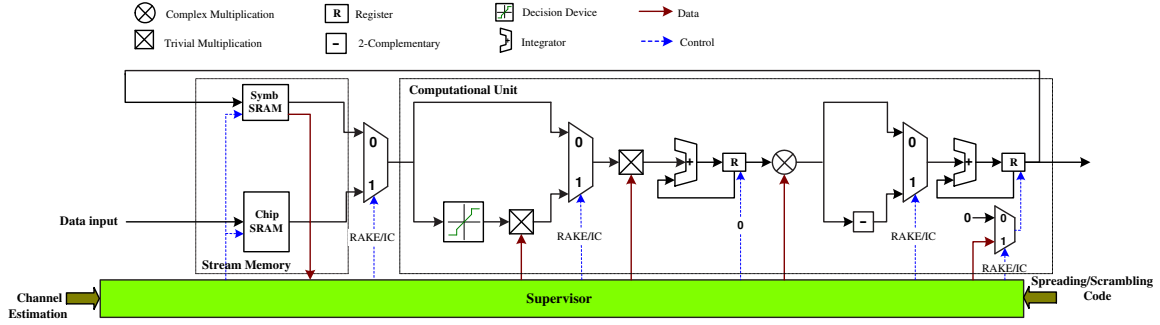


Figure C.12: Architecture reconfigurable.

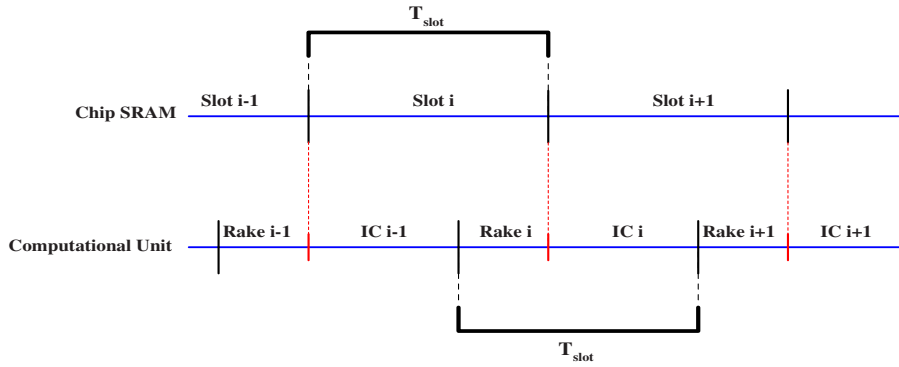


Figure C.13: Chronogramme des flux de donnée en réception et en traitement.

Dans la configuration RAKE, les données proviennent de la mémoire d'échantillons SRAM et vont directement sur l'unité de "désatelement", correspondant à un intégrateur, pour générer les symboles. Ceux-ci sont ensuite multipliés par les coefficients complexes du canal estimé. Un intégrateur final réalise la combinaison MRC des différents doigts de RAKE calculés.

En mode annulateur d'interférences, les données d'entrée sont les symboles issus du RAKE ou de l'étage précédent. Ces symboles sont ensuite soumis à la fonction de décision et étalés, ce qui différencie cette fonction du RAKE, puis il sont ré-étalés avec un retard $\tau_{l,q}$, généré par le superviseur, comme exprimé dans l'équation C.5. Après multiplication par les coefficients de canal, le symbole est ensuite complémenté et accumulé pour générer le terme d'interférence global à retirer du symbole.

Le superviseur de configuration

Le superviseur SPV fournit les signaux de contrôle à l'unité de calcul et à la mémoire. Il est en charge de reconfigurer l'unité de calcul pour passer d'un mode à l'autre. Il effectue la traduction des temps estimés par l'estimateur de canal, en adresse données de façon à lire de la SRAM les échantillons correspondant aux retards estimés. Le superviseur peut avoir un algorithme de reconfiguration dynamique de façon à trouver les meilleurs paramètres (L, n) en fonction du canal et du nombre de cycles de traitement dont on dispose.

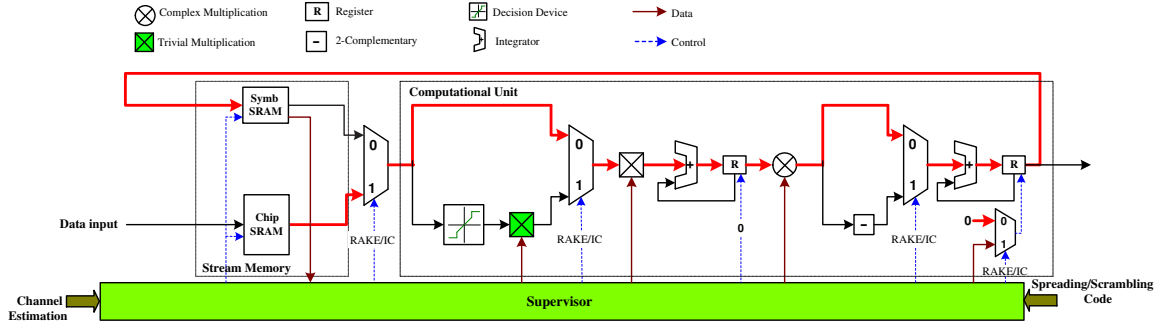


Figure C.14: Configuration en mode Rake.

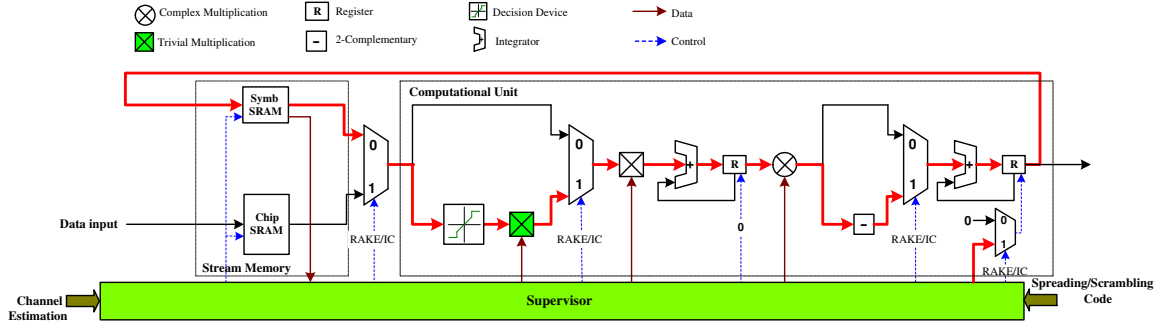


Figure C.15: Configuration en mode IC.

En considérant un opérateur synchrone et en “pipeline” fonctionnant à la fréquence F_{clk} , le temps de calcul d’une itération est de SF/F_{clk} du fait qu’il faut intégrer SF chips dans le symbole pour “désétaler”. Le temps de traitement est donc de $[L + VL(L - 1)]SF/F_{clk}$ en considérant L trajets et V étages pour l’annulateur d’interférences.

Le nombre d’itérations possibles est limité par le temps symbole de SF/F_{clk} . Les valeurs de L et V sont donc contraintes suivant l’équation:

$$L + VL(L - 1) < \frac{F_{clk}}{F_{chip}} \quad (C.10)$$

Il est possible d’obtenir de l’ordre de 100 en considérant les technologies CMOS $0.9\mu m$, ce qui donnerait les valeurs possibles de $L = 5$ et $V = 4$.

Dans le cas où les trajets du canal ont des puissances très différentes, le superviseur peut choisir un algorithme d’annulation partielle, ce qui a pour conséquence de diminuer le terme $VL(L - 1)$ et ainsi la consommation du calcul.

C.3.5 Implémentation

Premièrement on a implémenté l’algorithme proposé sur un DSP commercial, le TigerSHARC d’Analog Devices, avec une fréquence d’horloge égale à 250 MHz. Malgré sa possibilité de parallélisme (quatre multiplications de 16-bit en parallèle) et sa flexibilité, le DSP utilisé est

Nombre des trajets (L)	RAKE	IPI-IC
1	103	-
2	208	458 (2 termes ; 1 étage, 2 termes ; 2 étages)
3	315	262 (2 termes ; 1 étage)
4	362	262 (2 termes ; 1 étage)
5	472	-
6	536	-
7	638	-
8	627	-

Table C.1: Le temps nécessaire (en μsec) pour le traitement d'un "slot" UMTS .

Utilisation	Pourcentage
Slices	501/19968 (2%)
Flip Flops	378/39936 (0%)
LUTs	885/39936 (2%)
IOBs	133/410 (3%)
GCLKs	1/32 (3%)
DSP48s	2/48 (4%)

Table C.2: L'utilisation du FPGA Virtex 4 pour l'unité de calcul.

incapable de supporter les calculs posés par l'algorithme. La Table C.1 et la figure C.16 résument la performance du DSP pour différents environnements (nombre de trajets).

Pour supporter les calculs de l'algorithme proposé, une implémentation matérielle est alors nécessaire. On a implémenté l'algorithme sur un FPGA commercial, le Virtex 4 de Xilinx, en utilisant l'approche itérative. Les résultats obtenus montrent que l'architecture proposée utilise une petite partie du FPGA et alors correspond à un circuit très fin. La Table C.2 présente les résultats de synthèse pour l'unité de calcul.

Par le trajet critique, on a trouvé que la fréquence d'horloge maximale, supportée par le circuit, est égale à 168.8 MHz, ce qui correspond à 44 itérations. Ce nombre d'itérations est efficace pour supporter les conditions d'opérations moyennes.

Les FPGAs sont utilisés pour des raisons de test et pour implémenter des circuits "prototypes". Aux terminaux mobiles réels, on utilise la technologie ASIC pour implémenter l'architecture proposée. L'implémentation sur un ASIC 0.13 μm (CMOS HCMOS9 STMicroelec) a donné un trajet critique qui peut supporter une fréquence d'horloge égale à 500 MHz, ce qui correspond à 130 itérations. Ce nombre d'itérations est efficace pour supporter tous les calculs nécessaires introduits par l'algorithme. La figure C.17 compare les trois implémentations pour un environnement avec un canal de trois trajets et un IC avec $V=3$.

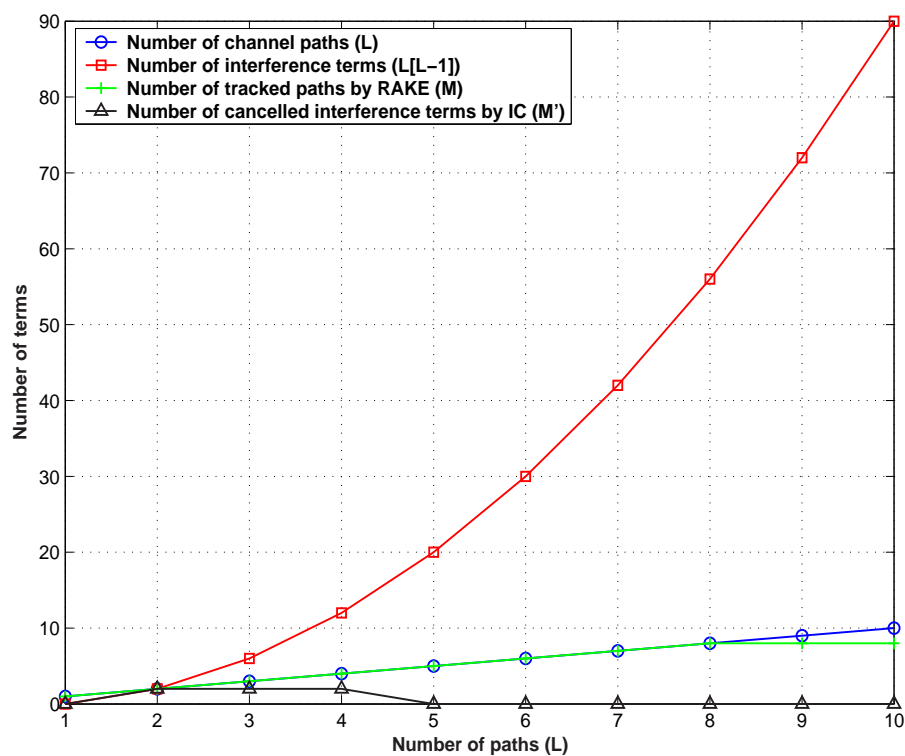


Figure C.16: L'efficacité de DSP en utilisant comme mesure le nombre de calculs effectuées pour différent nombre de trajets.

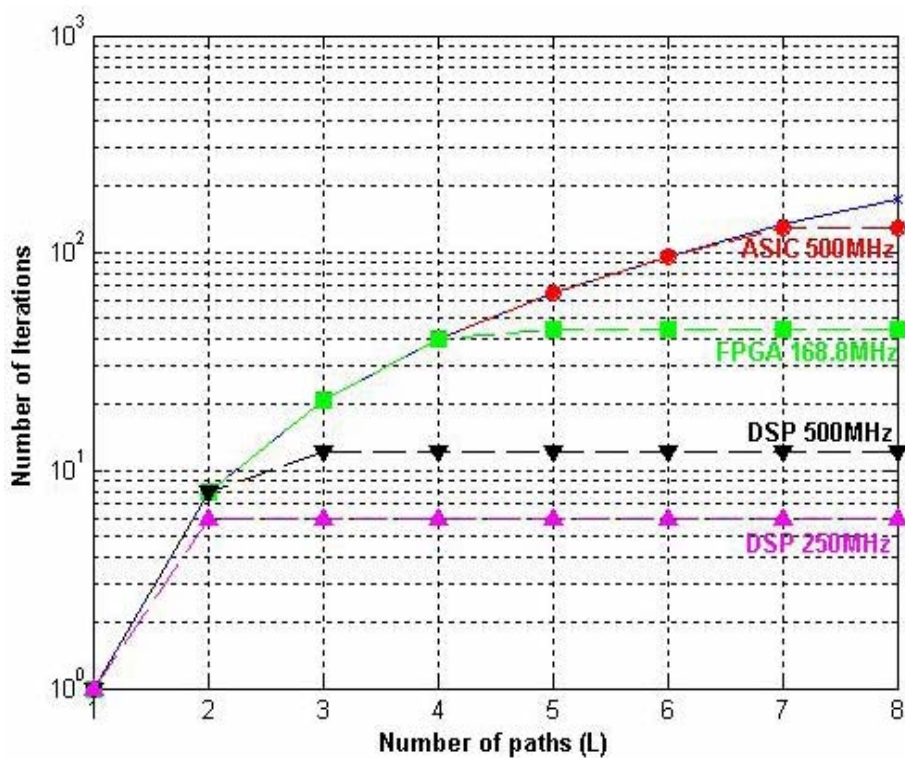


Figure C.17: La comparaison de trois implémentations.

C.4 Un annulateur d'interférences basé sur une estimation de canal

Dans le premier algorithme, nous avons considéré une connaissance parfaite du canal afin de mettre en oeuvre la détection cohérente et le processus d'annulation d'interférences. Dans cet exemple, l'algorithme prend en compte l'opération d'estimation de canal dont la qualité a un impact direct sur la performance d'un terminal. L'algorithme proposé est aussi adapté aux connections haut débit DS-CDMA. Basé sur une approche multi étages, il vise à supprimer les interférences du canal pilote et du canal de données.

C.4.1 Formulation du problème

Le modèle de communication descendante considéré est basé sur un système DS-CDMA mono utilisateur sur un canal à évanouissement du type Rayleigh et avec estimation de canal par pilote. Le signal transmis peut être exprimé par

$$s(t) = A_0 c_0(t) + A_1 b(t) c_1(t), \quad (\text{C.11})$$

où $c_0(t)$, $c_1(t)$ dénotent les séquences de signature pour le pilote ($n=0$) et les données ($n=1$); A_0 et A_1 sont, respectivement, les amplitudes de pilote et des données. Il faut remarquer qu'aucun symbole de données n'est présent sur le canal pilote.

Ainsi, le signal reçu peut être écrit comme C.12, où $n(t)$ est un bruit blanc gaussien avec densité spectrale de puissance bilatérale $N_0/2$.

$$r(t) = \sum_{l=1}^L h_l(t) [A_0 c_0(t - \tau_l) + A_1 b(t - \tau_l) c_1(t - \tau_l)] + n(t) \quad (\text{C.12})$$

On se place dans le cas d'une transmission non codée, avec des retards τ_l parfaitement connus. Afin d'estimer les coefficients du canal, nécessaires à la démodulation cohérente, le signal reçu est "désétaillé" à partir du code d'étalement du canal pilote. Ainsi, le coefficient complexe de canal d'évanouissement estimé pour le j -ème chemin et le k -ème symbole transmis est donné comme

$$\begin{aligned} \hat{h}_j(k) &= \frac{1}{N A_0} \int_{(k-1)T_b + \tau_j}^{kT_b + \tau_j} r(t) c_0^*(t - \tau_j) dt \\ &= h_j(k) + S_j(k) + C_j(k) + \mu_j(k), \end{aligned} \quad (\text{C.13})$$

où S_j , C_j , sont respectivement les interférences avec les chemins de données et avec les chemins du canal pilote, et μ_j est la composante de bruit. Par ailleurs, chaque composante d'interférence peut être re-écrite comme

$$S_j(k) = \frac{A_1}{N A_0} \sum_{\substack{l=1 \\ l \neq j}}^L h_l(k) \int_{(k-1)T_b + \tau_j}^{kT_b + \tau_j} b(t - \tau_l) c_1(t - \tau_l) c_0^*(t - \tau_j) dt, \quad (\text{C.14})$$

$$C_j(k) = \frac{1}{N} \sum_{\substack{l=1 \\ l \neq j}}^L h_l(k) \int_{(k-1)T_b + \tau_j}^{kT_b + \tau_j} c_0(t - \tau_l) c_0^*(t - \tau_j) dt, \quad (\text{C.15})$$

$$\mu_j(k) = \frac{1}{NA_0} \int_{(k-1)T_b+\tau_j}^{kT_b+\tau_j} n(t)c_0^*(t-\tau_j)dt \quad (C.16)$$

La qualité de l'estimation de canal basée sur les symboles pilotes est diminuée par les termes d'interférence multi chemin S_j et C_j qui introduisent une erreur d'estimation.

L'opération de démodulation est similaire à celle d'estimation de canal, excepté le fait que le désétalement est fait avec le code d'étalement de données. Le récepteur en râteau est le mode de démodulation conventionnel. En supposant l'algorithme MRC (maximum ratio combining) et L doigts de démodulation disponibles associés aux L chemins de canal, la décision variable du m-ème symbole de données est

$$\begin{aligned} \hat{b}(k) &= \frac{1}{NA_1} \sum_{l=1}^L \hat{h}_l^*(k) \int_{(k-1)T_b+\tau_l}^{kT_b+\tau_l} r(t)c_1^*(t-\tau_l)dt \\ &= D(k) + I(k) + F(k) + \eta(k), \end{aligned} \quad (C.17)$$

où D est la composante de signal de diversité générée; I , F , représentent respectivement les interférences entre chemins de données et celles entre les chemins de données et le pilote, et η est la composante de bruit. Les différentes composantes en sortie du corrélateur s'écrivent ainsi:

$$D(k) = b(k) \sum_{l=1}^L h_l(k) \hat{h}_l^*(k), \quad (C.18)$$

$$I(k) = \frac{A_0}{NA_1} \sum_{l=1}^L \sum_{\substack{q=1 \\ q \neq l}}^L \hat{h}_l^*(k) h_q(k) \int_{(k-1)T_b+\tau_l}^{kT_b+\tau_l} c_0(t-\tau_q) c_1^*(t-\tau_l) dt, \quad (C.19)$$

$$F(k) = \frac{1}{N} \sum_{l=1}^L \sum_{\substack{q=1 \\ q \neq l}}^L \hat{h}_l^*(k) h_q(k) \int_{(k-1)T_b+\tau_l}^{kT_b+\tau_l} b(t-\tau_q) c_1(t-\tau_q) c_1^*(t-\tau_l) dt, \quad (C.20)$$

$$\eta(k) = \frac{1}{NA_1} \sum_{l=1}^L \hat{h}_l^*(k) \int_{(k-1)T_b+\tau_l}^{kT_b+\tau_l} n(t) c_1^*(t-\tau_l) dt \quad (C.21)$$

Ces équations montrent que la détection de signal est pénalisée par l'IPI et est aussi sensible aux erreurs d'estimation. Pour de très hauts débits, l'IPI et l'erreur d'estimation générée peuvent conduire à une dégradation de la performance du système plutôt qu'à un gain de la diversité multi chemins.

C.4.2 Algorithme proposé

La structure générale de l'annulateur d'interférences multi étages proposé est montrée dans la figure C.18. L'idée de base est de reproduire les termes d'interférence et après les soustraire afin de générer des estimations de canal et données "nettoyées". L'application successive de ce processus devrait conduire à une importante amélioration de la performance du système

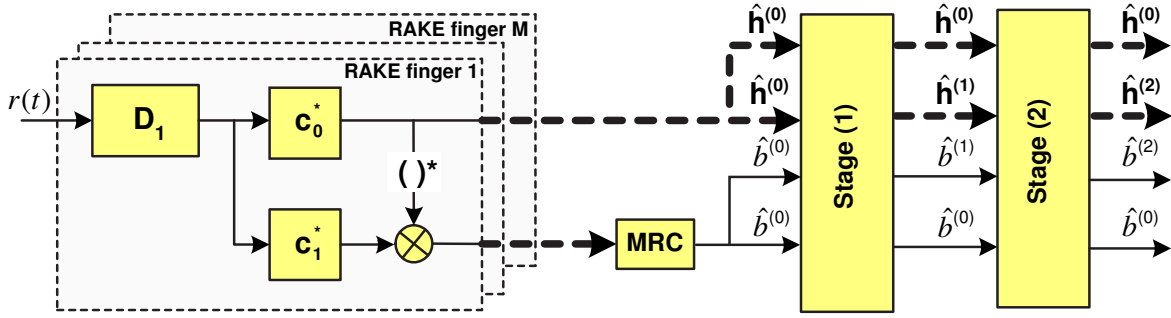
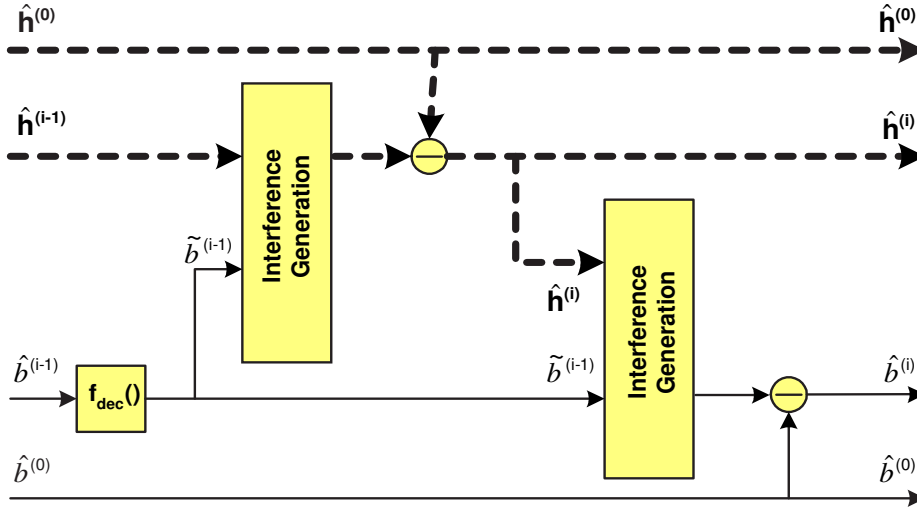


Figure C.18: IC inter chemin avec estimation de canal.

Figure C.19: Structure du i -ème étage d'annulation.

Dans l'étage initial, le récepteur - qui travaille en mode conventionnel - démodule et désétale le signal reçu. Les sorties initiales du corrélateur ($\hat{b}^{(0)}(k)$) et de l'estimateur de canal ($\hat{\mathbf{h}}^{(0)}(k) = [\hat{h}_1^{(0)}(k) \dots \hat{h}_M^{(0)}(k)]$) sont utilisées à chaque étage de l'annulateur multi étages. L'objectif de l'annulateur proposé est l'atténuation de ces interférences.

La structure du i -ème étage d'interférence est présentée dans la figure C.19. L'opération principale est l'estimation du signal transmis. Afin d'y parvenir, une décision est prise sur la sortie de l'étage $i - 1$. Cela peut être exprimé comme

$$\tilde{b}^{(i-1)}(k) = f_{dec}(\hat{b}^{(i-1)}(k)) \quad (\text{C.22})$$

Cette estimation est combinée avec l'estimation de canal de la sortie de l'étage $i - 1$ afin de reproduire les termes d'interférence compris dans l'estimation de canal initiale. Si $\hat{\mathbf{S}}^{(i)}(k) = [\hat{S}_1^{(i)}(k) \dots \hat{S}_M^{(i)}(k)]$ et $\hat{\mathbf{C}}^{(i)}(k) = [\hat{C}_1^{(i)}(k) \dots \hat{C}_M^{(i)}(k)]$ sont les répliques des termes d'interférence, l'estimation de canal mise à jour est

$$\hat{\mathbf{h}}^{(i)}(k) = \hat{\mathbf{h}}^{(0)}(k) - \hat{\mathbf{S}}^{(i)}(k) - \hat{\mathbf{C}}^{(i)}(k) \quad (\text{C.23})$$

et sont utilisés afin de reproduire les répliques des termes d'interférence présentés dans la sortie de corrélation initiale. En assumant que $\hat{I}^{(i)}(k)$ et $\hat{F}^{(i)}(k)$ sont les termes d'interférence générés, la nouvelle sortie de corrélation est

$$\hat{b}^{(i)}(k) = \hat{b}^{(0)}(k) - \hat{I}^{(i)}(k) - \hat{F}^{(i)}(k) \quad (\text{C.24})$$

Il est nécessaire de disposer de bonnes estimations de données et de canal pour que ce schéma itératif puisse bien fonctionner. Autrement, les performances seront dégradées par une erreur de propagation générée.

C.4.3 Evaluation de la performance

L'environnement de simulation est basé sur les spécifications du standard FDD-UMTS pour le cas de communication descendante. Le canal radio est constitué de $L = 2$ multi chemins Rayleigh indépendants avec puissance moyenne $P_t = [0 \ 0]dB$ et retard $\tau = [0 \ 2]T_c$. Le rapport de puissances entre canal pilote et canal de données est égal à 6.5 dB. Enfin, le récepteur râteau utilisé considère $L = 2$ doigts.

Des comparaisons au sens MSE (mean square error) entre l'estimation conventionnelle ($V=0$) et la méthode proposée pour différents nombres d'étages sont présentées dans la figure C.20. Nous pouvons remarquer que l'estimation de canal s'améliore avec l'augmentation du nombre d'étages. Après un certain nombre d'itérations ($V>6$), la méthode converge à la valeur 13 dB.

La figure C.21 donne le taux d'erreur binaire (TEB ou BER) du schéma proposé pour différents nombres d'étages. Les courbes confirment les observations faites précédemment et témoignent de la précision de la méthode proposée. Nous pouvons également voir que l'erreur de propagation sur les courbes MSE ne se traduit pas en une dégradation de la performance au sens BER.

C.4.4 Architecture reconfigurable

De même que pour l'exemple 1, le processus de génération pour les différents types d'interférence présente beaucoup de similarité entre eux et avec les opérations de démodulation RAKE et d'estimation de canal. Toutes ces fonctions sont itératives et il est possible de tirer profit de l'approche de reconfiguration de type "itération". La structure du récepteur est très proche de celle présentée pour l'exemple 1. Elle contient les 3 blocs de calcul, mémoire et supervision de configuration. Le schéma fonctionnel de l'unité de calcul est présenté en figure C.22. Le superviseur se charge de générer les signaux $\{Q_1, \dots, Q_6\}$ pour chaque mode. La table C.3 donne le type de signaux et de coefficients à générer pour chaque mode.

En considérant un opérateur synchrone et en "pipeline" fonctionnant à la fréquence F_{clk} , le temps de calcul d'une itération est de SF/F_{clk} du fait qu'il faut intégrer SF chips dans le symbole pour "désétaler". Le temps de traitement total est donc de $[2L + 4VL(L-1)] \frac{SF}{F_{clk}}$ en considérant L trajets et V étages pour l'annulateur d'interférences avec estimation de canal.

Le nombre d'itérations possibles est limité par le temps symbole de SF/F_{chip} . Les valeurs de L et V sont donc contraintes suivant l'équation:

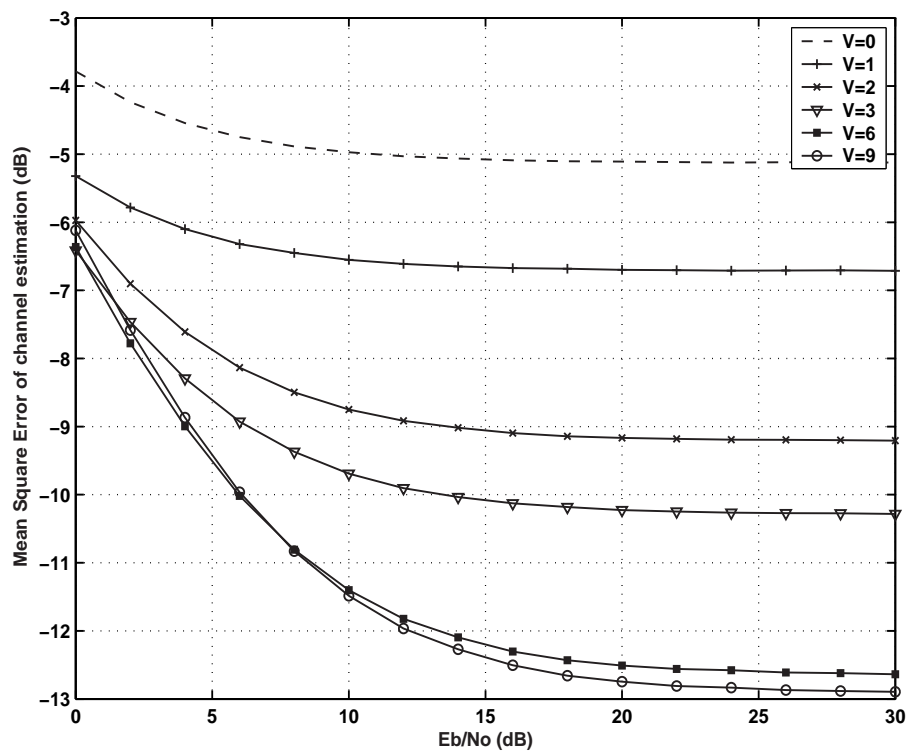


Figure C.20: Performance au sens MSE du schéma d'estimation proposé versus le schéma classique ($V=0$).

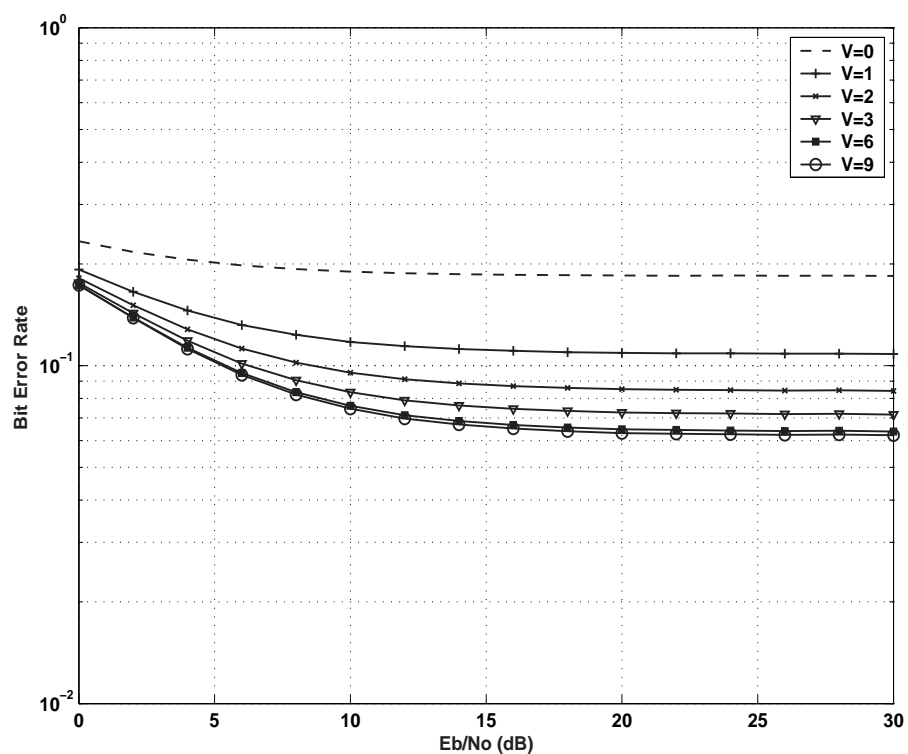


Figure C.21: Performance au sens BER du schéma d'estimation proposé versus le schéma classique ($V=0$).

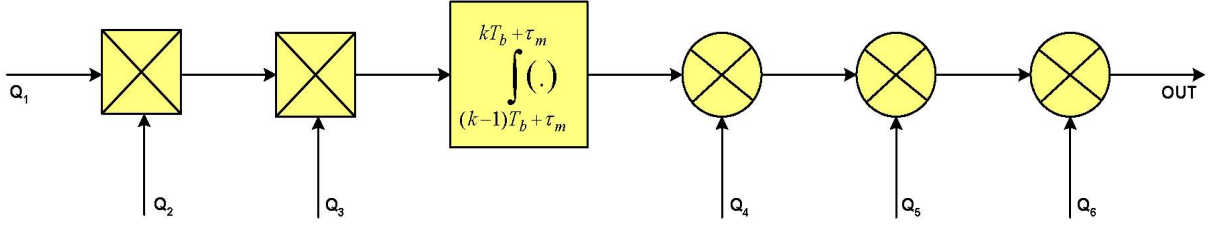


Figure C.22: de traitement de base pour le IC avec estimation de canal.

Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	OUT
$\widehat{b}(t - \tau_j)$	$c_1(t - \tau_j)$	$c_0^*(t - \tau_m)$	$\widehat{h}_j(k)$	1	A_1/NA_0	$\widehat{S}_{m_j}(k)$
1	$c_0(t - \tau_j)$	$c_0^*(t - \tau_m)$	$\widehat{h}_j(k)$	1	$1/N$	$\widehat{C}_{m_j}(k)$
$\widehat{b}(t - \tau_j)$	$c_1(t - \tau_j)$	$c_1^*(t - \tau_m)$	$\widehat{h}_j(k)$	$\widehat{h}_m^*(k)$	$1/N$	$\widehat{F}_{m_j}(k)$
1	$c_0(t - \tau_j)$	$c_1^*(t - \tau_m)$	$\widehat{h}_j(k)$	$\widehat{h}_m^*(k)$	A_0/NA_1	$\widehat{I}_{m_j}(k)$
$r(t)$	1	$c_1^*(t - \tau_m)$	$\widehat{h}_m^*(k)$	1	$1/NA_1$	m -th RAKE finger
$r(t)$	1	$c_0^*(t - \tau_m)$	1	1	$1/NA_0$	$\widehat{h}_m(k)$

Table C.3: Les différentes Configurations.

$$2L + 4VL(L - 1) < \frac{F_{clk}}{F_{chip}} \quad (C.25)$$

Il est possible d'obtenir de l'ordre de 100 en considérant les technologies CMOS $0.9\mu m$, ce qui donnerait les valeurs possibles de $L = 2$ et $V = 10$.

C.5 Conclusion

Dans ce chapitre nous avons vu qu'il était possible d'appliquer des approches de reconfiguration matérielle adaptées au traitement du signal pour les communications numériques. Les approches génériques de reconfiguration de type "multiplexage" ou "pagination" peuvent être optimisées en tirant parti de la nature des calculs quasi-similaires et itératifs. Les architectures peuvent donc se limiter à des opérateurs communs à l'ensemble des fonctions (approche factorisation). De plus elles peuvent disposer d'un "grain" de calcul très fin de façon à opérer des calculs itératifs propres à de nombreux algorithmes de communications (approche itération). Des exemples ont été donnés concernant l'égalisation effectuée dans un récepteur en râteau RAKE où les algorithmes chargés de diminuer les interférences et d'estimer le canal reposent sur un raffinement des informations à chaque itération. Les intérêts de cette approche sont d'une part une plus grande flexibilité algorithmique en jouant sur le nombre d'itérations et les modes de configuration, et d'autre part un coût de silicium, une consommation statique, non négligeable pour les nanotechnologies, et un temps de développement moindre dû à la petite taille de l'architecture à reconfigurer.

Bibliography

- [ADA97] F. Adachi, M. Sawahashi, K. Okawa, "Tree-structured generation of orthogonal spreading codes with different lengths for forward link of DS-CDMA mobile," *IEE Electronics Letters*, Vol. 33, No. 1, pp. 27-28, 1997.
- [ALO97] M. -S. Alouini, S. W. Kim, A. Goldsmith, "RAKE reception with maximal-ratio and equal-gain combining for DS-CDMA systems in Nakagami fading," *IEEE Proc. 6th International Conference on Universal Personal Communications*, Vol. 2, pp. 708-712, 1997.
- [ATH93] P. M. Athanas, H. F. Silverman, "Processor reconfiguration through instruction-set metamorphosis," *IEEE Computer Magazine*, Vol. 26, pp. 11-18, 1993.
- [BAI95] R. Baines, "The DSP Bottleneck," *IEEE Communication Magazine*, Vol. 33, pp. 46-54, 1995.
- [BAN93] J. Banker, A. Shanbhag, N. Sherwani, "Physical design tradeoffs for ASIC technologies," *Proceedings of Sixth Annual IEEE International ASIC Conference and Exhibit*, pp. 70-78, 1993.
- [BAR00] A. N. Barreto, M. Mecking, G. Fettweis, "A flexible air interface for integrated broadband mobile systems," *Proc. IEEE Vehicular Technology conference*, Vol. 3, pp. 1899-1903, 2000.
- [BEA00] N. C. Beaulieu, A. S. Toms, D. R. Pauluzzi, "Comparison of four SNR estimators for QPSK modulations," *IEEE Communication Letters*, Vol. 4, pp. 43-45, 2000.
- [BLA04] D. C. Black, J. Donovan, "SystemC: from the ground up," Kluwer Academic Publishers, 2004.
- [BOG02] H. Bogucka, K. Gzik, "Performance and complexity of a reconfigurable OFDM/CDMA transceiver," *Proc. 10th URSI National Symposium of Radio Science*, Poznan, Poland, pp. 70-75, 2002.
- [BOU02] D. Boudreau, G. Caire, G. E. Corazza, R. D. Gaudenzi, G. Gallinaro, M. Luglio, R. Lyons, J. Romero-Garcia, A. Vernucci, H. Widmer, "Wide-band CDMA for the UMTS/IMT-2000 satellite component," *IEEE Trans. on Vehic. Technol.*, Vol. 51, No. 2, pp. 306-331, 2002.
- [BOU01] H. Boujemaa, "Récepteur UMTS optimisé," PhD thesis, ENST editions, 2001.

- [BRA99] W. R. Braun, U. Dersch, "A physical mobile radio channel model," *IEEE Transactions on Vehicular Technology*, Vol. 40, pp. 472-482, 1999.
- [BRA02] J. Brakensiek et al., "Software Radio Approach for Reconfigurable Multi-standard Radios," *Proc. 13th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Vol. 1, pp. 110-14, 2002.
- [BRO92] S. D. Brown, R. J. Francis, J. Rose, Z. G. Vranesic, "Field-programmable gate arrays," Kluwer Academic Publishers, 1st Ed, 1992.
- [BUR03] P. Burns, "Software defined radio for 3G," Artech House Editions, 2003.
- [BUR03b] S. Burykh and K. Abed-Meraim, "Blind Interference Cancellation for Downlink DS/CDMA," *IEEE Workshop on Signal Processing Advances in Wireless Communications*, pp. 279-283, 2003.
- [CHA99] C. D. Charalambous, N. Menemenlis, "Stochastic models for short-term multipath fading channels: chi-square and Ornstein-Uhlenbeck processes," *Proceedings of the 38th IEEE Conference on Decision and Control*, Vol. 5, pp. 4959-4964, 1999.
- [CHA01] K. Chadha, J. R. Cavallaro, "A reconfigurable Viterbi decoder architecture," *IEEE Proc. of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, Vol. 1, pp. 66-71, 2001.
- [CHI99] C. Chien, M. B. Strivastana et al., "Adaptive radio for multimedia wireless links," *IEEE Journal on selected areas in communications*, Vol. 17, No. 5, pp. 793-813, 1999.
- [COO83] C. Cook, H. Marsh, "An introduction to spread spectrum," *IEEE Communications Magazine*, Vol. 21, pp. 8-16, 1983.
- [COR99] M. S. Corson, J. P. Macker, G. H. Cirincione, "Internet-based mobile ad hoc networking," *IEEE Internet Computing*, Vol. 3, pp. 63-70, 1999.
- [DAG03] I. Dagres, A. Polydoros, "Dynamic transceivers: adaptivity and reconfigurability at the signal-design level," *Proc. Software Defined Radio Technical Conference and Product Exposition*, Orlando, USA, 2003.
- [DAG05] I. Dagres, A. Zalonis, N. Dimitriou, K. Nikitopoulos, A. Polydoros, "Flexible radio: a framework for optimized multimodal operation via dynamic signal design," *Eurasip Journal on Wireless Communications and Networking*, pp. 284-297, 2005.
- [DEB00] M. Debbah, R. R., "MIMO channel modeling and the principle of maximum entropy," *IEEE Transactions on Inf. Theory*, Vol. 51, pp. 1667-1690, 2005.
- [DEM04] P. Demestichas, G. Vivier, K. El-Khazen, M. Theologou, "Evolution in wireless systems management concepts: from composite radio environments to reconfigurability," *IEEE Communications Magazine*, Vol. 42, No. 5, pp. 90-98, 2004.
- [DIV98] D. Divsalar, M. K. Simon, D. Raphaeli "Improved parallel interference cancellation for CDMA," *IEEE Trans. on commun.*, Vol. 46, No. 2, pp. 258-268, 1998.

- [DON99] T. Donnelly, K. Jackson, J. Moreno, "Reconfigurable devices for error-detection/correction applications," *IEE Proc. Reconfigurable Systems*, pp. 7/1-7/4, 1999.
- [DOU02] L. P. Douglas, "VHDL programming by example," McGraw-Hill Editions, 4th ed, 2002.
- [DRE01] N. J. Drew, M. M. Dillinger, "Evolution toward reconfigurable user equipment," *IEEE Communications Magazine*, Vol. 39, pp. 158-164, 2001.
- [DUE95] A. Duel-Hallen, J. Holtzman, Z. Zvonar, "Multiuser detection for CDMA systems," *IEEE Personal Communications*, Vol. 2, pp. 46-58, 1995.
- [ESM93] R. Esmailzadeh, M. Nakagawa, "Pre-rake diversity combination for direct sequence spread spectrum communications systems," *Proc. IEEE International conference on communications*, Vol. 1, pp. 463-467, 1993.
- [FRI00] J. Fridman, Z. Greenfield, "The TigerSHARC DSP architecture," *IEEE Micro*, Vol. 20, pp. 66-76, 2000.
- [GAR00] V. K. Garg, "IS-95 cdma and cdma2000," Prentice Hall PTR editions, 1st Edition, 2000.
- [GAR00b] V. K. Garg, O. T. W. Yu, "Integrated QoS support in 3G UMTS networks," *IEEE Proc. Wireless Communications and Networking Conference WCNC'00*, Vol. 3, pp. 1187-1192, 2000.
- [GAU99] R. De Gaudenzi, G. Gallinaro et al., "ESA satellite wideband CDMA radio transmission technology for the IMT-2000/UMTS satellite component: features and performance," *IEEE Proc. GLOBECOM'99*, Vol. 5, pp. 2699-2703, 1999.
- [GOL97] J. Goldsmith, S. G. Chua, "Variable-rate variable-power MQAM for fading channels," *IEEE Transactions on Communications*, Vol. 45, pp. 1218-1230, 1997.
- [GRA95] S. D. Gray, M. Kocic, D. Brady, "Multi-User Detection in Mismatched Multiple-Access Channels," *IEEE Transaction on Communications*, Vol. 43, No. 12, pp. 3080-3089, 1995.
- [GRA00] E. Grayver, B. Daneshrad, "A reconfigurable 8 GOP ASIC architecture for high-speed data communications," *IEEE Journal on selected areas in communications*, Vol. 18, No. 11, pp. 2161-2171, 2000.
- [HAL98] S. Halter, M. Oberg, P. M. Chau, P. H. Siegel, "Reconfigurable signal processor for channel coding and decoding in low SNR wireless communications," *IEEE Proc. Workshop on Signal Processing Systems- SIPS'98*, pp. 260-274, 1998.
- [HAN02a] J. -K. Han, M. -W. Lee, H. -K. Park, "Principal ratio combining for pre/post-rake diversity," *IEEE Communication Letters*, Vol. 6, No. 6, pp. 234-236, 2002.
- [HAN00] L. Hanzo, W. Webber, T. Keller, "Single- and Multi-carrier quadrature amplitude modulation: principles and applications for personal communications, WLANs and broadcasting," IEEE Press-John Wiley & Sons editions, 2000.

- [HAN02] L. Hanzo, C. H. Wong, M. S. Yee, "Adaptive wireless transceivers- turbo codes, turbo equalized and space time coded, TDMA, CDMA and OFDM systems," IEEE Press-John Wiley & Sons editions, 1st ed., 2002.
- [HAY02] S. Haykin, "Adaptive filter theory," Prentice Hall Editions, Fourth Edition, 2002.
- [HAR01] L. Harju, M. Kuulusa, J. Nurmi, "A flexible rake receiver architecture for WCDMA mobile terminals," *Proc. 3rd IEEE Workshop on Signal Processing Advances in Wireless commun.*, Taoyuan, Taiwan, pp. 9-12, 2001.
- [HAR02] L. Harju, M. Kuulusa, J. Nurmi, "Flexible implementation of a WCDMA rake receiver," *Proc. 3rd IEEE Workshop on Signal Processing Systems*, pp. 177-182, 2002.
- [HAU97a] S. Hauck, T. W. Fry, M. M. Hosler, "The Chimaera reconfigurable functional unit," *IEEE Proc. of 5th Symposium on FPGAs for Custom Computing Machines*, pp. 87-96, 1997.
- [HAU98] S. Hauck, "The roles of FPGAs in reprogrammable systems," *Proceedings of the IEEE*, Vol. 86, pp. 615-638, 1998.
- [HAU97] J. R. Hauser and J. Wawrzynek, "Garp: a MIPS Processor with a Reconfigurable Coprocessor," *IEEE Proc. 5th Annual Symp. FPGAs for Custom Comp. Machines*, pp. 12-21, 1997.
- [HAS03] M. Hasan, T. Arslan, J. S. Thompson, "A delay spread based low power reconfigurable FFT processor architecture for wireless receiver," *IEEE Proc. International Symposium on System-on-Chip*, pp. 135-138, 2003.
- [HEN96] J. L. Hennessy, D. A. Patterson, "Computer architecture a quantitative approach," Morgan kaufmann publishers, Second edition, 1996.
- [HEY02] P. M. Heysters et al., "A Reconfigurable Function Array Architecture for 3G and 4G Wireless Terminals," *Proc. 2002 World Wireless Cong.*, San Francisco, CA, pp. 399- 404, 2002.
- [HOL99] H. Holma, K. Heiska, "Performance of high bit rates with WCDMA over multipath channels," *IEEE 49th Vehicular Technology Conference*, Vol. 1, pp. 25-29, 1999.
- [HOL02] H. Holma, A. Toskala, *WCDMA for UMTS- radio access for third generation mobile communications*, John Wiley & Sons editions, second ed., 2002.
- [HOO99] K. Hooli, M. Juntti, M. Latva-aho, "Inter-path Interference Suppression in WCDMA Systems with Low Spreading Factors," *Proc. IEEE Vehicular Technology conference*, Amsterdam, Netherlands, pp. 421-425, 1999.
- [HUI98] A. L. C. Hui, K. B. Letaief, "Successive Interference Cancellation for Multiuser Asynchronous DS/CDMA Detectors in Multipath Fading Links," *IEEE Trans. on Communications*, Vol. 46, no. 3, pp. 384-391, 1998.

- [KAR99] M. S. Karaliopoulos, F. N. Pavlidou, "Modeling the land mobile satellite channel: a review," *IEEE Electronics & Communication Engineering Journal*, Vol. 11, pp. 235-248, 1999.
- [KAR04] M. Karaliopoulos et al., "Satellite radio interface and radio resource management strategy for the delivery of multicast/broadcast services via an integrated satellite-terrestrial system," *IEEE Communications Magazine*, Vol. 42, pp. 108-117, 2004.
- [KLE96] A. Klein, G. Kaleh, P. W. Baier, "Zero forcing and minimum mean-square-error equalization for multiuser detection in code-division multiple-access channels," *IEEE Trans. on Vehic. technol.*, Vol. 45, No. 2, pp. 276-287, 1996.
- [KLE97] A. Klein, "Data detection algorithms specially designed for the downlink of CDMA mobile radio systems," *Proc. IEEE Vehicular Technology conference*, Phoenix, AZ, Vol. 1, pp. 203-207, 1997.
- [KO01] K. Ko, S. Choi, Y. Lee, C. Kang, D. Hong "More accurate performance analysis of interference canceler in an asynchronous DS-CDMA system over the multipath fading channel," *Proc. Military Communications Conference-MILCOM'01*, Vol. 2, pp. 1414-1417, 2001.
- [KOU98] D. Koulakiotis, A. H. Aghvami, "Evaluation of a DS/CDMA multiuser receiver employing a hybrid form of interference cancellation in Rayleigh-fading channels," *IEEE Communications Letters*, Vol. 2, pp. 61-63, 1998.
- [KOU00] D. Koulakiotis, A. H. Aghvami, "Data detection techniques for DS/CDMA mobile systems: A review," *IEEE Personal communications*, Vol. 7, pp. 24-34, 2000.
- [KOU00b] S. Kourtis, P. McAndrew, P. Tottle, "Technology requirements of the 3GPP-TDD terminal," *Proc. 1st IEE International conference on 3G mobile communication technologies*, London, UK, pp. 89-93, 2000.
- [LET98] P. Lettieri, M. B. Strivastana, "Adaptive frame length control for improving wireless link throughput, range, and energy efficiency," *Proc. 7th IEEE Conf. INFOCOM Computer Communications*, San Francisco, USA, pp. 564-571, 1998.
- [LI94] Y. Li, R. Steele, "Serial interference cancellation method for CDMA," *IEE Electronics Letters*, Vol. 30, pp. 1581-1583, 1994.
- [LUN01] D. Lund et al., "Convolutional decoding for reconfigurable mobile systems," *Proc. Second IEEE Int. Conf. on 3G Mobile Communication Technologies*, London, England, pp. 297-301, 2001.
- [MAI01] L. Mailaender, "Low-complexity implementation of cdma downlink equalization," *Proc. 2nd IEE 3G Mobile Communication Technologies*, 3G'2001, pp. 396-400, 2001.
- [MAM02] A. Mammela, A. Polydoros, P. Jarvensivu, "Data and channel estimators: a systematic classification," *Proceedings of the X National Symposium of Radio Science (URSI 2002)*, Poznan, PL, pp. 13-25, 2002.

- [MAN02] G. Mandyam, J. Lai, "Third-generation cdma systems for enhanced data services," Academic press editions, 1st edition, 2002.
- [MAN04] U. Manzoli, M. L. Merani, "Comparison between the performance of conventional and selective RAKE receiver schemes," *IEEE Transactions on Vehicular Technology*, Vol. 53, pp. 621-625, 2004.
- [MAR98] J. Mar, H. -Y. Chen, "Performance analysis of cellular CDMA networks over frequency-selective fading channel," *IEEE Trans. on Vehic. Technol.*, Vol. 47, No. 4, pp. 1234-1244, 1998.
- [MEH99] A. Mehrnia, H. Hashemi, "Mobile satellite propagation channel. Part 1-a comparative evaluation of current models," *Proc. IEEE Vehicular Technology conference*, Vol. 5, pp. 2775-2779, 1999.
- [MES03] D. Mesquita, F. Moraes, J. Palma, L. Moller, N. Calazans, "Remote and partial re-configuration of FPGAs tools and trends," *Proc. IEEE Parallel and Distributed Processing Symposium*, pp. 1-8, 2003.
- [MIL00] S. L. Miller, M. L. Honig, L. B. Milstein, "Performance analysis of MMSE receivers for DS-CDMA in frequency-selective fading channels," *IEEE Transactions on Communications*, Vol. 48, pp. 1919-1929, 2000.
- [MIT95] J. Mitola, "The software radio architecture," *IEEE Communication Magazine*, Vol. 33, pp. 26-38, 1995.
- [MIT99] J. Mitola, "Software radio architecture: A mathematical perspective," *IEEE- Journal on selected areas in communications*, Vol. 17, No. 4, pp. 514-538, 1999.
- [MIT99b] J. Mitola, G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, Vol. 6, pp. 13-18, 1999.
- [MOS96] S. Moshavi, "Multi-user detection for DS-CDMA communications," *IEEE Communications Magazine*, pp. 124-135, 1996.
- [MOT99] D. Mottier, D. Castelain, "A Doppler estimation for UMTS-FDD based on channel power statistics," *Proc. IEEE Vehicular Technology conference VTC'99*, Vol. 5, pp. 3052-3056, 1999.
- [MOU04] B. Mouhouche, K. Abed-Meraim, S. Burykh, "Spreading code detection and blind interference cancellation for DS/CDMA downlink," *Proc. IEEE Eighth International Symposium on Spread Spectrum Techniques and Applications*, Sydney, Australia, pp. 774-778, 2004.
- [NOG04] D. Noguet, "A reconfigurable systolic architecture for UMTS/TDD joint detection real time computation," *Proc. IEEE Eighth International Symposium on Spread Spectrum Techniques and Applications*, Sydney, Australia, pp. 957-961, 2004.

- [NOG04b] D. Nogu t, J. -P. Bouyoud, L. Zaghdoudi, D. Varreau, B. Jechoux, P. Le Corre, X. Lagrange, "A hardware testbed for UMTS/TDD joint detection base-band receivers," *Proc. IEEE Eighth International Symposium on Spread Spectrum Techniques and Applications*, Sydney, Australia, pp. 972-976, 2004.
- [OLU94] K. A. Olukotun, R. Helaihel, J. Levitt, R. Ramirez, "A software-hardware cosynthesis approach to digital system simulation," *IEEE Micro*, Vol. 14, pp. 48-58, 1994.
- [PAL00] J. Palicot, C. Roland, "A two step architecture for an adaptive receiver," *IEE First International Conference on 3G Mobile Communication Technologies*, pp. 301-305, 2000.
- [PAL03] J. Palicot, "A new concept for wireless reconfigurable receivers," *IEEE Communications Magazine*, Vol. 41, No. 7, pp. 124-132, 2003.
- [PAP00] A. Papasakellariou, "Overview of interference cancellation for CDMA wireless systems," *IEEE International Conference on Information Technology: coding and computing*, Las Vegas, Nevada, pp. 86-91, 2000.
- [PAP02] A. Papoulis, S. U. Pillai, "Probability, random variables and stochastic processes," McGraw Hill Editions, 4th Edition, 2002.
- [PAT93] P. R. Patel, J. M. Holtzman, "Analysis of a simple successive interference cancellation scheme in a DS/CDMA system," *IEEE Selected areas in communications*, Vol. 12, pp. 796-807, 1994.
- [PIC82] R. Pickholtz, D. Schilling, L. Milstein, "Theory of spread-spectrum communications-A tutorial," *IEEE Trans. on Commun.*, Vol. 30, pp. 855-883, 1982.
- [POL03] A. Polydoros et al., "Wind-Flex: Developing a Novel Testbed for Exploring Flexible Radio Concepts in an Indoor Environment," *IEEE Commun. Mag.*, Vol. 41, pp. 116-122, 2003.
- [PRA01] O. Prator, C. Unger, A. Zoch, G. P. Fettweis, "Impact of channel estimation on the 3GPP-TD-CDMA," *IEEE Global Telecomm. Conf. - GLOBECOM'01*, San Antonio, TX, Vol. 6, pp. 3365-3369, 2001.
- [PRI58] R. Price and P. E. Green, "A communication technique for multipath channels," *Proc. IRE*, vol. 46, pp. 555-570, 1958.
- [PRO95] J. G. Proakis, "Digital Communications," McGraw-Hill International editions, third ed., 1995.
- [PUR77] M. B. Pursley, "Performance Evaluation for Phase-Coded Spread-Spectrum Multiple-Access Communication-Part I," *IEEE Trans. on Communications*, Vol. 25, pp. 795-799, 1977.
- [QAR01] K. A. Qaraqe, "Channel estimation algorithms for third generation W-CDMA communication systems," *Proc. IEEE Vehicular Technology Conf.*, Rhodes, Greece, pp. 2675-2679, 2001.

- [RAJ02] S. Rajagopal, S. Bhashyam, J. R. Cavallaro, B. Aazhang, "Real-time algorithms and architectures for multiuser channel estimation and detection in wireless base-station receivers," *IEEE Trans. on wireless commun.*, Vol. 1, No. 3, pp. 468-478, 2002.
- [RAO99] Y. S. Rao, A. Kripalani, "cdma2000 mobile radio access for IMT 2000," *IEEE International conference on Personal Wireless Communications*, pp. 6-15, 1999.
- [RAS00] L. K. Rasmussen, "On Ping-Pong Effects in Linear Interference Cancellation for CDMA," *IEEE Proc. International Symposium on Spread Spectrum Techniques and Applications*, New Jersey, USA, pp. 348-352, 2000.
- [RAT00] S. Ratanamahatana, H. M. Known, "Channel estimation for controlled 3G CDMA," *Proc. IEEE Vehicular Technology conference VTC'00*, Vol. 3, pp. 2429-2433, 2000.
- [SES99] I. P. Seskar, N. B. Mandayam, "A software radio architecture for linear multiuser detection," *IEEE Journal on Selected areas in communic.*, Vol. 17, No. 5, pp. 814-823, 1999.
- [SHA48] C. E. Shannon, "A mathematical theory of communications," *Bell Syst. Tech. J.*, Vol. 27, pp. 379-423, 623-656, July 1948.
- [SHE99] R. Shepherd, "Engineering the embedded software radio," *IEEE Communications Magazine*, Vol. 37, pp. 70-74, 1999.
- [SHI01] D. -J. Shin, W. Sung, I. -K. Kim, "Simple SNR estimation methods for QPSK modulated short bursts," *IEEE Proc. GLOBECOM'01*, pp. 3644-3647, 2001.
- [SHI02] S. Shin et al, "CDMA2000 1X Performance comparison with pilot power ratio," *IEEE Int. Seminar on Broadband Communications, Access, Transmission, Networking*, Zurich, Switzeland, pp. 52/1-52/6, 2002.
- [SIN02] D. Singh, A. Kumari, R. K. Mallik, S. S. Jamuar, "Analysis of RAKE reception with MRC and imperfect weight estimation for binary coherent orthogonal signaling," *IEEE Communications Letters*, Vol. 6, pp. 245-247, 2002.
- [SMI99] M. J. S. Smith, "Application-Specific Integrated Circuits," Addison Wesley Editions, 5th Edition, 1999.
- [SMI01] L. T. Smit, G. J. M. Smit, P. J. M. Havinga, "Parameter selection at run-time to optimize energy efficiency," *Proc. 2nd workshop on Embedded systems*, Veldhoven, Netherlands, pp. 227-236, 2001.
- [SRI00] S. Srikanteswara, J. H. Reed, P. M. Athanas, "Implementation of a reconfigurable soft radio using the layered radio architecture," *IEEE Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, Vol. 1, pp. 360-364, 2000.
- [SRI00a] S. Srikanteswara, J. H. Reed, P. Athanas, R. Boyle, "A soft radio architecture for reconfigurable platforms," *IEEE Communications Magazine*, Vol. 38, pp. 140-147, 2000.

- [SRI00b] S. Srikanteswara, M. Hosemann, J. H. Reed, P. M. Athanas, "Design and implementation of a completely reconfigurable soft radio," *IEEE Radio and Wireless Conference*, pp. 7-11, 2000.
- [SRI03] S. Srikanteswara, R. C. Palat, J. H. Reed, P. Athanas, "An overview of configurable computing machines for software radio handsets," *IEEE Communications Magazine*, Vol. 41, No. 7, pp. 134-141, 2003.
- [SUN02] Q. Sun, D. C. Cox, "A pipelined multi-stage parallel interference canceller for CDMA with realistic channel estimation," *IEEE Proc. WCNC'02*, Orlado, USA, pp. 369-373, 2002.
- [TOD05] T. J. Todman, G. A. Constantinides, S. J. E. Wilton, O. Mencer, W. Luk, P. Y. K. Cheung, "Reconfigurable computing: architectures and design methods," *IEE Proceedings Computers and Digital Techniques*, Vol. 152, pp. 193-207, 2005.
- [TSA96] M. K. Tsatsanis, G. B. Giannakis, "Optimal decorrelating receivers for DS-CDMA systems: a signal processing framework," *IEEE Transactions on Signal Processing*, Vol. 44, pp. 3044-3055, 1996.
- [UEB97] C.W. Ueberhuber, "Numerical Computation: Methods, software and analysis," Springer-Verlag (Berlin), 1997.
- [VAR90] M. Varanasi, B. Aazhang "Multistage detection in asynchronous code-division multiple-access communications," *IEEE Trans. on commun.*, Vol. 38, pp. 509-519, 1990.
- [VER86] S. Verdu, "Minimum probability of error for asynchronous gaussian multiple-access channels," *IEEE Trans. on information theory*, Vol. 32, No. 1, pp. 85-96, 1986.
- [VIT94] A. J. Viterbi, "The Orthogonal-Random Wave form Dichotomy for Digital Mobile Personal Communications," *IEEE Personal Commun.*, pp. 18-24, 1994.
- [VIT95] A. J. Viterbi, "CDMA: Principles of spread spectrum communications," Prentice Hall PTR editions, 1st edition, 1995.
- [WEB95] W. Webb, R. Steele, "Variable rate QAM for mobile radio," *IEEE Transactions on Commun.*, Vol. 43, pp. 2223-2230, 1995.
- [WON00] C. H. Wong, L. Hanzo, "Upper-bound performance of a wide-band adaptive modem," *IEEE Trans. on Commun.*, Vol. 48, No. 3, pp. 367-369, 2000.
- [XIE90] Z. Xie, R. T. Short, C. K. Rushforth, "A family of suboptimum detectors for coherent multiuser communications," *IEEE Journal on Select. Areas Commun.*, Vol. 8, pp. 685-690, 1990.
- [YOO93] Y. C. Yoon, R. Kohno, H. Imai, "A spread-spectrum multiaccess system with cochannel interference cancellation for multipath fading channels," *IEEE Journal on selected areas in communications*, Vol. 11, No. 7, pp. 1067-1075, 1993.
- [ZHA03] W. Zha, S. D. Blostein, "Soft-decision multistage multiuser interference cancellation," *IEEE Trans. on Vehicular Technol.*, Vol. 52, No. 2, pp. 380-389, 2003.

- [ZVO96] Z. Zvonar, D. Brady, "Linear multipath-decorrelating receivers for CDMA frequency-selective fading channels," *IEEE Transactions on Communications*, Vol. 44, pp. 650-653, 1996.
- [ETSI98] ETSI TR 101 112 (UMTS 30.03), Universal Mobile Telecommunication System (UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS, version 3.2.0, 1998.
- [E2R] IST-2003-507995 E^2R (End-to-End Reconfigurability) project, Web Site, www.e2r.motlabs.com
- [WINDFLEX] IST-1999-10025 WIND-FLEX (Wireless Indoor Flexible High Bitrate Modem Architecture) project, Web Site, www.vtt.fi/ele/research/els/projects/windflex.htm
- [3GPPa] Third Generation Partnership Project, 3GPP TS 25.101 version 6.7.0, User Equipment (UE) radio transmission and reception (FDD), March 2005.
- [3GPPb] Third Generation Partnership Project, 3GPP TS 25.101 version 6.6.0, Physical layer procedures (FDD), June 2005.
- [3GPPc] Third Generation Partnership Project, Web Site, www.3gpp.org
- [3GPP2] Third Generation Partnership Project 2, Web Site, www.3gpp2.org
- [ANALa] Analog Devices, Web Site, www.analog.com/
- [ANALb] Analog Devices, Web Site Document, www.analog.com/dsp, "Engineer to engineer note EE-147", pp. 1-11, 2001.
- [TI] Texas Instruments, Web Site, www.ti.com
- [XIL] Xilinx, Web Site, www.xilinx.com

List of publications

- [KRI03a] I. Krikidis, J. -L. Danger, L. Naviner, "CDMA2000 1X : Un récepteur reconfigurable minimisant la consommation de puissance," *Journées Nationales du Réseau Doctorale de Microélectronique - JNRDM'03*, Toulouse, France, pp. 389-391, 2003.
- [KRI03b] I. Krikidis, J. -L. Danger, L. Naviner, "UMTS Vs CDMA2000 for a satellite environment," *International Workshop on Computational Management Science, Economics, Finance and Engineering - CMSEFE'03*, Limassol, Cyprus, 2003.
- [KRI03c] I. Krikidis, J. -L. Danger, L. Naviner, "CDMA2000 for a satellite environment," *7th IEEE International Conference On Telecommunications - ConTel'03*, Zagreb, Croatia, Vol. 2, pp. 469-473, 2003.
- [KRI03d] I. Krikidis, J. -L. Danger, L. Naviner, "CDMA2000 1X: An adaptive low power rake receiver," *3rd IASTED International Conference on Wireless and Optical Communications - WOC'03*, Banff, Alberta, Canada, pp. 395-400, 2003.
- [KRI03e] I. Krikidis, J. -L. Danger, L. Naviner, "A reconfigurable RAKE receiver for high data rates," *IEEE International Symposium On Intelligent Signal Processing and Communication Systems - ISPACS'03*, Awaji Island, Japan, 2003.
- [KRI04a] I. Krikidis, J. -L. Danger, L. Naviner, "A finger configuration algorithm for a reconfigurable Rake receiver," *IEEE Wireless Communications and Networking Conference - WCNC'04*, Atlanta, Georgia, USA, Vol. 1, pp. 311-315, 2004.
- [KRI04b] I. Krikidis, J. -L. Danger, L. Naviner, "A DS-CDMA multi-stage inter-path interference canceller for high bit rates," *IEEE International Symposium on Spread Spectrum Techniques and Applications - ISSSTA'04*, Sydney, Australia, pp. 405-408, 2004.
- [KRI05a] I. Krikidis, J. -L. Danger, L. Naviner, "Approche itérative pour la reconfiguration matérielle : exemple du récepteur rake," pp. 135-160, Chapter in the book : G. Vivier, "Les systèmes radiomobiles reconfigurables," *Traité IC2- série Réseaux et Télécommunications*, Editions Hermes Lavoisier, France, 2005.
- [KRI05b] I. Krikidis, J. -L. Danger, L. Naviner, "Adéquation-Algorithmes-Architecture dans le cadre de reconfigurabilité: Egaliseur reconfigurable d'un système CDMA," *Journées Francophones sur l'Adéquation Algorithme Architecture - JFAAA'05*, Dijon, France, pp. 53-57, 2005.

- [KRI05c] I. Krikidis, J. -L. Danger, L. Naviner, "Flexible and reconfigurable receiver architecture for WCDMA systems with low spreading factors," *IEE-Electronics Letters*, Vol. 41, No. 1, pp. 22-24, January 2005.
- [KRI05d] D. Cardoso, I. Krikidis, L. Naviner, J. -L. Danger, M. Barros, B. Neto, "Implementation of a digital receiver for DS-CDMA communication systems using HW/SW codesign," *IEEE International Midwest Symposium on Circuits and Systems*, MWSCAS'05, Cincinnati, Ohio, 2005.
- [KRI05e] I. Krikidis, J. -L. Danger, L. Naviner, "Reconfigurable implementation issues of a detection scheme for DS-CDMA high data rate connections," *IEEE International Symposium on Personal Indoor and Mobile Radio Communications - PIMRC'05*, Berlin, Germany, 2005.
- [KRI05f] I. Krikidis, J. -L. Danger, L. Naviner, "A two-layer reconfigurability concept for DS-CDMA high data rate communications," *IEEE Wireless Communications Magazine*, (accepted for publication), 2006.
- [KRI05g] D. C. de Souza, I. Krikidis, L. Naviner, J.-L. Danger, M. A. de Barros, B. G. Aguiar Neto, "Heterogeneous Implementation of a Rake Receiver For DS-CDMA Communication Systems," *IEEE International Conference on Electronics, Circuits and Systems - ICECS'05*, Gammarth, Tunisie, Decembre 2005.