

Ibrahim A. Bello  
Basel Halak

# Algorithms and VLSI Implementations of MIMO Detection

 Springer

# Algorithms and VLSI Implementations of MIMO Detection

Ibrahim A. Bello • Basel Halak

# Algorithms and VLSI Implementations of MIMO Detection

 Springer

Ibrahim A. Bello  
Dyson Technology Ltd.,  
Bristol, UK

Basel Halak  
University of Southampton  
Southampton, UK

ISBN 978-3-031-04511-0      ISBN 978-3-031-04512-7 (eBook)  
<https://doi.org/10.1007/978-3-031-04512-7>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To our families and friends...*

# Preface

The use of multiple antennas at the transmitter and receiver of a wireless communication system, otherwise known as multiple-input multiple-output (MIMO), has been one of the most significant milestones in the field of wireless communications in the past few decades. MIMO offers several advantages to wireless communications systems, including increased diversity and an increased throughput using the same bandwidth as single-antenna systems. Presently, wireless communications standards supporting several megabits-per-second in throughput have been developed and are in use in many consumer products. However, despite these advantages, the complexity of the signal detection at the receiver in a MIMO system is exacerbated as a result of the multiple interferences at each receiver antenna. The maximum likelihood (ML) detector, which offers the optimal bit error rate performance, has a prohibitive complexity that scales exponentially with the number of antennas and is not suitable for hardware implementation.

Several algorithms have been proposed in the literature to implement MIMO signal detection with low complexity. These include linear detection algorithms, such as zero-forcing and minimum mean squared error, as well as non-linear detection schemes, such as lattice-reduction-based techniques and also tree search detection. The main objective of this book is to provide a broad overview of these algorithms while highlighting their VLSI implementation aspects. Given the importance of low power consumption in modern communications, we will pay special attention to the notable techniques proposed in the literature to achieve low-power and low-complexity hardware implementations of MIMO detection algorithms. Furthermore, we will provide hardware design examples and results, supplemented with several MATLAB code examples, which we hope will help the reader quickly gain an understanding and appreciation of this very interesting research topic.

Bristol, UK  
Southampton, UK  
February 2022

Ibrahim A. Bello  
Basel Halak

# Acknowledgements

We would like to thank colleagues and staff at the University of Southampton for their support towards the completion of this book. In particular, we would like to acknowledge the fruitful discussions and collaborations we had with Prof Mark Zwolinski and Dr Mohammed El-Hajjar on several publications in this research area. We would also like to acknowledge the Petroleum Technology Development Fund (PTDF), Nigeria, for their financial support. Finally, we would like to thank the Springer Nature editorial team for their guidance throughout the publication process.

# Contents

<b>1</b>	<b>Introduction</b>	1
1.1	Chapter Overview	1
1.2	Introduction	1
1.3	System Model	3
1.4	MIMO Detection	5
1.5	Hardware Implementation	7
1.6	Design Tradeoffs	8
1.7	Overview of the Book	10
1.8	Conclusion	10
	References	11
<b>2</b>	<b>Linear Detection Techniques for MIMO</b>	13
2.1	Chapter Overview	13
2.2	Introduction	13
2.3	Zero-Forcing	14
2.4	Minimum Mean Square Error	15
2.5	Complexity of Linear Detection	16
2.6	Successive Interference Cancellation Aided Linear Detection	16
2.7	Lattice Reduction	21
2.7.1	Lattice Reduction Aided Linear Detection	22
2.7.2	Lattice Reduction with Successive Interference Cancellation	23
2.8	Linear Detection-Based Preprocessing	24
2.9	BER Simulation	25
2.10	Matrix Operations	29
2.10.1	Matrix Inversion	30
2.10.2	QR Decomposition	32
2.11	Conclusion	40
	Appendix	41
	References	43



<b>3</b>	<b>Algorithm and VLSI Implementation of Sphere Decoding</b>	45
3.1	Chapter Overview	45
3.2	Introduction	45
3.3	Sphere Decoding Tree Search	46
3.4	Schnorr–Euchner Enumeration	49
3.4.1	Tabular Enumeration	50
3.4.2	Real-Valued Channel Decomposition	51
3.4.3	Orthogonal Real-Valued Channel Decomposition	53
3.5	Complexity of Sphere Decoding	53
3.5.1	Detection Ordering	54
3.5.2	Early Termination	55
3.6	Soft-Output Sphere Decoding	56
3.6.1	List Sphere Decoder	56
3.6.2	Single Tree-Search Soft-Output Sphere Decoder	57
3.7	Sphere Decoder Simulation	58
3.8	Hardware Implementation	60
3.8.1	Previous Works	60
3.8.2	Design Example	62
3.9	Design Considerations	70
3.10	Conclusion	71
	Appendix	71
	References	73
<b>4</b>	<b>Algorithm and VLSI Implementation of K-Best Detection</b>	75
4.1	Chapter Overview	75
4.2	Introduction	75
4.3	<i>K</i> -Best Algorithm	77
4.4	Real-Valued Channel Model <i>K</i> -Best Detection	80
4.5	Non-constant <i>K</i> -Best Detection	81
4.6	Sorting	82
4.6.1	Bubble Sort	82
4.6.2	Multi-Cycle Merge	83
4.6.3	Batcher’s Merge	84
4.6.4	Relaxed Sort	86
4.6.5	Schnorr–Euchner Enumeration	86
4.7	Preprocessing	87
4.8	Complexity of the <i>K</i> -Best Algorithm	87
4.9	Hardware Implementation	89
4.9.1	Previous Works	90
4.9.2	Design Example	91
4.10	Design Considerations	104
4.10.1	Throughput	104
4.10.2	Channel Model	105
4.10.3	Architecture	106
4.11	Conclusion	107

Appendix .....	108
References .....	109
<b>5 Design Methodology for MIMO Detection .....</b>	<b>111</b>
5.1 Chapter Overview .....	111
5.2 Introduction .....	111
5.3 Review of MIMO Detection .....	113
5.4 Modelling and Simulation .....	114
5.4.1 MATLAB Executable Files .....	115
5.4.2 Vectorised Operations .....	115
5.4.3 Multicore Operation .....	116
5.4.4 Computer Clusters .....	117
5.5 Register Transfer Level Implementation .....	117
5.5.1 Verilog .....	118
5.5.2 SystemVerilog .....	120
5.5.3 Number Representation .....	120
5.5.4 Integers .....	121
5.5.5 Fractional Numbers .....	122
5.5.6 Arithmetic Operations .....	124
5.5.7 Relational Operations .....	127
5.5.8 $Q$ Format Representation in SystemVerilog .....	128
5.5.9 Fixed-Point Simulation .....	128
5.5.10 Finite State Machines .....	130
5.5.11 State Encoding .....	132
5.5.12 Resets .....	133
5.6 Design Verification .....	134
5.6.1 Variable Assignments .....	135
5.6.2 Arithmetic Overflows and Underflows .....	135
5.6.3 Latches .....	136
5.6.4 Interconnections .....	137
5.7 Synthesis .....	138
5.7.1 Top-Down Approach .....	139
5.7.2 Bottom-Up Approach .....	140
5.7.3 Mixed-Mode Approach .....	141
5.8 Energy-Efficient MIMO Detection .....	141
5.8.1 Power Consumption .....	141
5.8.2 Throughput .....	142
5.8.3 Adaptive MIMO Detection .....	143
5.9 Conclusion .....	143
References .....	144
<b>Conclusion of the Book .....</b>	<b>147</b>
<b>Index .....</b>	<b>149</b>

## About the Authors

**Ibrahim A. Bello** obtained his bachelor's degree in electrical and electronics engineering from Abubakar Tafawa Balewa University, Nigeria, in 2006, MSc (with distinction) in microelectronic systems and telecommunications from the University of Liverpool in 2010, and PhD in electronics and electrical engineering from the University of Southampton in 2017. In 2019, he joined the Department of Engineering Science at the University of Oxford, where he conducted research on an EPSRC-funded project investigating smart energy systems. Dr Bello was previously at Ahmadu Bello University, lecturing on various courses in software development and digital systems design. In 2022, he joined Dyson Technology Ltd. as a senior embedded software engineer. He retains a keen research interest in wireless communications, hardware implementation of digital signal processing algorithms, and the Internet of Things.

**Basel Halak** is an associate professor and the director of the embedded systems and IoT program at the University of Southampton. Dr Halak is a visiting scholar at the Technical University of Kaiserslautern, an industrial fellow of the Royal Academy of Engineering, a senior fellow of the Higher Education Academy, and a national teaching fellow of Advance HE UK. He has written more than 100 refereed conference and journal papers and authored 4 books. His research expertise includes hardware security, digital design, and embedded systems. Dr Halak serves on several technical programme committees such as HOST, IEEE DATE, DAC, IVSW, ICCCA, ICCCS, MTV, and EWME. He is an associate editor of *IEEE Access* and an editor of the *IET Circuits, Devices and System* journal. He is also a member of the hardware security working group of the World Wide Web Consortium (W3C).

# Acronyms

API	Application Programming Interface
ASM	Algorithmic State Machine
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CMOS	Complementary Metal Oxide Semiconductor
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
Gbps	Gigabits per Second
HDL	Hardware Description Language
HLS	High Level Synthesis
IDE	Integrated Development Environment
LSB	Least Significant Bit
LTE	Long-Term Evolution
Mbps	Megabits per Second
MIMO	Multiple-Input Multiple-Output
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MSB	Most Significant Bit
PED	Partial Euclidean Distance
QAM	Quadrature Amplitude Modulation
RTL	Register Transfer Level
SD	Sphere Decoder
SINR	Signal-to-Interference-Plus-Noise Ratio
SNR	Signal-to-Noise Ratio
STBC	Space Time Block Code
V-BLAST	Vertical-Bell Laboratories Layered Space-Time
VCD	Value Change Dump
VLSI	Very-Large-Scale Integration
ZF	Zero-Forcing

# List of Figures

Fig. 1.1	Multiple-input multiple-output communication system .....	3
Fig. 1.2	Constellation diagram for 16-QAM.....	4
Fig. 1.3	Classification of MIMO detection algorithms .....	6
Fig. 1.4	Design tradeoffs .....	8
Fig. 2.1	Block diagram of conventional and LR-aided detection. <b>(a)</b> Conventional MIMO system model. <b>(b)</b> Lattice-Reduced MIMO system model .....	21
Fig. 2.2	Zero-forcing-based preprocessing .....	25
Fig. 2.3	BER versus SNR simulation results for 16-QAM. <b>(a)</b> $2 \times 2$ . <b>(b)</b> $4 \times 4$ .....	26
Fig. 2.4	BER versus SNR simulation results for 64-QAM. <b>(a)</b> $2 \times 2$ . <b>(b)</b> $4 \times 4$ .....	27
Fig. 2.5	Illustration of Givens rotation .....	35
Fig. 2.6	CORDIC datapath illustrating vectoring mode of operation .....	41
Fig. 3.1	Modified lattice search using the SD algorithm .....	46
Fig. 3.2	Depth-first tree search for a MIMO system with $N_T = 3$ using BPSK .....	48
Fig. 3.3	Tabular enumeration for 16-QAM .....	51
Fig. 3.4	Schnorr–Euchner enumeration for 16-QAM using the real channel model.....	52
Fig. 3.5	Block diagram of iterative MIMO detection .....	57
Fig. 3.6	BER versus SNR simulation results for the sphere decoder for a 64-QAM $4 \times 4$ MIMO system using different word lengths .....	63
Fig. 3.7	Block diagram for $4 \times 4$ sphere decoder .....	64
Fig. 3.8	ASM chart for the sphere decoder .....	66
Fig. 3.9	Reduced-complexity SE enumeration computation unit for 64-QAM.....	67
Fig. 4.1	Breadth-first tree search for a MIMO system with $N_T = 3$ using BPSK .....	77

Fig. 4.2	Fixed-complexity sphere decoder tree search for $N_T = 4$ using 16-QAM .....	79
Fig. 4.3	The distributed sort algorithm for 4-QAM and $K = 2$ .....	83
Fig. 4.4	Merge network for 64 candidates .....	84
Fig. 4.5	Illustration of Batcher's merge algorithms. (a) 8-input odd-even merge. (b) 8-input bitonic merge .....	85
Fig. 4.6	Implementation of relaxed sort algorithm .....	86
Fig. 4.7	BER versus SNR simulation results for different tree search detection algorithms .....	88
Fig. 4.8	$K$ -best detector architectures. (a) Single stage. (b) Multi-stage .....	89
Fig. 4.9	BER versus SNR simulation results for the $K$ -best detector for 64-QAM $4 \times 4$ using $K = 16$ and different values of $\lambda$ .....	92
Fig. 4.10	Overall architecture of the $K$ -best detector for a 64-QAM $4 \times 4$ system .....	93
Fig. 4.11	Processing element for the $K$ -best detector at a given level .....	93
Fig. 4.12	Merge unit using odd-even merge sort for $K = 16$ and $\lambda = 4$ .....	95
Fig. 4.13	Optimised implementation of U16 merge unit. (a) Odd-even merge sort implementation. (b) Bitonic merge sort implementation .....	96
Fig. 4.14	ASM chart for the fully pipelined $K$ -best detector for $4 \times 4$ MIMO .....	98
Fig. 4.15	Controller input and output signals for the $K$ -best detector .....	99
Fig. 4.16	Pipeline schedule for the $K$ -best detector .....	100
Fig. 4.17	Data movement of input signals through the $K$ -best pipeline for $N_T = 2$ .....	102
Fig. 4.18	BER versus SNR plot for the $K$ -best detector for 64-QAM $4 \times 4$ MIMO for different list sizes using $K = 16$ .....	104
Fig. 4.19	Pipeline schedule for the $K$ -best detector based on tree levels .....	106
Fig. 5.1	Simplified ASIC implementation flowchart .....	112
Fig. 5.2	Flowchart for adding two fixed-point numbers using $Q$ format .....	126
Fig. 5.3	Fixed-point simulation methodology .....	129
Fig. 5.4	Design synthesis strategies. (a) Top-down design synthesis. (b) Bottom-up design synthesis .....	140

# List of Tables

Table 1.1	Solution space size versus number of antennas in maximum likelihood detection.....	6
Table 2.1	Numerical complexity of linear detectors versus the ML detector.....	16
Table 2.2	Comparison of FLOPs for modified Gram–Schmidt (MGS), Householder Transformations (HH), and Givens Rotation (GR) QR decomposition techniques .....	36
Table 2.3	Step-by-step triangularisation of a matrix using Givens rotation .....	39
Table 3.1	Fixed-point $Q$ format for sphere decoder variables .....	63
Table 3.2	Computation of interference terms for $N_T = 3$ .....	65
Table 3.3	Tabular enumeration for 64-QAM .....	67
Table 3.4	Throughput of the sphere decoder at different SNRs using different runtime constraints .....	69
Table 4.1	Comparison of the $K$ -best algorithm with the sphere decoder .....	79
Table 4.2	Number of nodes expanded for the different tree search detectors for $N_T = 4$ and 64-QAM using a real channel model .....	87
Table 4.3	Computation of $r_{i,i} s_i$ for 64-QAM .....	94
Table 4.4	Comparison of hardware implementations results for the bubble sort (BS), multi-cycle merge (MC), relaxed sort (RS), hybrid merge (HM), and pipelined hybrid merge (PM).....	97
Table 4.5	VLSI implementation results of different $K$ -best architectures .....	107
Table 5.1	Three-bit binary numbers showing equivalent unsigned decimal (DC), one’s complement (OC), two’s complement (TC), and signed magnitude .....	122
Table 5.2	Adding two numbers with different $Q$ representations .....	127

# Chapter 1

## Introduction



### 1.1 Chapter Overview

The chapter is organised as follows. In Sect. 1.2, MIMO technology is introduced. In Sect. 1.3, we will discuss the basic setup of a multiple-antenna wireless system. In Sect. 1.4, we will discuss MIMO detection generally and present an overview of prominent MIMO detection algorithms. In Sect. 1.5, we will present the design flow for the algorithm-to-hardware conversion, which will be used throughout the book. In Sect. 1.6, we highlight a number of design tradeoffs to be considered when implementing the MIMO detector in hardware. The overview of the book and scope of the work is presented in Sect. 1.7. Finally, the chapter is concluded in Sect. 1.8.

### 1.2 Introduction

One of the main bottlenecks of the capacity of a wireless communication system is the channel bandwidth. The relationship between the channel capacity and the bandwidth is captured in the famous Shannon–Hartley channel equation as follows [1]:

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

where  $C$  represents the channel capacity in bits per second,  $B$  denotes the bandwidth in Hertz, and  $\frac{S}{N}$  is the signal-to-noise (SNR) ratio, which is a dimensionless ratio of the signal-to-noise power. It is clear that if the channel bandwidth can be increased, then the channel capacity, or the maximum data rate at which a wireless communication could occur at an arbitrarily low error rate, could be increased as well. Since bandwidth is both limited and expensive, there is a clear motivation to