Hardware Implementation of Baseband Processing for Massive MIMO

Prabhu, Hemanth

2017

Document Version:
Publisher's PDF, also known as Version of record

Hardware Implementation of Baseband Processing for Massive MIMO

Hemanth Prabhu

LUND UNIVERSITY

Doctoral Thesis
Electrical Engineering
Lund, March 2017

Hemanth Prabhu
Department of Electrical and Information Technology
Electrical Engineering
Lund University
P.O. Box 118, 221 00 Lund, Sweden

Series of licentiate and doctoral theses
ISSN 1654-790X; No. 100
ISBN 978-91-7753-194-4 (print)
ISBN 978-91-7753-195-1 (pdf)

Typeset in Palatino and Helvetica using LaTeX 2ε.
Printed in Sweden by Tryckeriet i E-huset, Lund University, Lund.

Abstract

In the near future, the number of connected mobile devices and data-rates are expected to dramatically increase. Demands exceed the capability of the currently deployed (4G) wireless communication systems. Development of 5G systems is aiming for higher data-rates, better coverage, backward compatibility, and conforming with "green communication" to lower energy consumption. Massive Multiple-Input Multiple-Output (MIMO) is a technology with the potential to fulfill these requirements. In massive MIMO systems, base stations are equipped with a very large number of antennas compared to 4G systems, serving a relatively low number of users simultaneously in the same frequency and time resource. Exploiting the high spatial degrees-of-freedom allows for aggressive spatial multiplexing, resulting in high data-rates without increasing the spectrum. More importantly, achieving high array gains and eliminating inter-user interference results in simpler mobile terminals.

These advantages of massive MIMO requires handling a large number of antennas efficiently, by performing baseband signal processing. Compared to small-scale MIMO base stations, the processing can be much more computationally intensive, in particular considering the large dimensions of the matrices. In addition to computational complexity, meeting latency requirements is also crucial. Another aspect is the power consumption of the baseband processing. Typically, major contributors of power consumption are power amplifiers and analog components, however, in massive MIMO, the transmit power at each antenna can be lowered drastically (by the square of the number of antennas). Thus, the power consumption from the baseband processing becomes more significant in relation to other contributions. This puts forward the main challenge tackled in this thesis, *i.e.*, how to implement low latency baseband signal processing modules with high hardware and energy efficiency.

The focus of this thesis has been on co-optimization of algorithms and hardware implementations, to meet the aforementioned challenges/requirements. Algorithm optimization is performed to lower computational complexity, e.g., large scale matrix operations, and also on the system-level to relax constraints on analog/RF components to lower cost and improve efficiency. These optimizations were evaluated by taking into consideration the hardware cost and device level parameters. To this end, a massive MIMO central baseband pre-coding/detection chip was fabricated in 28 nm FD-SOI CMOS technology and measured. The algorithm and hardware co-optimization resulted in the highest reported pre-coding area and energy efficiency of 34.1 QRD/s/gate and 6.56 nJ/QRD, respectively. For detection, compared to small scale MIMO systems, massive MIMO with linear schemes provided superior performance, with area and energy efficiency of 2.02 Mb/s/kGE and 60 pJ/b.

The array and spatial multiplexing gains in massive MIMO, combined with high hardware efficiency and schemes to lower constraints on RF/analog components, makes it extremely promising for future deployments.

Contents

Preface

This thesis summarizes my academic work carried out from August-2011 to February-2017 in the Digital ASIC group, at the department of Electrical and Information Technology, Lund University, Sweden. The main contributions are derived from the following articles sorted by publication date:

- H. Prabhu, J. Rodrigues, L. Liu, O. Edfors "A 60 pJ/b 300 Mb/s 128 × 8 Massive MIMO Precoder-Detector in 28 nm FD-SOI", International Solid-State Circuits Conference (ISSCC), San Francisco, 2017.

  Contribution This research work has been performed by first author under the guidance of the remaining authors. The first author has fabricated and measured a digital baseband chip for massive MIMO as part of the work.

- H. Prabhu, J. Rodrigues, L. Liu, O. Edfors "Algorithm and Hardware Aspects of Pre-coding in Massive MIMO Systems", Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California, 2015.

  Contribution This work serves an overview of pre-coding in massive MIMO, and also includes effects of hardware imparity.

- H. Prabhu, F. Rusek, J. Rodrigues, O. Edfors "High Throughput Constant Envelope Pre-coder for Massive MIMO Systems", IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, Portugal, 2015.

  Contribution The author has developed hardware architecture for constant envelope pre-coding along with synthesis results in ST-28 nm FD-SOI technology.

- C. Zhang, H. Prabhu, Y. Liu, L. Liu, O. Edfors, V. Öwall "Energy Efficient Group-Sort QRD Processor with On-line Update for MIMO Channel Pre-processing", IEEE Transactions on Circuits and Systems Part 1: Regular Papers, Vol. 62, No. 5, pp. 1220-1229, 2015.

  Contribution This work is an extension of the ISCAS-14 paper, which was invited for TCAS-I.

- Y. Liu, H. Prabhu, L. Liu, V. Öwall "Adaptive Resource Scheduling for Energy Efficient QRD Processor with DVFS", IEEE Workshop on Signal Processing Systems, Hangzhou, China, 2015.

  Contribution The author has supported the development of an improved scheduler for the channel tracking pre-processing algorithm.

- H. Prabhu, O. Edfors, J. Rodrigues, L. Liu, F. Rusek "A low complex peak-to-average power reduction scheme for OFDM based massive MIMO systems", International Symposium on Communications, Control and Signal Processing, Athens, Greece, 2014.

  Contribution This research work has been performed by the first author under the guidance of the remaining authors. A novel low complexity PAR aware pre-coding based on antenna reservation was developed.

- H. Prabhu, O. Edfors, J. Rodrigues, L. Liu, F. Rusek "Hardware Efficient Approximative Matrix Inversion for Linear Pre-Coding in Massive MIMO", IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, Australia, 2014.

  Contribution The first author has designed a hardware architecture for approximative Neumann series based pre-coding in 65 nm technology.

- C. Zhang, H. Prabhu, L. Liu, O. Edfors, V. Öwall "Energy Efficient SQRD Processor for LTE-A using a Group-sort Update Scheme", IEEE International Symposium on Circuits and Systems (ISCAS), 2014, Melbourne, Australia, 2014.

  Contribution: The author has developed algorithms and performed system level simulations. Specifically, a novel group-sorting mechanism was developed to perform channel tracking. This paper was awarded the best paper at ISCAS, 2014.

- H. Prabhu, J. Rodrigues, O. Edfors, F. Rusek "Approximative Matrix Inverse Computations for Very-large MIMO and Applications to Linear Pre-coding Systems", Wireless Communications and Networking Conference (WCNC), Shanghai, China, 2013.

Contribution The first author has developed with guidance from the remaining authors, a low complexity Neumann series based matrix inversion algorithm.

- C. Zhang, H. Prabhu, L. Liu, O. Edfors, V. Öwall "Energy Efficient MIMO Channel Pre-processor Using a Low Complexity On-Line Update Scheme", Norchip, Copenhagen, Denmark, 2012.

  Contribution The author developed a channel tracking algorithm and implemented an accelerator in 65 nm CMOS technology.

Furthermore, I have contributed in the following documents:

- H. Prabhu, J. Rodrigues, O. Edfors "MMSE linear pre-coding for very-large MIMO systems using rank-1 update", Swedish System-On-Chip Conference.
- H. Prabhu, J. Rodrigues, O. Edfors "A low-power flexible mixed radix-5/3/2 FFT for OFDM systems", Swedish System-On-Chip Conference.
- MAMMOET Technical Report, "D3.2 - Distributed and centralized baseband processing algorithms, architectures, and platforms", `https://mammoet-project.eu/` (visited on 2 Nov. 2016).
- MAMMOET Technical Report, "D3.3 - Hardware aware signal processing for MaMi", `https://mammoet-project.eu/`.

Acknowledgments

Undertaking this five-year long PhD journey has been a truly life-changing experience for me. It has been challenging, exciting, and more importantly joyfull, which could not have been possible without many people's help and support. First and foremost, my sincere gratitude and thanks to my supervisor Prof. Ove Edfors, for all the guidance and discussions during my PhD studies. Especially appreciate your patience, considering my mediocre skills in wireless communication and academic writing. In addition to research, it has overall been fun, and I will never forget the fast walking dumpling and Ouzo hunting trips in Shanghai and Athens, respectively.

I am indebted to my co-supervisor Joachim Rodrigues, who has guided and encouraged me since I was a master student. More importantly, convincing me that I could come this far when I myself was not confident. I would also thank Liang Liu and Fredrik Rusek for amazing insights, discussions and help throughout my doctoral studies. Special thanks to Rektor Viktor Öwall, who has been an inspiration, and over the years provided sharp feedbacks and suggestions. It has been fruitful academically, especially persuading me to write ISSCC paper. Also, thanks for hosting amazing summer bbq parties and some great trips, including the now legendary 'Wednesday@Vesuvio' in San Francisco with Ove.

I would like to thank all my colleagues in Digital ASIC group and EIT, for all the hard work, help, and joyful moments. I would thank Oskar and Rakesh for 'immense support during chip fabrication'; thank Chenxin for 'best paper in ISCAS'; Isael, Johan, Reza, Deepak, Yaseer, for 'hours of so-called scientific discussions'; Babak for 'some amazing travel in Australia and Thailand, lets not get into details'; Steffen for 'monitoring my office hours with German efficiency'; Yang and Xuhong for 'Xiaomi gadgets and tea supply'; Farrokh, Breeta and Dimitar for 'exploring restaurants on weekends', Eric Larsson,

Fredrik Tufvesson, Pietro Andreani, Mojtaba, Siyu, Xiaodong, Mohammed, Waqas, Ahmed, Muris, Nafiseh, Joao, Xiang, Rohit, Saeedeh, Hu Sha, and those I have forgotten to include, for all the times we shared during the studies. Also, would like to thank the administrative and technical staff, especially Anne Andersson, Pia Bruhn, Bertil Lindwall, Josef Wajnblom, Stefan Molund and Erik Johnsson, for all the help and support.

I would like to thank Ivo Bolsens and Kees Vissers for hosting me during my internship at Xilinx Inc. I would also thank my colleagues and friends I got to know there, Kristof Denolf, Jack Lo, Samuel Bayliss, Sneha Date, Naveen, Lisa and Ehsan for a short but joyful experience in the valley.

Last but not the least, I would like to express my utmost gratitude to my family, friends back home and in Sweden. Thank you Amma, Baba and Chinma for your unconditional support and sacrifices over the years, it has made me what I am today.

Hemanth Prabhu
Lund, March 2017

Acronyms and Mathematical Notations

ASIC	Application-Specific Integrated Circuit
ASIP	Application-Specific Instruction-set Processor
BER	Bit Error Rate
BIST	Built-In Self-Test
BS	Base Station
BSU	Backward Substitution Unit
CAGR	Compounded Annual Growth-Rate
CCDF	Complementary Cumulative Distribution Function
CD	Cholesky Decomposition
CE	Constant Envelope
CORDIC	COordinate Rotation DIgital Computer
CP	Cyclic Prefix
CSI	Channel State Information
DAC	Digital-to-Analog Converter
DMI	Direct Matrix Inversion
DPC	Dirty Paper Coding
DSP	Digital Signal Processor
DUT	Design Under Test
DVFS	Dynamic Voltage Frequency Scaling
ED	Euclidean Distance
FBB	Forward Body Biasing
FD-SOI	Fully-Depleted Silicon-on-Insulator

FER Frame Error Rate
FFT Fast Fourier Transform
FIFO First-In First-Out
FLOP Floating Point Operations
FPGA Field-Programmable Gate Array

GPU Graphics Processing Unit

HDL Hardware Description Language
HLS High-Level Synthesis

i.i.d. Independent and Identically Distributed
IC Integrated Circuit
ICI Inter-Carrier Interference
ICT Information and Communication Technology
IDFT Inverse Discrete Fourier Transform
IO Input Output
IoE Internet of Everything
ISI Inter-Symbol Interference

JTAG Joint Test Action Group

LOS Line-of-Sight
LTE Long Term Evolution
LTE-A LTE-Advanced
LUT Look-Up Table

MAC Multiply-Accumulate
MF Matched Filter
MGS Modified Gram-Schmidt
MIMO Multiple-Input Multiple-Output
ML Maximum Likelihood
MMSE Minimum Mean Square Error
MRT Maximum-Ratio Transmission
MS Mobile Station
MSE Mean Square Error

NHE Normalized Hardware Efficiency
NS Neumann Series

OBO Output Back-Off
OFDM Orthogonal Frequency-Division Multiplexing

PA	Power Amplifier
PAR	Peak-to-Average Ratio
PE	Processing Element
QRD	QR Decomposition
RB	Resource Block
RBB	Reverse Body Biasing
RF	Radio-Frequency
RNG	Random Number Generator
RTL	Register-Transfer Level
SD	Sphere Decoder
SER	Symbol Error Rate
SINR	Signal-to-Interference and Noise Ratio
SISO	Single-Input Single-Output
SNDR	Signal-to-Noise and Distortion Ratio
SNR	Signal-to-Noise Ratio
SQNR	Signal-to-Quantization-Noise Ratio
TAP	Test Access Port
TLM	Transaction-Level Modeling
VLIW	Very Large Instruction Word
VPU	Vector Projection Unit
ZF	Zero-Forcing

$(\cdot)^*$	Complex conjugate
$(\cdot)^T$	Vector/matrix transpose
$(\cdot)^H$	Hermitian transpose
$(\cdot)^{-1}$	Matrix inverse
$(\cdot)^\dagger$	Matrix pseudo-inverse
$(\cdot)_i$	Column vector
$(\cdot)_{i,j}$	$(i,j)^{th}$ matrix element
$\lvert\cdot\rvert$	Euclidean vector length
$\lVert\cdot\rVert_2$	ℓ^2-norm
$\propto$	Proportional
∞	Infinity
$\approx$	Approximation
$\mathcal{O}$	Order of computational complexity
$x \in \mathbb{S}$	The element x belongs to the set $\mathbb{S}$
$det(\boldsymbol{A})$	Determinant of $\boldsymbol{A}$

1

Introduction

This thesis presents a multidisciplinary study of wireless communication and digital hardware design. More specifically, the study is on co-optimized Integrated Circuit (IC) implementations of baseband algorithms for future wireless communication systems. Development of such systems is crucial to meet the communication demands of envisioned billions of devices widely marketed as Internet of Everything (IoE) [1], [2]. On the other hand, the limited availability and the highly inflated price of spectral resources conflict with the ever-increasing data-rate demands [3], [4]. Subsequently, tackling this conflict requires efficient spectrum usage, which calls for more advanced communication schemes and algorithms.

In general, improving spectral efficiency by advanced schemes, in turn, comes with an increase in signal processing complexity. Furthermore, the overall latency of these systems are expected to reduce by a factor of 10x for 5G [5]. Also, future deployment strategies like femtocell Base Stations (BSs) are expected to be produced in large volumes, which makes lowering hardware cost very desirable. This brings forth one of the main challenges of the thesis, *i.e.*, to achieve a low latency and high-performance signal processor, while still keeping the hardware cost and power consumption low.

Reduction in power consumption has become ever-so-important considering the environmental impact cellular BSs have around the world [6]. Traditionally, transmit signal amplifiers contributed to the bulk of the BS power consumption. However, this changes drastically in case of femtocells, where the baseband processing contributes to a significant portion of the total power [7]. In the case of battery-powered Mobile Stations (MSs), performing efficient processing is even more important due to limited energy budget. Therefore, lowering the power consumption of the baseband signal processing becomes a crucial aspect and challenge, for both BSs and MSs.

In this thesis, the main focus is on massive Multiple-Input Multiple-Output (MIMO) baseband processing on the BS side. Massive MIMO is a key candidate for 5G, offering very high spectral and transmit power efficiency. In massive MIMO systems, the BSs are equipped with a very large number of antennas (in 100s), serving a relatively low number of users (say in 10s), simultaneously in the same frequency and time resource. To reach the full potential of massive MIMO, it is critical to perform efficient processing to utilize all the antennas at the BS. This inherently results in performing operations on large matrices, however, the challenge is to meet the previously mentioned requirements on latency, power and hardware cost.

This thesis addresses these challenges by co-optimization at both algorithm and hardware level. At algorithmic level various techniques were employed, e.g., complexity vs performance trade-offs, schemes to relax power amplifier requirements, and exploitation of special computational characteristics occurring in systems. Hardware optimizations were performed on architecture and device to achieve high area and energy efficiency.

1.1. SCOPE OF THE THESIS

The goal of the research topic is to explore efficient hardware architectures for next generation massive MIMO systems. The focus has been on baseband processing both for the BS and the MS, considering the high performance requirements with constraints on hardware cost and power consumption.

The central part of the thesis is addressing the following questions:

- What is the area and energy cost of performing massive MIMO baseband processing?
- Are there special system level characteristics which can be exploited to lower the computational complexity, leading to efficient hardware?
- Is it possible to exploit processing and modulation schemes to relax constraints while designing low cost and power efficient massive MIMO systems?

Exploration of these questions, invariably requires a hardware oriented study. Throughout the work an algorithm-hardware co-design approach is adopted with special attention to the following:

- Improving hardware efficiency, e.g., by extensive hardware re-usage and time multiplexed architectures.
- Lowering design/verification complexity by focusing on simplification of data-flow and control logic.

- Utilizing various low power techniques and exploiting modern CMOS technology nodes.

- Flexible and parameterized implementation using high-level synthesis languages such as SystemC/C++.

Digital baseband processing is a wide topic which includes many modules such as digital front-end, Orthogonal Frequency-Division Multiplexing (OFDM) modulation, channel estimation, MIMO processing, forward error-correction, and interleaving. Among these, the thesis mainly focuses on MIMO processing for uplink detection and downlink pre-coding for large-scale (massive) MIMO BSs. At the mobile terminal side the focus has been on MIMO channel pre-processing.

1.2. OUTLINE AND CONTRIBUTION

This work is divided into four parts, the first three chapters constitute the background, to give an overview of the research field. Chapter 2 provides an introduction to the field of wireless communication, followed by details of MIMO-OFDM techniques. Later the promising large-scale or massive MIMO technology is presented. Chapter 3 provides an overview of hardware platforms, followed by a discussion of possible trade-offs and challenges involved during implementation.

Part I presents aspects of downlink pre-coding for large-scale MIMO, including implementation on different hardware platforms followed by measurement results. Other alternative downlink pre-coding strategies to tackle Peak-to-Average Ratio (PAR) and IQ imbalance are also implemented and discussed.

Part II presents uplink detection for large-scale MIMO. The two chapters in this part describe detection techniques followed by a massive MIMO adaptive framework supporting both linear and non-linear detection.

Part III focuses on Long Term Evolution (LTE) mobile terminals, targeting low complex channel pre-processing techniques. This is followed by a chapter with conclusions and outlook for future work.

PART I: DOWNLINK PROCESSING FOR MASSIVE MIMO

Conventional small-scale MIMO systems have been extensively analyzed and are well established. These systems are also backed by plethora of implementations performing channel estimation, pre-processing, detection, etc., [8], [9]. However, naive scaling of these designs for a large scale MIMO system will

lead to inefficient and expensive hardware implementations. The contributions of this thesis (hereafter referred to as this work) are to propose low complexity algorithms to handle the inherent large matrices in massive MIMO. The algorithms are tuned for different hardware platforms, and measurements were performed as a proof-of-concept.

This work also includes novel pre-coding schemes relaxing constraints on the analog circuits, which is in-line with the vision of employing 100's of low-cost Radio-Frequency (RF)-chains in massive MIMO BSs [10]. Furthermore, analysis of IQ imbalance during downlink transmission is also presented.

The contents of part-I are based on the following publications:

- H. Prabhu, J. Rodrigues, O. Edfors, F. Rusek "Approximative Matrix Inverse Computations for Very-large MIMO and Applications to Linear Pre-coding Systems", IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, 2013.
- H. Prabhu, O. Edfors, J. Rodrigues, L. Liu, F. Rusek "Hardware Efficient Approximative Matrix Inversion for Linear Pre-Coding in Massive MIMO", IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, Australia, 2014.
- H. Prabhu, O. Edfors, J. Rodrigues, L. Liu, F. Rusek "A low-complex peak-to-average power reduction scheme for OFDM based massive MIMO systems", International Symposium on Communications, Control and Signal Processing (ISCCSP), Athens, Greece, 2014.
- H. Prabhu, F. Rusek, J. Rodrigues, O. Edfors "High Throughput Constant Envelope Pre-coder for Massive MIMO Systems", IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, Portugal, 2015.

PART II: UPLINK PROCESSING FOR MASSIVE MIMO

This part of the work presents detection techniques for massive MIMO systems. The key contribution is an adaptive framework, which based on the system conditions can perform either linear detection or pre-processing for non-linear detection. The core module enabling this is a Cholesky decomposition processor, that has been fabricated and measured in 28 nm Fully-Depleted Silicon-on-Insulator (FD-SOI).

The content of part-II is based on the following publication:

- H. Prabhu, J. Rodrigues, L. Liu, O. Edfors "A 60 pJ/b 300 Mb/s 128 $\times$ 8 Massive MIMO Precoder-Detector in 28nm FD-SOI", International Solid-State Circuits Conference (ISSCC), San Francisco, 2017.

PART III: ADAPTIVE CHANNEL PROCESSING

Constraints on battery lifetime and low-cost requirements in mobile terminals have resulted in numerous low power hardware implementations for LTE systems. On the other hand, supporting flexibility for various system requirements and standards necessitates a flexible architecture. In part-II, low complex channel pre-processing algorithms are presented and evaluated on an in-house reconfigurable wireless processor [11].

The contents of part-III are based on the following publications:

- C. Zhang, H. Prabhu, Y. Liu, L. Liu, O. Edfors, V. Öwall "Energy Efficient Group-Sort QRD Processor with On-line Update for MIMO Channel Pre-processing", IEEE Transactions on Circuits and Systems Part 1: Regular Papers, Vol. 62, No. 5, pp. 1220-1229, 2015.
- C. Zhang, H. Prabhu, L. Liu, O. Edfors, V. Öwall "Energy Efficient MIMO Channel Pre-processor Using a Low Complexity On-Line Update Scheme", Norchip, Copenhagen, Denmark, 2012.

2

Wireless Communication Concepts

From smoke signals used along the Great Wall of China, to transatlantic cables used by New York traders in the 19^{th}-century [12], information transfer and retrieval has been a key component in human civilization. The "information-age" in the mid 20^{th}-century powered by the computer miniaturization, has seen an incredible growth in Information and Communication Technology (ICT) industry. The early Nordic Mobile Telephone services had voice only data-rate of 1.2 kbits/s and around 110 k subscribers in 1985 [13]. ICT industry in 2015 supports multimedia streaming and world-wide roaming (4G), with data-rate of 40 Mbits/s and around 3.2 billion users. This growth is just a beginning, the data traffic is expected to have Compounded Annual Growth-Rate (CAGR) of 45% with 25 billion IoE connected devices by 2021 [1].

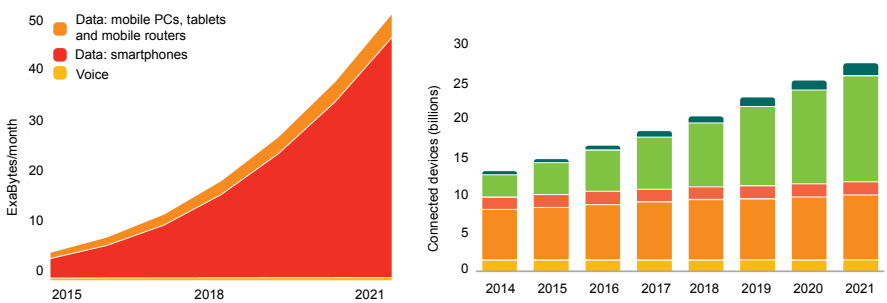


Figure 2.1. Expected total mobile traffic and number of connected devices. Devices in the bar graph are stacked in following order fixed phone, mobile phones, laptop/tablet, non-cellular IOT and cellular IOT. Data/figure from Ericsson mobility report [1].

Sustaining this growth requires much higher data-rates, coverage, and reliability than the current 4G standard can deliver. Massive MIMO is a technology that has the potential to fulfill these requirements and visions. Before getting into the details of massive MIMO, a brief introduction of a wireless communication system is presented.

2.1. WIRELESS SYSTEM

An image of a wireless system is depicted in Fig. 2.2. The transmitter includes digital processing and the RF-chain for transmission of a modulated signal. Channel is a medium in which information transfer between the transmitter and receiver occurs. The electromagnetic signal leaves the transmission antenna, propagates through the medium, and is picked up by the receiver. The received electromagnetic waves experience a reduction in power or attenuation by various factors, e.g., the expansion of radio wavefront in free space taking the shape of a growing sphere. Furthermore, the signal can be scat-

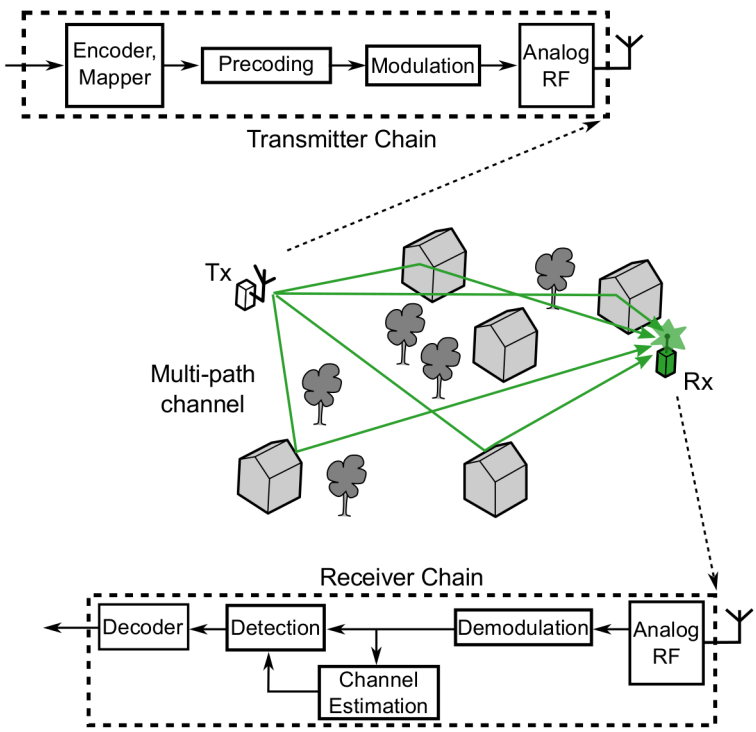


Figure 2.2. A simplified wireless system in a propagation environment.

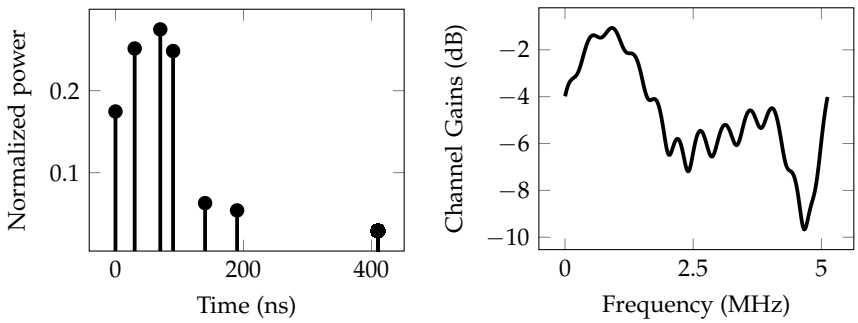


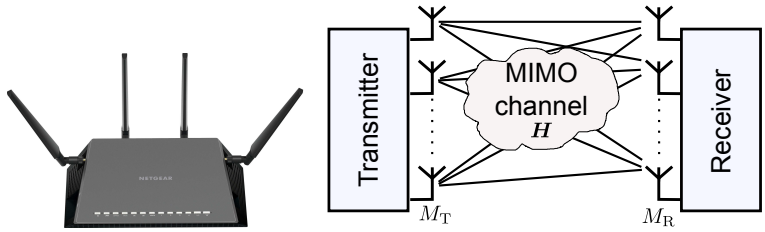
Figure 2.3. Multi-path channel impulse and frequency response.

tered and diffracted by different objects in the paths between the transmitter and receiver. This would result in several clones of the signal with different interactions and paths reaching the receiver at different times. This, in general, is referred to as a multi-path channel [14].

The different arrival times due to scattering along with the corresponding attenuation and phase results in the channel being frequency selective. Intuitively this can be visualized as a filter (channel taps), and the corresponding frequency equivalent of the impulse response is the channel frequency response, as shown in Fig. 2.3. Another effect not illustrated here is the changes in the propagation environment, by movement of receiver, transmitter and/or scatterers. In addition to changing the channel tap, it also effects the perceived frequency (Doppler spread) of the transmitted signal. These changes depend on the speed e.g., a receiver with walking speed or in a moving car will have different channel variations and Doppler frequency shifts. These propagation effects make wireless communication a very difficult and challenging task. In the next section key technologies enabling modern wireless standards are described.

2.1.1. ORTHOGONAL FREQUENCY-DIVISION MULTIPLEXING (OFDM)

In general, increasing the bandwidth of the communication system can improve the data-rates. Basically more information can be transmitted in any time interval, however, a higher bandwidth increases the complexity of the system as well. The main reason for this is that the channel becomes increasingly frequency selective. One approach of reducing frequency selectivity is to multiplex data over frequency. The key idea in frequency division multiplexing (FDM) is to divide bandwidth into a series of non-overlapping sub-bands (or sub-carriers), which carry different signals with minimal interference to each other. Orthogonal Frequency-Division Multiplexing (OFDM) is one such



(a) Wireless access point. (b) MIMO system model.

(Source Netgear Inc., www.netgear.com)

Figure 2.4. MIMO technology in every day life.

technique, which is similar to FDM with a constraint that sub-carriers are chosen to be orthogonal to each other. The orthogonality under ideal conditions results in zero interference between sub-carriers and also does not require inter sub-carrier guard bands. Hence, allowing OFDM system to achieve high spectral usage. Apart from this, the orthogonality allows for an efficient modulation/demodulation implementation by using Fast Fourier Transform (FFT)/IFFT at receiver/transmitter.

The propagation effects are still present in OFDM and cause Inter-Symbol Interference (ISI) and Inter-Carrier Interference (ICI). Introducing a Cyclic Prefix (CP) with a length greater than the delay spread of the channel can reduce the impact of the effects of both ISI and ICI [15]. However, in practice, there are several hardware imperfections like IQ-imbalance, PAR issues, carrier frequency and timing offsets, which effects OFDM systems [16]. Details on tackling some of the mentioned imperfections will be presented in later parts of the work.

2.1.2. MULTIPLE-INPUT MULTIPLE-OUTPUT (MIMO)

MIMO is a physical layer technique that allows more data to be transferred within the same available bandwidth [17]. The technique has been incorporated into various wireless standards like IEEE 802.11n, LTE-Advanced, WiMAX, etc., [18], [19]. Products as shown in Fig. 2.4(a) are quite common nowadays. These MIMO systems allow multiple data streams in different spatial paths simultaneously to improve data-rates compared to a Single-Input Single-Output (SISO) system.

Transmitting multiple streams simultaneously through wireless channels would result in mixing of the signals at the receivers. Therefore, additional signal processing needs to be performed, at either receiver or transmitter or often at both, to separate the data streams, which is generally termed as MIMO processing.

MIMO SYSTEM MODEL Fig. 2.4(b) shows a MIMO system with M_{T} transmit and M_{R} receive antennas. We consider a narrow band channel to simplify our analysis. Denoting the $M_{\mathrm{R}} \times M_{\mathrm{T}}$ channel transfer matrix by $\boldsymbol{H}$, the input-output relation for the MIMO channel is given by

$$\boldsymbol{y} = \sqrt{P}\boldsymbol{H}\boldsymbol{s} + \boldsymbol{w}\,, \tag{2.1}$$

where $\boldsymbol{y}$ is the $M_{\mathrm{R}} \times 1$ received signal vector, $\boldsymbol{s}$ the $M_{\mathrm{T}} \times 1$ pre-scaled transmit signal vector, $\boldsymbol{w}$ the additive Independent and Identically Distributed (i.i.d.) noise vector, and P the total transmit power. The covariance matrix of $\boldsymbol{s}$, $\boldsymbol{R}_{\boldsymbol{ss}} = \mathbb{E}[\boldsymbol{s}\boldsymbol{s}^H]$, need to satisfy $\mathrm{Tr}(\boldsymbol{R}_{\boldsymbol{ss}}) = 1$ in order to constraint the total power transmitted.

A mobile receiver equipped with M_{R} antennas receives $\boldsymbol{y}$, and may require performing MIMO processing to separate the data-streams. The MIMO processing typically involves complex matrix operations like matrix inversion, QR Decomposition (QRD), Cholesky Decomposition (CD). For battery operated mobile terminals, optimization at both algorithm and hardware are required due to the stringent energy constraints. Different techniques to tackle MIMO processing for MS are presented in the last part of this work.

MIMO CAPACITY The capacity of the MIMO system is given by

$$C = \max_{\mathrm{Tr}(\boldsymbol{R}_{\boldsymbol{ss}})=1} \log_2 \det\left(\boldsymbol{I}_{M_{\mathrm{R}}} + \frac{P}{N_o}\boldsymbol{H}\boldsymbol{R}_{\boldsymbol{ss}}\boldsymbol{H}^H\right) \text{ bps/Hz}. \tag{2.2}$$

The capacity C is also referred to as the error-free spectral efficiency, or the data-rate per unit bandwidth that can be sustained reliably over the MIMO link. For a given bandwidth W Hz, the maximum achievable data-rate over this bandwidth using the MIMO channel is to scale C with W.

Considering $\boldsymbol{s}$ to be non-preferential, *i.e.*, $\boldsymbol{R}_{\boldsymbol{ss}} = \boldsymbol{I}_{M_{\mathrm{T}}}/M_{\mathrm{T}}$, $\boldsymbol{H}$ being an unitary channel and $M_{\mathrm{T}} = M_{\mathrm{R}} = M$, results in a capacity

$$C = M \log_2\left(1 + \frac{P}{N_o}\right). \tag{2.3}$$

The above expression for capacity highlights the data-rate gains for MIMO systems, indicating that increasing M increases the number of parallel data streams. It is important to note that, the capacity increases linearly with the number of antennas, but only logarithmically with the power. In the current wireless standards like LTE-advanced up to $M = 8$ streams are supported. These techniques provide high data-rates in the range of 3 Gbps [18], however, future requirements far-surpass the currently supported 4G data-rates.

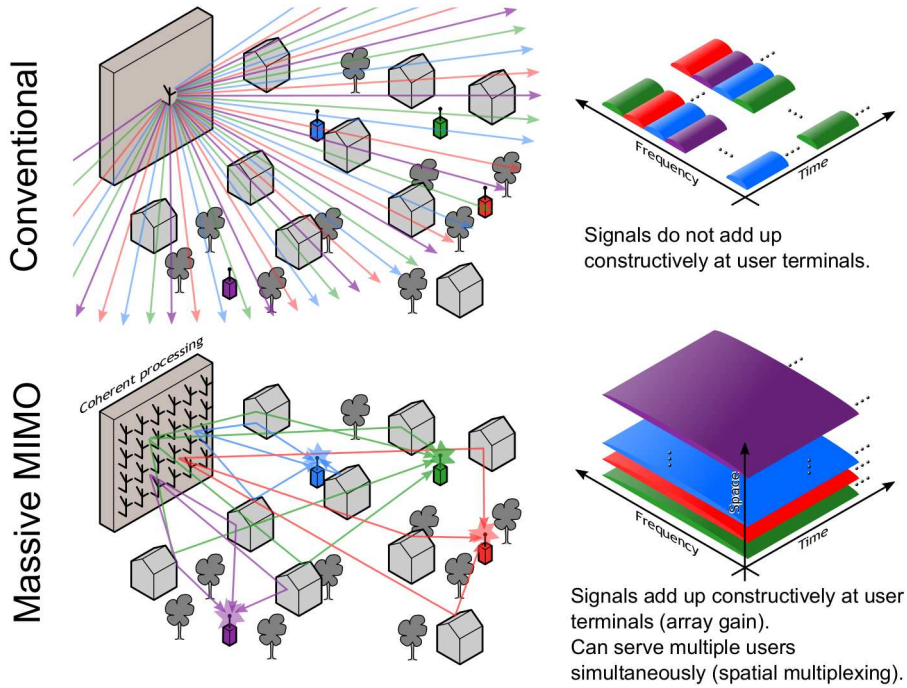


Figure 2.5. Massive MIMO provides huge data-rate improvement over conventional systems. (Artwork from Ove Edfors)

2.2. ADVENT OF MASSIVE MIMO

The high cost and lack of availability of spectrum, and the increasing demand for data-rates meant that further exploitation of spatial domain is inevitable. The article "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas", by Thomas L. Marzetta from Bell Labs, pointed out the possibility of further exploiting spatial domain. Analysis of the effects of increasing the number of antennas without limit in ideal conditions at the BS was performed, where it was shown that the effects of additive receive noise and small-scale fading disappear, as does interference among users [20].

Scaling up MIMO provides higher degrees of freedom in the spatial domain than existing wireless communication (4G) systems. Conventional MIMO systems usually have up to 8 antennas, whereas the term "massive MIMO", "large-scale MIMO" or "very-large MIMO" are systems equipped with much larger number of antennas (in 100s) at the BS. Such massive MIMO systems typically operate in a Multi-User MIMO (MU-MIMO) scenario, wherein a BS

serves many terminals in the same time-frequency resource. Fig. 2.5 illustrates the stacking of users in the same time and frequency grid, wherein each user can use the entire time-frequency resource, resulting in a high spectral efficiency.

Another important distinction is that the multi-path components are exploited in massive MIMO to improve signal strength at the receiver. This is possible due to the high spatial resolution and array gains in massive MIMO, thanks to the large number of antennas at the BS. Moreover, processing efforts are mostly performed at the BS side, while battery operated mobile terminals can have low hardware cost and power consumption. One can also think of such systems as pushing the bulk of baseband computational complexity from mobile terminals to the BS.

2.2.1. MULTI-USER MASSIVE MIMO SYSTEM MODEL

Consider a massive MIMO BS with M antennas serving K single antenna users. The downlink signal model for a narrow band system is

$$\boldsymbol{y} = \sqrt{P}\boldsymbol{H}\boldsymbol{p} + \boldsymbol{w}\,, \tag{2.4}$$

which is similar to (2.1), except for $\boldsymbol{p}$, which is a $M \times 1$ vector of pre-coded transmit signals across the antennas. The user symbols $\boldsymbol{s}$ are pre-coded before transmission and this operation is represented as

$$\boldsymbol{p} = \boldsymbol{W}\boldsymbol{s}\,, \tag{2.5}$$

where $\boldsymbol{W}$ is the pre-coding matrix with appropriate power scaling. This model is for a narrow band system, and introducing OFDM modulation will require an Inverse Discrete Fourier Transform (IDFT) after pre-coding. With OFDM, several such parallel channels will exist for each sub-carrier. The uplink signal model is similar to (2.4), with the channel matrix $\boldsymbol{H}^T$, due to the channel reciprocity. The goal of pre-coding and detection is to separate user data streams with little or no inter-user interference. Details on different schemes will be described in latter parts of the work.

Although the very-large MU-MIMO model is similar to a standard MIMO model, the increased number of BS antennas has several advantages. Things that were random before, now start to look deterministic. For example, the distribution of the singular values of the channel matrix approaches a deterministic function [21]. Another observed property is that very wide (or tall) matrices under certain conditions tend to be well conditioned.

2.2.2. MASSIVE MIMO ADVANTAGES

In this section, some of the advantages of massive MIMO systems are described in line with [22]. The benefits are analyzed by allowing the number of

antennas to grow large, which provides some interesting features and trends.

IMPROVEMENT IN SPECTRAL AND ENERGY EFFICIENCY The capacity increase results from the aggressive spatial multiplexing used in massive MIMO. The number of spatial streams depend upon the rank of the Gram matrix $\boldsymbol{H}\boldsymbol{H}^H$. Under ideal conditions, the rank of Gram matrix is equal to the number of users K. The fundamental principle that makes the dramatic increase in energy efficiency possible is that with a large number of antennas, energy can be focused into small regions in space. This is mainly due to the increase in spatial resolution and the coherent superposition of wavefronts. By shaping the signals sent out by the antennas, the BS can make sure that the wavefronts emitted by antennas add up constructively at the locations of users. Interference between users can be suppressed even further by using an appropriate pre-coding scheme.

INEXPENSIVE AND LOW POWER COMPONENTS Massive MIMO reduces the constraints on accuracy and linearity of each individual amplifier and RF chain. In a way, massive MIMO relies on the law of large numbers to make sure that noise, fading, and hardware imperfections average out when signals from a large number of antennas are combined in the channel.

With massive MIMO, expensive ultra linear (40 W) amplifiers used in conventional systems are replaced by hundreds of low-cost amplifiers with output power in the milliwatt range. Further component cost reduction is possible by tackling hardware imperfections and constraining transmission signals by exploiting the large degree-of-freedom e.g., with constant envelope pre-coding at the BS.

LINEAR PROCESSING PROVIDES CLOSE TO OPTIMAL PERFORMANCE Under favorable channel conditions, increasing M results in the diagonal elements of the Gram matrix ($\boldsymbol{H}\boldsymbol{H}^H$) becoming more dominant. This in turn, means that user channels become orthogonal. Inter-user interference vanishes, and the BS can communicate with the users simultaneously.

For practical systems with a limited number of antennas, the inter-user interference can still be removed by performing signal processing (pre-coding) at the BS. The price-to-pay would be that additional radiation power would be required to explicitly suppress interference. However, such pre-coding schemes are still linear, and in the case of massive MIMO systems, performs close to optimal non-linear pre-coding [23].

REDUCTION OF LATENCY The performance of wireless communication systems can be limited by fading, wherein the signal strength can reduce dras-

tically. This occurs in multi-path channels where signals on arrival add up destructively. Fading makes it hard to build low latency wireless links, if the MS is trapped in a fading dip, it has to wait until the propagation channel has sufficiently changed until any data can be received. Massive MIMO relies on the law of large numbers and beamforming to avoid fading dips, so that fading no longer limits latency.

2.2.3. MASSIVE MIMO CHALLENGES

The advantages of massive MIMO systems are impressive, especially asymptotic gains when M increases without bounds. However, for practical systems with a limited number of antennas at BS (still large compared to traditional systems), some of the main implementation challenges are listed below.

HARDWARE IMPERFECTIONS The number of RF-chains and analog components in massive MIMO is high. Therefore, it is important to lower the analog component costs. On the one hand, it is expected that massive MIMO systems can handle imperfections, due to the averaging effects. On the other hand, for a limited number of antennas in a practical system, there is still a need to combat these imperfections. In this work, amplifier constraints are relaxed by performing PAR aware pre-coding, and effects of IQ-imbalance are studied.

CHANNEL CONDITIONS For analysis of massive MIMO gains, a typical assumption is to consider the channel to be i.i.d. Rayleigh fading. For such channels increasing M results in the Gram matrix becoming diagonally dominant, leading to an interference free transmission. However, there can be channel conditions in practical scenarios with a strong correlation between users [24], e.g., in dense (stadium, shopping mall) scenarios, where a lot of users are physically close to each other. Furthermore, during peak traffic (rush hour) the ratio of M and K may not be high. Thus, assuming a diagonal dominance of the Gram matrix may not hold for all channel conditions. When developing a hardware architecture, the handling of different channel situations efficiently is important. In this work, one such approach is proposed, wherein detection is performed adaptively based on channel conditions. The framework allows only using as much detection complexity as necessary and therefore saves power.

BASEBAND PROCESSING COST A crucial challenge in massive MIMO is to perform signal processing operations to efficiently utilize the large antenna arrays. This inherently requires handling large matrices with dimensions depending on M and K. In addition to handling large matrices, the channel

matrix may need to be updated frequently, depending on the frequency selectivity of channel and Doppler spread. In this work, the focus is on reducing down-/up-link MIMO processing computational complexity and its implementations on different platforms.

3

Digital Hardware Design

The Nobel prize-winning invention of the integrated circuit in 1950's, revolutionized the electronics industry. No other field in the history of civilization has achieved an incredible performance of 55% Compounded Annual Growth-Rate (CAGR) for such an extended period of time. This trend of exponential growth was famously observed by Gordon E. Moore and later came to be known as "Moore's law" [25]. The growth has impacted all aspects of life and society, and enabled numerous industries from communication, banking to space-travel. To get a perspective, a single Google search query performs more computations than the whole of NASA Apollo (11 years 17 missions) program [26]. Even comparing to some recent and probably the most famous supercomputer ever, "Deep Blue" (11.38 G Floating Point Operations (FLOP) benchmarked on June-1997), is off no match to today's off-the-shelf laptop or smartphone (Single Exynos chip in Samsung S5 delivers 142 GFLOPS).

The field of ICT has been a direct beneficiary of the growth of semiconductor industry, so much so that today around 35% of semiconductor sales are in the ICT sector. This impact has been in all aspects of ICT industry, from the digitalization of switches to advanced communication schemes. The demand for higher performance and spectral efficiency combined with shrinking of transistor size and cost has led to implementation of communication schemes/algorithms which were deemed too expensive a decade earlier.

As mentioned previously in the introduction, this work mainly focuses on hardware trade-offs involved in the implementation of future wireless communication systems. Before dealing with implementation details in coming chapters, a brief introduction of hardware platforms and various optimizations involved during hardware implementation are explained.

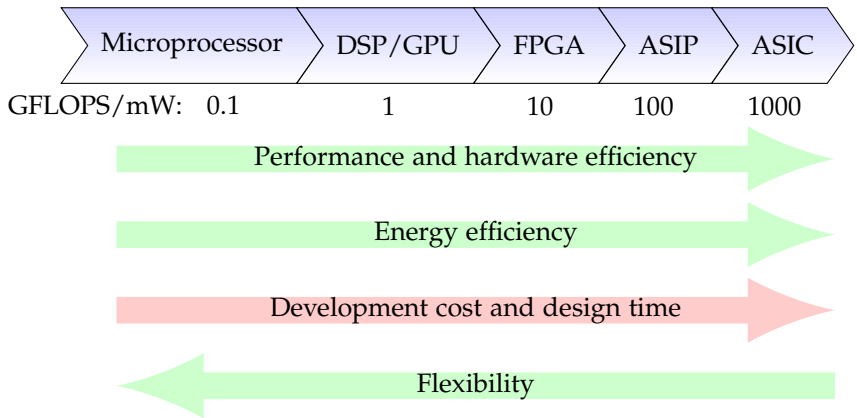


Figure 3.1. Design features for various hardware platforms.

3.1. HARDWARE PLATFORMS

Semiconductor industry growth is backed up by research and development in all aspects of silicon chip design. The industry is constantly evolving into different branches/variety of chips, which can be broadly classified as programmable processors, reconfigurable architectures and Application-Specific Integrated Circuits (ASICs) as shown in Fig. 3.1. The programmable devices include general purpose processors (e.g., Intel i7), microcontrollers (e.g., PIC-18f) and some domain specific processors like Digital Signal Processors (DSPs) (e.g., Ti C66), Graphics Processing Units (GPUs) (e.g., Nvidia GTX). The reconfigurable architectures differ from the programmable processors by exposing both data and control paths to the user and are configurable in hardware. Field-Programmable Gate Arrays (FPGAs) are one the most powerful and well-known examples of this architecture, but not limited to it. There are other products like Complex Programmable Logic Device (CPLD) and Programmable Array Logic (PAL) used for low-cost products. Application-Specific Instruction-set Processors (ASIPs) are highly optimized processors with customized data-paths with a limited degree of reconfiguration, offering programmability with small instruction sets. They are often assisted by a smaller scalar processor used to control the data-flow paths and scheduling. The performance of ASIPs is usually better than FPGAs in the domain of interest, due to limited overhead in data and control paths. ASICs are the most efficient hardware platform, mainly due to full customization of data-paths, scheduling and computational blocks. This makes the flexibility of ASICs quite limited. The selection of a particular hardware platform depends upon different design space aspects which are briefly described in the next section.

3.2. HARDWARE DESIGN ASPECTS

Power, performance, and cost are three major and highly dependent aspects of hardware design. It is desirable to achieve low power, low-cost, and high-performance devices, but unfortunately, these aspects do not accompany each other. Fortunately, shrinking of transistors and corresponding improvements in switching speed assists in lowering cost and improving performance. However, power consumption, especially static (leakage) power is becoming more dominant with shrinking technology. Also, the power density of chips increases dramatically, requiring advanced and expensive cooling systems.

Total power consumption of CMOS consists of two components, dynamic power (P_{dyn}) and static power (P_{stat}), and is expressed as

$$P_{\text{total}} = P_{\text{dyn}} + P_{\text{stat}}. \tag{3.1}$$

The dynamic power consumption (proportional to clock frequency) originates from different sources, of which the dominant is charging and discharging of internal and load capacitances. Other components are dynamic hazards caused by glitches and short-circuit currents. In this work, only the main component of dynamic power is considered and expressed as

$$P_{\text{dyn}} = \alpha C_{\text{L}} V_{\text{DD}}^2 f\,, \tag{3.2}$$

where α is circuit switching activity, C_{L} represents total capacitance, V_{DD} supply voltage and f operating clock frequency [27].

Traditionally dynamic power has dominated the power budget, but lately with scaling technology static power is also becoming important. The static power is in general defined as

$$P_{\text{stat}} = I_{\text{OFF}} V_{\text{DD}}\,, \tag{3.3}$$

where I_{OFF} is static current, consisting of sub-threshold drain-source leakage, gate leakage, and junction leakage. Among these three leakage sources, the sub-threshold leakage plays the most important role, with an exponential increase in leakage current when the threshold lowers [27].

In addition to the three main design parameters, power, performance, and cost, there are two more important hardware design aspects, which are flexibility and design time (time to market). Flexibility was mentioned as part of the design platforms in the previous section. The time to market is sometimes closely coupled with flexibility and is a crucial aspect in the industry. Also, implementing simpler algorithms (may impact performance), architectures which are easy to verify, etc., are efforts helping to shorten time to market. In the next section more details of such trade-offs are discussed by describing a generic hardware design optimization flow.

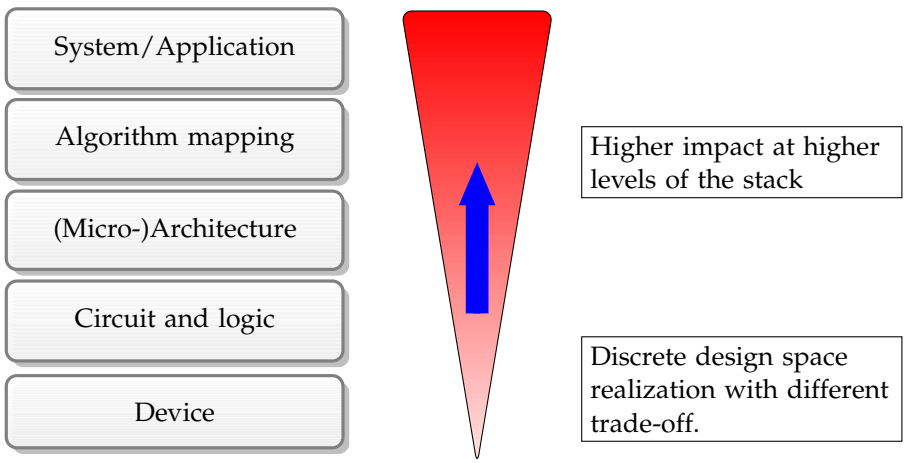


Figure 3.2. Hardware design trade-off [27].

3.3. DESIGN OPTIMIZATION

Fig. 3.2 shows the different stages, or stacks, involved in a hardware design flow. The optimization impact is higher at the higher stack levels, however, it should be noted that none of the stacks are independent from the other. The knowledge of lower level stacks like transistor speed, leakage power, has to be propagated to higher stacks, and unlike a top-down strategy a "meet-in-middle" strategy is often employed in modern design flow.

3.3.1. SYSTEM/APPLICATION

The choice of performance requirements, design costs, and its impact on the application or end-users is a crucial aspect of hardware design. For some scenarios these requirements may be rigid and quite critical, e.g., control circuits in nuclear reactors, cryptography for banking. However, the choice of requirements is quite flexible for numerous mass products such as music players and smartphones. Targeting a very high precision music quality may be desirable, however, it may not be noticed by end-user as much as the cost and battery-life. Therefore, the choices of requirements and trade-offs need not be some technical numbers, but rather involve more complex intuition and end-user real-life experiences. Therefore, this stack sets the requirements based on multitude of choices and has a high impact in the design space.

3.3.2. ALGORITHM MAPPING

Algorithm mapping involves selection of algorithm and corresponding mapping to a hardware platform. Trading performance and the choice of algo-

rithms is probably the most well-known aspect of hardware implementation. This involves various features such as design time, complexity, arithmetic and data operations involved, fixed point performance. Typically complexity of an algorithm in terms of FLOPs is proportional to hardware cost. However, there can be scenarios where an algorithm with low complexity has higher cost due to excessive data shuffling [28], division/square-root operations etc. Therefore, choice of the algorithm requires information from lower stacks as well, and drawing general conclusions on trade-offs are difficult.

Fixed point word-length optimization is another important aspect of algorithm mapping. Typically, microprocessors opt for floating point representation due to the high dynamic range required to support various applications. Reconfigurable hardware like FPGAs and some of the ASIPs support both floating-point and fixed-point. ASICs, on the other hand, are tuned for a specific application with known dynamic range, making fixed point implementations a natural choice. Lowering word-length can reduce critical paths, storage requirements, and hardware cost at the price of reduced accuracy. Therefore, the word-length should be long enough to give the necessary accuracy, but not longer.

3.3.3. MICRO-ARCHITECTURE

Optimization techniques become more independent of the application or algorithm when moving down the stack as shown in Fig. 3.2. At the architectural level, various circuit transformation and optimization techniques can be employed to trade between hardware cost, power consumption, and performance. Some of these techniques are used in this thesis, as briefly explained in the following.

TIME MULTIPLEXING Hardware re-use, or resource sharing, and time multiplexing are similar methods used to share hardware among several functional blocks to lower area cost. The sharing of resources typically requires a controller, additional muxes in data-paths and some storage. Also, depending upon the utilization factor of the modules, a higher clock frequency may be required to maintain the same throughput. An example is shown in Fig. 3.3, where f1 and f2 blocks are merged, and to maintain the same throughput the merged block may need to run at a higher clock frequency.

Although the total number of transistors are lowered, the operating clock frequency is higher and the possibility of scaling down supply voltage is reduced. Therefore, the total power consumption remains the same or increases if the design is dominated by dynamic power. For such designs optimization for power/energy would be to perform the opposite of time-multiplexing known as parallelism, albeit at a higher area cost. On the con-

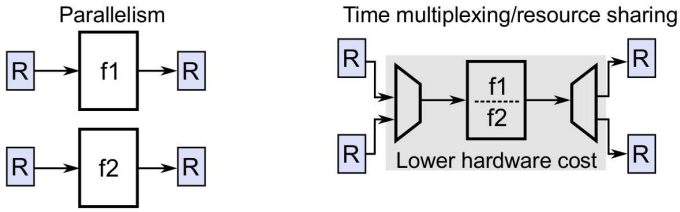


Figure 3.3. Time multiplexing, merging blocks to share resource.

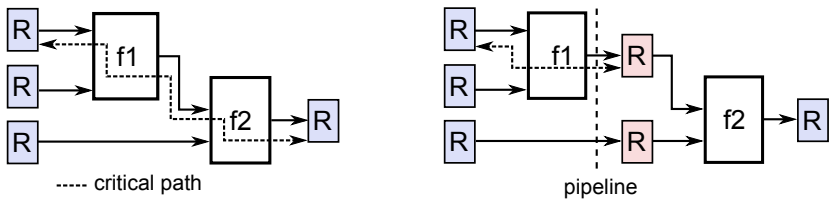


Figure 3.4. Pipelining, increase throughput at cost of latency.

trary, leakage dominant design operating on low clock frequencies gain by time-multiplexing due to lower leakage currents and still have a possibility to scale down supply voltage.

PIPELINING Pipelining is an important technique, wherein improvement in throughput is achieved at the cost of latency by inserting extra registers between logic as shown in Fig. 3.4. The area overhead of pipelining is much smaller than parallelism, mainly consisting of pipeline registers compared to replicating the design. The reduction in critical path can also be leveraged to scale supply voltage to lower power.

Pipelining at architecture level can also be at a higher functional or task level, using memories or FIFOs as pipeline stages. The main objective of such pipelining is to improve the utilization of hardware blocks, and are usually called a coarse pipeline. This is important in OFDM based wireless applications, where computations have to be performed repetitively across subcarriers.

On the other hand, fine grain pipelining or circuit level pipelining is used to shorten the critical path. In some cases the best position for a pipeline register is inside a multiplier (or adder) and inserting registers manually is difficult. In such cases modern synthesis tools are able to perform circuit retiming, where the pipeline register are moved to balance the timing paths in a design.

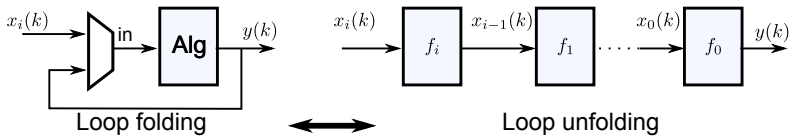


Figure 3.5. Folding, area reduction by using loop transformations.

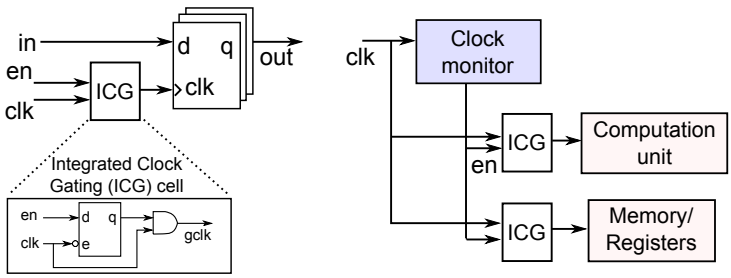


Figure 3.6. Local and global clock gating.

FOLDING/UNFOLDING Fig. 3.5 illustrates the folding and unfolding of an algorithm implementation. This mainly involves loop transformations such as loop unrolling and loop pipelining/retiming. The transformation can also make use of algebraic properties like associativity, distributivity, and commutativity. Design exploration using transformation is quite cumbersome, especially when using Hardware Description Language (HDL). However, the field of High-Level Synthesis (HLS) has created some major breakthrough in this area, especially in the signal processing domain. Providing constraint in HLS can easily fold/unfold loops and pipeline unrolled loops based on throughput requirements. Details on HLS-based design flow is described in Appendix-A.

CLOCK GATING Clock gating is a low power technique, which does not impact throughput, and is important to be considered during architecture development. Functionality or task-based clock gating has a bigger impact on reducing power since the controller knows the utilization or schedule of the blocks. For some scenarios it is hard to know the behavior of the blocks, hence local clock gating is required as shown in Fig. 3.6. In general, it is more efficient to group modules that are idle simultaneously for clock gating. There is no doubt that clock gating is a truly effective means of reducing dynamic power, ranging from 20%-60% reduction [29]. The area cost for global clock gating is minimal, and there is no excuse for not using it in modern digital designs.

3.3.4. CIRCUIT AND LOGIC

This stage of optimization is one level above the actual physical device, and it mainly involves parameters like supply voltage, threshold, device sizing, logic family. Some of these parameters are continuous parameters and give rise to a continuous optimization space curves. Discrete parameters like logic family or cell libraries result in a set of points in the optimization space. In this work, many of the parameters like cell libraries, logic families, transistor sizing, stacking are kept to default or adhere to standard digital ASIC flow. For example, the synthesis tool picks the correct multiplier topology based on the timing constraint. Similarly, gate drive strengths are dependent on the timing and power constraints provided to the tool. Some of these techniques are specific for ASIC flow (body-bias) but not limited to it, e.g., voltage scaling can be used in FPGAs and microprocessors but are dependent on vendors providing these options. The main parameters used in this work for optimization in this stage are described in the following.

VOLTAGE SCALING A highly effective way of exploiting performance and power trade-off is to opt for both voltage and frequency scaling. Voltage scaling provides a quadratic factor reduction in power consumption (3.2). Therefore, choosing a minimum operating point with required frequency is important to keep the average energy per operation low and still meet all the performance needs. Well known techniques like Dynamic Voltage Frequency Scaling (DVFS) perform such optimization at run-time. In this work (ASIC implementations), support for such optimization is provided and also measured, but the additional infrastructure (voltage sources, software framework, etc.) are not part of the implementation.

BODY-BIASING Typically transistors have a fourth terminal (in addition to gate, source, and drain), which can be used to influence the threshold voltage. Increasing of the threshold voltage, known as Reverse Body Biasing (RBB), reduces leakage current exponentially at the cost of lower operational frequency. On the other hand, Forward Body Biasing (FBB) lowers threshold voltage resulting in higher operating frequency. Dynamic Body-Biasing (DBB) introduced in [30], performs both FBB and RBB for active and standby modes, respectively.

Although this looks attractive at first glance, there are some overheads, mainly consisting of bias voltage generators (independent of process, temperature, voltage variations) and the distribution network. In [31] the total area overhead of all the bias units and the wiring turned out to be approximately 8%. In modern technologies, body-bias is quintessential to compensate for process/threshold variations. In this work, body-biasing is performed with

external voltage generators and is mainly used as a fine-tuning knob to handle process variations and performance-power trade-offs.

3.3.5. DEVICE

The lowest stack in the optimization mainly deals with the actual physical components. The choices involved are device platforms, technology nodes, type of devices (FDSOI or Bulk), transistor thresholds, cost, etc. These choices propagate up the optimization stack to make architectural decisions. In this work, Xilinx Kintex-7 FPGA's, ST-65 nm bulk technology, and ST-28 nm FD-SOI is used for implementations.

TRANSISTOR OPTIONS The transistors in a design kit or technology usually come in various flavors, mainly consisting of different physical parameters like size, doping levels (threshold level), gate-oxide thickness etc. Transistor size supported in digital design kits are usually a set of fixed lengths greater than or equal than the minimum channel length. Selecting a larger transistor than offered by technology may look sub-optimal at first glance, but it is a important option for reducing leakage and improving reliability while staying in the same technology. This can be cost effective for an organization, which can have licensing of the same technology node and cater for different applications.

Another important option available in the design kit is the different threshold levels. For example, ST-65 nm CMOS technology provides three options, high-V_{TH}, standard-V_{TH} and low-V_{TH}. As discussed earlier, the threshold level impacts maximum operational frequency as well as leakage current. The choice of transistor threshold type is an effective device level design parameter since there is no overhead in area, level conversions, and bias voltage generators. Moreover, multiple thresholds helps in reducing leakage by up to 50% without impacting timing [27]. The real burden is pushed to the manufacturing process, and also the challenge of optimal technology mapping. In this work, single threshold transistors are opted for according to design requirements, and when not mentioned explicitly high-V_{TH} with minimum length transistors of the corresponding technology is used. Details specific to ST-28 nm FD-SOI technology and the choice of transistor types is provided in Appendix-B.

Part I

Downlink Processing for Massive MIMO

Results and discussion in this part are from the following papers:

- H. Prabhu, J. Rodrigues, L. Liu, O. Edfors "A 60pJ/b 300Mb/s 128×8 Massive MIMO Precoder-Detector in 28nm FD-SOI", ISSCC, 2017.
- H. Prabhu, F. Rusek, J. Rodrigues, O. Edfors "High Throughput Constant Envelope Pre-coder for Massive MIMO Systems", ISCAS, 2015.
- H. Prabhu, O. Edfors, J. Rodrigues, L. Liu, F. Rusek "A low-complex peak-to-average power reduction scheme for OFDM based massive MIMO systems", ISCCSP, 2014.
- H. Prabhu, J. Rodrigues, O. Edfors, F. Rusek "Approximative Matrix Inverse Computations for Very-large MIMO and Applications to Linear Pre-coding Systems", WCNC, 2013.

4

Pre-coding Techniques

The assumption of an unlimited number of BS antennas greatly simplifies the theoretical analysis of massive MIMO systems [20]. However, it is obvious that in a practical system the number of antennas cannot be arbitrarily large due to factors such as physical area, cost, and power constraints. The theoretical analysis in [20], assumes that inner products between propagation vectors of different users grow at a slower rate than inner products of the propagation vectors with themselves when the number of antennas grows, *i.e.*, the user channels are asymptotically orthogonal. In [32], measurements in a realistic propagation environment for large antenna arrays at a BS (up to 128 antennas at the BS and 26 different single antenna users) were performed. It was shown that it is possible to separate single user channels. Furthermore, in [23], residential area measurements were performed, showing linear pre-coding sum rates of up to 98% of those achieved by Dirty Paper Coding (DPC), for BS to MS antenna ratios as low as 10.

Although there is a clear benefit of scaling up the number of BS antennas, including a near-optimal linear pre-coding, the hardware cost and signal processing complexity can still be high. Employing linear pre-coding such as Zero-Forcing (ZF) requires inverting a $K \times K$ matrix, where K is the number of single antenna MSs. In addition to large matrices, the pre-coding matrix may need to be updated frequently, based on the Doppler spread of the channels. Therefore, simultaneously delivering high throughput, accuracy, and flexibility to support different MIMO configurations with reasonable cost requires optimization at both algorithm and hardware level. The various optimization strategies are described in subsequent chapters. This chapter introduces pre-coding techniques and hardware implementation challenges.

4.1. PRE-CODING IN MASSIVE MIMO

In line with the system model described earlier in chapter 2.2.1, the downlink signal model for a narrow band system is

$$\boldsymbol{y} = \sqrt{P}\boldsymbol{H}\boldsymbol{p} + \boldsymbol{w}\,. \tag{4.1}$$

The key idea here is to transmit signals which propagates through the channel and constructively adds up at the users. For favorable channels the users can be spatially separated to create parallel data streams. Compared to a simple broadcasting, the high spatial multiplexing gains require additional processing. In general linear pre-coding can be seen as a domain transformation, where $K\times 1$ user symbols ($\boldsymbol{s}$) are mapped to $M\times 1$ pre-coded transmit vector ($\boldsymbol{p}$) as

$$\begin{aligned} &\boldsymbol{p} = f_{precode}(\boldsymbol{H}_{K\times M},\ \boldsymbol{s},\ \text{SNR, hardware constraints}) \\ &\text{such that } \ \mathbb{E}[\boldsymbol{p}^H\boldsymbol{p}] \leq 1 \\ &\qquad\qquad \boldsymbol{s} = \boldsymbol{H}\boldsymbol{p}\,. \end{aligned} \tag{4.2}$$

The expression is an informal representation of pre-coding, since there exists many arbitrary solutions. Typically, the pre-coding schemes target reduction of inter-user interference, improve power efficiency, array gain, etc. The hardware constraints for pre-coding in (4.2) can be various factors such as maximum possible signal amplitude, hardware impairments, PAR limitations. Linear pre-coding schemes depend on channel matrix and Signal-to-Noise Ratio (SNR), and are described in the following:

MF PRE-CODING SCHEME A Matched Filter (MF) linear pre-coding, also known as Maximum-Ratio Transmission (MRT) or Hermitian pre-coding, is a simple low complexity pre-coder given as

$$\boldsymbol{W}_{\text{MF}} = \boldsymbol{H}^H\,. \tag{4.3}$$

Therefore, the pre-coding can be performed in a distributed way, with a very small hardware cost. Although MF is a low complexity scheme, to attain certain performance levels would require a lot more antennas at BS than other schemes like ZF [33].

ZF PRE-CODING SCHEME Zero-Forcing linear pre-coding transmits user signals towards the intended user with nulls steered in the position of other users as illustrated in Fig. 4.1. The ZF pre-coder is given as

$$\boldsymbol{W}_{\text{ZF}} = \boldsymbol{H}^{\dagger}\,, \tag{4.4}$$

where $\boldsymbol{H}^{\dagger} = \boldsymbol{H}^{H}(\boldsymbol{H}\boldsymbol{H}^{H})^{-1}$ is the pseudo-inverse of the channel matrix. Perfect Channel State Information (CSI) at the transmitter, full rank of $\boldsymbol{H}$, and nulling makes this scheme interference free. The sum rate expression for ZF is described in [23], wherein sum-rates can be maximized by performing optimal power allocation. However, in this work, a fair power allocation is performed such that all users more or less have the same performance (SNRs). As the number of BS antennas M increases, $\boldsymbol{H}$ tends to have nearly orthogonal columns as the terminals are not correlated due to their physical separation. This often results in performance of ZF pre-coding being close to that of optimal DPC pre-coding. However, ZF pre-coding requires computation of the pseudo-inverse (in (4.4)), which requires the hardware expensive inversion of a $K \times K$ matrix.

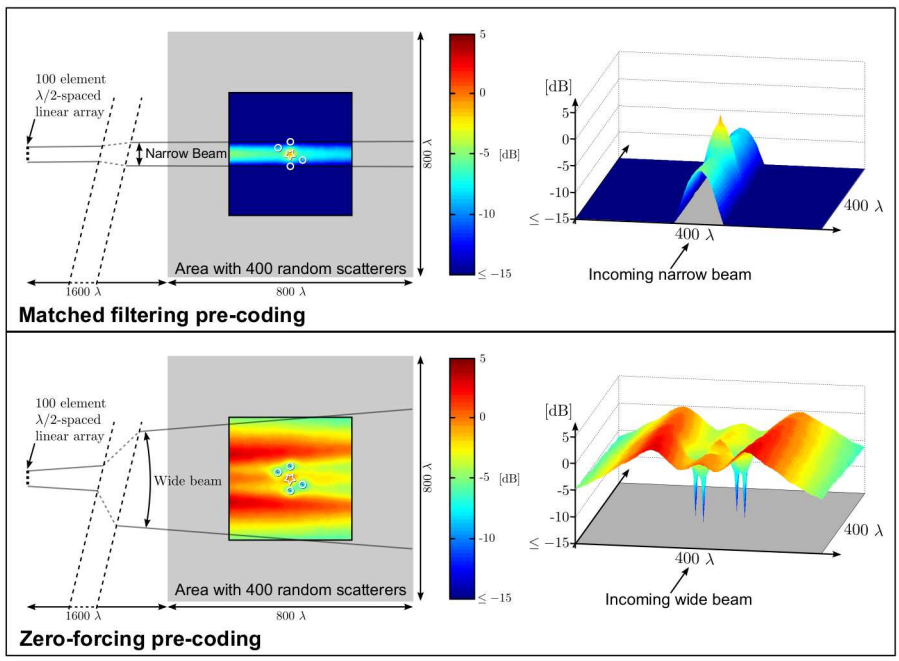


Figure 4.1. Relative field strength around terminals in a scattering environment for MF and ZF pre-coding. The left image is pseudo-color plots of average field strengths, with target user positions at the center ($\star$), and four other users nearby ($\circ$). Right plot is average field strengths as surface plots. (Artwork from Ove Edfors), © 2014 IEEE. Reprinted, with permission, from [10].

MMSE PRE-CODING SCHEME Minimum Mean Square Error (MMSE) pre-coding can trade interference reduction for signal power inefficiency. The MMSE pre-coder is given as

$$\boldsymbol{W}_{\text{MMSE}} = \boldsymbol{H}^H \left(\boldsymbol{H}\boldsymbol{H}^H + \frac{\boldsymbol{I}_K}{\rho} \right)^{-1}, \tag{4.5}$$

where ρ is the Signal-to-Noise Ratio (SNR) at the MS. At low SNRs, the MMSE approaches MF pre-coder and at high SNRs, it approaches the ZF pre-coder. In hardware, the implementation of MMSE is similar to that of ZF pre-coder. In this work, although ZF pre-coders are implemented, it can be modified to perform MMSE with minimal architectural changes.

4.2. HARDWARE DESIGN ASPECTS FOR PRE-CODING

As mentioned earlier, operating on large matrices in a limited time frame is an implementation challenge. In addition to latency issues, it is also important to keep the hardware cost and system-bus bandwidth requirements low. In the next sub-section, first the computational complexity for different pre-coding strategies are analyzed, which is followed by system-bus communication overhead.

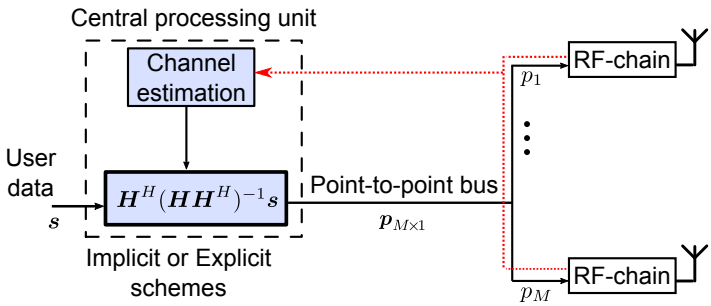
4.2.1. COMPUTATIONAL COMPLEXITY

The attractiveness of MF is that it can be performed in a distributed fashion, independently of each antenna unit, as shown in Fig. 4.2. While ZF requires a centralized processing, it reaches the same performance levels as MF with a lot lower number of antennas. An order of magnitude reduction in the number of RF-chains is quite significant, and a BS capable of performing ZF can always switch to MF. Therefore, the focus of this work is on ZF. The computational complexity and system-bus requirements of ZF pre-coding vary based on the chosen architectural strategies. These strategies, in turn, translate to different hardware costs and hence is important to analyze. The key idea is that ZF (and MMSE) which requires a pseudo inverse, can be performed either by inverting matrices or by solving a system of linear equations or merging both (Hybrid) approaches. Table 4.1 shows the computational complexity for different approaches and are described in the following:

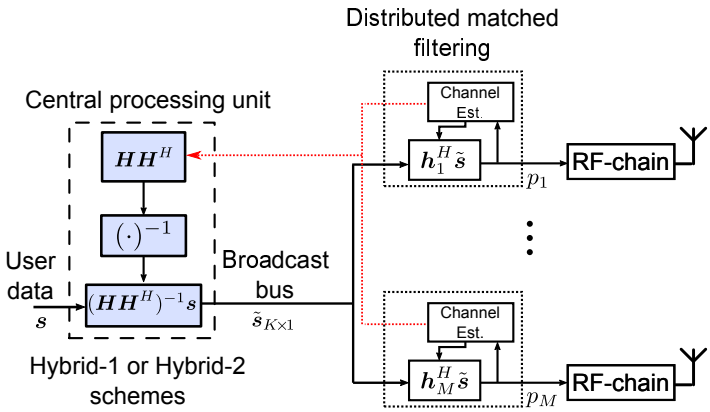
EXPLICIT ZF Explicit computation basically generates the pseudo inverse of the channel matrix, which includes all three steps, *i.e.*, Gram matrix generation, matrix inversion, and matrix multiplication. After explicitly computing the pseudo inverse ($\boldsymbol{H}^{\dagger}$), a matrix-vector multiplication is performed with the

user data. At a first glance, explicit ZF seems to have the highest complexity. However, when the channel coherence bandwidth/time is large, the rate of computing matrix pseudo-inverse lowers, since the computed pseudo-inverse can be reused over frequency and time.

IMPLICIT ZF In the case of implicit ZF pre-coding, no matrix inversions are performed. Basically, the pre-coding vector is computed by solving linear equations using iterative methods. The complexity depends on the convergence of the iterative method. At a first glance, this approach may look more favorable than explicit ZF in terms of complexity. However, the factor b in Table 4.1 is typically greater than 2, and more importantly, implicit ZF can not exploit the channel coherence bandwidth and coherence time. This is due to the fact that, every realization of user data requires solving a new set of linear equations.



(a) Fully central, implicit or explicit ZF.



(b) Partially distributed, Hybrid-1 or Hybrid-2 ZF.

Figure 4.2. ZF pre-coding architecture/schemes.

Table 4.1. Complexity for different pre-coding approaches.

		per-channel realization			per-user data
		Gram Matrix Generation	Gram Matrix Inversion	Pseudo Inverse Generation	Matrix vector operation
Matched Filtering		-	-	-	MK
ZF/MMSE	Explicit	$\frac{1}{2}MK^2$	aK^3 *	MK^2	MK
	Hybrid-1	$\frac{1}{2}MK^2$	aK^3	-	$K^2 + MK$
	Hybrid-2	$\frac{1}{2}MK^2$	-	-	$bK^2 + MK$ **
	Implicit	-	-	-	bMK

* a is constant depending on inversion algorithm.
** b is constant depending on iterative methods.

HYBRID ZF SCHEMES An alternative approach would be to merge explicit and implicit pre-coding. In addition, the hybrid ZF schemes exploit the channel coherence bandwidth/time to lower computational complexity. In both of the hybrid methods, Gram matrix ($\boldsymbol{Z} = \boldsymbol{H}\boldsymbol{H}^H$) is computed explicitly. However, matrix inversion is performed explicitly only in the Hybrid-1 scheme, where the value of a in Table 4.1 depends on the inversion algorithm. Hence, an additional term is added to the vector operations, which relates to the user data vector multiplied with the inverse of Gram matrix. The Hybrid-2 approach avoids explicit inversion and instead opts for iterative methods.

Complexity of the different schemes depends on M, K, and the rate of change of the channel matrix with respect to user-data. In OFDM based systems the channel matrix can be assumed to be constant over a certain number of sub-carriers and symbols, depending on the scenario. This makes it hard to generalize the selection of any particular scheme. In this thesis, the algorithms are generic, however, some of the implementations have requirements derived from the LuMaMi frame-structure [34]. Consider a system with $M = 128$ BS antennas serving $K = 8$ users. The OFDM frame-structure of the testbed assumes the channel matrix to remain same over 5 OFDM symbols and over K sub-carriers. For such a system, the Hybrid-1 strategy can exploit the coherence time/bandwidth of the channel efficiently, and in turn lower the complexity. In addition to lower complexity, the Hybrid-1 strategy has another important advantage in terms lowering system-bus communication overhead as described in the next sub-section.

4.2.2. SYSTEM BUS COMMUNICATION OVERHEAD

In massive MIMO, data transfer between the large number of RF-chains inevitably exerts a huge strain on the system bus. Lowering the system bus

Table 4.2. System-bus overhead for different architectures.

		Data Transfer [num. of samples] +	Broadcast
Matched Filtering		$\gamma_d K$	Yes
ZF/MMSE	Explicit	$\gamma_h MK + \gamma_d M$	No
	Hybrid-1	$\gamma_h MK + \gamma_d K$	Yes
	Hybrid-2	$\gamma_h MK + \gamma_d K$	Yes
	Implicit	$\gamma_h MK + \gamma_d M$	No

+ γ_h - channel estimation rate, γ_d is user vector transmission rate, and $\gamma_d > \gamma_h$.

requirements is challenging, especially considering the large channel matrices and data which needs to be transferred to a centralized processing unit. In case of MF, system-bus bandwidth requirements are much lower, since the user data is broadcast to all the remote RF-chains which perform pre-coding locally, as shown in Fig. 4.2(b). The remote units are integrated with the RF-chains, and are capable of performing vector-dot product operation and channel estimation. In case of ZF, the channel estimates from the remote units need to be transferred to a central processing unit. The user symbols or data-vector to be transferred depend on the pre-coding strategy, as shown in Table 4.2. For both of the hybrid schemes $(\boldsymbol{H}\boldsymbol{H}^H)^{-1}\boldsymbol{s}$ is first performed centrally. The resulting vector ($\tilde{\mathbf{s}}$) is then broadcasted to remote units. Where a multiplication with ($\boldsymbol{H}^H$) is performed, similar to MF. Hence, these hybrid schemes have a lower system bus bandwidth requirement compared to the other ZF architecture/strategies.

As an example, consider a system with $M = 128, K = 8$, and a channel coherence time $T_{\text{ch}} = 1$ ms (pedestrian model). The rate of channel estimation is around $\gamma_h = 1/T_{\text{ch}}$, and if the data rates to each user is $\gamma_d = 10^6$ symbols per sec (if 16-QAM, leads to a moderate 4 Mbps/user). The bandwidth requirement on the system-bus for MF is around 8 MSamples/sec, whereas in case of ZF opting implicit or explicit strategy requires 129 MSamples/sec. However, by using Hybrid schemes, which utilize the same broadcasting mechanism as MF, the bandwidth requirement on the system-bus is lowered to 9 MSamples/sec. This highlights the importance of architecture/algorithm selection, which has to encompass many of the system level aspects.

4.2.3. LATENCY REQUIREMENTS

A LTE-A OFDM based frame-structure is a reasonable assumption for initial development before a standard is established. The frame-structure for the LuMaMi testbed is shown in Fig. 4.3. The frame-structure is a bit more re-

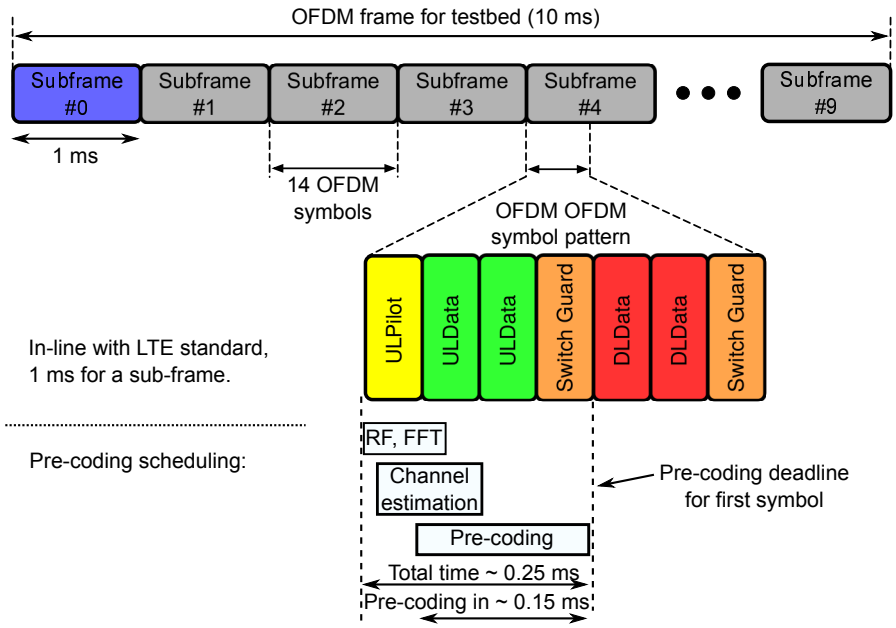


Figure 4.3. OFDM frame structure for LuMaMi massive MIMO testbed.

laxed than the LTE-A standard, especially when considering the two guard frames in a sub-frame. Also, there is in an assumption that the channel remains almost constant over $N_{\text{int}} = 10$ sub-carriers and over 5 OFDM symbols. These simplifications are valid since massive MIMO deployments would be in dense urban areas (shopping malls, stadiums) with pedestrian users, provide internet to homes in rural area, etc. However, the frame-structure is not limited to slow moving MSs, practical experiments have shown that it can easily support MSs with speeds of around 70 km/h.

The simplification in the frame-structure aids in lowering the necessary number of matrix operations, and also lowers the throughput requirements. For a 20 MHz bandwidth 128×8 massive MIMO system, 120 channel inversions and 1200 user data-vectors need to be pre-coded in less than 150 μs. In the testbed, due to the communication overhead, there are 8 instances of pre-coder each handling 150 sub-carriers. Thus, the challenge is to design highly tuned (not over designed) hardware architectures handling large matrices and strict timing deadlines. Furthermore, design time, reconfigurability, and scalability, are also desirable hardware traits for the future (*i.e.*, as and when the standards mature).

5

Algorithms and Implementation

This chapter describes different algorithms and hardware optimizations for massive MIMO pre-coders, in particular focusing on matrix inversion. It is divided into three sections based on the targeted hardware platforms. The first section covers a FPGA implementation using a novel Neumann series based low-computational complexity pre-coding. The aim of this design was to implement a rapid prototype of matrix inversion accelerator for the LuMaMi testbed. The second section covers a much more elaborate and highly optimized ASIC implementation. A new modified QR Decomposition (QRD) is used for pre-coding, which lowers the inversion complexity with negligible performance impact and, in turn, provides high hardware efficiency. The third section provides details of the mapping of a Modified Gram-Schmidt (MGS) based QRD on a reconfigurable vector processor. This, unlike the previous two designs, provides a more software-centric solution with a potential for almost ASIC like performance.

5.1. FPGA PROTOTYPING BASED ON NEUMANN SERIES

The section on massive MIMO pre-coding in [35], briefly introduced an idea for a low-complexity matrix inversion approach based on Neumann Series (NS). This was further analyzed for pre-coding in [36], followed by many other investigations for both detection and pre-coding in [37–40].

The key idea for reducing pre-coding computational complexity is to exploit certain special properties occurring in massive MIMO systems. Although the massive MU-MIMO model is similar to a standard MIMO model, the increased number of BS antennas has several consequences. Things that were random before, now start to look deterministic. For example, with increasing BS antennas, the Gram matrix $\boldsymbol{H}\boldsymbol{H}^H/M$, asymptotically tends to a diagonal

matrix for certain "nice enough" channel conditions. This diagonal dominance of the Gram matrix is exploited during matrix inversion by employing NS, which in turn results in simpler hardware.

5.1.1. NEUMANN SERIES IN MASSIVE MIMO

In massive MIMO, as the number of antennas at BS (M) and single antenna users (K) increases, the eigenvalues of the Gram matrix $\boldsymbol{Z} = \boldsymbol{H}\boldsymbol{H}^H$ converges to a fixed deterministic distribution, known as the Marchenko-Pastur distribution [21]. This can be utilized to lower the computational complexity of Gram matrix inversion. Following the analysis in [35], the largest and the smallest eigenvalues of $\boldsymbol{Z}$ converge to

$$\lambda_{\max}(\boldsymbol{Z}) \to \left(1 + \frac{1}{\sqrt{\beta}}\right)^2, \lambda_{\min}(\boldsymbol{Z}) \to \left(1 - \frac{1}{\sqrt{\beta}}\right)^2,$$

where ($\beta = M/K$), as M and K grows to infinity. Simple scaling and rearrangement leads to an interesting property about the distribution of eigenvalues. Eigenvalues are important, as they give an insight on the convergence of the Neumann series. The faster the convergence of Neumann series the lower the computational complexity to reach a certain inversion accuracy. The arithmetic operations are in line with [35], wherein $\boldsymbol{Z}$ is scaled with a factor $\left(\frac{\beta}{1+\beta}\right)$, and the eigenvalues are found in the region

$$\lambda_{\max}\left(\frac{\beta}{1+\beta}\boldsymbol{Z}\right) \to \left(1 + 2\frac{\sqrt{\beta}}{1+\beta}\right),$$

$$\lambda_{\min}\left(\frac{\beta}{1+\beta}Z\right) \to \left(1 - 2\frac{\sqrt{\beta}}{1+\beta}\right). \tag{5.1}$$

Hence, the eigenvalues of $\boldsymbol{I}_K - \beta/(1+\beta)\boldsymbol{Z} = \boldsymbol{I}_K - \boldsymbol{Z}/(M+K)$ lie in the range $[-2\sqrt{\beta}/(1+\beta), 2\sqrt{\beta}/(1+\beta)]$, where $\boldsymbol{I}_K$ is an $K \times K$ identity matrix. By asymptotically increasing β, the eigenvalues of $\boldsymbol{I}_K - \boldsymbol{Z}/(M+K)$ lie in the range

$$\lim_{\beta\to+\infty}\left[\left(-2\frac{\sqrt{\beta}}{1+\beta}\right), \left(2\frac{\sqrt{\beta}}{1+\beta}\right)\right] \to [-0, 0]. \tag{5.2}$$

Therefore, as β grows, the faster is the convergence of

$$\lim_{n\to\infty}\left(\boldsymbol{I}_K - \frac{1}{M+K}\boldsymbol{Z}\right)^n \simeq \boldsymbol{0}_K. \tag{5.3}$$

It is known that if a matrix satisfies (5.3), its inverse can be expressed by Neumann series [41] as

$$\boldsymbol{Z}^{-1} \approx \frac{\delta}{M+K} \sum_{n=0}^{L} \left(\boldsymbol{I}_K - \frac{\delta}{M+K} \boldsymbol{Z} \right)^n , \tag{5.4}$$

with equality when L grows to infinity, and $\delta < 1$ is an attenuation factor introduced, since for finite M and K the eigenvalues of channel realizations may lie outside the range specified in (5.1). For a feasible implementation of a matrix inversion using Neumann series the number of iterations (L) needs to be finite and small.

The inverse of $\boldsymbol{Z}$ is approximated by a summation of powers of a matrix (or matrix multiplications) (5.4), which has a complexity order $\mathcal{O}((L-1) \cdot K^3)$. Although the computational complexity can be equal or higher (depending on L) than computing the exact inverse, performing matrix inversion using matrix multiplications are simpler to implement in hardware.

The convergence of (5.3) is based on the fact that the eigenvalues lie in the range given by (5.1) as M and K grows asymptotically. However, for practical systems with finite M and K the eigenvalues may lie outside this range. In addition to what is described in [35], we introduce one modification of the Neumann series inversion. It is based on the fact that the closer the eigenvalues of the inner matrix are to zero, the faster the convergence of the series in (5.4). The modification is described as follows. The scalar multiplication by $\delta/(M+K)$ in (5.4) can be represented as a diagonal (inverse) matrix

$$\boldsymbol{X}_{\mathrm{MP}}^{-1} = \frac{\delta}{M+K} \boldsymbol{I}_K .$$

Using this notation, (5.4) is rewritten as

$$\boldsymbol{Z}^{-1} \approx \sum_{n=0}^{L} \left(\boldsymbol{I}_K - \boldsymbol{X}_{\mathrm{MP}}^{-1} \boldsymbol{Z} \right)^n \boldsymbol{X}_{\mathrm{MP}}^{-1} , \tag{5.5}$$

and the accuracy of the approximation, for a given number of terms (L), depend on the size of the eigenvalues of $(\boldsymbol{I} - \boldsymbol{X}_{\mathrm{MP}}^{-1} \boldsymbol{Z})$. The smaller their magnitude, the faster the convergence. Thus, the goal is to pre-condition $\boldsymbol{Z}$ so that it will lead to a fast convergence for a finite M and K system.

Now, assume that we want to pre-condition with a diagonal matrix $\boldsymbol{X}_d$, with non-zero diagonal entries. In principle, we would like to calculate the eigenvalues of $(\boldsymbol{I} - \boldsymbol{X}_d^{-1} \boldsymbol{Z})$ and optimize $\boldsymbol{X}_d$ so that the magnitudes of the eigenvalues are as small as possible. This, however, is a complex and non-trivial task. Therefore, the Gershgorin's circle theorem [42] is used to derive an upper bound of the magnitude of the eigenvalues. Keeping this bound small,

by selecting $\boldsymbol{X}_d$, will also guarantee that the magnitude of the eigenvalues are small. This derivation of the $\boldsymbol{X}_d$ assumes that the Hermitian matrix $\boldsymbol{Z}$ is diagonally dominant, meaning that the magnitudes of the diagonal elements z_{ii} are larger than the sum of the magnitude of the off-diagonal elements in the same row, $z_{ij}, i \neq j$, namely that $|z_{ii}| > \sum_{i \neq j} |z_{ij}|$. The largest magnitude of any eigenvalue of $(\boldsymbol{I} - \boldsymbol{X}_d^{-1}\boldsymbol{Z})$ is upper bounded by

$$\max_i |\lambda_i| \leq \max_i \left(|1 - \frac{z_{ii}}{x_i}| + \frac{\sum\limits_{i \neq j} |z_{ij}|}{x_i} \right), \tag{5.6}$$

and under the condition of a diagonally dominant $\boldsymbol{Z}$, the smallest upper bound is obtained if $x_i = z_{ii}$. For this selection of $\boldsymbol{X}_d$ we also have that $\max\limits_i |\lambda_i| < 1$, which guarantees convergence of the Neumann series. Hence, the final approximation of the inverse of a diagonally dominant $\boldsymbol{Z}$ is by using a pre-conditioning matrix $\boldsymbol{X}_d = \text{diag}(z_{11}, z_{22}, ..., z_{kk})$, and the inverse can be expressed using Neumann series as

$$\boldsymbol{Z}^{-1} \approx \sum_{n=0}^{L} \left(\boldsymbol{I}_K - \boldsymbol{X}_d^{-1}\boldsymbol{Z} \right)^n \boldsymbol{X}_d^{-1}. \tag{5.7}$$

A fast (or accelerated) way to compute the series (5.4) and (5.7), up to $L = 2^P - 1$ terms, where P is an integer, is to use the identity

$$\boldsymbol{Z}^{-1} \approx \sum_{n=0}^{L} \left(\boldsymbol{I}_K - \boldsymbol{X}_d^{-1}\boldsymbol{Z} \right)^n \boldsymbol{X}_d^{-1} = \left(\prod_{n=0}^{P-1} (\boldsymbol{I} + (\boldsymbol{I} - \boldsymbol{X}_d^{-1}\boldsymbol{Z})^{2^n}) \right) \boldsymbol{X}_d^{-1}, \tag{5.8}$$

which leads to a numerical complexity proportional to the logarithm of the number or terms in the truncated series. In terms of number of matrix multiplications, the brute force computation of the inverse using (5.7) (or (5.4)) would require $L - 1$ matrix multiplications, whereas computation using (5.8) would require only $2(P - 1)$ matrix multiplications, where $P = \log_2(L + 1)$. This acceleration methods although very promising is only beneficial if $L > 3$.

In [43], a method to solve linear systems using Operator Perturbation Technique (OPT) is described. It can be shown that this method is essentially the same as Neumann series, and the proposed acceleration can be applied to OPT for an exponential convergence. In [44], another method to accelerate OPT is developed, which could be merged with these techniques. In [37], we proposed an improvement in convergence, which was achieved by using a tri-diagonal matrix ($\boldsymbol{X}_t$) as an initial matrix. Although, not as trivial as inverting a diagonal matrix, employing a tri-diagonal initial matrix leads to fewer iterations and in turn reduced complexity. Moreover, inverting a tri-diagonal

Table 5.1. Kintex-7 FPGA results for 100×10 massive MIMO pre-coder based on Neumann series.

FPGA Resource	Num. of instances	Utilization Percent.	Operation	Clock Cycles	Latency at 150 MHz
DSP48	216	14%	Gram Matrix	360	2.4 μs
BRAM	181	22%	NS Inverse	330	2.2 μs
Slice Register	28 k	5%	Matrix-vector	110	0.73 μs
LUT	53 k	20%	multiplication		

matrix could be performed in a streaming hardware friendly data-flow architecture. However, the common issue with the approximative inversions (NS) still prevails, *i.e.*, the accuracy depends on the convergence (5.8) with iterations (P). Therefore, it is important to have a trade-off between complexity and the accuracy of the approximation.

5.1.2. PERFORMANCE OF NEUMANN SERIES

Fig. 5.1 shows the Bit Error Rate (BER) performance for different MIMO configurations. NS performance is not as good as exact ZF when the ratio of BS to MS antennas ($\beta = 2$) is small. However, for massive MIMO it is expected that a large number of BS antennas serves a few single antenna users. Thus, the ratio of operation is expected to be around or greater than 8, in which case the performance of NS gets very close to the exact ZF.

5.1.3. NEUMANN SERIES BASED PRE-CODING IMPLEMENTATION

The pre-coder implementation is targeted for Xilinx®Kintex-7™ FPGA used in the LuMaMi testbed. The design used Vivado®- HLS flow with source code written in C/C++. The main target of the implementation was to meet the timing requirements (Fig. 4.3) of the frame-structure described in the previous chapter. Also, it was expected to be rapid prototyping for the testbed to evaluate or perform initial studies, before the development of pre-coder in National Instruments®LabView™. This requirement for rapid prototyping goes along with Neumann series, since it has a simple data-flow and designing matrix multiplication units are trivial in hardware, especially when using HLS (see Appendix-A).

The hardware architecture shown in Fig. 5.2, is highly pipelined to perform operations across sub-carriers. The external interface module handles the protocol to integrate the pre-coder to the LuMaMi testbed (LabView) interface. The channel First-In First-Out (FIFO), buffers the channel matrix ($\boldsymbol{H}$) which arrives in a serial fashion (element-wise). The user data is stored in a separate FIFO since the arrival time of channel and user data is different. Furthermore,

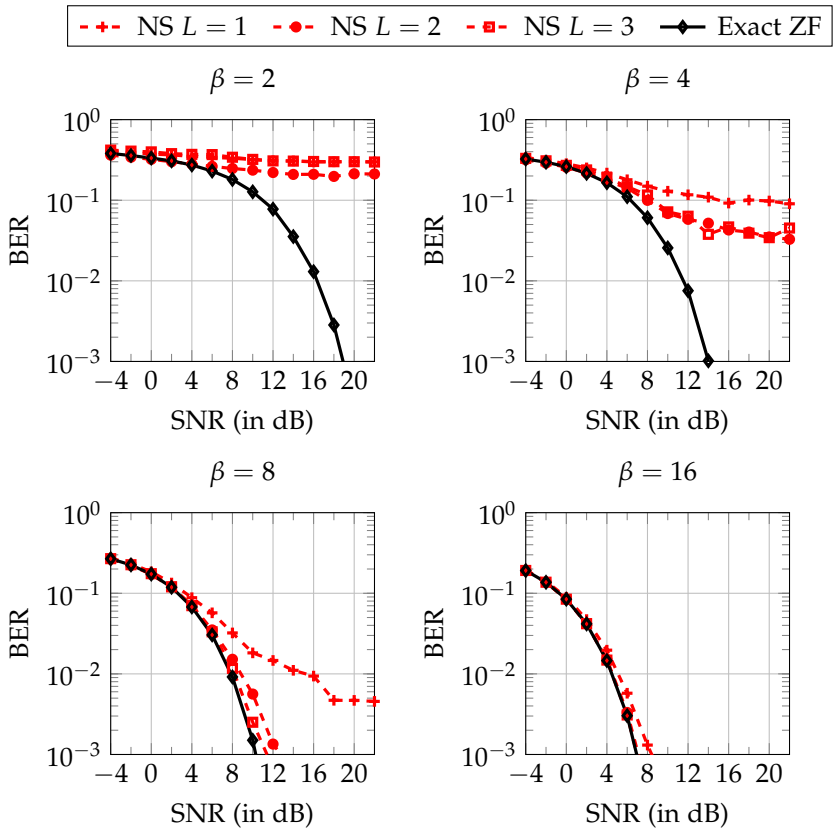


Figure 5.1. NS performance for different antenna ratio β and SNR at MS. Simulations with uncoded 16-QAM and i.i.d. Rayleigh channel.

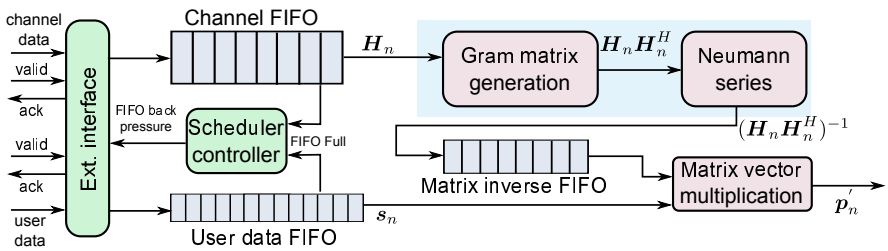


Figure 5.2. Neumann Series implementation on FPGA

the FIFOs are over-designed, since the system DMA burst/data rate are not guaranteed to be consistent. There is an option for FIFO to stall external interface, however, by setting the size of FIFOs to worst case size, this feature acts only as a safety measure. The testbed uses 8 instances of the pre-coder, the

Table 5.2. FPGA results for pre-coding module.

	QR Decomposition [45]	Neumann series
Complexity	$MK^2 + MK$	$0.5MK^2 + K^3$
Clock Freq. [in MHz]	200	150
Registers	46470 (9.1%)	16000 (3.1%)
LUT	49315 (20.3%)	28700 (11.8%)
DSP48	596 (38.7%)	216 (14%)
Latency [in μs]	5.1	2.4

FIFOs buffers all the user data for corresponding (1200/8=) 150 sub-carriers and (150/10=) 15 channel matrices required for 1 sub-frame. In total, the FIFO size is around 72 Kbytes. The processing is straight forward with two matrix multiplication blocks and one matrix-vector product unit. The Gram matrix and Neumann series based matrix inversion unit are merged together and share the hardware resources (multipliers and adders). This is another important advantage of NS, *i.e.*, it does not require a special dedicated inversion and can use the existing matrix multiplication Multiply-Accumulates (MACs) for inversion. The matrix-vector multiplication unit is separated for parallel operations and uses 10 complex multipliers. The hardware utilization and latency numbers are shown in Table 5.1. The resource utilization is reasonable considering the matrix dimensions and the over-designed FIFOs.

For a fair comparison with National Instruments®QRD implementation [45], the FIFOs in the design are removed and only the core processing block responsible for matrix inversion is accounted. The results of the NS pre-coder implemented in Kintex-7-410T with 150 MHz clock is shown in Table 5.2. Performing QRD directly on the large channel matrix $\boldsymbol{H}$ is very expensive in hardware. More importantly, the QRD design operates at a higher clock rate and still has a much higher latency. Therefore, it is beneficial to lower the dimension by first performing matched filtering and then using approximative inversion.

Before moving to the next hardware platform some takeaways from NS based implementation are as follows:

- High hardware re-usage between Gram matrix and NS inversion, since both perform matrix multiplication operations.
- Design time is low, hence suitable for rapid prototyping. Coupled with HLS, it is only a few lines of C++ code.
- Accuracy of the approximation depends on channel conditions and ratio of BS to MS antennas β.

5.2. ASIC IMPLEMENTATION BASED ON APPROXIMATE QRD

In the previous section, an approximative NS based pre-coder was implemented on an FPGA. Though it has some advantages like simpler architecture, the implementation is not robust against channel conditions. In this section a novel approximative QRD algorithm is described, followed by architectural details and measurement results.

5.2.1. APPROXIMATE QRD

QRD is a popular approach for solving linear equations and to perform matrix inversion in hardware. QRD of $\boldsymbol{Z}$ with dimension $K \times K$, is a decomposition of the matrix into a product

$$\boldsymbol{Z} = \boldsymbol{Q}\boldsymbol{R}, \tag{5.9}$$

where $\boldsymbol{Q} \in \mathbb{C}^{K \times K}$ is unitary and $\boldsymbol{R} \in \mathbb{C}^{K \times K}$ is upper triangular. This computation can be arranged in several ways e.g., Gram-Schmidt orthogonalization [46], Householder transformation [47] or Givens rotations [48]. All these approaches require around $\mathcal{O}(l\,K^3)$ multiplications, where $1 < l < 1.5$. Explicit inversion after QRD is given by

$$\boldsymbol{Z}^{-1} = \boldsymbol{R}^{-1}\boldsymbol{Q}^{H}, \tag{5.10}$$

which requires further computation of $\boldsymbol{R}^{-1}$ and a matrix multiplication. High complexity of (5.9) and (5.10) is lowered by performing approximative QRD, computed utilizing the properties of massive MIMO. In particular, as mentioned earlier, when increasing the number of BS antennas, the Gram matrix $\boldsymbol{Z} = \boldsymbol{H}\boldsymbol{H}^{H}$, asymptotically tends to a diagonal matrix for certain "nice enough" channel conditions as

$$\lim_{M \to \infty} \frac{1}{M}\boldsymbol{H}\boldsymbol{H}^{H} \to \boldsymbol{I}_K. \tag{5.11}$$

The property of diagonal dominance of the Gram matrix in massive MIMO systems can be utilized by all the three previously mentioned QRD approaches. However, Givens rotation reaps the most gains in terms of complexity reduction and is also favorable in hardware [49].

Givens rotations has the capability of zeroing selected elements of a matrix [42], by performing matrix operations given as,

$$\begin{bmatrix} c & s \\ -s^* & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}, \tag{5.12}$$

where,

$$c = a/r$$

$$s = b^*/r, \tag{5.13}$$

and a, b are real and complex numbers, respectively, and $r = \sqrt{a^2 + |b|^2}$.

The selective zeroing, in Givens rotation is performed in an orderly fashion. Starting from the first column, all the elements below diagonal element are zeroed. Then this process is repeated by moving to next column and so-on, resulting in an upper triangular matrix. In case of massive MIMO, the pivot or the diagonal element is dominant, which would result in a scenario where $a \gg |b|$. Also, $\boldsymbol{Z}$ is a Hermitian matrix with real diagonal elements. In this work, an approximate Givens rotation based QRD is developed, by leveraging on these properties, to lower the computational complexity. One such simple approximation technique is to set

$$\begin{aligned} c &\approx c_{\text{const}} \\ s &= b^*/a, \end{aligned} \tag{5.14}$$

where c_{const} is a constant, as illustrated in Fig. 5.3. It can be observed that as the value of pivot (a) becomes dominant, the required rotation to nullify b is smaller, and the resulting vector length is also close to a.

The reduction in computing Givens matrix is evident from above equation, however, there is further gain achieved from the row operations as well. This is mainly due to the fact that in hardware, a constant multiplication is implemented using shifts and adds, which results in lower area costs. In the following, we discuss the performance of the Givens rotation approximation along with analysis of the effect of c_{const}. Latter look into the complexity reduction.

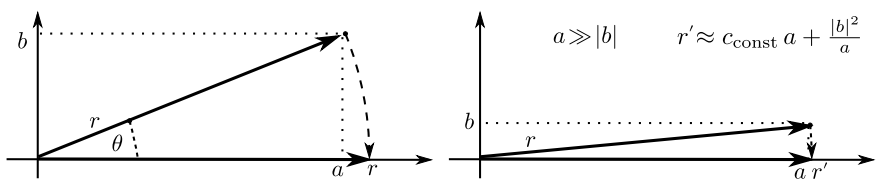
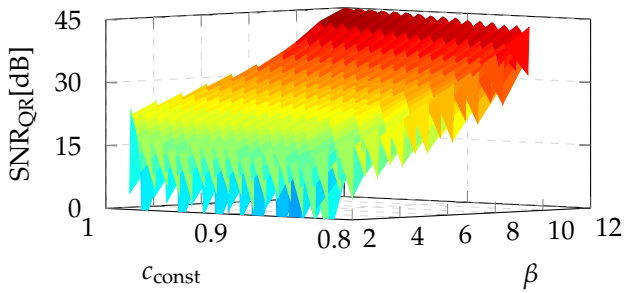
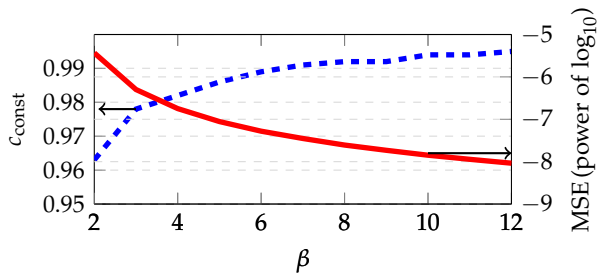


Figure 5.3. Illustrates standard Givens rotations and the low complexity approximation which can be performed when there is a dominant pivot. The larger the dominant pivot, smaller is the required rotation.



(a) QRD performance for different β and c_{const}.



(b) Value of c_{const}, and MSE reduction with increasing β.

Figure 5.4. Performance of approximative QRD.

PERFORMANCE ANALYSIS

To measure the performance of the approximation, the following yardstick is used

$$\text{SNR}_{\text{QR}} = \frac{||\boldsymbol{Z}^{-1}||_{\text{F}}^2}{||\boldsymbol{Z}^{-1} - \boldsymbol{R}_{approx.}^{-1}\boldsymbol{Q}_{approx.}^{H}||_{\text{F}}^2}, \tag{5.15}$$

where denominator term is the inversion MSE and $||\cdot||_{\text{F}}$ is the Frobenius norm. The reasoning behind this is that the approximate QRD is used to perform the matrix inversion and hence compared with the exact inverse.

The performance of the approximation is mainly dependent on two intertwined factors. Firstly, it depends on the ratio of the number of antennas at BS to that of MS *i.e.*, $\beta = M/K$. For an increasing β, the diagonal elements of matrix $\boldsymbol{Z}$ gets dominant, enabling the approximation described in (5.14) and Fig. 5.3. Furthermore, performance depends on the selected value of c_{const}, which is in the range

$$0 \leq c_{\text{const}} \leq 1\ , \tag{5.16}$$

for all values of a and b. The above bound is derived from the expression

$c = 1/\sqrt{1+|b|^2/a^2}$, which is a rearrangement of (5.13), and the fact that for a Gram matrix ($\boldsymbol{Z}$) the diagonal elements (a) are always real. It is evident that if the diagonal elements of $\boldsymbol{Z}$ get more dominant, it is more effective to select c_{const} closer to 1.

The performance analysis of an approximate QR is shown in Fig. 5.4. The 3-D plot shows the SNR_{QR} for different c_{const} and β. As β increases, the approximation in general performs better. However, it is important to select c_{const} as close as possible to the actual c. To assist in this selection, the value of c which yields the best SNR_{QR} (or lowest MSE) for different antenna ratios β is shown in the second plot of Fig. 5.4. The plot further confirms that as β increases c_{const} gets closer to 1. Also, the MSE decreases with an increase in β, since diagonal elements of $\boldsymbol{Z}$ become more dominant, which in turn lowers the variations in c. From a hardware perspective, a Look-Up Table (LUT) with c_{const} for a range of β can be implemented. This would translate to different instances of constant multipliers, sharing some of the adders. However, a single value of c_{const} can be chosen based on the system configuration, e.g., if the system is operating with β more than 8 most of the time, $c_{\text{const}} = 0.991$ would provide SNR_{QR} above 35 dB.

COMPLEXITY ANALYSIS

The algorithm that performs the proposed low-complexity approximate QR using Givens rotation is shown in Alg5.1. After the initialization, the two *for-loops* run through the matrix in a triangular form. In each inner iteration, Givens coefficients are computed, by updating variables a and b, after which the length r is computed, and followed by s and c. The Givens rotation coefficients are then used to perform row operations. There are two row operations, first the i-th row is updated (*i.e.*, $\boldsymbol{R}(i,:)$) followed by the j-th row. For each of these row operations, elements from the i-th column to the K-th (last) column of these rows are updated. The upper triangular matrix is formed in-place after completion of the two nested *for-loops*. The vector rotations leverage on the already computed Givens coefficients, consequently, avoiding to generate $\boldsymbol{Q}$ explicitly.

Computing the Givens coefficients, r requires three multiplications, since a is always real, and the square root operation is merged with the division as a single operation attributed to the Newton-Raphson method [50]. The computation of c becomes obsolete, *i.e.*, $c = c_{\text{const}}$, which saves 2 multiplications. Moreover, during the row updates the complexity is reduces from $8(K-i+1))$ to $4(K-i+1)$. This is because c is a constant, and realized by shift and add instruction synthesized logic. Overall this translates to a complexity reduction from $\mathcal{O}(\frac{8K^3}{3}+\frac{9K^2}{2}-\frac{97K}{6})$ to $\mathcal{O}(\frac{4K^3}{3}+\frac{3K^2}{2}-\frac{31K}{3})$, which is around 50% less for reasonable values of K. The complexity reduction of

Algorithm 5.1: Approximative QRD, the last two operations are for the user vector rotations.

	// Initialization per channel realization	$\mathcal{O}$
1	$\boldsymbol{R} = \boldsymbol{Z}$	
2	$\boldsymbol{u} = \boldsymbol{s}$	
3	**for** $i = 1 \rightarrow K-1$ **do**	
4	**for** $j = i+1 \rightarrow K$ **do**	
	// Compute Givens Rotations	
5	$a = \boldsymbol{R}(i,i)$	
6	$b = \boldsymbol{R}(j,i)$	
7	$r = \sqrt{a^2 + \lvert b \rvert^2}$	3
8	$s = b^*/r$	3
9	$c = \mathrm{C}_{\mathrm{val}}$	
	// Update rows	
10	$\boldsymbol{R}(i,i{:}K) = c\boldsymbol{R}(i,i{:}K) + s\,\boldsymbol{R}(j,i{:}K)$	$4(K-i+1)$
11	$\boldsymbol{R}(j,i{:}K) = -s^*\boldsymbol{R}(i,i{:}K) + c\,\boldsymbol{R}(j,i{:}K)$	$4(K-i+1)$
	// vector rotation of user data	
12	$\boldsymbol{u}(i) = c\boldsymbol{u}(i) + s\boldsymbol{u}(j)$	2
13	$\boldsymbol{u}(j) = -s^*\boldsymbol{u}(i) + c\boldsymbol{u}(j)$	2
14	**end**	
15	**end**	

the proposed method along with the performance is compared with other approaches in the following.

ALGORITHM COMPARISON

In Table 5.3 the complexity of inversion for different algorithms along with the user data vector operations is described. In particular, comparison is done with the Cholesky Decomposition (CD) [42] inversion approach, and the previously described approximate NS based inversion. The algorithms are classified into implicit and explicit approaches, where the former method does not compute the final inversion but indirectly performs the inverse on the data vectors. Operations per-channel realization involve the decomposition and post-processing, which needs to be performed when the channel matrix changes. The post processing in case of QRD and CD is the inversion of an upper triangular matrix, which has an $\mathcal{O}(2K^3)$ complexity. The vector operations also play an important role in the complexity, especially when the channel is almost constant over a range of sub-carriers. This would then result in lowering the number of matrix inversions, which is performed only at regular sub-carrier intervals (N_{int}). For modified-QRD, the complexity of

Table 5.3. Detailed complexity for different inversion algorithms.

Algorithm	type*	per-channel realization: decomposition process	per-channel realization: post-process	per-user data: vector operations
CD	I	$\frac{2K^3}{3} + 4K^2 + 8K$	$2K^3$	$4K^2 + 4K$
NS#	E	$8K^2 + 4(L-1)K^3$	-	$4K^2$
QRD	I	$\frac{8K^3}{3} + \frac{9K^2}{2} - \frac{97K}{6}$	$2K^3$	$6K^2 - 4K$
Modified QRD	I	$\frac{4K^3}{3} + \frac{3K^2}{2} - \frac{31K}{3}$	$2K^3$	$4K^2$

* Type of inversion either E - explicit or I - Implicit.
# L is iterations in Neumann series [36].

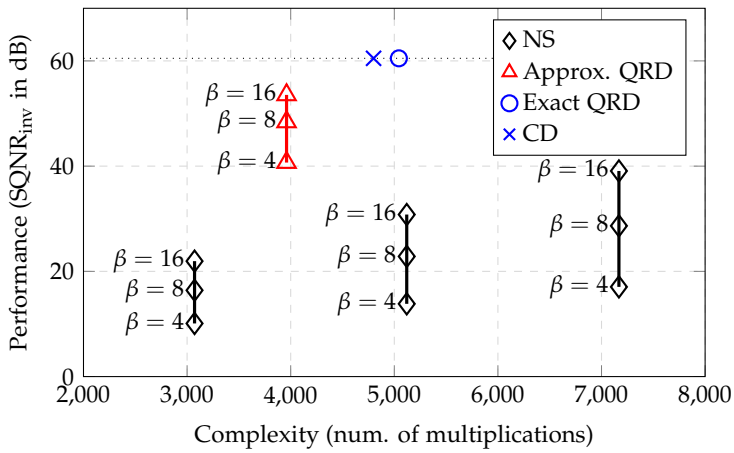


Figure 5.5. Performance vs complexity for algorithms with 12-bit precision, different β and iterations (L), where $\text{SQNR}_{\text{inv}} = 10\log_{10}\left(\frac{||\mathbf{Z}^{-1}_{Float}||^2_{\text{F}}}{||\mathbf{Z}^{-1}_{Float} - \mathbf{Z}^{-1}_{Approx.}||^2_{\text{F}}}\right)$.

vector operations is also lowered compared to standard QRD. Consequently, the proposed approximative QRD not only has a lower processing complexity, but also performs better with increasing sub-carrier intervals.

The detailed analysis depends on K and the coherence bandwidth, however, as an illustration and in line with the massive MIMO (LuMaMi) testbed frame-structure [34], we choose the sub-carrier interval as $N_{\text{int}} = K$ and $K = 8$ for analysis in Fig. 5.5. The SQNR performance serves as an illustration of the accuracy of algorithms with respect to exact inversion. Later in this section, system level performance in terms of BER is presented. The NS improves in performance with increasing β and iterations (L), having the drawback of a higher complexity when high performance is required. Both exact QR and CD have fixed performance which is the benchmark, however, they have higher

complexity. Contrarily, the complexity of a modified QRD is lower with a robust performance, which improves with an increased β. This complexity reduction along with high performance makes the proposed modified QRD a promising candidate for pre-coding in massive MIMO. The next section details the hardware architecture, which utilizes the proposed algorithm.

5.2.2. HARDWARE ARCHITECTURE

Systolic array consist of a homogeneous hardcoded network of nodes or Processing Elements (PEs), with each PE usually performing the same sequence of tasks. Due to the homogeneity, these architectures are easily scalable and have a relatively low design time. For the downlink pre-coding, systolic array are used for all stages as shown in Fig. 5.6 and described as follows:

- The first operation is to perform matrix multiplication to generate $\boldsymbol{Z}$. Several systolic arrays for matrix multiplication are described in the literature [51], [52]. In the case of generating Hermitian or Gram matrix, the same systolic array can be used, but with only half the PEs in a triangular form. This is achieved due to the symmetrical property of Hermitian matrix. In [52], both K^2 (2-D) and K (1-D) systolic arrays are described, which has a time complexity of K and K^2, respectively. In this work, we employ a 2-D systolic array with $K^2/2 + K$ PE, resulting in a time complexity of K.

- The second operation is to triangularize the Gram matrix, which is performed by a 2-D triangular systolic array. The concepts for the triangularization using systolic arrays was proposed in [53], and we utilize this to perform QRD. The approximative QRD can be leveraged to either lower the number of multipliers (gate count) or lower clock cycles (latency).

- After the QRD, the user data vector is multiplied with the unitary matrix

$$\boldsymbol{u} = \boldsymbol{Q}^H \boldsymbol{s}. \tag{5.17}$$

  Generating the unitary matrix $\boldsymbol{Q}$ is an expensive procedure in hardware. This would require another systolic array with K^2 nodes. However, to reduce the gate count a 1-D systolic array is proposed, which performs the rotation operations on the data vectors (implicit). The Givens co-efficient are buffered for each PE and the sequence of rotations are matched to that of QRD.

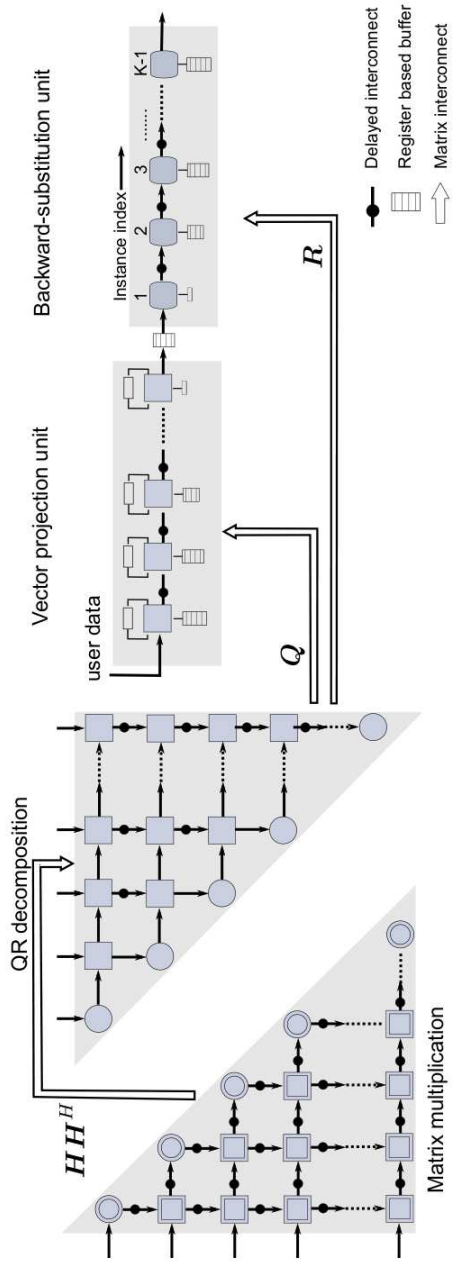


Figure 5.6. Top level architecture of QRD based pre-coder.

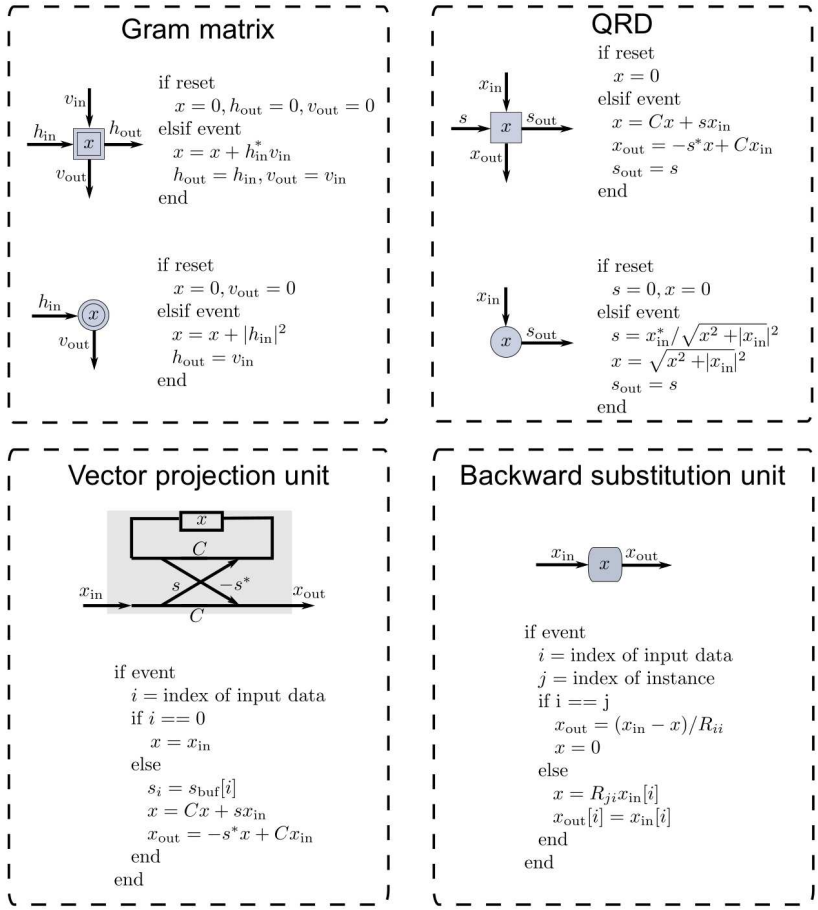


Figure 5.7. Functionality of the processing elements.

- After the rotation, the vector is multiplied with the triangular matrix

$$\boldsymbol{v} = \boldsymbol{R}^{-1}\boldsymbol{u}. \tag{5.18}$$

To avoid explicitly computing the inverse of a triangular matrix and then performing the matrix-vector multiplication, we employ a backward substitution based linear systolic array.

All these operations are envisioned to run in parallel on different sub-carriers to fully utilize the hardware.

PROCESSING ELEMENTS The four highly scalable modules consist of homogeneous PEs described in Fig. 5.7. The PEs share the data and control signals with each other by a hardcoded wire interconnect. Some of the interconnects also act as delay elements, these are registers along with a counter, if multi-clock delays are required. The interconnect between modules are large arrays, which pass the whole matrix simultaneously.

The processing nodes for the matrix multiplication are of two types, boundary nodes (circular) and internal nodes. Both the PEs have a MAC unit internally and are initially reset. The internal node has both horizontal and vertical input-output connections and accumulates the product of incoming data. The boundary nodes perform an accumulate operation as well as a directional shift, *i.e.*, the horizontal inputs are pushed out vertically. A scheduler ensures that each row multiplies with the Hermitian of itself and other rows.

The QRD module although has a similar triangular form as Gram matrix generation, has different PEs, interconnects and scheduling. The boundary nodes perform the computation of Givens rotation coefficients and broadcast it to internal nodes in the same row. The internal elements perform the row operation in parallel, and results are pushed to lower rows.

The vector projection unit performs (5.17) on a streaming user data vector. The matrix $\boldsymbol{Q}$ is not explicitly computed, instead Givens rotation is performed on the data vector in the same sequence. This requires storing the Givens rotation coefficients in buffers with size of the buffers decreasing from $K-1$ to 1. The user data vector $\boldsymbol{s}$ is streamed element-wise into the linear systolic array. The first element entering the PE is stored in the register ($x = x_{\text{in}}$, see pseudo code in Fig. 5.7). The remaining elements of the vector are rotated with the corresponding buffered coefficients. The elements after rotation are streamed to the next PE, until the $(K-1)$-th PE. The final resulting vector $\boldsymbol{u}$ is in the corresponding registers (x) in the systolic array. To pipeline the operations further, it is necessary to store the resulting vector elements in a buffer. This frees up the PEs to operate on the next vector stream. Apart from pipelining the operations, buffering is used to reverse the order of data stream.

The final systolic array performs (5.18) using the standard backward substitution method. The elements with lower indices than the instance index are multiplied with the corresponding row elements of matrix $\boldsymbol{R}$, and thereafter, accumulated (in x, see Fig. 5.7). A division operation is performed when the element index is equal to the instance index, resulting in the solution $\boldsymbol{v}$. The result element is streamed out in-place to the next PE. The higher indices compared to the instance index are not processed, and are passed through to the next PE.

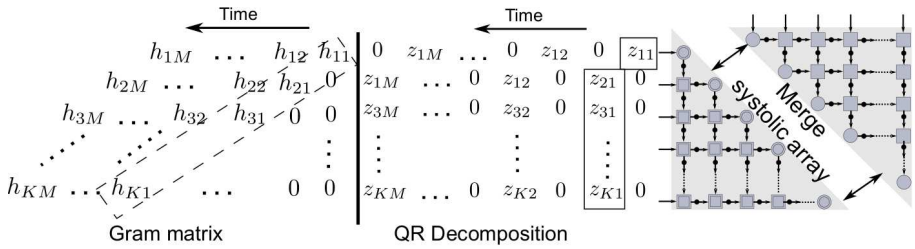


Figure 5.8. Scheduling schemes for Gram matrix generation and QRD.

SCHEDULING A generalized *event* was used to describe the pseudo code for the PE in Fig. 5.7. The *event* can be either time, valid signal, or 2-way handshake protocol based hardware triggers. In this architecture, we use valid signals to start operations and time-based scheduling to process sequential data. In Fig. 5.8 the scheduling for both Gram matrix generation and QRD is shown. For the Gram matrix generation, after the valid start signal is asserted, the elements of the first row of $\boldsymbol{H}$ ($h_{11}, h_{12}...h_{1M}$) are pushed into the systolic array. The successive rows are delayed by one time unit (latency of nodes) with respect to the previous row. Hence, the last row's first element (h_{K1}) enters the systolic array after $K-1$ time units. The total time unit from the first data (h_{11}) into the systolic array till the last one (h_{KM}) is $K+M$ time units and will require in total $2K+M$ units to move through the systolic array.

The QRD has a standard scheduling scheme, where the pivot element (z_{11}) is first provided to the boundary node. The boundary node will compute the rotation coefficients which are then transmitted to the internal nodes. This mechanism necessitates that the non-pivot elements be delayed both when entering the systolic array and internally when moving to the next row. Although there is different scheduling for both operations, the data flow and architecture has some similarities which can be exploited to merge them to reduce silicon cost. In the next section, the implementation details of the proposed algorithm based on the generic hardware architecture described in this section is presented.

5.2.3. IMPLEMENTATION DETAILS

Until now, the described algorithm and hardware architectures were platform independent and generic. In this part implementation details of the fabricated chip in 28 nm FD-SOI CMOS technology are described. This implementation involved various techniques, e.g., word-length optimizations, hardware re-usability, folding, testing strategies (covered in Chapter-2). The implementation is driven by area and I/O constraints, but word-length and folding factor can be easily modified without major changes to the architecture (or

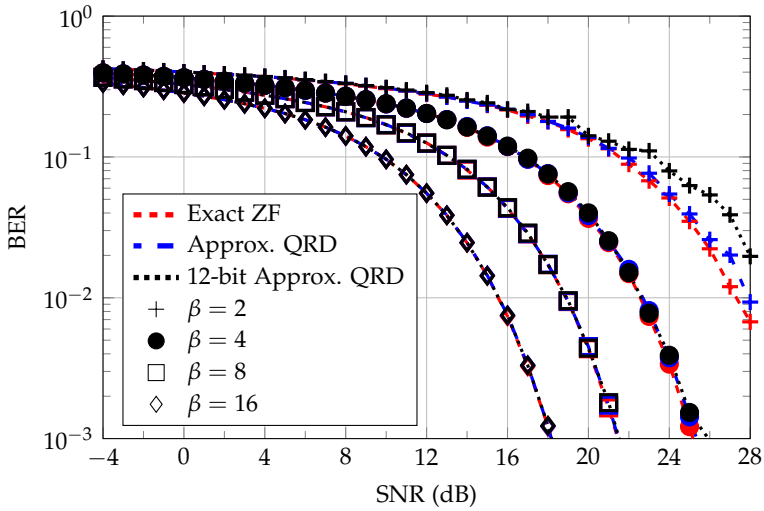
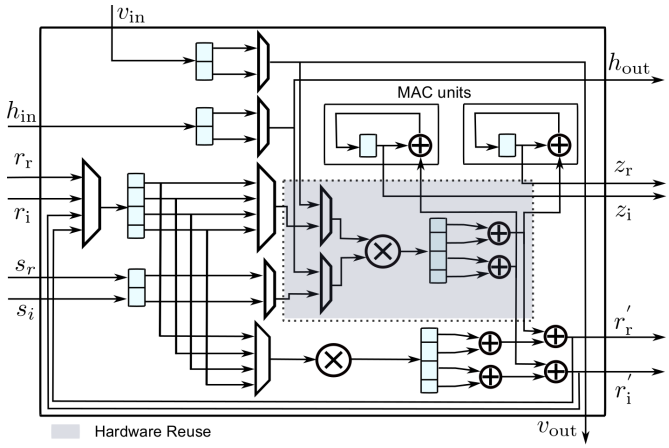


Figure 5.9. BER performance for 12-bit fixed point approximative QRD, for a 256-QAM uncoded system.

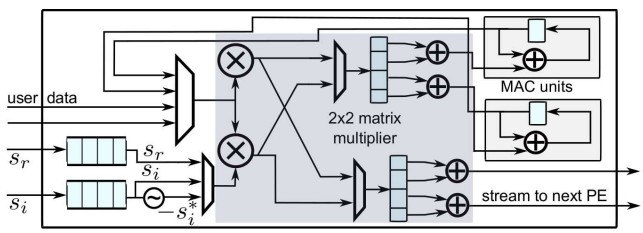
code). Some of the implementation details and techniques are mentioned in the following.

WORD-LENGTH OPTIMIZATION The internal word-length is a critical aspect for efficient hardware implementation and system performance. Fig. 5.9 shows the BER performance with different antenna configurations (β) for exact ZF, approximate QRD, and fixed-point approximate QRD. The performance loss of approximative QRD reduces rapidly with increasing β. Simulations showed that a 12-bit word-length provided almost negligible performance loss when targeting 10^{-3} BER for a 256-QAM uncoded system.

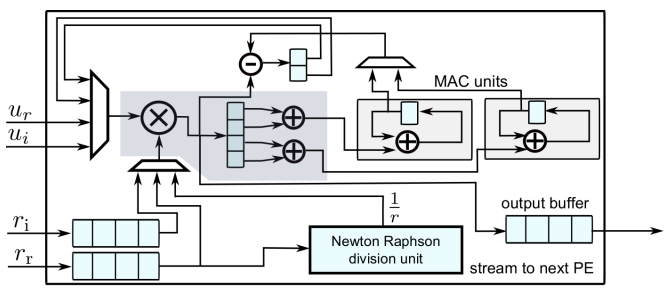
HARDWARE RE-USABILITY To reduce the hardware cost of the implementation, Gram matrix and QRD systolic arrays are merged, see Fig. 5.8. To support this the PEs have two modes. Since both modes are not active simultaneously, the hardware resources are shared between them to reduce cost. The dotted box in Fig. 5.10(a) shows the hardware resources which are reused for internal nodes. This merging reduces the hardware cost by 40%. Moreover, the scheduling schemes for both remain the same, albeit not simultaneously. This hardware re-usability can also be interpreted as a 20% overhead in the QR systolic array to support matrix multiplication operations.



(a) Implementation of internal PE for performing both QRD and Gram matrix generation by re-using hardware.



(b) PE for vector rotation module.



(c) PE for backward substitution module.

Figure 5.10. A highly time-multiplexed implementation.

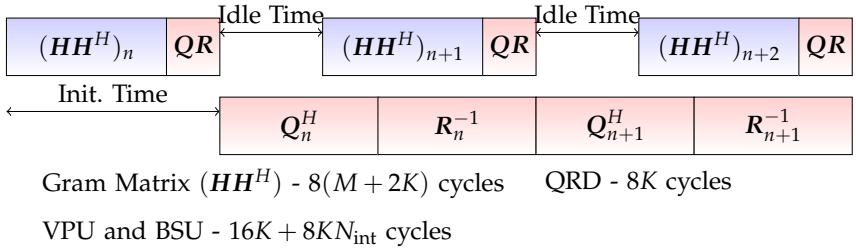


Figure 5.11. Schedule for performing operations over sub-carrier with $M = 32$, $K = 8$, $N_{\text{int}} = 10$ configuration.

TIME MULTIPLEXED ARCHITECTURE The architecture in Fig. 5.6 describes the operations (pseudo-code) for different PEs. To reduce hardware cost, an aggressive time multiplexed design is implemented. The computation and hardware cost is mainly dominated by multipliers. Consequently, the PEs were implemented by employing a single multiplier and time-multiplexing, resulting in a high utilization factor.

In Fig. 5.10(b), the PE for performing vector projection is shown. The main operation of Vector Projection Unit (VPU) is a 2×2 matrix multiplication. Since the diagonal elements are constant in (5.12), the total multiplications are performed in 8 clock cycles using a single multiplier. A MAC unit is employed to perform the rotation operation and to store the pivot element. The final result after all the rotation operations is in the MAC and is copied into the buffer.

The PE for Backward Substitution Unit (BSU) also employs a folded complex multiplier unit as shown in Fig. 5.10(c). The sum of products of row elements of $\boldsymbol{R}$ and the data vector ($\sum_{j=i+1}^{K} \boldsymbol{R}_{ij}\boldsymbol{v}_j$) is implemented using folded multiplier and MAC units. The result is deducted from the data vector and then solved by performing a division, *i.e.*, $\boldsymbol{v}_i = (\boldsymbol{u}_i - \sum_{j=i+1}^{K} \boldsymbol{R}_{ij}\boldsymbol{v}_j)/\boldsymbol{R}_{ii}$. The division is performed using Newton-Raphson, with initial estimate computed by first order curve fitting. High hardware efficiency is attained by reusing the multiplier for performing division. This is possible due to the fact that each BSU PE only requires inverse of the corresponding $\boldsymbol{R}_{ii}$, and also leveraging on the latency of the VPU when $\boldsymbol{R}$ changes.

PIPELINING The modules are fully pipelined to handle sub-carrier processing, *i.e.*, during the n-th tone QRD, the vector projection and backward substitution units operate on user data of previous tones. The first two matrix operations are performed in parallel with the two vector operations, as shown in Fig. 5.11. This is possible by introducing buffers storing results of the previous QRD. In total K^2 register are required in both units to store the QRD.

The latency of the VPU and the BSU is $16K$, and it processes 1 element every 8 cycles. Similarly, the Gram matrix multiplication and QRD in total takes $8M + 24K$ cycles, see Fig. 5.11. Hence, based on the system configuration, either the QR unit or the VPU along with the BSU has a higher latency.

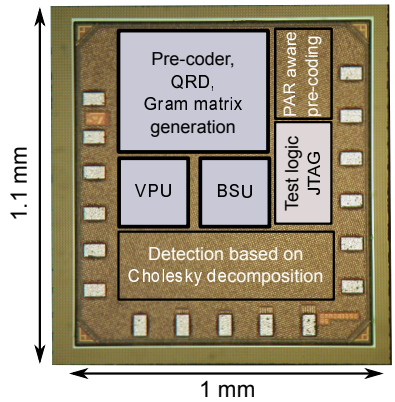
LOW POWER TECHNIQUES Due to support for different system configurations, modules can be in an idle state. To lower the power consumption during an idle state, a clock-gating with 2-way handshake protocol is implemented. Based on functionality two clock-gating units are employed, one for the QRD and another for the VPU and the BSU. Furthermore, the implementation exploits body biasing to either lower power consumption by performing RBB or improve performance by FBB.

TESTABILITY To perform the functional verification and measurements a test sub-system based on Joint Test Action Group (JTAG) is implemented [54]. In addition to high flexibility and debugging features, the JTAG interface requires only 5 Input Output (IO) pads. A fully functional verification can be performed by shifting stimulus into scan-chain and then triggering the Design Under Test (DUT). After completion of processing, the results are captured and shifted out for comparison. Furthermore, a Built-In Self-Test (BIST) controlled by JTAG is implemented, which is used to perform validity checks and power measurements. Further details on the test subsystem are provided in Appendix-B.

5.2.4. HARDWARE RESULTS AND DISCUSSION

Fig. 5.12 shows the chip micrograph containing different modules along with the specification. The gate-count for the individual modules is shown in Fig. 5.13(a), with break down of sequential and combinational logic. The Gram multiplication module is not implemented in the chip but shown as a reference for the gains achieved by merging with the QRD module. This huge reduction in gate-count (100 k) is a design choice aimed at lowering hardware cost. However, based on system requirements separate instances can be envisioned with minimal changes to the architecture.

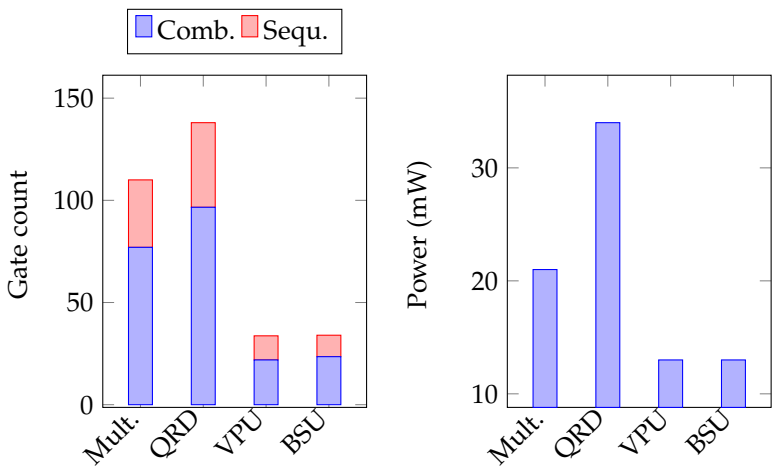
The power consumption for various modules is shown in Fig. 5.13(b), with the QRD module having higher power requirement due to complex control logic, computations and buffers. The performance of pre-coding along with the QRD unit was functionally verified for multiple frequencies, core and body-bias voltages, as shown in Fig. 5.14. The maximum measurable frequency is 300 MHz, due to lab setup, which creates a rooftop in the plots. The chip is functional from 0.5 V to 1.0 V core supply, and at lower voltage FBB is used to improve performance at the cost of increased power consump-



Fabricated chip details	
Technology	28nm FD-SOI
Total die area	1.1 mm x 1 mm
Supply voltage	0.5 V to 1.0 V
Body-bias voltage	-0.3 V to 1.2 V

8x128 Massive MIMO pre-coder	
Gate count (QRD)	138 k
Max. freq.	300 MHz
Clock cycles	64 or 128
Power	31 mW

Figure 5.12. Chip photo, with marked modules active during pre-coding.



(a) Gate Count. (b) Total Power.

Figure 5.13. Results for individual modules. Gram matrix multiplication is merged with QRD, and is shown here to highlight the reduction in gate count.

tion. The implementation also supports RBB of up to 0.3 V, which increases the threshold voltage of transistors, in turn lowering static power consumption at the cost of performance. The maximum clock frequency of 300 MHz (minimum latency) is achieved for different core and body-bias voltage combinations. Fig. 5.15(a) shows the measured minimum latency and corresponding minimum energy per 8×8 QRD operation for different core supply voltages. Applying V_{DD}=0.9 V and FBB of V_{BB}=0.2 V achieves a minimum la-

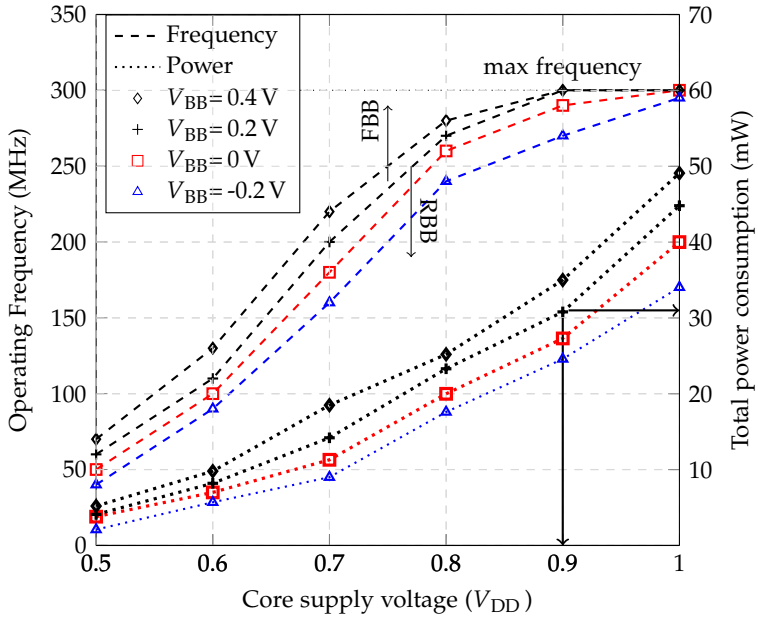
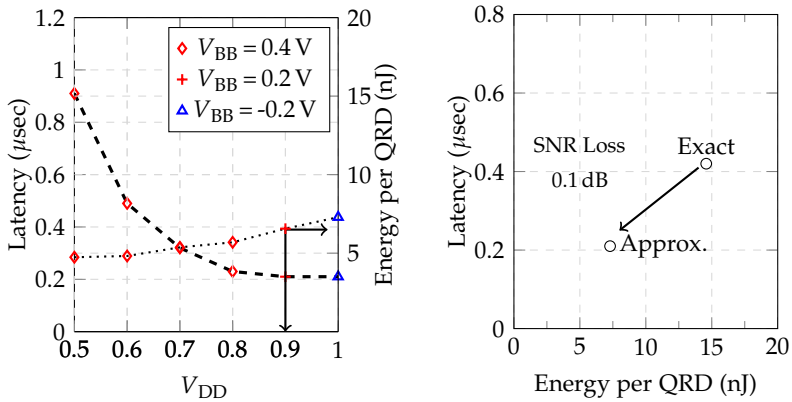


Figure 5.14. QRD measurements for different core voltages and body-bias.



(a) Min latency and energy per QRD (b) Exact vs aprrox QRD gains.

Figure 5.15. Measurement results.

tency and energy of .21 μs and 6.56 nJ, respectively, per QRD. The high gains when opting for approximative QRD for minimal performance loss is shown in Fig. 5.15(b).

Table 5.4. Hardware Results Comparison.

	This Work		Chun	Rakesh	Shabany	Huang	Chiu
	Aprox.	Exact	[55]	[47]	[48]	[49]	[46]
Matrix Dimension (K×K)	8×8		8×8	4×4	4×4	4×4	4×4
Algorithm	Adp. GR		GR	HH	Hy.	GR	GS
Gram Matrix Generation	✓		✗	✗	✗	✗	✗
Technology [nm]	28		90	65	130	180	90
Gate Count [kGE]	138		300	328	36	111	505 [a]
Clock Freq. [MHz]	300		55	72	278	100	114
Power [mW]	31		27.2	-	48.2	319	56.8
QRD Latency [Cycles]	64	128	-	1	40	4	-
Throughput [M-QRD/s]	4.7	2.34	1.1	72	6.95	25	28.5
N.E.E [nJ/QRD] [b]	6.56	13.1	-	-	11.9	21.9	-
N.H.E [QRD/s/GE][c]	34.1	17	3.67	27.4	24.13	28.2	7.1

[a] Includes SIC based detector, QRD takes around 65% according to micrograph.
[b] Normalized Energy Efficiency = $(\text{Power}/(\text{Throughput} \times (8^3/K^3))) \times 28/\text{Tech}$.
[c] Normalized Hardware Efficiency = $(\text{Throughput} \times (K^3/8^3))/\text{Gate-Count}$ as in [55]

As an overview, comparison with different QRD implementation is shown in Table 5.4. It is important to note that our design performs other functionalities like Gram matrix generation and approximate/exact QRD as well. The results of QRD implementations may be targeted for different system scenarios and optimization goals, hence, the focus of the comparison is to highlight the gains of approximative QRD. Moreover, the matrix dimensions for most of the implementations is lower, and the complexity grows as $\mathcal{O}(K^3)$. Therefore, a metric Normalized Hardware Efficiency (NHE) is defined based on the growth of the complexity with matrix dimension, which is normalized to a 8×8 matrix, in line with [55]. The ability to reconfigure QRD to perform Gram-matrix generation is an optimization choice to reduce the total system level gate-count. This additional functionality restricts utilization of the highly efficient custom PE based on COordinate Rotation DIgital Computer (CORDIC) computations as in [49], [48]. Although this restriction results in exact QRD having a slightly lower NHE, it is still comparable to other reported implementations. The approximative QRD lowers the total complexity, resulting in an improved throughput, which in turn improves NHE. For massive MIMO systems, this approach is highly beneficial, since the performance loss is minimal for a large improvement in hardware efficiency. The implementation has the highest NHE of the once compared. However, the proposed approximation is generic and can also be incorporated into other CORDIC architectures.

The vector processing blocks (VPU and BSU) achieve a maximum throughput of 37.5 MSamples/s when operating at 300 MHz with total power consumption of 26 mW. This results in hardware and energy efficiencies of 0.56

MSample/s/kGE and 0.7 nJ/Sample. The vector processing block supports up to 256-QAM, which results in data-rate of 300 Mbits/s. Higher throughputs can easily be achieved by unfolding the PE, however, a folded design was opted in line with [34], where the IO bottle-neck lead to multiple instances of the pre-coder to operate on separate sub-carrier groups. The highly time-multiplexed design achieved high hardware efficiency by exploiting the properties of matrices in massive MIMO at algorithm level and optimizing circuits to adaptively perform pre-coding based on channel conditions.

Before moving to the next hardware platform some takeaways from approximative QRD based ASIC implementation are as follows:

- Approximative QRD provides better performance and robustness compared to NS. However, more effort at the hardware level is required to exploit these gains.
- Architectural optimizations in ASIC provide high efficiency. This work has the highest reported QRD efficiency.
- Flexibility, although limited in ASIC, is still reasonable, e.g., implementation supports a fixed range of users $K \leq 8$, adaptability to perform exact or approximative QRD.
- The major price-to-pay is the high design time, mainly consisting of back-end flow, verification (becomes even more critical), and developing test infrastructure.

5.3. RECONFIGURABLE PLATFORM

Previous two sections presented a hardware-centric approach to meet the massive MIMO pre-coding requirements. In this section a software-centric approach is described, mainly targeting a reconfigurable vector processor. The architecture of the vector processor is not part of this work and is frozen *i.e.*, assumed to be an off-the-shelf vector processor. So we analyze scheduling strategies for a generic vector processor architecture.

5.3.1. VECTOR PROCESSOR ARCHITECTURE

In addition to throughput and power requirements, reconfigurability/flexibility is also becoming important design aspect. This is mainly driven by shorter time-to market and the need to support various standards. The most promising reconfigurable platforms are vector processor based architectures. These vector processors, when finely tuned for a particular domain, provide high performance, which is almost like ASIC [11], [56]. Interestingly, if one

ignores domain specific modifications, most of the vector processors have a very similar architecture. Fig. 5.16 shows a wireless communication baseband vector processor, in line with [11]. The vector core processor has N_{MAC} MAC units with 16-bit fixed point precision. This can be reconfigured to perform as $N_{\text{MAC}}/4$ complex-number MAC units, which is a typical mode for MIMO processing. The pre-processing module performs vector additions and scaling operations, which are useful for setting up the vectors before core processing. Post-processing also performs vector additions, shifting, and some specific operations like sorting, min/max. The final resulting vector is stored in the output registers, which can be written into data memory or sent back to input vector registers. The scalar processor is an off-the-shelf standard RISC processor (ARM, RISC-V) used to control/sequencing the vector operations. The scalar unit also consists of accelerators for division, square-root and trigonometric functions. To fully exploit all the MAC units in the vector core the following features are quintessential:

- Multi-stage computing: Most vector processing involves several tightly coupled operations, such as scaling, conjugation, offset, before and after vector computations. Mapping of these operations onto hardware require a Very Large Instruction Word (VLIW)-style multi-stage computation chain to accomplish several consecutive data manipulations in one single instruction.

- Flexible data access: A major design challenge is to rate-match the memory bandwidth with the vector core data consumption rate. For simple memory access schemes, the bandwidth of memory equal or greater than the vector size should suffice. However, special access

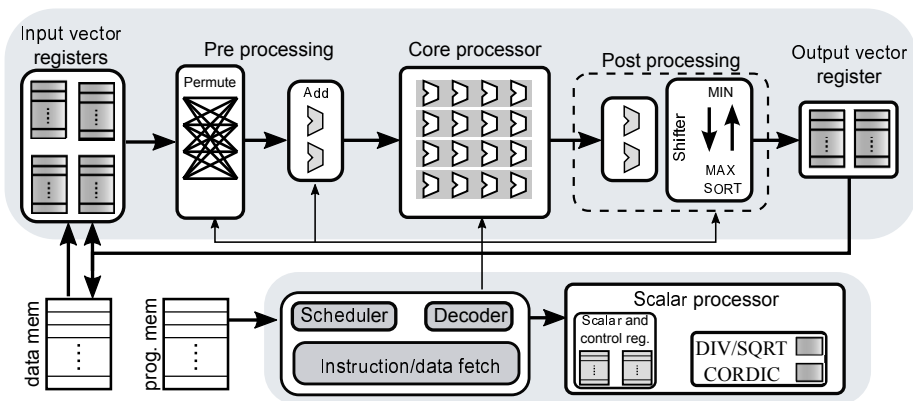


Figure 5.16. Reconfigurable vector processor architecture, based on [11].

patterns may create a bottleneck. One approach is to load data from memory and perform selection operations in the pre-processing using a permutation (mux) network. Another approach would be to push the complexity to a memory controller or DMA, which can handle various access patterns. Evaluation with different use-case kernels helps in exposing architectural bottlenecks, memory access pattern requirements etc. A well known approach to perform such analysis is to use the roofline models [57].

- Zero-delay loop control: Typically vector processing involves loops over the matrix dimensions. In the case of OFDM systems, the loops exist over sub-carriers as well. A zero-delay loop control lowers the execution latency by avoiding dedicated cycles for condition checks.

- Pipelining: High performance is achieved at a cost of a slight increase in latency by inserting pipeline registers. Performance increase is due to the shortening of the critical path, which allows for operation at higher clock frequency. Although it is beneficial and almost a must to pipeline a processor, it can lower MAC utilization if there are dependencies in the execution.

5.3.2. ALGORITHM AND SCHEDULING

Considering a generic vector processor with the features mentioned earlier, the challenge of mapping a massive MIMO pre-coder is to select appropriate algorithm and scheduling efficiently. In addition to complexity, memory access patterns is an important criterion for algorithm selection. Gram matrix multiplication is straight forward and hence the focus is on matrix inversion. The goal of approximate approaches used in previous sections was to lower complexity, thereby reducing hardware cost. However, for an off-the-shelf

Algorithm 5.2: MGS based QRD.

```
   // Iterate over matrix columns
1  for i = 1 → K do
2  |   r_ii = ||v_i||_2          // 2-norm of a column vector
3  |   q_i = v_i / r_ii          // Vector scaling operation
4  |   for j = i + 1 → K do
5  |   |   r_ij = q_i^* v_j      // Vector-dot product operation
6  |   |   v_j = v_j − r_ij q_i  // Vector update : scaling and subtraction
7  |   end
8  end
```

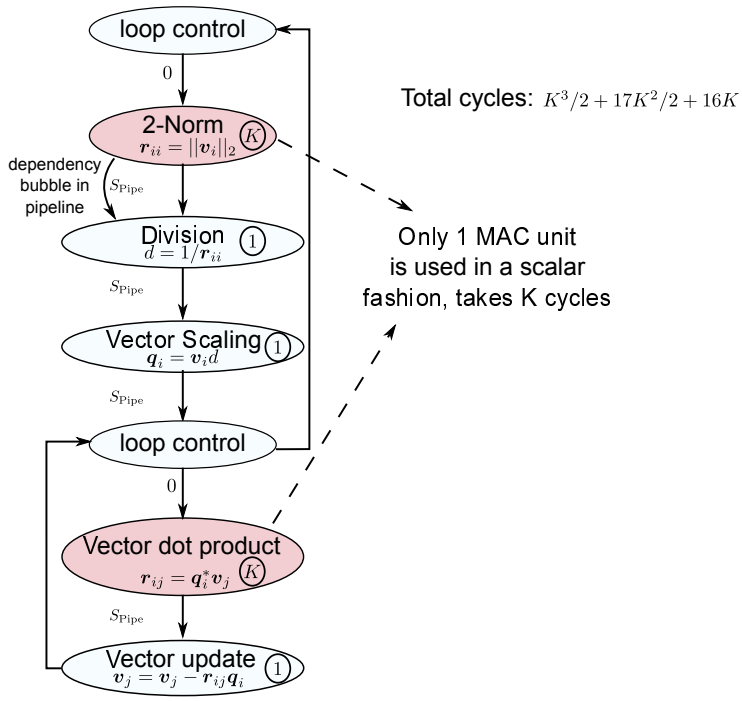


Figure 5.17. MGS QRD dependency and scheduling on a vector processor with deep (=S_{pipe} stage) pipeline. Straight forward MGS implementation has low MAC utilization.

vector processor the hardware cost is fixed. Hence, an exact inversion strategy was used, specifically Modified Gram-Schmidt (MGS) based QRD with its consistent memory access was used for performing matrix inversion.

The MGS based QRD algorithm is described in Alg.5.2. The outer *for-loop* iterates over each column, while the inner loop starts from the current column

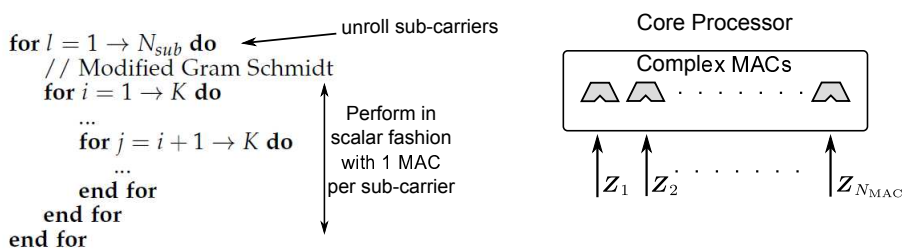


Figure 5.18. Unrolling over sub-carrier improves MAC utilization.

till the right most column. The memory accesses are simple and consistent, *i.e.*, only column-wise access are performed. The four main vector operations are vector 2-norm, vector scaling, vector-dot product, and vector-update. Vector scaling and update are operations which can be performed in parallel on each MAC separately. However, vector-dot product (and 2-norm) require an adder tree (reduction operation), which is typically not available, since it increases the critical path of the core. Therefore, these reduction operations creates a dependency bubble in the pipeline as shown in schedule Fig. 5.17. Also, the vector dot-product is now computed in a scalar fashion using one MAC hence requires multiple (K) cycles. Overall, a straightforward scheduling of the MGS will lead to low utilization of the vector core unit. For a vector processor with $N_{\text{MAC}} = 32$ MACs and $S_{\text{pipe}} = 8$ pipeline stages, 8×8 QRD would take around 930 cycles with only 13% core utilization.

An approach to avoiding such bottlenecks would be to change the architecture, *i.e.*, introduce hardware support for vector reduction operations like norm and dot-product. However, for an off-the-shelf vector processor, this fix is not an option. Thus, efficient scheduling based on architecture is very important for improving performance. Typically, in OFDM based communication systems there is another outer loop which runs over the sub-carriers. This can be exploited during scheduling to hide the bubble in the pipeline. Fig. 5.18 shows a scheduling scheme where multiple matrices are fed to each MAC unit to perform QRD in parallel. This can be envisioned as if each MAC is an independent scalar processor. The latency of each matrix increases. However, due to the parallelism, the overall throughput also increases. The number of clock cycles to perform QRD reduces from 930 to 160. The gains are mainly due to lowering the impact of the dependency bubble by performing QRD over more matrices in parallel. For a clock rate of 500 MHz, the efficient scheduling provides a throughput of 3.13 M-QRD/sec. The gate count and power consumption require a much more detailed analysis and implementation, which is not part of this work. On the other hand, the high performance, flexibility and software implementation offered by vector processor makes it an attractive platform for massive MIMO systems.

6

Hardware Impairments

In the previous chapter, low complex pre-coding algorithms which reduced the overall digital (signal processing) hardware cost, were presented. However, it is also important to consider the RF chains for an efficient massive MIMO implementation. One of the critical components of the transmitting RF-chain, both in terms of hardware cost and power consumption, are the Power Amplifiers (PAs). It would be desirable to have low Peak-to-Average Ratio (PAR), which would both reduce the hardware cost of the PA and, more importantly, improve its power efficiency. OFDM based systems, are known to suffer from high PAR. Hence, it requires a linear PA with high dynamic range to avoid in-band distortion and out-of-band components due to non-linearity and signal clipping. Linear PAs are much more expensive and typically have a lower power efficiency than their non-linear counterparts. To counter this, many techniques for handling high PAR in OFDM systems are described in the literature [58]. A well known and low-complex approach is the tone-reservation technique [59]. It relies on reserving bandwidth (around 20% for 10 dB reduction in PAR), which, also, reduces spectral efficiency significantly. The main reason for the impact on spectral efficiency is the fact that the bandwidth has a linear relation to capacity, since it is outside the logarithmic term in (2.2).

In massive MIMO there is inherently a large degree-of-freedom (due to the large number of antennas at the BS), that can be utilized to reduce the PAR. In the first section of this chapter, a novel low complexity PAR aware pre-coding is presented which reduces PAR by around 4 dB with only a 15% increase in pre-coding complexity compared to ZF. The idea is to reserve antennas (”antenna-reservation” analogy to ”tone-reservation”), which will be used to compensate for a (deliberate) clipping of the signals on the remaining antennas. In the second section, a stringent amplitude constrained (with only

phase changes) transmission scheme known as constant Envelope (CE) is presented. This allows using a highly efficient non-linear PA, since a very strict constraint on the amplitude (constant) results in a 0 dB PAR before Digital-to-Analog Converter (DAC). The last section of the chapter briefly looks into the effects of IQ-imbalance in massive MIMO.

6.1. PAR AWARE PRE-CODING

The narrow-band massive MIMO system model from the previous chapter is extended to an OFDM massive MIMO model, which is then followed by the description of the proposed low-complexity PAR aware pre-coding.

SYSTEM DESCRIPTION

The system model depicted in Fig. 6.1 is an OFDM-based multi-user massive MIMO system for downlink. The BS consists of M antennas serving $K(\ll M)$ single antenna users. The K user symbols on tone n, of a total of N tones, is represented by the $K \times 1$ vector $\boldsymbol{s}_n$. The user symbol normally has an allocation pattern, consisting of guard-bands (unused tones at the edges) and user data. In this work, we use ψ as the set of all data tones. The complementary set ψ^c hence contains the guard band tones, such that $\boldsymbol{s}_n = 0_{K\times 1}$ for $n \in \psi^c$.

The user symbols on the n-th tone are pre-coded as

$$\boldsymbol{p}_n = \boldsymbol{W}_n \boldsymbol{s}_n \,, \tag{6.1}$$

prior to transmission in order to cancel inter-user interference, where $\boldsymbol{p}_n$ is the pre-coded vector of size $M \times 1$ and $\boldsymbol{W}_n$ the corresponding pre-coding matrix for the n-th OFDM tone. In this study, we use a standard ZF linear pre-coder

$$\boldsymbol{W}_n \propto \boldsymbol{H}_n^H (\boldsymbol{H}_n \boldsymbol{H}_n^H)^{-1} \,, \tag{6.2}$$

where $\boldsymbol{H}_n$ is the n-th tone channel matrix of size $K \times M$. To satisfy an average power constraint $\mathbb{E}\{||\boldsymbol{p}_n||_2^2\} = 1$ on the pre-coded vector, a normalization factor is applied.

The overall input-output relation in the frequency-domain, after removal of the cyclic prefix, is described as

$$\boldsymbol{y}_n = \sqrt{P_n}\boldsymbol{H}_n \boldsymbol{p}_n + \boldsymbol{w}_n \,, \tag{6.3}$$

where P_n is the transmit power on tone n (constrained by total transmit power $P_T = \sum P_n$) and $\boldsymbol{w}_n$ contains unit-variance zero-mean complex Gaussian white noise elements.

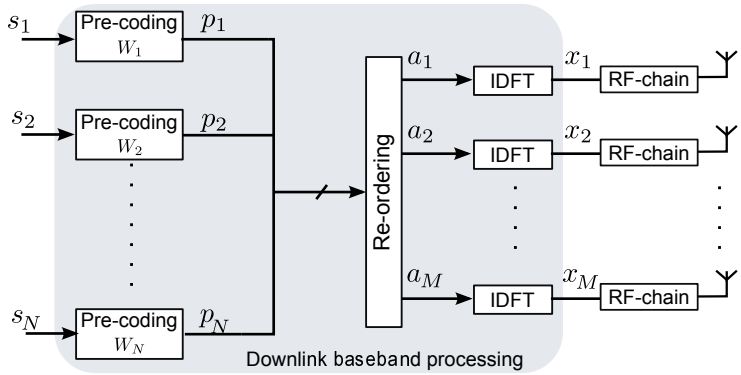


Figure 6.1. Massive MIMO system model for downlink. The PAR problem occurs after the IDFT, and is related to the pre-coding matrix and the input symbols.

To describe the transmitted signals in the time-domain, the pre-coded vectors $\boldsymbol{p}_n$ are reshaped as

$$[\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_M] = [\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N]^T, \tag{6.4}$$

where $\boldsymbol{a}_m$ is an $N \times 1$ vector containing the frequency-domain signal transmitted from antenna m and $(\cdot)^T$ denotes the transpose operator. An IDFT is performed on the transposed pre-coded signal as

$$\boldsymbol{x}_m = \boldsymbol{F}\boldsymbol{a}_m\,, \tag{6.5}$$

where $\boldsymbol{F}$ is the $N \times N$ unitary IDFT matrix, $\boldsymbol{x}_m$ the $N \times 1$ time-domain OFDM symbol transmitted on antenna m. With the addressed system specified, different approaches to handle PAR are described before introducing the new low-complexity PAR pre-coding approach.

6.1.1. PAR AWARE PRE-CODING TECHNIQUES

OFDM transmission causes the signal $\boldsymbol{x}_m$ in (6.5) to have a high dynamic range, due to the nature of the transformation. This would, in turn, require expensive linear PAs made to operate at a certain power back-off. Increasing the transmit power implies an increased probability of saturating the PAs and thus generation of both in-band distortion and out-of-band emissions. To avoid this, PAR reduction techniques are essential. The PAR for the OFDM symbol on antenna m is defined as

$$\mathbf{PAR}_m = \frac{||\boldsymbol{x}_m||_\infty^2}{||\boldsymbol{x}_m||_2/N}\,, \tag{6.6}$$

and the global PAR as

$$\mathbf{PAR}_{\text{gbl}} = \frac{||\bar{\boldsymbol{x}}||_{\infty}^{2}}{||\bar{\boldsymbol{x}}||_{2}/(NM)}, \tag{6.7}$$

where, $||\cdot||_{\infty}$ is the infinity norm, which is the vector element with maximum absolute value, $\bar{x}$ is an $NM\times 1$ vector containing all time-domain OFDM symbols transmitted on all antennas.

CONVEX OPTIMIZATION BASED PAR PRE-CODING In massive MIMO, the high degree-of-freedom can be used to select a frequency-domain transmission signal $\boldsymbol{a}_m$ with very strict requirements on the dynamic range of the corresponding time-domain signal $\boldsymbol{x}_m$, while at the same time canceling the multi-user interference. An example of this is the joint pre-coding, modulation, and PAR reduction (PMP) scheme presented in [60], where an optimization problem is formulated along the lines of

$$\underset{\bar{x}}{\text{minimize}} \;\; ||\bar{\boldsymbol{x}}||_{\infty} \;\; \text{subject to} \;\; \bar{\boldsymbol{s}} = \bar{\boldsymbol{G}}\bar{\boldsymbol{x}}. \tag{6.8}$$

where all time-domain symbols are stacked in $\bar{\boldsymbol{x}}$, all user data symbols, including nulls on guard-band frequencies, are stacked in $\bar{\boldsymbol{s}}$, and the structure of $\bar{\boldsymbol{G}}$ implements the constraints on no user interference and no energy in the guard bands [60].

Different methods to solve this optimization problem exist [61], including the method described in [60]. This approach gives an optimal solution with high control on the PAR levels. However, the complexity and the number of dimensions involved in such large scale optimization poses an implementation challenge in hardware. Moreover, the optimization problem is non-linear and depend on the user data, thus requires solving of (6.8) frequently, which implies a high computational latency.

PEAK CANCELLATION An approach that is contrary to the optimization problem (6.8) is clipping the signal peaks. Either the clipping is performed in the analog domain (*i.e.*, by the PA), which introduces both out-of-band components and in-band distortion, or it can be performed in the digital domain. The benefit of clipping in the digital domain is that it does not cause any out-of-band components, only in-band distortion which also includes guard band. The idea is to compute a signal $\boldsymbol{r}$, to be added to $\boldsymbol{x}$, that reduces the PAR,

$$\text{PAR}(\boldsymbol{x}+\boldsymbol{r}) < \text{PAR}(\boldsymbol{x}), \tag{6.9}$$

in line with the tone-reservation technique [59]. When the signal $\boldsymbol{r}$ is mapped to all tones it is known as the peak-cancellation technique. Remaining part of this section describes the proposed low-complex approach to reduce PAR in massive MIMO systems.

6.1.2. LOW-COMPLEX PAR AWARE PRE-CODING

Clipping the transmit signals is simple, but suffers from a certain amount of distortion. An approach to compensate this distortion is to dedicate a subset of the antennas which transmits signals to mitigate the resulting distortion. Hence, the overall system for the PAR reduction would be a set of antennas (χ) transmitting user data with the clipping technique, and the remaining antennas in the set χ^c are used to compensate this distortion as shown in Fig. 6.2. The peak-cancellation functional block of Fig. 6.2 basically imposes a saturation logic (clipping), based on the desired PAR level by applying

$$x^{'} = \begin{cases} x, & \text{if } |x| < T \\ Te^{j\angle x} & \text{otherwise,} \end{cases} \tag{6.10}$$

element-wise to the signals, where T is the clipping amplitude threshold and $x^{'}$ the resulting distorted signal value. The clipping signal on the m-th antenna, with notation analogous to (6.10),

$$\boldsymbol{c}_m = \boldsymbol{x}_m - \boldsymbol{x}^{'}_m, \tag{6.11}$$

is the residual after peak-cancellation. Reducing the desired PAR level would mean more energy in the clipping signal which, in turn, needs to be compensated by antennas in the set χ^c.

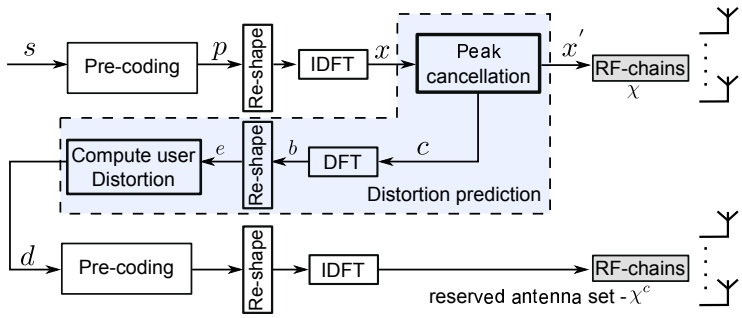


Figure 6.2. Low complexity PAR reduction, where the dedicated set of compensation antennas χ^c counteracts the distortion.

DISTORTION PREDICTION The distortion arising by clipping signal due to peak-cancellation needs to be predicted before being compensated by a reserved set of antennas. This is done by the distortion prediction block shown in Fig. 6.2, wherein first the clipped signal ($\boldsymbol{c}_m$) is transformed back to the frequency domain as

$$\boldsymbol{b}_m = \boldsymbol{F}^H \boldsymbol{c}_m \,. \tag{6.12}$$

After which it is reshaped (transposed) similar to (6.4) as

$$[\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_N] = [\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_M]^T \,, \tag{6.13}$$

and used to compute the distortion

$$\boldsymbol{d}_n = \boldsymbol{H}_n^{(\chi)} \, \boldsymbol{e}_n \,, \tag{6.14}$$

on tone n at the user end, for all K users, where $\boldsymbol{H}_n^{(\chi)}$ is the channel matrix from the antenna set χ to the users. The distortion can now be compensated by transmitting $-\boldsymbol{d}_n$ through the χ^c antenna set.

COMPLEXITY ANALYSIS There are two sets of pre-coding which needs to be performed, for user symbols $\boldsymbol{s}_n$ and distortion cancellation $-\boldsymbol{d}_n$. If we consider the set χ to contain M_1 antennas with a total of M antennas, the compensation set χ^c contains $M - M_1$ antennas. The total complexity in terms of multiplications for computing the two pre-coding matrices per tone is $\mathcal{O}(M_1K^2 + K^3) + \mathcal{O}((M - M_1)K^2 + K^3) = \mathcal{O}(MK^2 + 2K^3)$, which is almost the same as performing ZF pre-coding on all M antennas (except for one additional $K \times K$ matrix inversion).

The main overhead, in terms of complexity, is from the distortion prediction module. A DFT is performed efficiently using fast Fourier transform, hence requiring $\mathcal{O}(N \log_2 N)$ multiplications. The matrix-vector multiplication with $\boldsymbol{H}_n$ requires $\mathcal{O}(M_1K)$ multiplications per tone, resulting in a total complexity overhead of $\mathcal{O}(M_1N \log_2(N) + NM_1K + NK^3)$ to perform the proposed PAR reduction as compared to performing only zero-forcing pre-coding on all antennas. Consider a system with $M = 100$ antennas serving $K = 10$ users simultaneously, using OFDM modulation with $N = 512$ tones. If we reserve $M_2 = 25$ antennas for distortion compensation resulting from the peak cancellation on the remaining 75 antennas, the complexity overhead is around 15% to that of performing ZF using M antennas. This is a minor overhead and what remains to investigate is the performance of this method.

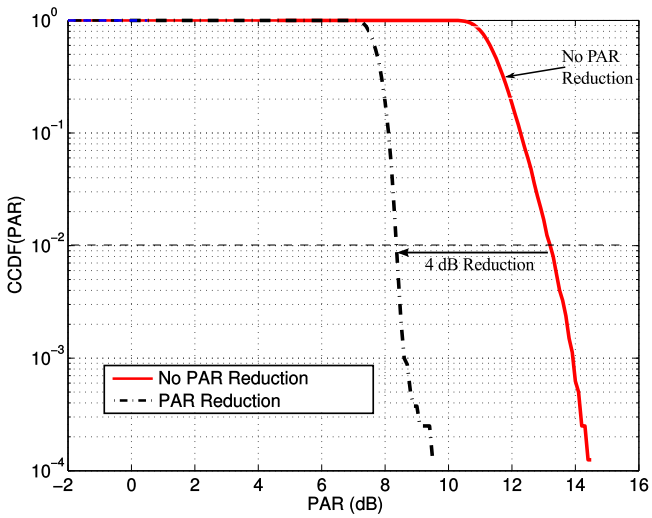


Figure 6.3. CCDF of PAR_{gbl} for a MIMO system with $M = 100$, $K = 10$, $N = 512$, using 64-QAM modulation and with $M_2 = 25$ reserved antennas for compensation.

PERFORMANCE ANALYSIS

To measure the PAR characteristics of the proposed method, the Complementary Cumulative Distribution Function (CCDF) of the PAR defined as

$$\text{CCDF}(\text{X}) = \Pr(\text{PAR} < \text{X}) = 1 - \text{CDF}(\text{X}),$$

in used. A tolerance level of 99%, *i.e.*, CCDF(PAR)=1% is set as a benchmark for all the antennas.

Fig. 6.3 demonstrates the PAR_{gbl} characteristics of the proposed low complexity scheme. There is a reduction of around 4 dB when using 25% of the antennas for distortion compensation. Reserving antennas has a negative effect on the spectral efficiency, but it is much lower than when using techniques like tone-reservation to achieve similar PAR reductions. This is because the bandwidth is linearly proportional to capacity, whereas array gain which depends on number of transmit antennas is logarithmically proportional to capacity. For the antenna set χ, the PAR_m is controlled by the amplitude threshold applied in the peak cancellation on these. For the antennas in the compensation set χ^c the PAR_m gets worse by around 3-5 dB, which may appear a bit concerning. However, it should be noted that the average power of these antennas is considerably lower (around 8 dB lower for $M_2 = 25$) than on the other antennas, resulting in an absolute peak power on a comparable level.

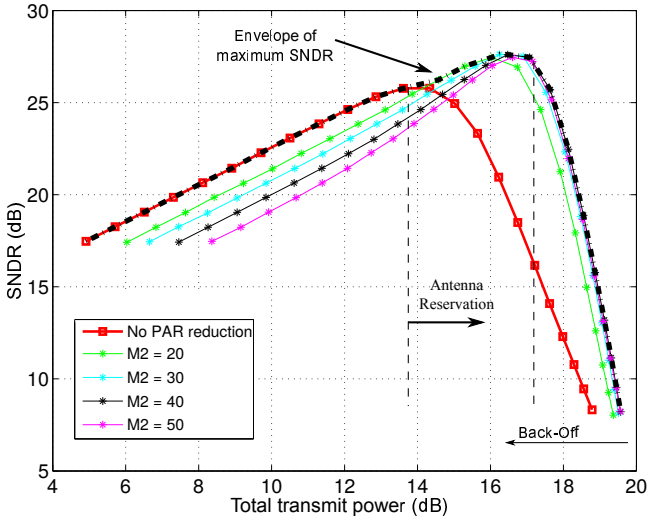


Figure 6.4. Antenna-reservation impact on SNDR for varying transmit power. The simulation was performed for a MIMO system with $M = 100$, $K = 10$, $N = 512$, noise variance of 1 and a hard-saturation model (maximum amplitude of 1) for PA.

To analyze the impact of antenna-reservation (χ^c) on the system performance, we look at the Signal-to-Noise and Distortion Ratio (SNDR) at the user terminals

$$\text{SNDR} = 10\log_{10}\left(\frac{P_\text{s}}{P_\text{d} + \sigma_w^2}\right), \tag{6.15}$$

where P_s is the received signal power, P_d the power of the distortion introduced by clipping and σ_w^2 the receiver noise variance. In Fig. 6.4, it can be seen that by reserving antennas there is a drop in SNDR at low transmit powers, because of the associated loss in the array gain. As the transmit power is increased, beyond some point, the PAs saturate, and a higher SNDR is achievable by deliberate clipping and using reserve antennas to mitigate the clipping distortion. This results in a linear extension of SNDR by around 4 dB as shown in Fig. 6.4. However, further increasing the transmit power results in more power in the reserved antennas, which also start to saturate, hence lowering the SNDR.

A more intuitive and generalized measure is the Output Back-Off (OBO), the difference between operating power and the 1 dB compression point of the PA, required to guarantee enough linearity. The proposed PAR reduction schemes allows for around 4 dB lower back-off. This, in turn, would result in the the PAs operating in a more efficient region.

6.2. CONSTANT ENVELOPE PRE-CODING

In the previous section, a low complexity OFDM based PAR aware pre-coding was presented. In this section, a stringent transmission scheme is presented which, due to the constant amplitude (envelope), has 0 dB PAR before going through a DAC. The narrow-band system model in this section is similar to that of the previous chapter, *i.e.*,

$$\boldsymbol{y} = \sqrt{P}\boldsymbol{H}\boldsymbol{x} + \boldsymbol{w}\,. \tag{6.16}$$

The channel gain between the m-th BS antenna and the k-th user is denoted by h_{km}. Channel matrix to all users is denoted as $\boldsymbol{H} \in \mathbb{C}^{K\times M}$, where h_{km} is the (k,m)-th entry. Let $\boldsymbol{x} = [x_1, x_2, ..., x_M]^{\mathrm{T}}$ denote the pre-scaled transmit vector from the M BS antennas, which is normalized to satisfy $\mathbb{E}[\boldsymbol{x}^{\mathrm{H}}\boldsymbol{x}] = 1$.

ZF pre-coding can be viewed as a constrained least-squares solution for an under determined system, *i.e.*, ZF cancels (zeros) all inter-user interference with least transmit energy (min $||\boldsymbol{x}||_2$, subject to $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{x}$). The constant envelope pre-coding is similar to ZF, *i.e.*, inter-user interference is suppressed, but with a strict constraint on the amplitude. The optimization problem for the CE pre-coder can be mathematically formulated as

$$\begin{aligned} &\text{maximize} && \alpha \\ &\text{subject to} && \alpha\boldsymbol{s} = \boldsymbol{H}\boldsymbol{x} \\ &&& |x_m|^2 = 1, \text{for } m = 1, \cdots M, \end{aligned} \tag{6.17}$$

where α is scaling factor which improves the transmission power-efficiency by utilizing the array gain more effectively.

CE can also be treated as a phase only (fixed amplitude, 0 dB PAR) transmission, with received signal at the k-th user as

$$y_k = \sqrt{P}\sum_{m=1}^{M} h_{km}e^{j\phi_m} + w_k, \tag{6.18}$$

where $\sqrt{P}e^{j\phi_m}$ is the signal transmitted on antenna m. The solution to (6.17) is not trivial, and in this work, we try to simplify it in two steps. First, we compute an appropriate value of α and then we minimize the inter-user interference. In the next section, we will look into these simplifications and discuss the performance and complexity.

6.2.1. PROPOSED CE PRE-CODER

Increasing α increases the signal strength (*i.e.*, increase Signal-to-Interference and Noise Ratio (SINR)), but a too high α hinders the ability to cancel interference (hence decrease SINR). Finding the optimal scaling factor α is a

non-convex optimization problem, and is also dependent on the data $\boldsymbol{s}$. Furthermore, frequent changes in α makes it hard for the users (receivers) to keep a track or to predict the scaling factor for detection.

APPROXIMATION FOR SCALING FACTOR Due to the above factors, an approximation of α as a long-term constant (varying slowly over multiple channel estimations) is preferred. We came up with a low-complexity approximation to find the scaling factor

$$\alpha_{\mathrm{Tr}} = \sqrt{\frac{\mathrm{Tr}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{H}})}{K}}, \tag{6.19}$$

where Tr is the trace of a matrix. This approximation was evaluated with the simulated heuristic optimal α as shown in Fig. 6.5. The low complexity approximation is close to a heuristic α, and the computation is performed as part of the CE pre-coder with limited hardware overhead (initialization step in Algorithm 6.1). By computing α from (6.19), we can simplify (6.17) as

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & ||\alpha_{\mathrm{Tr}}\,\boldsymbol{s} - \boldsymbol{H}\boldsymbol{x}||_2 \\ \text{subject to} \quad & |x_m|^2 = 1, \text{where } m = 1, \cdots M. \end{aligned} \tag{6.20}$$

The solution of (6.20) has multiple local-minima, but fortunately in case of massive MIMO, where $M \gg K$, even the local-minima tend to be close to optimal [62]. To solve the CE pre-coder we use the coordinate-descent algorithm, which is analogous to gradient-descent, barring that the optimization

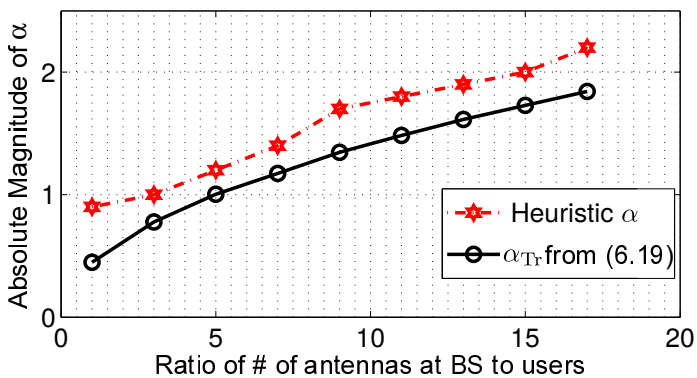


Figure 6.5. Approximated scaling factor (α) for a $K = 10$, and 4-QAM modulation. The heuristic-α is determined by sweeping over a range of values and solving (6.17).

is performed on one coordinate (variable) at a time. This is quite similar to [62], except that $\alpha = 1$ in the latter and also we avoid explicit computation of phases, due to high hardware cost.

After performing appropriate scaling of user symbols, coordinate-descent algorithm is used to solve (6.20), as shown in Alg.6.1. To reduce the complexity and hardware cost, the previous results and residual vectors are used to avoid straight out matrix-vector computations for each iteration. This results in an $\mathcal{O}((9K+5)MP)$ real-valued multiplications per user symbol $\boldsymbol{s}$.

PERFORMANCE OF CE PRE-CODER Fig. 6.6 shows the per-user sum-rate for the CE pre-coder with different number of iterations (P). When the scaling factor $\alpha = 1$, the CE pre-coding is same as in [62], and has a lower performance compared to our proposed pre-coding. The proposed CE has a lower sum-rate than that of ZF for smaller ratios, due to the inability to suppress inter user interference, while satisfying the strict amplitude constraint. However, increasing number of antennas provides a large degree-of-freedom (massive MIMO), which improves the performance of CE. In particular for high ratios (>10) only $P = 2$ iterations are needed to achieve performance close to the ZF. In the following the hardware architecture of the CE pre-coder is described, followed by discussion on implementation results.

Algorithm 6.1: Proposed CE pre-coder.

	// Initialization per channel realization	$\mathcal{O}$
1	**for** $m = 1 \rightarrow M$ **do**	
2	$\quad \boldsymbol{a}_m = \frac{\boldsymbol{h}_m}{\|\|h_m\|\|_2^2}$	$4K+1$
3	$\quad \alpha = \alpha + \|\|h_m\|\|_2^2$	
4	**end**	
	// Main loop (P iterations over M antennas)	
5	**for** $\boldsymbol{r} = \boldsymbol{s}$, $\boldsymbol{x} = \boldsymbol{0}$, $l = 1 \rightarrow P$ **do**	
	$\quad$ // Inner-Update loop	
6	$\quad$ **for** $m = 1 \rightarrow M$ **do**	
7	$\quad\quad update\_x = \boldsymbol{a}_m^H \boldsymbol{r}$	$4K$
8	$\quad\quad x_{\text{temp}} = x_m + update\_x$	
	$\quad\quad$ // Truncation	
9	$\quad\quad x_{\text{trunc}} = \frac{x_{\text{temp}}}{\|x_{\text{temp}}\|}$	5
	$\quad\quad$ // Update residual vector, $\boldsymbol{r}$	
10	$\quad\quad \boldsymbol{r} = \boldsymbol{r} - \boldsymbol{h}_m(x_{\text{trunc}} - x_m)$	$4K$
11	$\quad\quad x_m = x_{\text{trunc}}$	
12	$\quad$ **end**	
13	**end**	

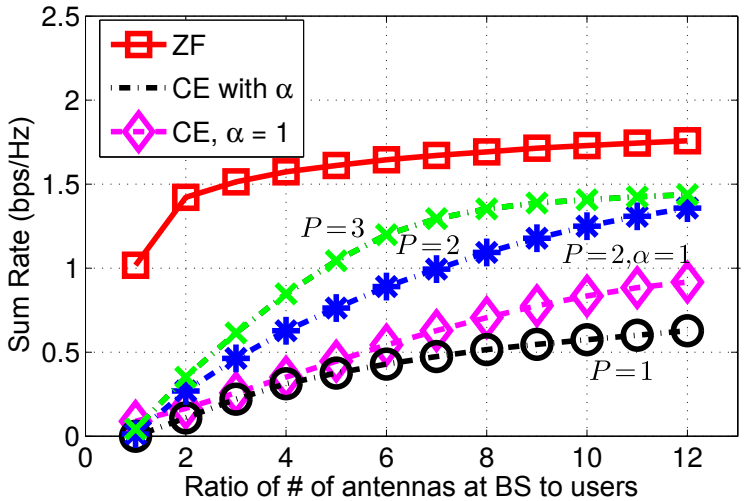
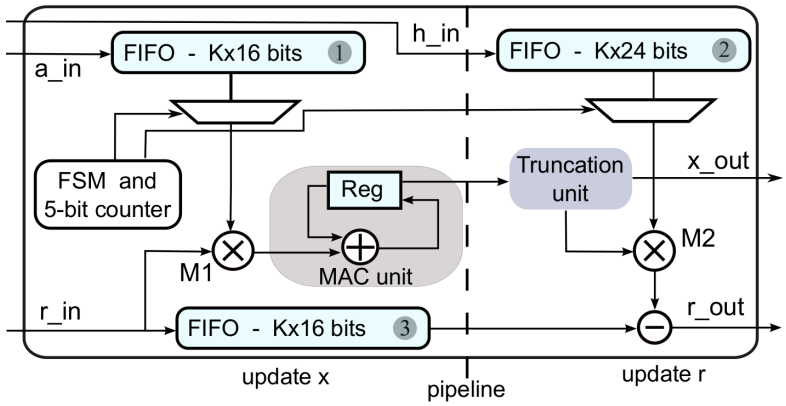


Figure 6.6. Per-user ergodic sum-rate for CE pre-coder with respect to ideal ZF pre-coder for an i.i.d. channel with $\sigma^2 = 0.01$.

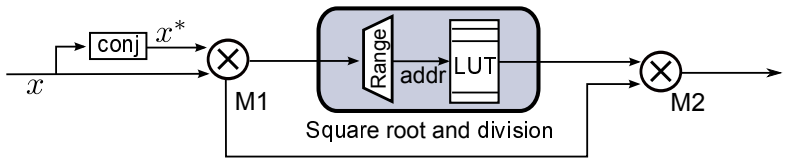
6.2.2. HARDWARE ARCHITECTURE

The CE pre-coder in Alg.6.1 is mapped to a high throughput and pipelined systolic array. This mapping exploits the fact that optimization is performed on one variable at a time, resulting in vector dot-products, rather than matrix-vector products. The vector dot-product and (update residual) vector-scalar operation are streamlined, resulting in a highly efficient processing element.

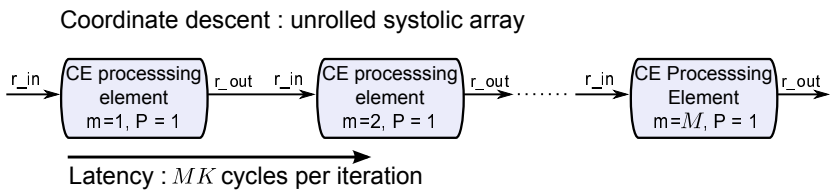
PROCESSING ELEMENT In Fig. 6.7(a), the hardware architecture of the PE for CE pre-coder is shown. The PE implements the *inner-update* loop (Alg.6.1), in a streamlined architecture. The operations start with a *load-phase*, where the channel column vector ($\boldsymbol{h}_m$) is shifted serially into the FIFO-1. After the *load-phase*, the PE enters a *wait-phase* and the MAC unit is cleared. A valid data (r_in_valid) triggers the MAC units to perform vector-dot product (*compute-state*) between the serial data (r_in) and normalized column vector (a_in). The serial data is also pushed into the FIFO-3 simultaneously. After the completion of the vector-dot product, the result is truncated as mentioned in Alg.6.1. The PE is pipelined such that while in the *compute-state* the computation of residual vector $\boldsymbol{r}$ is performed in parallel to the previous data. The utilization of the PE is 100%, with both multipliers M1 and M2 continuously active, since M1 operates on new data vector and M2 operates simultaneously on the old vector.



(a) Processing element for CE pre-coder.



(b) Performing truncation by re-using multipliers.



(c) Uni-directional systolic array of the CE pre-coder with M antennas.

Figure 6.7. Hardware details of CE pre-coder.

TRUNCATION UNIT The truncation unit adjusts the amplitude of the result from the MAC unit to a fixed value. This is performed by dividing by the absolute value ($|x_{\text{temp}}|$) and scaling appropriately based on the number of antennas. In hardware, the truncation unit can be implemented as an independent hardware unit with one real valued division unit and two real valued multipliers. However, the utilization of this unit will be low, since it will be used only once every K cycles *i.e.*, once after *compute-state*. To avoid this low utilization, the multipliers M1 and M2 are re-used to compute the absolute value and perform the appropriate truncations. For hardware re-use, a *trucate-state* is inserted in the state flow after the *compute-state*. In this state, the multipliers M1 and M2 are multiplexed to perform the absolute value computation and scaling, respectively, as shown in Fig. 6.7(b). Overall this re-use costs two

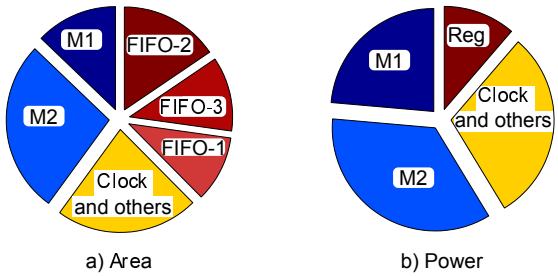


Figure 6.8. Area and power breakdown of PE with 12-bit precision.

additional cycles, but the expensive truncation logic is performed with almost no additional hardware.

SQUARE ROOT AND DIVISION UNIT In this implementation, the user symbols are 8-bits wide (modulation of up to 256-QAM), and this in turn means that the square-root and division unit can be accurately performed without multipliers e.g., by a simple LUT. The number of elements in the LUT is optimized by dividing the total range based on statistics, and the fact that smaller numbers need to be more accurately represented when performing division. In the next section, the optimized PE is used to implement a high throughput systolic CE pre-coder, and the results of the latter are discussed.

6.2.3. IMPLEMENTATION AND RESULT

The CE pre-coder is implemented by connecting the PEs in series, forming a systolic array as shown in Fig. 6.7(c), with first PE(m=1) performing computations for the first antenna. The systolic array is generic and is easily generated based on the requirements, *i.e.*, BS antennas (M), and iterations (P).

HARDWARE COST The proposed architecture is implemented in RTL, and synthesized in 65 nm CMOS. The gate-count break-up of the PE is shown in the pie-chart in Fig. 6.8. The multipliers (M1 and M2) take the majority of combinational logic, along with the FSM and counters. The non-combinational logic mainly consists of 3 FIFOs, with FIFO-2 being the biggest. The other non-combinational logic mainly consists of pipeline registers required to perform the streaming operations. A major concern with this architecture is the number of registers in the design, which would increase the total power consumption. To combat this, a clock-gating scheme is used.

POWER REDUCTION AND CLOCK-GATING Clock gating of the FIFOs are performed in each PE separately, based on its local state-machine. The gating

Table 6.1. Hardware results of CE pre-coder in 65 nm CMOS.

		Per Processing Element	For $K = 10$, $M = 100$, $P = 2$
Area [mm^2]$^{\#}$		.030	6.02
Gate Count [kGE]		14.1	-
Max. Clock [MHz]		500	500
Latency	[cycles]	$K+2$	2400
	[μsec]	$(K+2)/500$	4.8
Throughput [G samples/sec]		$0.5/K$	4.1
Information rate [Gbps]		-	3.33
Power [mW]*		3.96	792

$^{\#}$ Includes post layout clock tree synthesis.
* Power numbers are extracted by performing post-layout simulations.

is done by using a latch-based clock-gating circuit [27]. The power reduction due to the clock-gating technique is around 20%, with a negligible area overhead.

LATENCY AND THROUGHPUT The latency of the PE depends on the number of users K (Table 6.1), since the operations in the PE are performed serially. However, the systolic array provides a high throughput with a maximum clock rate of 500 MHz. The high clock-rate is mainly due to the critical path being isolated inside each PE. Furthermore, the PE can handle one user-symbol every cycle in a fully unrolled implementation. As an example, we consider a massive MIMO system with $K = 10$ (up to 16), $M = 100$, $P = 2$, with 256-QAM modulation, as in Table 6.1. This would result in total user information data-rate of 3.33 Gbps with a power consumption of 792 mW. The power consumption is high compared to ZF pre-coding, however, it is expected that the fixed amplitude of transmit signal gets translated to efficient power amplifiers. In the next section, we will look into effects of IQ imbalance in massive MIMO system.

6.3. EFFECTS OF IQ IMBALANCE

Most of the transceivers today have an in-phase (I branch) and quadrature (Q-branch) component which are passed through two mixers with a phase difference of 90°. IQ imbalance arises when there is a mismatch in amplitude or phase between the mixers of transmitter and receiver. This effect can be modeled by two parameters, *i.e.*, ϵ amplitude and $\delta\phi$ phase mismatch, as shown in Fig. 6.9. The effects and compensation of IQ imbalance are well studied [63], [64]. In line with these works, two variables a and b are defined,

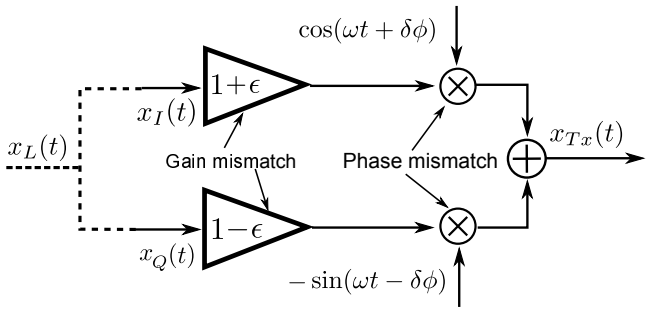


Figure 6.9. Transmitter IQ imbalance model, with ϵ and $\delta\phi$ the physical mismatch parameters, $x_L(t)$ time domain baseband IQ signal and $x_{Tx}(t)$ is transmitted signal.

which are calculated from the physical parameters as

$$\begin{aligned} a &= \cos(\delta\phi) + j\epsilon\sin(\delta\phi) \\ b &= \epsilon\cos(\delta\phi) + j\sin(\delta\phi), \end{aligned} \tag{6.21}$$

where $a \to 1$ and $b \to 0$ with decreasing ϵ and $\delta\phi$. The signal received at a receiver with no IQ imbalance, when there is frequency independent IQ imbalance at a transmitter, becomes

$$x_{\mathrm{Rx}}(t) = a x_{\mathrm{Tx}}(t) + b x^*_{\mathrm{Tx}}(t), \tag{6.22}$$

which, in the corresponding frequency domain is expressed as

$$X_{\mathrm{Rx}}(f) = a x_{\mathrm{Tx}}(f) + b x^*_{\mathrm{Tx}}(-f), \tag{6.23}$$

indicating a dual effect. There is both an attenuation of the correct signal and interference from a frequency mirrored copy of the signal.

Various studies on the effects of hardware impairments for massive MIMO systems have been [64], [65], however, these do not consider any hardware cost. In the following an analysis of IQ imbalance in the downlink is performed, which show that there is a need for pre-compensation.

EFFECTS OF IQ IMBALANCE IN MASSIVE MIMO To evaluate the effects of IQ imbalance, we look at the SNDR at the user terminals

$$\mathrm{SNDR} = 10\log_{10}\left(\frac{P_{\mathrm{s}}}{P_{\mathrm{d}} + \sigma_w^2}\right), \tag{6.24}$$

where P_{s} is the signal strength, P_{d} is the distortion due to IQ imbalance at the transmitter and σ_w^2 is the additive noise variance at the receiver. For a fixed

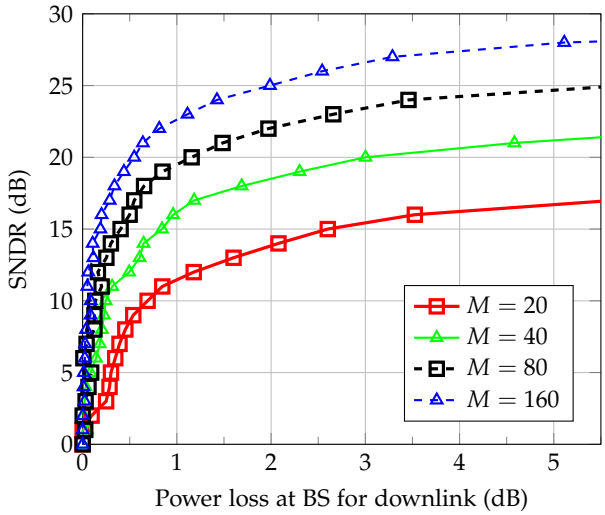


Figure 6.10. Simulated IQ imbalance for $K = 10$ users, with 6% amplitude and 6° degree phase mismatch.

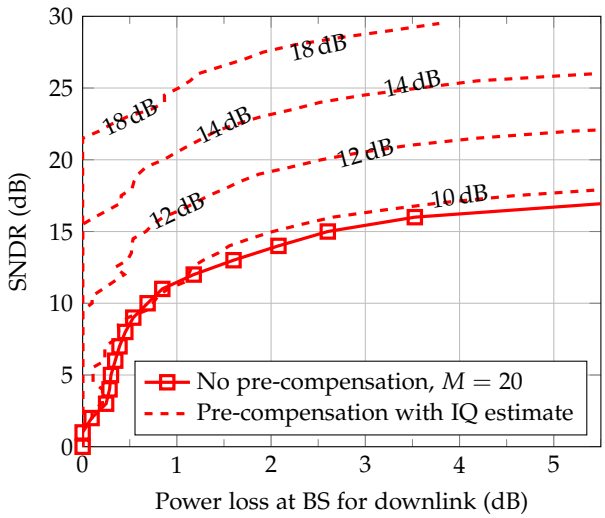


Figure 6.11. Pre-compensation for $M = 20$, $K = 10$ system, with different IQ imbalance estimation accuracy.

transmission power budget, signal power increases linearly with the number of antennas, due to the array gain. However, the IQ distortion increases at a much slower rate, mainly due to the fact that the phase of distortion is negated (x^*_{Tx} in (6.22)), and rotated (multiplying by b). Hence, the IQ distortion is unlikely to add-up constructively at the receiver. These effects can be

seen in Fig. 6.10, where the x-axis is the loss in power compared to a system with no IQ imbalance. For a fixed configuration, the SNDR will saturate if the distortion dominates over noise, and further increasing transmission power has little effect on the SNDR. One way to improve the SNDR is to increase the number of antennas, as seen in Fig. 6.10. The x-axis in the plot shows the loss in transmission power to achieve certain SNDR, which is defined as the additional transmission power required in a BS suffering from IQ imbalance compared to an ideal BS without IQ imbalance. The lower the effects of IQ imbalance, the smaller the value of the transmission power loss (moves leftwards to 0). As can be seen in the plot, increasing the number of antennas moves the curves towards the left, e.g., to achieve 15 dB SNDR a BS (suffering from IQ imbalance) with $M = 20$ antennas has a transmission power loss of 2.5 dB whereas with $M = 160$ the power loss lowers to 0.3 dB. This observation goes in line with the claims about massive MIMO being able to inherently handle hardware impairments [65]. However, increasing the number of antennas for mitigating IQ imbalance could be quite expensive, and digital pre-compensation may be a better option to limit this particular effect.

PRE-COMPENSATION ARCHITECTURE Increasing the number of antennas is a robust approach to tackle IQ imbalance, as no knowledge of the IQ imbalance parameters are required. However, increasing the number of antennas only for this purpose may not be the most cost effective solution. In Fig. 6.11 we show how the achieved SNDR of $M = 20$ antennas system increases with digital pre-compensation and different quality of the estimated IQ imbalance parameters. Higher improvements are achieved for fairly low estimation accuracies, and low-energy digital pre-compensation can be a truly viable alternative.

The IQ imbalance pre-compensation is performed after pre-coding as shown in Fig. 6.12. The main idea of pre-compensation is to transmit the signal w such that after the mixer with IQ imbalance the transmitted signal is the desired signal x. As described in (6.23), mirroring effects the n-th and $-n$-th tone, which needs to be considered during pre-compensation. Therefore the two sets of linear equations are grouped and expressed in the real domain as

$$\begin{pmatrix} a_r^n & -a_i^n & b_r^{-n} & b_i^{-n} \\ a_i^n & a_r^n & b_i^{-n} & -b_r^{-n} \\ b_r^n & b_i^n & a_r^{-n} & -a_i^{-n} \\ b_i^n & -b_r^n & a_i^{-n} & a_r^{-n} \end{pmatrix} \begin{pmatrix} w_r^n \\ w_i^n \\ w_r^{-n} \\ w_i^{-n} \end{pmatrix} = \begin{pmatrix} x_r^n \\ x_i^n \\ x_r^{-n} \\ x_i^{-n} \end{pmatrix}, \tag{6.25}$$

where the subscripts r, i indicate real and imaginary parts of complex signals.

The pre-compensation scheme involves solving (6.25). One technique is to perform a brute force inversion and a matrix vector multiplication. However,

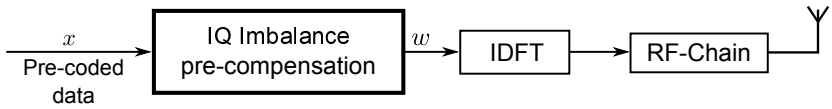


Figure 6.12. Top level block diagram of IQ imbalance pre-compensation.

Table 6.2. IQ imbalance pre-compensation in 28 nm FD-SOI.

	Per Instance	For $M = 100$
Area [mm^2]$^{\#}$	.008	0.8
Gate Count [kGE]	27	2700
Max. Clock [MHz]	200	200
Latency [cycles] *	2	2
Power [mW]	0.6	60

$^{\#}$ Synthesis results
* Latency is for 1 pair of tones per iteration.

since a and b are close to 1 and 0, respectively, an iterative method of solving linear equations is favorable. This approach is more hardware friendly and in this work we use Jacobi iterative approach [42]. To illustrate, we define the 4×4 matrix in (6.25) as $\boldsymbol{A}$, the 4×1 vectors $\boldsymbol{w}$ and $\boldsymbol{x}$. The matrix $\boldsymbol{A}$ is split into two matrices $\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{R}$, where $\boldsymbol{D}$ contains only diagonal elements of $\boldsymbol{A}$. The initial value of $\boldsymbol{w}$ is set with values of $\boldsymbol{x}$. In total 12 multipliers are used to perform matrix vector ($\boldsymbol{Rw}$) multiplications. The resulting vector is subtracted with input vector using 4 subtracters ($\boldsymbol{x} - \boldsymbol{Rw}$). The residual vector is then divided by the diagonal elements *i.e.*, $\boldsymbol{D^{-1}}(\boldsymbol{x} - \boldsymbol{Rw})$. Division is performed when updating the estimates by using the Newton-Raphson method [50] and 4 multipliers. The hardware has a flexible iterative path, and the input vector is loaded with the residual vector for the next iterations. For a low IQ mismatch parameters, the numerical accuracy of the solver is around 27 dB and 38 dB with only one and two iterations, respectively. The pre-compensation was implemented in 28nm FD-SOI technology and the power simulations are performed on a gate level netlist with back annotated timing and toggle information. The corresponding hardware results are shown in Table 6.2. Although the numbers when multiplied by a factor $M = 100$ antennas looks huge, they are not in practice, as they are distributed over RF-chains. Also, note that the implementation is fully unrolled, and folding the design to the required data-rate will reduce gate-count. When considering the energy overhead, the IQ compensation consumes only around 9 pJ/bit. Thus, the performance improvement for an extremely low energy consumption makes digital IQ compensation highly attractive.

Summary of Part-I

This part dealt with various aspects of downlink pre-coding for a multi-user massive MIMO systems. Exploiting special properties occurring in massive MIMO resulted in a reduction in computational complexity of the pre-coder. The evaluation was performed by implementing low complexity algorithms on different hardware platforms. The Neumann series based approximative matrix inversion performed well for diagonally dominant matrices, with minimal performance (BER) loss. However, for an ASIC implementation, modified Givens rotation was opted, which provided a much more robust performance for various channel conditions. Moreover, circuit level optimizations (like resource sharing –merging Gram matrix and QRD–, time-multiplexing) and device level optimizations resulted in the highest reported hardware and energy efficiency for a QRD implementation.

Lowering cost of analog components is an important aspect, considering the large number of antennas in massive MIMO systems. Two PAR aware pre-coding schemes and an IQ imbalance correction scheme were proposed in this part. It was shown that the digital hardware cost and power consumption for mitigating hardware impairments is relatively low.

Part II

Uplink Processing for Massive MIMO

Results and discussion in this part are from the following paper:

- H. Prabhu, J. Rodrigues, L. Liu, O. Edfors "A 60pJ/b 300Mb/s 128×8 Massive MIMO Precoder-Detector in 28nm FD-SOI", ISSCC, 2017.

7

Detection Techniques

Symbol detection is one the most complex tasks in wireless receivers. The channel and noise (as mentioned in Chapter2) modify the transmitted signal, and the receiver has to correctly estimate the original transmitted signal. In general there two ways of performing symbol detection. The output of the symbol detector can be a constellation point, in which case it is know as hard detection. The output can also be several possible constellation points with corresponding probabilities, in which case it is known as soft detection. Soft detection compared to hard detection may require much more hardware to compute the probabilities.

Consider a simple narrow band SISO system

$$y = hs + w\,, \tag{7.1}$$

where s and y are the transmitted and received signal, respectively, h the channel gain, and w is the additive i.i.d. noise. Typically, channel estimation is performed to aid in receiver detection. In practice, there will be estimation error which effect performance and these have been analyzed in various studies. In this work, a quasi-stationary viewpoint is taken such that the channel time variation is negligible over a few symbols, and the channel is estimated accurately by use of a training sequence embedded in the symbols. Thus for brevity and to explore the detection hardware architectures, the channel and its estimate is assumed to be same. If the channel is known (h) at the receiver, a simple approach for detection is to invert the channel

$$y/h = s + w/h\,, \tag{7.2}$$

which is illustrated in Fig. 7.1, also known as Zero-Forcing (ZF). ZF in this context is for a SISO system, and is not the same as ZF for canceling inter-user interference. After channel equalization the received data can be used

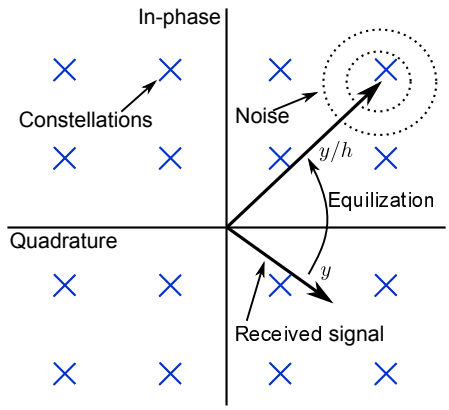


Figure 7.1. ZF detection illustration and the possibility of noise enhancement.

to perform either hard or soft detection. Hard detection locates the closest constellation point whereas soft detection computes the probability for each constellation point. Both these approaches can be parallelized since there is no dependencies between different constellation points.

The problem with ZF is that it enhances noise if the channel is close to zero or in low SNR scenarios. An approach to avoid noise enhancement is to employ MMSE equalizer which uses the SNR information. The noise spectral density cannot go to infinity due to a perturbation factor [66]. In practice, MMSE is slightly more complex than ZF and requires hardware to perform SNR estimation.

7.1. MIMO DETECTION

MIMO detection is a much more complex operation compared to a SISO detection with a single stream. The multiple streams are usually not orthogonal to each other and will interfere with each other. The streams in case of multi-user MIMO can be from different independent users. Concisely, the problem is to recover the transmitted user vector $\boldsymbol{s}$ from an observation (received vector) of the form

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{w}\,, \tag{7.3}$$

where $\boldsymbol{H}_{M\times K}$ is the channel matrix, and $\boldsymbol{w}$ the i.i.d. noise vector. The elements of $\boldsymbol{s}$ belong to a finite alphabet $\mathbb{S}$ of size $|\mathbb{S}|$, hence, there are $|\mathbb{S}|^K$ possible vectors $\boldsymbol{s}$. The assumption is that the system is not under-determined *i.e.*, $M \geq K$, and the $\boldsymbol{H}$ is full rank. To detect $\boldsymbol{s}$ in the Maximum Likelihood (ML)

sense is equivalent to

$$\underset{s \in \mathbb{S}^K}{\text{minimize}} \quad ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{s}||_2^2, \tag{7.4}$$

which is a finite-alphabet-constrained least-squares problem and is known to be NP-hard [61]. The optimal detection in terms of lowering Symbol Error Rate (SER) is the ML detection. This method calculates the distance from the received vector to all possible MIMO constellation points. Hence, the complexity ($\mathcal{O}(|\mathbb{S}|^K)$) becomes unmanageable with increasing MIMO configurations and modulation order.

ZERO FORCING The ZF detector solves (7.4) without considering the finite constraints on $\boldsymbol{s}$ as in

$$\underset{s \in \mathbb{C}^K}{\text{minimize}} \quad ||\boldsymbol{y} - \boldsymbol{H}\boldsymbol{s}||_2^2. \tag{7.5}$$

The solution for (7.5) is the well known least-squares (pseudo inverse of channel matrix) give as

$$\tilde{\boldsymbol{s}} = \boldsymbol{H}^{\dagger}\boldsymbol{y} = s + \boldsymbol{H}^{\dagger}\boldsymbol{w}. \tag{7.6}$$

ZF then approximates (7.4) by pushing each $\tilde{\mathbf{s}}_k$ onto the nearest constellation point. An important observation here is that the inter-user (MIMO stream) interference is completely removed, hence the name Zero-Forcing. However, similar to the SISO channel, the ZF works poorly unless $\boldsymbol{H}$ is well conditioned. To avoid this, a regularized ZF linear detection can be opted, which trades between interference reduction and signal power inefficiency, given as $\widetilde{\boldsymbol{s}} = (\boldsymbol{H}^H\boldsymbol{H} + \delta\boldsymbol{I})^{-1}\boldsymbol{H}^H\boldsymbol{y}$, where δ is a perturbation or scalar factor. When an optimal value of δ is chosen based on SNR to maximize detection performance, it is known as MMSE detection. For low SNR (large δ) MMSE approaches MF and for high SNR (low δ) it approaches ZF. In hardware, other than the perturbation unit, MMSE utilizes the same modules and data-flow as ZF.

ZF WITH DECISION FEEDBACK (DF) Assume ZF is computed by using Gaussian elimination. At each step, the symbol computed is projected to the constellation point before solving the next user symbol. This detection method is called DF-ZF, and in decision-tree perspective, it is equivalent to examining one single path down from the root. The problem with ZF-DF is error propagation, *i.e.*, if due to noise or bad channel conditions a symbol is detected incorrectly, the noise will propagate and thereby affect subsequent decisions.

To counter this the symbols with most reliable detection is first computed in every step. However, even with optimal ordering, error propagation can severely limit the performance. Due to this, in this work, we opt for more advanced tree-search algorithms when analyzing the performance of non-linear detection.

TREE-SEARCH DETECTION ALGORITHMS A way to reduce the complexity of ML is to limit the search space of the detector. The idea behind the Sphere Decoder (SD) is to limit the search space to a hypersphere around the received vector [67]. Thus, depending on the radius of hypersphere the complexity and performance vary. The main problem is finding an optimal radius of the hypersphere.

SD in its basic form can be improved by a mechanism called pruning. The idea is that every time a leaf node with cumulative metric is reached with a lower value than the hypersphere radius, the radius is updated [68]. There are other improvements and flavors of SD, e.g., similar to ZF-DF user symbol order can be optimized. In this work, the implementation of SD is not considered. The focus is mainly on the channel pre-processing which is required to efficiently perform sphere decoding.

To reduce the distance calculations and cancellations in SD, a well-known procedure is to perform QRD on the channel matrix, *i.e.*, $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{Q}$ is unitary and $\boldsymbol{R}$ an upper-triangular matrix. Subsequently, the detection problem in (7.4) can also be reformulated as

$$\underset{s \in S^K}{\text{minimize}} \quad ||\tilde{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{s}||_2^2, \tag{7.7}$$

where $\tilde{\boldsymbol{y}} = \boldsymbol{Q}^H \boldsymbol{y}$. The triangularized least-square constraint is a much more efficient approach to implement in hardware.

A drawback of sphere decoding is the variable time and computational complexity needed for the detection to complete. An alternative is the K-best SD algorithm which has a fixed complexity [69], [70]. This detection algorithm detects a certain number of (k) best candidates in each layer, as shown in Fig. 7.2. These fixed best candidates are further expanded in subsequent layers and the result of all expansions are then compared. The lowest cumulative metric of the (k) sub-sets is selected as the best candidate. The drawback of this compared ML or SD is that it cannot guarantee an optimal detection. A larger search (k) increases the chance of optimal detection, but will also lead to higher computational complexity. Overall the implementation of K-best algorithm is simpler, since it is a strictly feedforward algorithm, and also can be parallelized to a certain extent.

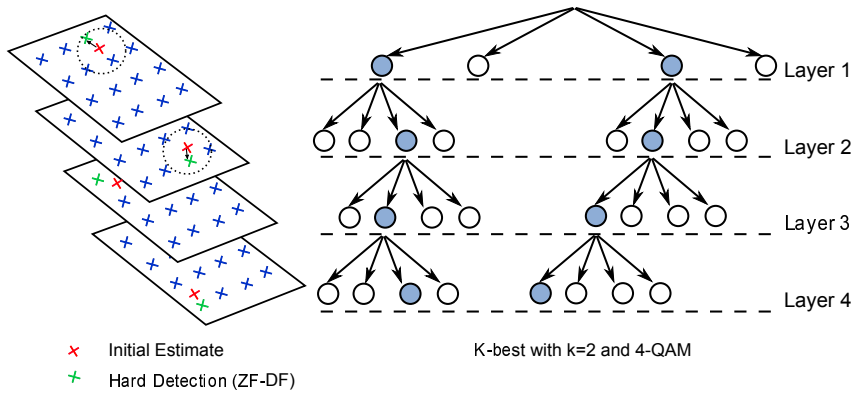


Figure 7.2. Illustration of MIMO ZF hard detection and K-best detection.

7.2. DETECTION IN MASSIVE MIMO

In massive MIMO systems, the number of observations of the transmitted signal in the uplink is much greater than in previously considered systems, *i.e.*, $M >> K$ so $\boldsymbol{H}$ is a tall matrix (over-dimensioned). It can also be interpreted as the BS having higher spatial resolution, which in turn means the multi-user streams can be resolved with greater precision.

In the next chapter, architecture and algorithm details of detection in massive MIMO is described. This is followed by a discussion on the measurement results of the implementation. The detection techniques mentioned in this chapter is also applicable for small scale, 4×4 MIMO (LTE-A) systems. In the third part of this work optimization of the channel pre-processing is performed to reduce complexity.

8

Algorithms and Implementation

Linear detection techniques provide good performance in massive MIMO systems under favorable channel conditions. Increasing number of BS antennas makes the diagonal elements of Gram matrix more dominant, resulting in lower inter-user interference. However, for lower antenna ratios ($\beta = M/K$) or for highly correlated channels, the diagonal elements of Gram matrix may not be dominant. For such scenarios, high-performance requirements call for non-linear detection schemes. On the other hand, for an efficient hardware implementation, it is important to not over-compute when the channel conditions are good. This makes detection implementation challenging, and in this chapter an adaptive Cholesky-based decomposition is proposed, which supports both linear and pre-processing for non-linear detection.

8.1. LINEAR DETECTION SCHEMES

The system model is similar to the one in previous chapter with a notable change in dimension of the channel matrix. An OFDM based multi-user massive MIMO system model is considered, with BS consisting of M antennas receiving signals from K single antenna users. The K user symbols are represented by the $K \times 1$ vector $\boldsymbol{s}$. Each antenna at the BS receives these signals, and the uplink can be modeled as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{w}\,, \tag{8.1}$$

where $\boldsymbol{y}$ is the received $M \times 1$ vector signal at the BS, $\boldsymbol{H}$ the $M \times K$ channel matrix, $\boldsymbol{s}$ the $K \times 1$ vector of user symbols to be detected at the BS.

For favorable channel conditions and high antenna ratios ($M >> K$), linear detector performance is close to optimal. The advantage of linear detection

is the relatively low complex algorithms and in turn hardware friendly implementations. MF is a simple linear detection scheme given as $\widetilde{\boldsymbol{s}}_{\mathrm{MF}} = \boldsymbol{H}^H \boldsymbol{y}$, which is multiplying the received signals with the Hermitian of the channel matrix. MF, although simple, requires much more antennas at BSs to attain close to optimal performance. ZF, on the other hand, requires a central processing unit and a higher processing cost to perform detection, given as $\widetilde{\boldsymbol{s}}_{\mathrm{ZF}} = (\boldsymbol{H}^H \boldsymbol{H})^{-1} \boldsymbol{H}^H \boldsymbol{y}$. The ZF detection scheme is viewed as a two-step procedure: firstly, the multiplication with $\boldsymbol{H}^H$, which is similar to MF, transforms the over dimensioned $M \times K$ linear system (detection problem - (8.1)) to an equivalent $K \times K$ linear system as

$$\hat{\boldsymbol{y}} = \boldsymbol{Z}\boldsymbol{s} + \hat{\boldsymbol{w}}\,, \tag{8.2}$$

where $\hat{\boldsymbol{y}} = \boldsymbol{H}^H y$, $\boldsymbol{Z} = \boldsymbol{H}^H \boldsymbol{H}$ and $\hat{\boldsymbol{w}} = \boldsymbol{H}^H \boldsymbol{w}$. The second step involves detecting $\boldsymbol{s}$ either by explicit or implicit inversion of the Gram matrix $\boldsymbol{Z}$. For large antenna ratio (β) and in favorable channel conditions, $\boldsymbol{Z}$ tends to be diagonally dominant, and low complex approximative techniques can be used for inversion. However, for unfavorable channel conditions, $\boldsymbol{Z}$ may not be diagonally dominant and requires exact inversion. In hardware, typical inversion strategies include Cholesky decomposition [71], QRD [48] or Direct Matrix Inversion (DMI) [72]. These algorithms have similar $\mathcal{O}(K^3)$ complexity. However, Cholesky decomposition in addition to performing linear detection, assists in setting up a triangularized linear system with whitened noise, which is crucial for tree-search based detection.

8.2. ADAPTIVE DETECTION BASED ON CHOLESKY DECOMPOSITION

Low complexity and near optimal performance makes linear detection an obvious design choice. However, the claims of high performance assume an uncorrelated channel and high antenna ratio β. This may not hold true in practical system, e.g., in the case of highly correlated Line-of-Sight (LOS) or large K as in stadium scenarios. For these scenarios, non-linear detection techniques like tree-based algorithms are essential for achieving full performance.

The application of tree-based detection algorithms on (8.2) needs to handle the colored noise $\hat{\boldsymbol{w}}$. An exhaustive depth-search considering the noise variance will not impact performance but is expensive in hardware. On the other hand, the colored noise impacts performance of sub-optimal pruned breadth-search algorithms with fixed complexity e.g., the K-best algorithm [69]. The standard approach in MIMO systems to setup tree-based detection without coloring noise is to perform QRD on the channel matrix. This is expensive in massive MIMO considering the large dimension of M in the 100s and the corresponding precision required for computation [45]. Hence, it is desirable

to operate on a smaller $K \times K$ linear system in hardware, after a simple MF is performed in a distributed way. However, performing MF will color the noise and an approach to whiten noise is to first perform Cholesky decomposition on the Gram matrix $\boldsymbol{Z} = \boldsymbol{L}\boldsymbol{L}^H$, where $\boldsymbol{L}$ is a lower triangular matrix. Afterwards both sides of (8.2) are multiplied with $\boldsymbol{L}^{-1}$ as

$$\bar{\boldsymbol{y}} = \boldsymbol{L}^H \boldsymbol{s} + \bar{\boldsymbol{w}}\,, \tag{8.3}$$

where $\bar{\boldsymbol{y}} = \boldsymbol{L}^{-1}\hat{\boldsymbol{y}}$ and $\bar{\boldsymbol{w}} = \boldsymbol{L}^{-1}\hat{\boldsymbol{w}}$. Computing $\boldsymbol{L}^{-1}$ explicitly is avoided by employing a forward-substitution module. Performing back-substitution on (8.3) is equivalent to ZF linear detection. Furthermore, in (8.3), noise $\bar{\boldsymbol{w}}$ is whitened, *i.e.*,

$$\begin{aligned}
\mathbb{E}(\bar{\boldsymbol{w}}\bar{\boldsymbol{w}}^H) &= \mathbb{E}((\boldsymbol{L}^{-1}\hat{\boldsymbol{w}})(\boldsymbol{L}^{-1}\hat{\boldsymbol{w}})^H) \\
&= \mathbb{E}((\boldsymbol{L}^{-1}\boldsymbol{H}^H)\boldsymbol{w}\boldsymbol{w}^H(\boldsymbol{L}^{-1}\boldsymbol{H}^H)^H) \\
&= \mathbb{E}((\boldsymbol{L}^{-1})\boldsymbol{H}^H\boldsymbol{H}(\boldsymbol{L}^{-1})^H) \\
&= \mathbb{E}(\boldsymbol{L}^{-1}\boldsymbol{L}\boldsymbol{L}^H(\boldsymbol{L}^H)^{-1}) = \boldsymbol{I}_K\,.
\end{aligned}$$

Hence, using Cholesky decomposition for linear detection has an added advantage/ability of switching over to tree-based detection techniques for higher performance.

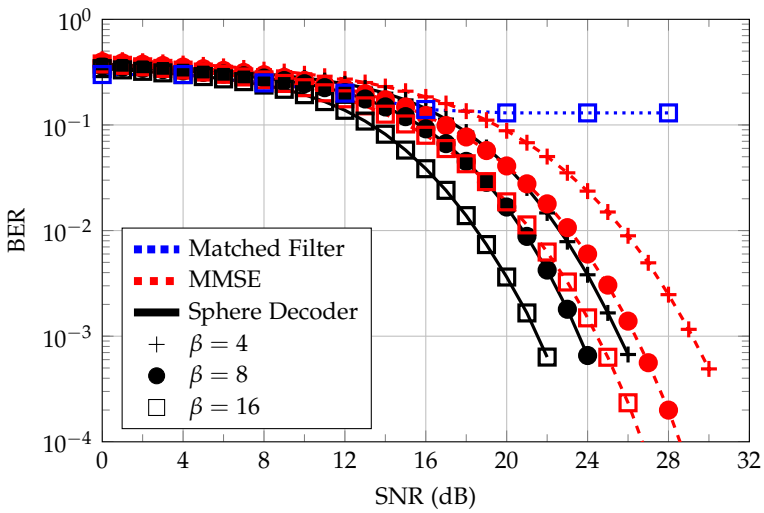


Figure 8.1. Simulated detection performance with measured outdoor channels [24]. The high correlation between users in LOS scenario impacts linear detection. For lower β the performance loss is around 4 dB to achieve 10^{-3} BER in a 16-QAM uncoded system.

8.2.1. PERFORMANCE ANALYSIS

The performance evaluation was simulated using the real measured channels from [24]. The outdoor measurements were carried out with a virtual linear array in a rich scattering scenario and LOS scenario. Uncoded BER is used as a metric for performance, and the simulation is performed for different antenna ratios β. For a rich scattering scenario or i.i.d. channel, the performance of linear detection is close to optimal [36]. Furthermore, the performance improves with an increasing ratio of BS to MS antennas. This is due to the fact that in rich scattering scenarios the channels are highly uncorrelated, resulting in a well-conditioned channel. However, the performance of linear detection is poor in cases where the channel is correlated, e.g., in LOS scenario in Fig. 8.1. There is a loss of around 2 dB to achieve BER of 10^{-3} when $\beta = 16$, and is higher for lower β. This dependency of performance on channel conditions, antenna configuration, and number of users, calls for an adaptive detection architecture.

8.2.2. FRAMEWORK DESCRIPTION

The proposed framework for adaptive detection is shown in Fig. 8.2, wherein switching between linear and non-linear detection is accomplished depending on channel conditions and performance requirement. The following modes can be envisioned for the architecture

Mode (1): The first detection option in the architecture is MF, which is multiplying the incoming signal with the Hermitian of the channel estimate.

Mode (2): For ZF or MMSE the output of MF is used for further processing. This involves computing the Gram matrix followed by Cholesky Decomposition and then performing forward and backward substitution on (8.2).

Mode (3): In this mode, the output after forward substitution *i.e.*, (8.3) is used for non-linear detection schemes. Due to the whitened noise, standard tree search implementations [73], [74] can be employed.

The selection of these modes can also be a trade-off between complexity and performance. Also, the higher non-linear detection performance can be leveraged for antenna selection and turn-off antennas at BS, with an increased detection processing cost.

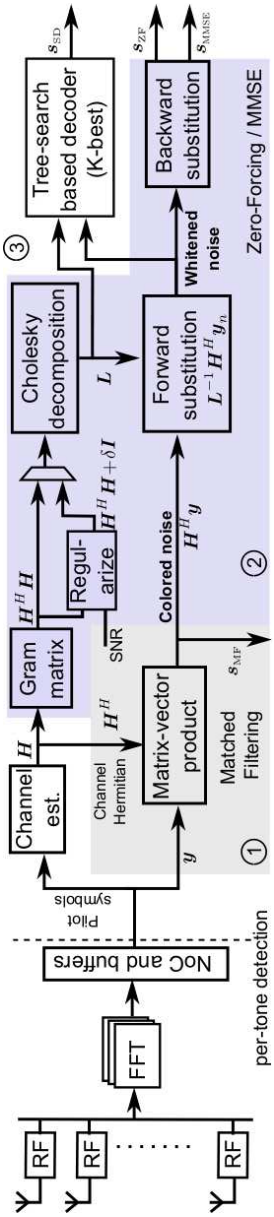


Figure 8.2. Top level framework of adaptive detection.

8.2.3. COMPLEXITY ANALYSIS

The total computational complexity for each mode is shown in Table 8.1. As expected, MF has the least complexity. Non-linear detection compared to ZF has an additional complexity factor due to the tree-search algorithm. The common computations for both are matrix-vector product, Gram matrix computation, and Cholesky decomposition, of which the first two are trivial implementations in hardware. Tree-search algorithm implementation is not covered in this work, and an existing standard implementation is used for analysis. The main focus of this work is on the Cholesky decomposition implementation which is described in the next section.

Table 8.1. Computational complexity for different detection modes.

	Complexity per realization of $\boldsymbol{H}$	Complexity for each realization of $\boldsymbol{y}$
Mode 1: MF	0	MK
Mode 2: ZF[1]	$0.5MK^2 + aK^3$	$MK + K^2$
Mode 3: Non-Linear[2]	$0.5MK^2 + aK^3$	$MK + 0.5K^2 + K^2\|S\|^K$

[1] a is constant which depends on the opted matrix decomposition strategy.
[2] Sphere decoder complexity depends on the algorithm, constellation symbols $\mathbb{S}$, search radius, number of users K.

Algorithm 8.1: Cholesky decomposition of a $K \times K$ Hermitian symmetric positive definite $\boldsymbol{Z}$ to a lower triangular $\boldsymbol{L}$.

	// Per Gram matrix realization	$\mathcal{O}$
1	**for** $p = 0 \rightarrow K-1$ **do**	
2	**for** $q = 0 \rightarrow p$ **do**	
3	**for** $a = 0, r = 0 \rightarrow q-1$ **do**	
4	$a += \boldsymbol{L}[p][r] * (\boldsymbol{L}^H[q][r])$	$K^3/6 + K^2/2$
5	**if** $p == q$ **then**	
6	$\boldsymbol{L}[p][q] = \sqrt{\boldsymbol{Z}[p][q] - a}$	K
7	**else**	
8	$\boldsymbol{L}[p][q] = (\boldsymbol{Z}[p][q] - a)/\boldsymbol{L}[q][q]$	$0.5K^2 + K$
9	**end**	
10	**end**	
11	**end**	
12	**end**	

8.2.4. HARDWARE IMPLEMENTATION

For an efficient Cholesky decomposition implementation, various intertwined trade-offs between area, throughput, and accuracy needs to be considered. To evaluate these aspects by traversing between different design space parameters requires a flexible architecture.

HARDWARE DESIGN SPACE The Cholesky algorithm in Alg. 8.1 is used for mapping onto hardware. It consists of 3 for-loops, the outer main loop has K iterations, the inner loops iterate over the index of the previous loop. The inner-most loop performs an accumulation and has an $\mathcal{O}(K^3/6)$ complexity. This accumulated value is used to compute elements of $\boldsymbol{L}$, and requires either a square root or division operation. Different implementations in hardware can be envisioned, based on parallelization and pipelining, by unrolling the for-loops.

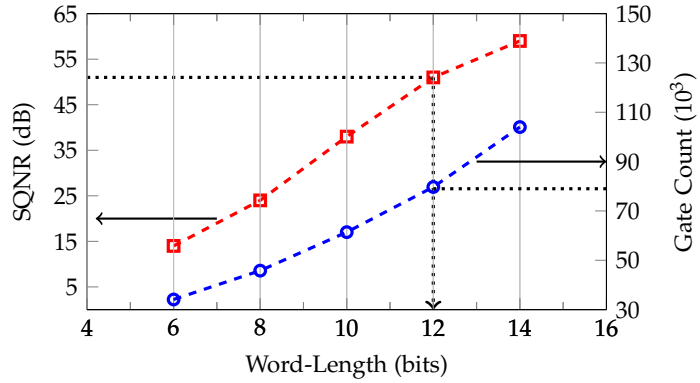
Word-length optimization is a crucial aspect of an efficient hardware implementation. Fig. 8.3(a) shows the accuracy and corresponding gate count of performing fixed point Cholesky decomposition for different word-lengths. Signal-to-Quantization-Noise Ratio (SQNR) is used as an accuracy metric and is computed as

$$\text{SQNR} = 10\log_{10}\left(\frac{||\boldsymbol{L}_{\text{Float}}||^2_{\text{Fro}}}{||\boldsymbol{L}_{\text{Float}} - \boldsymbol{L}_{\text{FP}}||^2_{\text{Fro}}}\right), \tag{8.4}$$

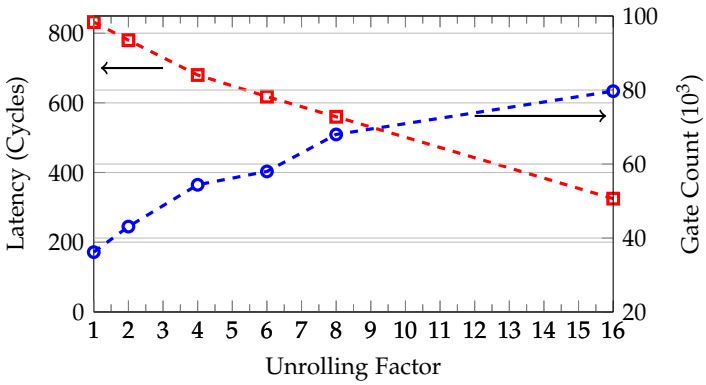
where $\boldsymbol{L}_{\text{Float}}$ is floating point precision and $\boldsymbol{L}_{\text{FP}}$ is fixed point precision computation of Cholesky decomposition. As expected increasing the word-length improves the SQNR, however, it also increases the hardware cost.

Another important hardware trade-off is between parallelization and cost. In general, reducing the computation time demands a higher hardware cost. The actual design space has many more parameters, e.g., pipelining factor, targeted frequency, power, area, and high speed/low power libraries. For illustration purpose, a simplified design space is shown in Fig. 8.3(b), where different implementations are realized based on the unrolling factor. The unrolling factor represents the speedup in processing the loops in Alg.8.1, hence, increasing the unrolling factor reduces the processing latency at the cost of higher gate count.

IMPLEMENTATION DETAIL As a case study, an unrolling factor of 16 and word-length of 12-bits is used for implementation, corresponding to a latency of 325 clock cycles and an SQNR of around 50 dB. The high accuracy is achieved by employing a bit accurate division and square root units. A standard sequential restoring arithmetic algorithm is used for both square



(a) SQNR vs hardware cost for different word-length



(b) Latency and hardware cost at 12-bit word-length.

Figure 8.3. Design space exploration, gate-count is extracted post synthesis at 300 MHz operating frequency. The unrolling factor is defined as the integer product of the unrolling of the 3 for-loops in Alg.8.1.

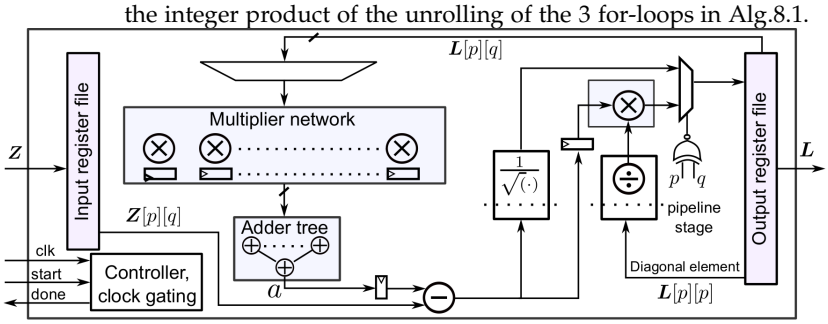
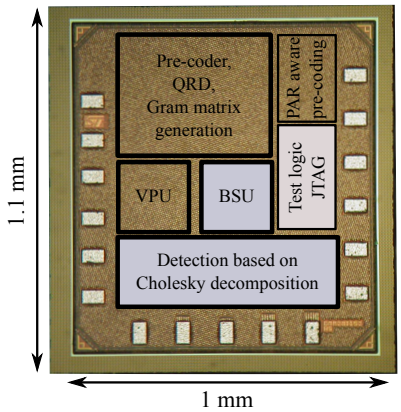


Figure 8.4. Top level architecture for Cholesky decomposition.



Fabricated chip details	
Technology	28nm FD-SOI
Total die area	1.1 mm x 1 mm
Supply voltage	0.5 V to 1.0 V
Body-bias voltage	-0.3 V to 1.2 V

8x8 Cholesky decomposition	
Gate count	79.7 k
Max. freq	300 MHz
Clock cycles	325
Power	20 mW @ 0.9V

Figure 8.5. Detection microphotograph.

root and division implementation [50]. This approach has a long critical path mainly due to the repetitive subtractions and comparisons. To overcome the speed compared to approximative fast techniques (Newton-Raphson), a two stage pipelining is performed as shown in Fig. 8.4. The architecture consists of a multiplexer network which feeds data from the register file to the multipliers. A generic adder tree network performs the vector-dot product to compute the scalar value. This scalar value (a) is used for further computations and updates the output register file based on the element pointers. The forward and backward substitution units are systolic arrays re-used from the pre-coder (described in Section 5.2).

Two techniques are employed to lower power consumption, namely global clock gating and body-biasing. The implementation supports different clock gating modes, e.g., automatic clock gating based on module activity. As mentioned earlier in the pre-coding chapter, in FD-SOI technology, the planar back-side of a gate allows for a higher electrostatic control and body biasing efficiency [75]. The implementation exploits body-biasing to either lower power consumption by performing RBB or improve performance by FBB. In the next section, measurement results of a 28 nm FD-SOI ASIC are presented, focusing on energy and latency.

8.2.5. MEASUREMENT RESULTS

Fig. 8.5 shows the chip micrograph containing different modules, which are configured by test logic and programmed through a standard JTAG interface. The performance of the detection unit was functionally verified for different core and body-bias voltages, and frequency as shown in Fig. 8.6. The maximum measurable frequency is 300 MHz due to lab setup limitations, which

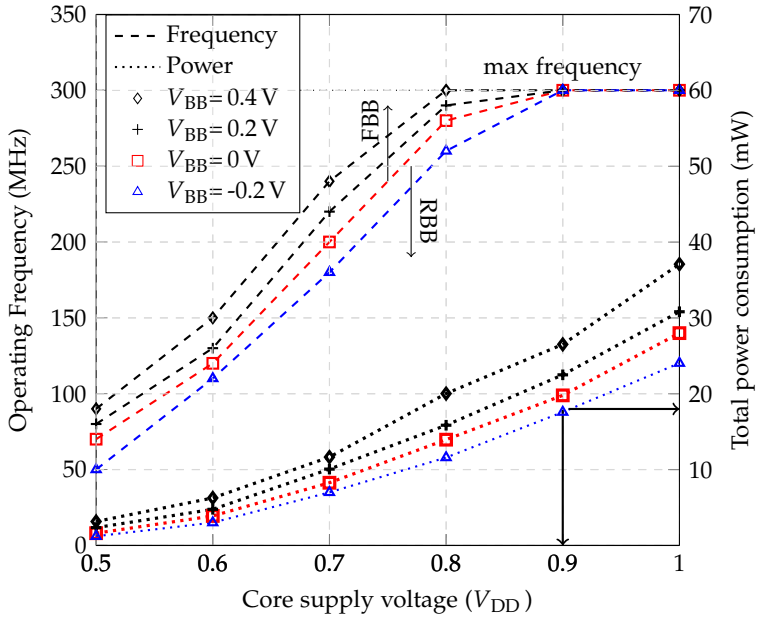
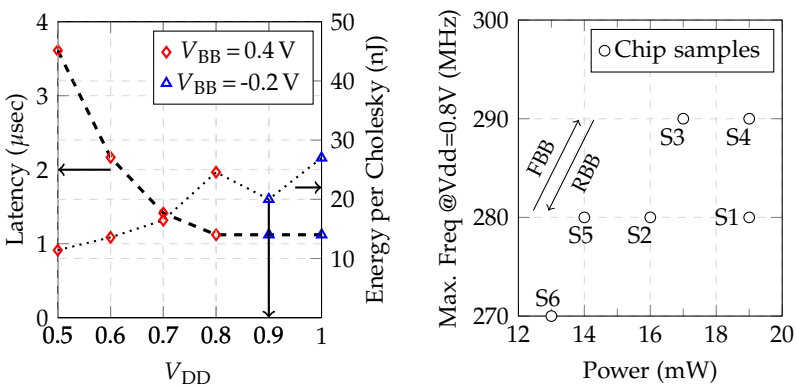


Figure 8.6. Measurement results for different core voltages and body-bias.



(a) Min latency and energy.

(b) Sample variations, $V_{BB} = 0$.

Figure 8.7. Measurement results.

creates a rooftop in the plots. The chip is functional from 0.5 V to 1.0 V core supply, and at lower voltages FBB is used to improve performance for a minor increase in power consumption. The implementation also supports RBB of up to 0.3 V, which lowers power consumption. A maximum clock frequency

Table 8.2. Hardware Results for different inversion algorithms.

	This Work	LDL [71]	QRD [48]	DMI [72]	NS [37]
Matrix Dimension ($K \times K$)	8×8	4×4	4×4	4×4	16×16
Gate Count [kGE]	79.7	89.8	36	73	104
Normalized Latency[1][μsec]	1.08	0.48	1.067	0.56	0.34
Hardware Efficiency[2]	0.013	0.023	0.026	0.024	0.028

[1] Normalized latency scaled for 8×8 with $8^3/K^3$ at clock rate of 300 MHz.
[2] Hardware Efficiency = Normalized Throughput/Gate Count

of 300 MHz or minimum latency is achieved for different core and body-bias voltage combinations. The measurements show that to minimize power consumption it is better to apply RBB than lowering V_{DD} and using FBB. Fig. 8.7(a) shows the measured minimum latency and corresponding minimum energy for different core supply voltages. For $V_{\mathrm{DD}} = 0.9\,\mathrm{V}$ and applying RBB provides lower energy consumption compared to $V_{\mathrm{DD}} = 0.8\,\mathrm{V}$ with FBB. The measurements were performed on 6 chip samples, and Fig. 8.7(b) shows the performance at $V_{\mathrm{DD}} = 0.8\,\mathrm{V}$ without body-bias. The sample variations are low and further tuning can be performed by body-bias to reach the desired performance.

As an overview, we compare the implementation with different matrix inversion and decomposition strategies in Table 8.2. A fair comparison is difficult since each of these implementations are optimized for different design targets like latency, accuracy, and power. The goal here is to show that in general, the hardware cost of Cholesky decomposition is expected to be in the same order as other approaches, due to similar $\mathcal{O}(K^3)$ complexity, with an added advantage of adaptive detection. The hardware efficiency of this work is lower than other reported designs, mainly due to the flexibility in supporting variable users ($K = 2$ to 8), wide dynamic range and multi-cycle bit-accurate division leading to lower multiplier utilization. These design choices in the architecture were targeted towards the in-house massive MIMO (LuMaMi) testbed [34]. The uplink detection has a latency requirement around 3.5 μsec, which is well supported. The NS based inversion, as expected has higher efficiency, due to the approximative inversion computations. QRD utilizes highly optimized CORDIC units which in turn improves the hardware efficiency compared to Cholesky implementations. However, NS, QRD and DMI approaches do not offer the flexibility of switching between linear and non-linear detection.

The overall detection, including the systolic array for forward/backward substitution (same module as for pre-coder), takes 148 k gates. A clock-rate

Table 8.3. Comparison of measurement results for MIMO detection.

	This Work	Chen [9]	Noethen [8]	Winter [76]
MIMO ($M \times K$)	128×8	4×4	4×4	4×4
Detection Algorithm	ZF/MMSE	MMSE	SD SISO	SD SO
Modulation	256	256	64	64
Technology [nm]	28	65	65	65
Power [mW]	18 [a]	26.5	87	38
Frequency [MHz]	300	517	445	333
Detection Area [kGE]	148	347	383	215 [b]
Detection Data-rate [Mb/s]	300	1379	396	296-807
Area Efficiency [c]	2.02	0.99	0.26	0.34-0.94
Energy Efficiency [pJ/b][d]	60	76.8	878.8	188.3

[a] Post MF power, MIMO dimension lowers to 8×8

[b] Pre-processing not included

[c] Area Efficiency = (Data-rate/Gate Count)×$(K^2/8^2)$

[d] Energy Efficiency scaled by $(8^2/K^2)\times 28/\text{Tech}$. Note that, usually higher efficiency number is better, however, in case of energy efficiency lower number is better. This is the convention used literature.

of 300 MHz achieves a throughput of 300 Mb/s, at a power consumption of 18 mW (V_{DD} = 0.9 V, V_{BB} =-0.2 V). Table 8.3 shows results for small scale MIMO detection implementations. Note that the MF area and power cost is not accounted in the central detection chip, since, it is assumed to be performed in a distributed way close to the RF chains. The massive MIMO uplink detection utilizes a 1.08 μs latency 8×8 Cholesky unit, with comparable detection area and energy efficiency of 2.02 Mb/s/kGE and 60 pJ/b. The reason for such high performance is that the small scale MIMO implementation require a lot more complex processing (SD, iterative MMSE) to achieve high performance. However, in case of massive MIMO, simple linear detectors like zero-forcing or MMSE are able to achieve high performance.

The energy efficiency can be further improved by body-bias, voltage, and frequency scaling. The array and spatial multiplexing gains in massive MIMO require handling large matrices, however, the reported higher hardware efficiencies, and ability to switch between linear and non-linear schemes makes it promising for future deployments.

Summary of Part-II

This part dealt with various aspects of uplink detection for multi-user massive MIMO systems. The limitations in the transmit power levels of battery based MSs and variable channel conditions, makes uplink detection quite challenging. Performing non-linear detection for all the scenarios is expensive and not scalable with increasing modulation order and number of users. In this part, an adaptive detection framework was proposed, which can switch between linear and non-linear detection schemes. The adaptive framework also supports different linear detection schemes like MF, ZF, and MMSE. The adaptability arises by opting to perform Cholesky decomposition on the Gram matrix, which has an ability to whiten the post matched-filtered noise, and in turn enables using standard tree-search based detection algorithms. This could be achieved by performing QRD on the channel matrix directly, however, performing QRD of a large channel matrix also requires more memory and higher accuracy computations. Thus, it is beneficial to first perform MF to lower the dimension.

The Cholesky decomposition, along with forward/backward substitution unit, was implemented in ASIC. The Cholesky decomposition implementation in this part is not the most efficient, since it employs a multi-cycle bit-accurate division and supports variable number of users ($K \leq 8$). Optimization of the division unit and using block based Cholesky decomposition leads to a much more efficient hardware ([77] in-house improved implementation). Nevertheless, the systolic forward/backward substitution unit along with Cholesky decomposition still results in high area and energy efficiencies of 2.02 Mb/s/kGE and 60 pJ/b, respectively.

Part III

Adaptive Channel Processing for Wireless Systems

Results and discussion in this part are from the following papers:

- C. Zhang, H. Prabhu, Y. Liu, L. Liu, O. Edfors, V. Öwall "Energy Efficient Group-Sort QRD Processor with On-line Update for MIMO Channel Pre-processing", IEEE Transactions on Circuits and Systems Part 1: Regular Papers, 2015.
- C. Zhang, H. Prabhu, L. Liu, O. Edfors, V. Öwall "Energy Efficient SQRD Processor for LTE-A using a Group-sort Update Scheme", ISCAS, 2014.
- C. Zhang, H. Prabhu, L. Liu, O. Edfors, V. Öwall "Energy Efficient MIMO Channel Pre-processor Using a Low Complexity On-Line Update Scheme", Norchip, 2012.

9

Channel Pre-processing

The earlier parts of the thesis presented signal processing and digital hardware implementation for massive MIMO BSs. In this part, the focus is shifted to the MS, which has a much more stringent energy constraints. Although the techniques are focused mainly on LTE-Advanced (LTE-A) MSs, they are not limited to it. Massive MIMO BSs could also adopt these techniques once the frame structures are standardized.

Efficient MIMO signal detection relies on the knowledge of CSI and channel pre-processing. For instance, the QRD of channel matrices is a key prerequisite for advanced tree-search algorithms such as sphere and K-best decoders, as discussed earlier in Chapter 7. Channel matrix inversion or other approaches of solving the system of linear equations is crucial for linear detectors such as ZF/MMSE. However, despite their importance, channel preprocessing units are often excluded from conventional signal detectors [78], as their computations are considered to be less frequent due to the common assumption of block-stationary data transmissions [79]. Unfortunately, CSI of real-world radio channels is rarely constant because of channel changes and multi-path propagation. Outdated CSI introduces additional interference to signal detection, which will drastically degrade MIMO performance. Thus, frequent CSI update and the corresponding channel pre-processing are highly desirable in wireless communication systems to provide signal detectors with adequate channel knowledge.

Using the channel's time correlation, tracking of channel state changes can be achieved using decision-directed algorithms such as least mean squares (LMS), recursive least square (RLS), and Kalman filtering [80] [81]. Nevertheless, continuous CSI tracking has not been widely adopted in practical systems, as each CSI update requires computationally intensive channel preprocessing, either QRD or channel matrix inversion, which has a comparable

complexity to that of MIMO signal detectors [81]. Moreover, considering energy consumption, frequent CSI updates result in an increased power budget for channel pre-processing operations, which is not always affordable in practical systems, especially for portable devices. For instance, QRD of one 4×4 channel matrix may be accomplished in 4 clock cycles with power consumption of 318.66 mW, corresponding to 12.76 nJ energy at 100 MHz clock frequency [49]. Although this QRD scheme achieves 20% higher processing throughput than the state-of-the-art signal detector presented in [78], each decomposition consumes 10 times more energy than performing MIMO signal detection on one received vector. To reduce energy consumption, QRD update schemes are proposed in open literature to lower the frequency of brute-force QRD computations by using either LMS [81] or matrix perturbation algorithm [82]. Although those schemes involve low complexity QRD update operations, additional error evaluation criteria are as complex as the QRD itself, which goes against the intention of complexity and energy reduction. To tackle these issues, energy efficient update schemes for performing channel pre-processing operations are described as follows:

- The first section in next chapter covers a time based energy efficient update scheme for performing channel pre-processing operation upon CSI updates. Since matrix inversion can be efficiently computed via QRD and $\boldsymbol{R}^{-1}$ [49], we focus on those updates to serve both tree-search based and linear signal detectors. To the best of our knowledge, this is the first study in open literature targeting both QRD and $\boldsymbol{R}^{-1}$ update and has hardware implementation and energy figures reported. By taking advantage of time correlation between adjacent channels, exact tone-by-tone QRD computation during successive channel matrix updates can be avoided by holding unitary matrix $\boldsymbol{Q}$ fixed while only updating the upper triangular matrix $\boldsymbol{R}$. For $\boldsymbol{R}^{-1}$ update, Neumann series approximation is adopted instead of computing exact matrix inversion. To minimize detection errors caused by outdated $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices, on-line condition check of the proposed scheme is performed prior to each QRD update.

- The second section presents a hybrid decomposition scheme with a novel group-sort QR-update strategy to efficiently implement a low-complexity Sorted-QRD (SQRD) processor for a 4×4 MIMO LTE-A system. This scheme exploits the property of the LTE-A pilot pattern to perform QR-update by computing exact $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices using only one Givens rotation. To obtain the low-complexity benefit of the introduced update scheme in the context of SQRD a novel group-sort algorithm for channel reordering is presented.

SYSTEM MODEL

Before getting into the details of the low-complexity channel pre-processing techniques it is important to understand the LTE-A pilot patterns. The system model for a LTE-A $N \times N$ MIMO is

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w}, \tag{9.1}$$

as described in Chapter 2. Each component of $\boldsymbol{x}$ is obtained by mapping a set of information bits, encoded by error-correcting codes (ECC), onto a Gray-labeled complex constellation such as M-QAM. In this work, an in-house LTE-A downlink simulator is used, and we use Frame Error Rate (FER) to evaluate the proposed algorithms. The ECC scheme adopted in simulations is a rate 1/2 parallel concatenated turbo code with coding block length of 5376 and 6 decoding iterations. Furthermore, we assume the receiver has perfect channel knowledge.

To help receivers track frequent channel changes, LTE-A inserts scattered orthogonal pilot tones to multiple antenna ports [18]. The pilot pattern for four antenna ports is sketched in a time-frequency grid, as shown in Fig. 9.1. One special property to notice, pilot tones allocated in the middle of each LTE-A Resource Block (RB) are only available for antenna ports 0 and 1. This corresponds to an update of first two columns in the channel matrix $\boldsymbol{H}$, denoted as *half-H renewal* hereafter. Benefiting from this property, the QRD updates can be simplified and is further described in the next chapter.

The decision to either perform a QRD updates or brute-force QRD in LTE system depends mainly on the channel coherence time and coherence bandwidth. This requires developing an on-line decision making or condition check scheme to evaluate if the channels have not changed drastically. For-

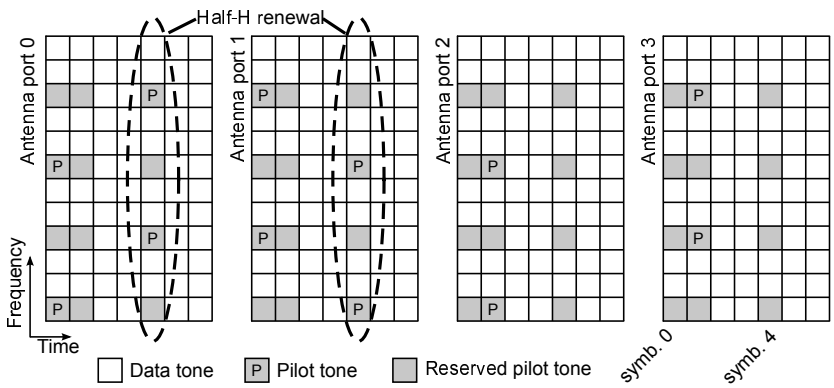


Figure 9.1. Pilot pattern for antenna ports in LTE-A resource block.

Table 9.1. LTE channel models.

	EPA $f_d = 5$		EVA $f_d = 70,300$		ETU $f_d = 70,300$	
Tap	$\tau[ns]$	P_{dB}	$\tau[ns]$	P_{dB}	$\tau[ns]$	P_{dB}
1	0	0.0	0	0.0	0	-1.0
2	30	-1.0	30	-1.5	50	-1.0
3	70	-2.0	150	-1.4	120	-1.0
4	90	-3.0	310	-3.6	200	0.0
5	110	-8.0	370	-0.6	230	0.0
6	190	-17.2	710	-9.1	500	0.0
7	410	-20.8	1090	-7.0	1600	-3.0
8			1730	-12.0	2300	-5.0
9			2510	-16.9	5000	-7.0

tunately evaluating this through simulations is straight forward since LTE-A provides standard channel models for different scenarios [83]. The three power delay profiles shown in Table 9.1 represents low, medium and high delay-spread environments, respectively. All the taps have a classical (or Jakes) Doppler spectrum [66], and a specified maximum Doppler frequency for each multi-path fading propagation condition. The three Doppler frequencies of 5, 70, and 300 Hz, say for a carrier frequency of 2.6 GHz corresponds to speeds of 2, 29 and 125 km/hr, respectively. The high speed model are not considered in the update schemes, since smaller MIMO configurations (e.g., 2×2) or lower modulation schemes (e.g., QPSK) are expected to be used in such channel scenarios to mitigate serious interference induced by fast channel variations.

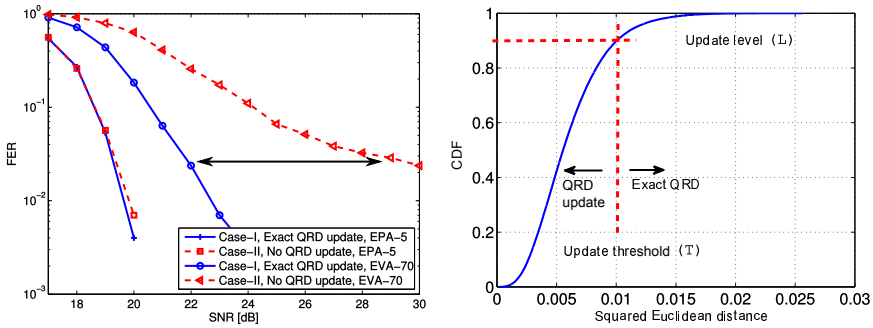
10

Adaptive CSI Tracking

Before presenting channel pre-processing update schemes, different baseband processing responses to CSI changes are analyzed using 3GPP channel models with various mobile speeds. In the next sections, we treat half-H renewals as CSI updates which happen on the 5-th OFDM symbol of a LTE-A RB, see Fig. 9.1. A full-H update happens at the beginning of each RB. Using QRD for channel pre-processing, two cases or strategies are developed:

- Case-I : Also referred to as *exact QRD update*, is the scenario when exact QRD is performed during half-H renewals.
- Case-II : In contrast to *exact QRD update*, *no QRD update* case assumes the receiver stays in a static channel environment, thus has no channel pre-processing operation is performed during CSI updates.

Utilizing K-Best signal detector with $K = 10$, FER performance for the above cases in a 4×4 MIMO downlink with 64-QAM modulation (system settings of the simulator is mentioned in previous chapter) is illustrated in Fig. 10.1(a). EPA-5 and EVA-70 channel models are used for simulations. It is interesting to note that Case-II achieves similar FER performance as Case-I in EPA-5 simulations, thanks to the high channel correlation caused by low mobility. In contrast, a significant performance degradation is observed with the EVA-70 model when no operation is performed during CSI updates (Case-II). The same analysis can be applied on $\boldsymbol{R}^{-1}$ update together with linear detectors such as MMSE, and similar results are obtained for the two cases. Hence, we can conclude that frequent QRD and $\boldsymbol{R}^{-1}$ updates only lead to small FER improvements in static and slow moving channel scenarios. However, channel pre-processing updates need to be performed explicitly upon CSI changes



(a) FER performance for different CSI updates.

(b) CDF of squared Euclidean distance for 3GPP EVA-70 channel model.

Figure 10.1. Update Scheme performance and condition check.

in channel models with moderate and high mobile speed, in order to prevent performance degradation at the cost of computation power. The above observations motivate us to propose an efficient channel pre-processing update scheme that can bridge the performance gap between Case-I and Case-II during CSI updates, and at the same time, have much lower computational complexity and energy consumption than performing the exact channel pre-processing operations.

10.1. TRACKING BY HOLDING THE UNITARY MATRIX

This section presents the proposed update scheme in conjunction with an on-line decision mechanism. In each channel matrix factorization, namely $\boldsymbol{H} = \boldsymbol{QR}$, the $\boldsymbol{Q}$ matrix contains orthogonalized unit (column) vectors, and $\boldsymbol{R}$ holds co-ordinates in the new base created by $\boldsymbol{Q}$. By utilizing the time correlation property of $\boldsymbol{H}$, we assume that the orthogonality of column vectors in $\boldsymbol{Q}$ remains unchanged during successive CSI updates. Also, any change in channel matrix is represented by value variations in elements of $\boldsymbol{R}$. With these assumptions, the demanding QRD computation can be simplified to updates of matrix $\boldsymbol{R}$, expressed as

$$\boldsymbol{H}_i = \boldsymbol{Q}_i \boldsymbol{R}_i \approx \boldsymbol{Q}_{i-t} \hat{\boldsymbol{R}}'_i, \quad t = 1 \ldots T, \tag{10.1}$$

$$\hat{\boldsymbol{R}}'_i = \boldsymbol{Q}^H_{i-t} \boldsymbol{H}_i, \tag{10.2}$$

where i and t are time indexes of channel matrix updates (both full- and half-H renewals), and the constant T defines the effective time interval for holding $\boldsymbol{Q}$ fixed in between two QRD computations. Because of the outdated $\boldsymbol{Q}$, $\hat{\boldsymbol{R}}'_i$

is an approximation of $\boldsymbol{R}_i$ and generally loses its upper triangular matrix form, which is not in favor of tree-search based signal detectors. Hence, post-processing of $\hat{\boldsymbol{R}}_i'$ is needed to transform it back to the expected matrix form. This is accomplished by forcing non-zero imaginary elements on the main diagonal and non-zero entries below the main diagonal to zero. This post-processed $\hat{\boldsymbol{R}}_i'$ matrix is hereafter denoted as $\hat{\boldsymbol{R}}_i$.

Comparing to an exact QRD, the total error induced by the update scheme includes inaccuracies caused by the outdated $\boldsymbol{Q}$ matrix and zero forcing applied on $\hat{\boldsymbol{R}}_i'$. To minimize these errors, an on-line decision check is applied prior to each QRD update. During this decision check, vector closeness between $\boldsymbol{H}_i$ columns and their predecessors in $\boldsymbol{H}_{i-t}$ is evaluated. QRD update scheme is applied only when the vector closeness is below a certain threshold value $\mathcal{T}$, otherwise exact QRD is performed. The closeness is measured as maximum squared Euclidean Distance (ED) between column vectors, formulated as

$$d^2 = \max \parallel \boldsymbol{H}_i^n - \boldsymbol{H}_{i-t}^n \parallel^2, \quad n = 1 \dots N, \tag{10.3}$$

where $\boldsymbol{H}_i^n$ is the n-th column vector in $\boldsymbol{H}_i$. The threshold value ($\mathcal{T}$) is determined by adjusting update level $\mathcal{L}$ in ED distributions (decided off-line and obtained by Monte-Carlo simulations), where $\mathcal{L}$ indicates an average ratio of QRD updates to the total number of channel matrices during half-H renewals. As an illustration, the cumulative distribution function (CDF) of ED values simulated for 3000 EVA-70 channel realizations is shown in Fig. 10.1(b). Together with the CDF, $\mathcal{T}$ and $\mathcal{L}$ values for an 90% update level are marked. $\boldsymbol{H}$ matrices with ED values larger than $\mathcal{T}$ refer to the channel matrices that require exact QRDs. Selections of update levels $\mathcal{L}$ is a design trade-off between performance degradation and complexity-energy reduction.

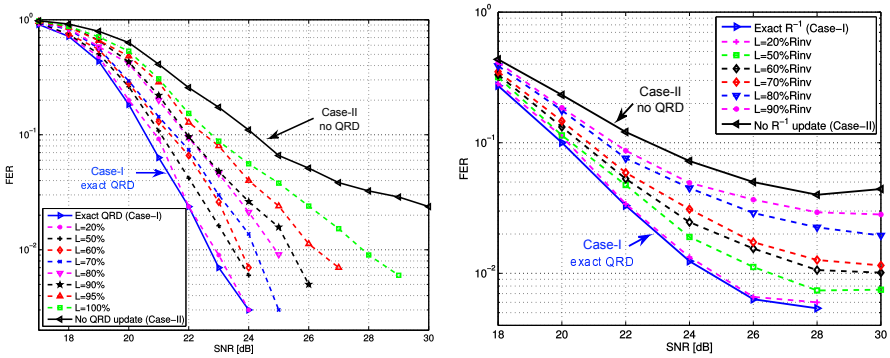
INVERSE UPDATE SCHEME

$\boldsymbol{R}^{-1}$ update is computed based on the result of QRD. By adopting Neumann series approximation (explained in Chapter 5), the explicit matrix inversion (from $\boldsymbol{R}$ to $\boldsymbol{R}^{-1}$) is avoided. Given the fact that matrix $\hat{\boldsymbol{R}}_i$ is an updated version of $\boldsymbol{R}_{i-1}$, and if the convergence condition of the Neumann series is satisfied, expressed as

$$\lim_{k \to \infty} (\boldsymbol{I}_N - \boldsymbol{R}_{i-1}^{-1} \hat{\boldsymbol{R}}_i)^k \approx \boldsymbol{0}_N, \tag{10.4}$$

then, assuming the inverse of $\boldsymbol{R}_{i-1}$ is computed together with the exact QRD, $\boldsymbol{R}^{-1}$ update ($\hat{\boldsymbol{R}}_i^{-1}$) can be calculated using Neumann series

$$\hat{\boldsymbol{R}}_i^{-1} = \sum_{j=0}^{\infty} (\boldsymbol{I}_N - \boldsymbol{R}_{i-1}^{-1} \hat{\boldsymbol{R}}_i)^j \boldsymbol{R}_{i-1}^{-1}. \tag{10.5}$$



(a) QRD update, for an 64-QAM system with K-best (k=10).)

(b) $\boldsymbol{R}^{-1}$ update for 16-QAM system with MMSE.

Figure 10.2. Update scheme performance for different threshold in a 4×4 coded MIMO LTE-A downlink.

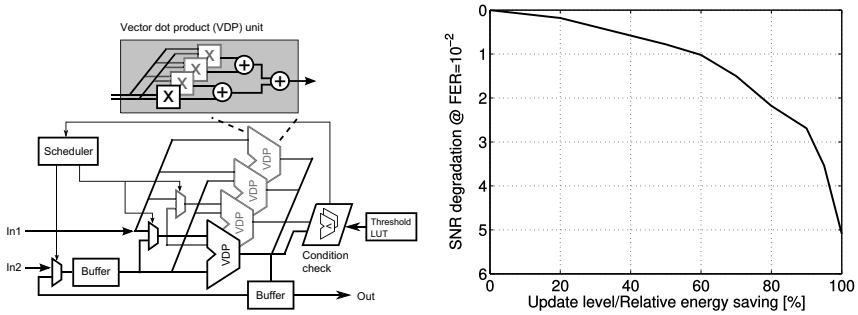
Since linear detectors do not utilize the property of upper triangular matrices, no post-processing is needed for $\hat{\boldsymbol{R}}_i^{-1}$. For practical implementations, the infinite matrix summation in (10.5) needs to be truncated. In this work, we use the first order Neumann series approximation due to its low complexity, written as

$$\begin{aligned}\hat{\boldsymbol{R}}_i^{-1} &\approx \boldsymbol{R}_{i-1}^{-1} + (\boldsymbol{I}_N - \boldsymbol{R}_{i-1}^{-1}\hat{\boldsymbol{R}}_i)\boldsymbol{R}_{i-1}^{-1}, \\ &= (2\boldsymbol{I}_N - \boldsymbol{R}_{i-1}^{-1}\hat{\boldsymbol{R}}_i)\boldsymbol{R}_{i-1}^{-1}. \end{aligned} \tag{10.6}$$

ALGORITHM EVALUATION

To evaluate the effectiveness of the proposed schemes, FER performance is simulated for a 5 MHz bandwidth LTE-A test environment. To evaluate the QRD update scheme, a K-Best signal detector with $K = 10$ is used together with 64-QAM modulation in FER simulations, whereas the $\boldsymbol{R}^{-1}$ update test uses a linear MMSE detector with 16-QAM.

Simulated FERs, as shown in Fig. 10.2, show that both update schemes fill up the performance gap between the two cases (Case-I and Case-II). Various design trade-offs can be obtained by varying the update level ($\mathcal{L}$) values. Generally, higher update levels provide higher complexity-energy reduction at the cost of performance degradation, whereas better FER performance can be achieved with lower update levels. Interesting to note is that the proposed schemes with $\mathcal{L} = 20\%$ exhibit similar FER performance as Case-I. Besides, a substantial FER improvement is presented in comparison to Case-II, e.g., for



(a) Update scheme architecture.

(b) Energy-performance trade-off for QRD update scheme.

Figure 10.3. QRD-update architecture and design trade-off.

QRD updates shown in Fig. 10.2(a), the interference noise floor observed at FER $= 10^{-2}$ in Case-II is completely removed.

With the adjustable update levels, the proposed schemes can be configured on-demand to meet different system requirements, e.g., performance- or energy-oriented design criterion. Additionally, when implementing these schemes on reconfigurable platforms, the update level can be adjusted dynamically adapting to channel condition. Hence, the proposed update schemes are highly flexible.

IMPLEMENTATION RESULTS AND COMPARISON

The proposed algorithms only involve vector processing, both QRD and the update schemes have the potential to be mapped onto one platform and to share computational resources. Therefore, hardware area is not our primary concern in this work. Instead, focus is on energy reduction in comparison with conventional QRD designs, and discuss design trade-offs between processing energy and FER performance.

A hardware architecture for the proposed update schemes is shown in Fig. 10.3(a). Resorting to the use of four complex-valued vector dot product (VDP) units, each $\boldsymbol{R}$ and $\boldsymbol{R}^{-1}$ update ((10.2) and (10.6)) for 4×4 channel matrices takes 8 and 16 clock cycles respectively, including on-line update level (($\mathcal{L}$)) based decision check (10.3). Threshold values are stored in a LUT, which are preloaded during system initialization or configured on-the-fly to adapt to the current channel scenario. The proposed hardware architecture is synthesized in a 65 nm low-leakage standard cell CMOS library. Hardware results are summarized in Table 10.1, the total core area is 0.242 mm^2 equivalent to a 116 K gate count. Operating at a normal 1.2 V core voltage, the

Table 10.1. Implementation comparison with conventional QRD.

	[49]	[84]	This work	
Algorithm	QRD Givens	QRD Interpolation	QRD update $\boldsymbol{R}$	$\boldsymbol{R}^{-1}$
Matrix size	4×4	4×4	**4×4**	
Technology [nm]	180	90	65	
Gate count [kGE]	111	317.95	116	
Max. freq. [MHz]	100	140.65	330	
Proc. cycles	4	4	8	16
Throughput [M-QRD/s]	12.5	35.16	41.25	20.625
Energy [nJ/QRD]	12.76	1.39	0.36	0.72
N.T. [M-QRD/s/kGE][#]	0.31	0.15	0.35	0.18
N.E. [nJ/QRD][*]	2.05	1.45	0.36	0.72

[#] N.T.: Throughput×(Technology/65nm)/(Gate count)
[*] N.E.: Energy×(65nm/Technology)

maximum clock frequency is 330 MHz, resulting in an update throughput of 41.25 M-QRD/s and 20.625 M$\boldsymbol{R}^{-1}$/s. The energy consumption of each QRD and $\boldsymbol{R}^{-1}$ update is 0.36 nJ and 0.72 nJ, respectively.

In Table 10.1, implementation results are compared with QRD designs reported in the literature. To ensure a fair comparison, normalized throughput (N.T.) and energy (N.E.) is considered to take into account of various design parameters, such as throughput, gate count, and technology. It shows that (Table 10.1) the proposed QRD update scheme achieves 14% higher throughput (normalized) than the QRD in [49]. More importantly, each QRD and $\boldsymbol{R}^{-1}$ update consumes 4 and 2 times less energy (normalized) compared an energy efficient QRD design [84]. Considering the simulation setup, *i.e.*, 5 MHz LTE-A with 300 data sub-carriers per OFDM symbol, a 38% energy reduction can be obtained with QRD updates during half-H renewals when using 50% update level, while performance degradation is less than 1 dB at FER $= 10^{-2}$.

In order to choose an appropriate update level value $\mathcal{L}$, a design trade-off analysis is carried out between FER and energy consumption. An evaluation of FER performance and corresponding SNR values of each $\mathcal{L}$ at the target level of FER $= 10^{-2}$ was performed. Taking the exact QRD case as a reference, SNR performance degradation of the proposed schemes is measured. As an illustration, Fig. 10.3(b) shows the energy-performance trade-off curve for QRD update simulations (Fig. 10.2(a)). The same analysis can be applied to the $\boldsymbol{R}^{-1}$ case. It is worth to note that up to 60% energy reduction can be achieved with the proposed QRD update scheme with an SNR degradation as low as 1 dB.

10.2. GROUP-SORTING

The previous section described a QRD update scheme based on tracking the upper triangular matrix (track-$\boldsymbol{R}$) and holding the unitary matrix (hold-$\boldsymbol{Q}$). The mechanism is generic and can be easily extended to other OFDM-MIMO standards. In this section, a highly LTE specific update scheme is presented. Sorted QR Decomposition (SQRD) is capable of improving detection performance by optimizing processing order based on the energy of spatial channels [85]. SQRD starts with a column permutation to the original channel matrix $\boldsymbol{H}$,

$$\tilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{P}, \tag{10.7}$$

where $\tilde{\boldsymbol{H}}$ and $\boldsymbol{P}$ denote the sorted channel and corresponding permutation matrix, respectively. After sorting, a QRD is performed on $\tilde{\boldsymbol{H}}$ to obtain the unitary matrix $\boldsymbol{Q}$ and upper-triangular matrix $\boldsymbol{R}$. In the following, we focus on the computational complexity of the QRD on $\tilde{\boldsymbol{H}}$.

For scenarios where only parts of the matrix columns change over time, QRD of the new matrix can be performed in a more efficient way than a brute-force computation (referred to as Ordering-I), *i.e.*, starting from scratch [42]. Inspired by this, a low-complexity QR-update scheme during half-H renewals is presented. Specifically, the scheme starts with the brute-force SQRD during full-H renewals, expressed with a subscript "old" as

$$\tilde{\boldsymbol{H}}_{\text{old}} = \boldsymbol{Q}_{\text{old}}\boldsymbol{R}_{\text{old}}. \tag{10.8}$$

During half-H renewals, $\tilde{\boldsymbol{H}}_{\text{new}}$ is obtained by updating two columns of $\tilde{\boldsymbol{H}}_{\text{old}}$. Although orthogonal vectors in $\boldsymbol{Q}_{\text{old}}$ may no longer triangularize $\tilde{\boldsymbol{H}}_{\text{new}}$, some of the column vectors may still point in the correct direction. As a consequence, the new $\boldsymbol{R}$ matrix, denoted as $\tilde{\boldsymbol{R}}_{\text{new}}$, can be expressed using $\tilde{\boldsymbol{H}}_{\text{new}}$ and $\boldsymbol{Q}_{\text{old}}$ as

$$\tilde{\boldsymbol{R}}_{\text{new}} = \boldsymbol{Q}_{\text{old}}^{H}\tilde{\boldsymbol{H}}_{\text{new}}. \tag{10.9}$$

Due to the outdated $\boldsymbol{Q}_{\text{old}}$, $\tilde{\boldsymbol{R}}_{\text{new}}$ is no longer an upper-triangular matrix but may still reveal some upper-triangular properties depending on the positions of the two renewed columns. Specifically, in cases where column changes take place at the right-most of $\tilde{\boldsymbol{H}}_{\text{new}}$, only one element in the lower triangular part of $\tilde{\boldsymbol{R}}_{\text{new}}$ (*i.e.*, $\tilde{r}_{\text{new}}(4,3)$) becomes non-zero. This implies that triangularization of $\tilde{\boldsymbol{R}}_{\text{new}}$ can be significantly simplified by nulling the single non-zero element instead of operating on all columns afresh as

$$\boldsymbol{G}\tilde{\boldsymbol{R}}_{\text{new}} = \boldsymbol{G}\boldsymbol{Q}_{\text{old}}^{H}\tilde{\boldsymbol{H}}_{\text{new}}, \tag{10.10}$$

where

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c & s^* \\ 0 & 0 & -s^* & c^* \end{bmatrix}. \tag{10.11}$$

In (10.11), $(\cdot)^*$ is the complex conjugation, c and s are defined as

$$\begin{aligned} c &= \tilde{r}^*_{\text{new}}(3,3)/z, \\ s &= \tilde{r}^*_{\text{new}}(4,3)/z, \\ z &= \left(|\tilde{r}_{\text{new}}(3,3)|^2 + |\tilde{r}_{\text{new}}(4,3)|^2\right)^{1/2}. \end{aligned} \tag{10.12}$$

After triangularizing $\tilde{\boldsymbol{R}}_{\text{new}}$, exact $\boldsymbol{Q}_{\text{new}}$ and $\boldsymbol{R}_{\text{new}}$ in the proposed QR-update scheme are obtained, expressed as

$$\boldsymbol{Q}_{\text{new}} = (\boldsymbol{G}\boldsymbol{Q}^H_{\text{old}})^H, \tag{10.13}$$

$$\boldsymbol{R}_{\text{new}} = \boldsymbol{G}\tilde{\boldsymbol{R}}_{\text{new}} = \boldsymbol{G}(\boldsymbol{Q}^H_{\text{old}}\tilde{\boldsymbol{H}}_{\text{new}}). \tag{10.14}$$

By combining the traditional brute-force approach (*i.e.*, computing QRD from scratch during half-H renewals) and the QR-update scheme, a hybrid decomposition algorithm is formed which dynamically switches between the two schemes to reduce the computational complexity, depending on run-time conditions of the channel reordering. Obviously, the complexity reduction depends on the applicability of the QR-update. Intuitively, we could fix the position of antenna ports 0 and 1 to the right-most part of $\tilde{\boldsymbol{H}}_{\text{new}}$ in order to obtain a maximum complexity gain, since it completely avoids brute-force computation during half-H renewals. However, the advantage of channel reordering (for improving detection performance) is lost and we refer to this as Ordering-II. On the other hand, where channel columns are permuted based on the optimal detection order without considering the position of renewed channel columns referred as Ordering-III, the applicability of the QR-update is dramatically reduced. For example, considering the 4×4 MIMO LTE-A, only $(2!2!)/4! = 1/6$ of sorting combinations meet the required update condition, thus limiting the complexity reduction. As a consequence, a smart scheduling strategy is needed to explore the low-complexity potential of the QR-update, while still retaining the performance gain of the optimal channel reordering.

GROUP-SORT ALGORITHM To fulfill the above mentioned requirements, an effective group-sort algorithm for channel reordering is developed. The key

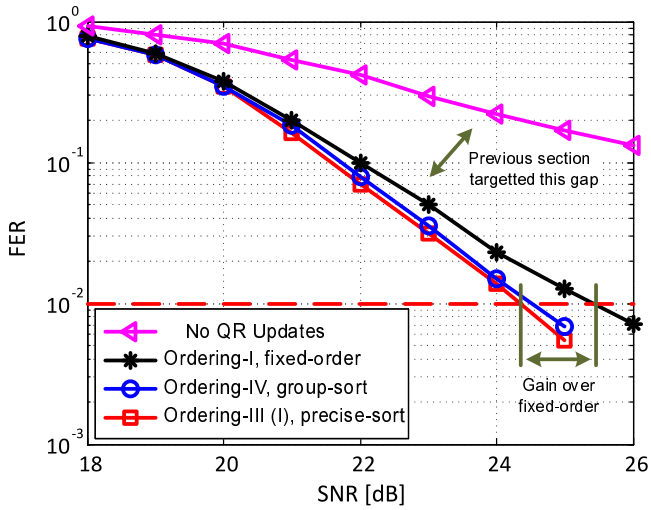


Figure 10.4. Simulated FERs in a 4×4 MIMO LTE-A downlink using 3GPP EVA-70 channel model with 64-QAM modulation.

idea is that instead of operating on individual columns, sorting of $\boldsymbol{H}$ is applied on two virtual groups, wherein columns associated with antenna ports 0 and 1 are tied together. This way, combinations of "columns" is reduced from 4! to 2!. Consequently, the probability of having both altered columns at the right-most part of $\tilde{\boldsymbol{H}}_{\text{new}}$ is increased by 3 times, *i.e.*, from $1/6$ to $1/2$. To reduce errors due to sub-optimal sorting sequences, a two-step sorting scheme is adopted. First, the sorting between groups is based on the total energy of bundled columns as

$$\mathcal{I} = \arg\max_{\mathcal{I}=\{\{0,1\},\{2,3\}\}} \sum_{i=\mathcal{I}} \|\boldsymbol{h}_i\|^2, \tag{10.15}$$

where $\mathcal{I}$ contains inter-sorted group indexes, e.g., $\mathcal{I} = \{0, 1\}$ if antenna ports 0 and 1 correspond to the strongest channels. Second, the two columns within each group, e.g., indexes within $\mathcal{I}$, are intra-sorted based on the energy of individual columns. To conclude, Table 10.2 summarizes all four scenarios of the hybrid SQRD algorithm and their corresponding applicability, wherein we denote the proposed group-sort method as Ordering-IV.

ALGORITHM EVALUATION To illustrate the effectiveness of the proposed algorithm, the 3GPP EVA 70 Hz model is used. In each FER simulation, 5000 LTE-A subframes are transmitted and decoded using a fixed-complexity sphere decoder [86]. Performance of the proposed group-sort QR-update and

Table 10.2. Ordering-I−IV of the proposed hybrid SQRD algorithm.

	Channel reordering	Brute-force	QR-update
Ordering-I	Optimal ordering with precise sorting	100%	0%
Ordering-II	Fixed order for antenna port 0 and 1	0%	100%
Ordering-III	Optimal ordering with precise sorting	83.33%	16.67%
Ordering-IV	Group-sort	50%	50%

Table 10.3. Computational complexity of hybrid SQRD algorithm.

Complexity	Computation	Multiplication	DIV/SQRT*
$\mathcal{C}_1$	QRD (10.8)	$N^3 + 2N^2$	$2N$
$\mathcal{C}_2$	$\boldsymbol{Q}_{\text{old}}^H \tilde{\boldsymbol{H}}_{\text{new}}$ (10.9)	$\frac{1}{2}N^3$	0
$\mathcal{C}_3$	Triangularization ((10.10))	$N^2 + 2N$	3
$\mathcal{C}_4$	Sorting (10.7)(10.15)	$4N$	0

* Division, square-root, and CORDIC, where the latter one is often used for generating GR matrices.

other ordering are shown in Fig. 10.4. Note that Ordering-III is the same performing brute-force and is used as a reference for FER measurements. Compared to the one where no QRDs are performed during half-H renewals (uppermost curve in Fig. 10.4), it clearly shows the importance of performing CSI and QR updates even for channels with moderate Doppler shifts. Additionally, adoption of channel reordering during QR decomposition improves performance over that of the fixed-order approach by, e.g., 1.1 dB between Ordering-II and III at FER $= 10^{-2}$. Furthermore, the group-sort approach has only small performance degradation of about 0.2 dB compared to Ordering-III, but with a large reduction in complexity as analyzed in the following.

We divide the operations required for performing channel pre-processing into four parts as shown in Table 10.3. The first operation denoted as $\mathcal{C}_1$, corresponds to complexity to perform brute-force QRD (10.8). Gram-Schmidt algorithm [42] is used for performing QRD and the total complexity is for a matrix with dimension $N \times N$. The second and third operation are related to the update scheme based on Givens rotation as mentioned in (10.10). The last operation ($\mathcal{C}_4$) is for sorting the columns as described in (10.15). Computations when performing precise- and group-sort update require both (10.9) and (10.10), which has a total complexity of $\mathcal{C}_2 + \mathcal{C}_3$. This is significantly lower than the complexity of performing QRD $\mathcal{C}_1$, e.g., by about 42% for $N = 4$. Note that the product of $\boldsymbol{Q}_{\text{old}}^H \tilde{\boldsymbol{H}}_{\text{new}}$ in (10.9) requires only half of the matrix computations during QR updates, since only two columns change in $\tilde{\boldsymbol{H}}_{\text{new}}$. The complexity of sorting in both precise- and group-sort approaches is relatively low.

Based on this analysis and in reference to Ordering-I, Table 10.4 shows the

Table 10.4. Complexity and performance comparisons of ordering-I−IV.

	Complexity	Complexity reduction	SNR degradation
Ordering-I	$\mathcal{C}_1 + \mathcal{C}_4$	− (ref.)	− (ref.)
Ordering-II	$\mathcal{C}_2 + \mathcal{C}_3$	50%	1.1 dB
Ordering-III	$\frac{5}{6}\mathcal{C}_1 + \frac{1}{6}(\mathcal{C}_2 + \mathcal{C}_3) + \mathcal{C}_4$	6%	0 dB
Ordering-IV	$\frac{1}{2}\mathcal{C}_1 + \frac{1}{2}(\mathcal{C}_2 + \mathcal{C}_3) + \mathcal{C}_4$	18%	0.2 dB

Table 10.5. Implementation summary of the vector processor [87].

	Brute-force QRD	Proposed QR-update
Gate count [kGE]	339	
Max. freq. [MHz]	500	
Execution cycles	7.5	5
Throughput [M-QRD/s]	67	100
Energy [nJ/QRD]	2.85	1.9

complexity reduction versus performance degradation of Ordering-II−IV for a 4×4 system. It shows that a 50% complexity reduction is obtained for Ordering-II. Moreover, combining the group-sort and the QR-update schemes results in more palatable trade-offs, *i.e.*, 18% complexity reduction for only 0.2 dB performance degradation.

Evaluation of the hardware friendliness of the algorithm and re-usability is performed in [87]. Implementation of the vector processor is not part of this work and is based on a in-house reconfigurable array framework presented in [88], [11]. Table 10.5 summarises implementation results for the brute-force and the QR-update computations. Operating at 500 MHz, processing throughput of the QR-update is 100 M-QRD/s and consumes 1.9 nJ per decomposition. This results in a 33% improvement compared to the brute-force counterpart, which is quite an impressive gains considering minimal (0.2 dB) performance degradation.

Summary of Part-III

This part dealt with lowering computational complexity of channel preprocessing (QRD) in MIMO systems, mainly focusing on battery operated MSs. The key idea is to exploit the time correlation of the channel to avoid computation from scratch. Evaluation of tracking channel by a hold-Q approach was performed for an OFDM based system, where up to 38% energy reductions was obtained for a performance degradation of less than 1 dB at FER$= 10^{-2}$. The tracking algorithm was further improved by utilizing the LTE-A frame structure, wherein a new group-sorting mechanism was developed along with Givens rotation updates. The group-sort algorithm was mapped to a reconfigurable platform and supports a wide range of trade-offs between performance and energy. Although, the algorithms where focused on tracking variations over time, it could also be used to exploit correlation over frequency. Moreover, the tracking algorithms mentioned in this part are not limited to mobile terminals, *i.e.*, it can be used in (massive MIMO) base stations as well.

Conclusion and Outlook

Exploiting the spatial domain to improve spectral efficiency is critical for meeting requirements of current and future wireless communication systems. Massive MIMO is a technology with the potential to fulfill these requirements. Massive MIMO requires handling a large number of antennas efficiently, by performing computationally expensive baseband processing. Typically base stations have slightly relaxed constraints on power and cost, however, it is still important to keep it low, considering the large volumes expected in future deployment strategies like femtocell networks. Furthermore, handling of the inherent large matrix dimensions in massive MIMO is a challenge, especially considering the latency requirements. In this thesis, different implementation strategies where explored including fabrication of an 128×8 massive MIMO chip. Algorithm and circuit co-optimization lead to a highly efficient pre-coding/detection implementation. The downlink pre-coding employs a QRD unit with the highest reported area and energy efficiency of 34.1 QRD/s/GE and 6.56 nJ/QRD, respectively. The Uplink utilizes a 1.08 μs latency 8×8 Cholesky decomposition unit, with the detection area and energy efficiency of 2.02 Mb/s/kGE and 60 pJ/b, respectively. The area and energy cost are similar to small-scale MIMO (LTE-A) systems, which is promising considering the advantages of massive MIMO systems.

Another important aspect of massive MIMO base stations is the energy efficiency and cost of RF components. In this thesis, pre-coding strategies to lower requirements of power amplifiers were performed. A novel "antenna reservation" based pre-coding lowered PAR by 4 dB, with only a 15% complexity overhead. A much more extreme pre-coding is the constant envelope, which has a PAR of 0 dB. Both these techniques have the potential of lowering the costs and improving efficiencies of power amplifiers. IQ imbalance in the

RF mixer were also analyzed, where the results showed that increasing the number of antennas at the base station lowered the impact of IQ imbalance. However, a much higher performance gains were achieved by performing a IQ pre-compensation which consumes only around 9 pJ/b.

This work also covered optimization for mobile station targeting currently deployed LTE-A system. In particular, tracking/update techniques were proposed to tackle the expensive MIMO processing on mobile stations by exploiting channel coherence time/bandwidth. It would be interesting as part of future work to use these low complexity update schemes for massive MIMO base stations as well.

This work covered various aspects of (massive) MIMO processing, with focus on algorithm and circuit co-optimization. This resulted in high hardware and energy efficient implementations, which is promising for future wireless communication systems. However, a well known issue with ASIC implementation is the design time, and with advancement in development tools evaluating multi-core reconfigurable vector processor with compiler support to accelerate algorithm development would be an interesting way forward.

Appendices

Appendix A

HLS and SystemC

The raising complexity of applications and corresponding implementation have forced design methodologies and tools to move to higher abstraction levels. An analogy of this is in software domain, where until the mid 1950s machine code (binary sequence) was the only language that could be used to program a computer. Later the concept of assembly language (assemblers) were introduced, which quickly progressed to high level languages (C/C++) with compilers. The productivity has improved dramatically in software domain, with higher abstraction levels, platform independent code, human readable syntax, and easier debugging. Assembly language are rarely used today, except for critical parts of a program (real time systems) where performance or code compactness is absolutely necessary. However, with the growing complexity of both software applications and hardware architectures, using high level languages clearly generates superior overall results. No one today would think of programming even a simple (forget complex) software application in assembly.

In hardware implementation, HDL did bring forth a major boost in the productivity. However, to utilize increasing silicon capacity requires even higher level of abstraction. The rapid growing demand for HLS is observed for the following reasons:

- Trend towards extensive use of accelerators: By 2024 the number of on-chip accelerators is expected to reach 3000. These accelerators have custom dedicated architectures and is incredibly expensive to maintain, update, and verify. Also, it requires dedicated teams working on these accelerators with specific know-how of the HDL implementations. Such accelerators are perfect for HLS.

- Easier system level verification: Transaction-Level Modeling (TLM) with

SystemC or C/C++ is a popular approach to describe virtual hardware platforms for early embedded software development, architectural exploration and functional verification. The availability of translating these models to generate Register-Transfer Level (RTL) automatically is attractive. Since, it saves the slow and error-prone process of manual RTL writing.

- Technology and platform independent: The behavioral level implementation of HLS provides ability to target different applications (performance), technology or even platforms. RTL implementations are fixed for a micro-architecture, and would require rewriting if there is a change in timing requirements, technology, platform (map to DSP units in FPGA) etc.

The above mentioned advantages and many other innovative design tools is expected to be the next big productivity boost in silicon industry. This chapter provides a tutorial on C/C++ based HLS followed by introduction to SystemC.

HARDWARE DESIGN USING HLS

Matrix multiplication is selected as an example for exploring some important HLS features. Computation of matrix multiplication requires 3 for-loops, with inner loop performing vector-dot product. In A.1 matrix multiplication code in HLS followed by design constraints are shown. The code for matrix multiplication (unlike HDL) is few lines and simple to understand. The implementation is highly parameterized e.g., matrix dimensions, data types, interface etc. Depending on requirements and constraints different implementations are synthesized by the HLS tool. In A.1 a folded design using one multiplier is synthesized which takes 4096 cycles to perform a 16×16 matrix multiplication.

Listing A.1 Matrix multiplication (folded)

```
---------------------------------
HLS Code (in C/C++):
void matrix_mult(int in_a[A_ROWS][A_COLS],
int in_b[B_ROWS][B_COLS], int out_c[A_ROWS][B_COLS]) {
a_row_loop : for(int i=0; i<A_ROWS; i++)
b_col_loop : for(int j=0; j<B_COLS; j++) {
int sum_mult = 0;
a_col_loop : for(int k=0; k<A_COLS; k++)
sum_mult += in_a[i][k]*in_b[k][j];
```

```
out_c[i][j] = sum_mult;
}
}
----------------------------------
HLS Constraints:
A_ROWS=16, A_COLS=16, B_ROWS=16, B_COLS=16
Data type - 32-bit integer, clock rate = 250 MHz
All IO mapped to corresponding single port RAM interface
Top level design pipelined with II=1
All loops are kept folded
----------------------------------
Synthesized Hardware:
Throughput - 4096, Latency clock - 4096
Slack - 1.74 ns, Total Area - 8900 μm²
```

In A.2 the same matrix multiplication code is used to synthesize a design with 8x throughput improvement by unfolding the inner loop. Unrolling by 8 generates hardware which performs vector-dot product in 2 cycles, with an obvious increase in area (8 multipliers). This design space exploration of folding/unfolding loops to generate different hardware is a powerful feature which is very hard to mimic in traditional HDL implementations.

Listing A.2 Matrix multiplication (unfolded)

```
----------------------------------
HLS Code (in C/C++): Same as previous
----------------------------------
HLS Constraints:
A_ROWS=16, A_COLS=16, B_ROWS=16, B_COLS=16
Data type - 32-bit integer, clock rate = 250 MHz
All IO mapped to corresponding single port RAM interface
Top level design pipelined with II=1
Inner loop unfolded by a factor of 8
RAM bandwidth increased from 32-bits to 256-bits
----------------------------------
Synthesized Hardware:
Throughput - 512, Latency clock - 512
Slack - 0.89 ns, Total Area - 58439 μm²
```

Using template and other high level abstraction in C++ (object oriented programming) provides high flexibility. In A.3 a complex 12-bit fixed point matrix multiplication is implemented. The unrolling of inner most loop increases critical path because a multiplier-adder tree is inferred. For complex data-types requiring 4 real multiplications and 2 addition, the critical path increases further. Therefore, unrolling the outer loop is more beneficial since

it infers multiple parallel *vector-dot product* units. The advantage of HLS is that only minor changes in design constraint assists in generating different hardware flavors. For further details on various design tricks and incredible power of HLS refer to [89].

Listing A.3 Matrix multiplication (unfolded complex fixed point)

```
----------------------------------
HLS Code (in C/C++):
template <typename T>
void matrix_mult(T in_a[A_ROWS][A_COLS],
T in_b[B_ROWS][B_COLS], T out_c[A_ROWS][B_COLS]) {
a_row_loop : for(int i=0; i<A_ROWS; i++)
b_col_loop : for(int j=0; j<B_COLS; j++) {
T sum_mult = 0;
a_col_loop : for(int k=0; k<A_COLS; k++)
sum_mult += in_a[i][k]*in_b[k][j];
out_c[i][j] = sum_mult;
}
}
----------------------------------
HLS Constraints:
T = ac_complex<ac_int12<12, 2, SIGNED, AC_RND, AC_SAT_SYM> >
(A_ROWS,A_COLS,B_ROWS,B_COLS)=16, clock rate = 250 MHz
All IO mapped to corresponding single port RAM interface
Top level design pipelined with II=1
b_col_loop unrolled by 8
Memory bandwidth of matrices (B, C) is increased by factor 8
----------------------------------
Synthesized Hardware:
Throughput - 512, Latency clock - 512
Slack - 2.37 ns, Total Area - 35912 μm^2
```

SYSTEMC

SystemC is a set of C++ classes and macros which provide an event-driven simulation kernel deliberately mimicking HDL [90]. These facilities enable a designer to simulate concurrent processes, each described using plain C++ syntax. SystemC has semantic similarities to both HDLs, with a small syntactical overhead when used for hardware synthesis. However, it offers a greater range of expression, similar to object-oriented design partitioning and template classes. This provides a wide usage range, spanning from architecture modeling to RTL level as shown in Fig. A.1.

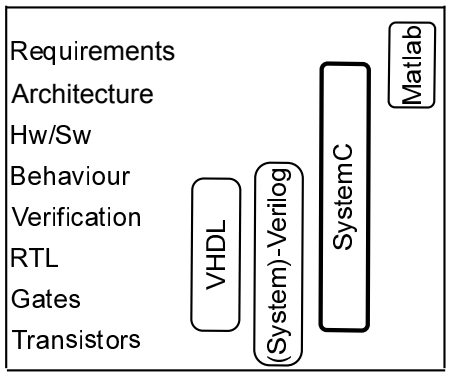


Figure A.1. Advantages and features of SystemC.

SYSTEMC EXAMPLE

The HLS implementation based on C/C++ in the previous section are compatible with SystemC. However, unlike C/C++, systemC has a major advantage that it also supports event based implementation similar to HDL. In A.4 a JTAG TAP is implemented using systemC, which looks very similar to a HDL implementation. Therefore, a single language can be used for a range of design stages (architecture to gate level implementation) as shown in Fig. A.1. In this thesis the fabricated ASIC was designed in SystemC. Critical blocks like systolic arrays, state machines where implemented in systemC with RTL like coding style. High level of abstraction was opted for implementing Cholesky decomposition. Verification and optimization (word-length) was performed by calling the top level systemC (design) function from Matlab.

Listing A.4 JTAG TAP SystemC

```
class jtag_tap_fsm : public sc_module {
  public :
    sc_in<bool> clk;
    sc_in<bool> trst;
    sc_in<bool> tms;
    // connecting signals
    sc_signal<TapState> tap_state_sig;
    // modules
    void tap_fsm();
    SC_HAS_PROCESS(jtag_tap_fsm);
    jtag_tap_fsm(const sc_module_name &name_) : sc_module(
        name_)
```

```
  {
    SC_CTHREAD(tap_fsm, clk.pos());
    reset_signal_is(trst, true);
  }
};

void jtag_tap_fsm::tap_fsm() {
  // trigger every pos edge of clock
  // sample tms and move states
  static TapState tap_state_reg = TAP_TEST_LOGIC_RESET;
  bool tms_tmp = 0;
  while(1) {
    wait();
    tms_tmp = tms.read();
    switch(tap_state_reg) {
      case TAP_TEST_LOGIC_RESET :
        if(tms_tmp == 0) tap_state_reg = TAP_RUN_TEST_IDLE;
        else tap_state_reg = TAP_TEST_LOGIC_RESET;
        break;
      case TAP_RUN_TEST_IDLE :
        if(tms_tmp == 1) tap_state_reg = TAP_SELECT_DR_SCAN;
        else tap_state_reg = TAP_RUN_TEST_IDLE;
        break;
      .
      .
      .  %fill up all the tap states (cases)
      .
      .
      case TAP_UPDATE_IR :
        if(tms_tmp == 1) tap_state_reg = TAP_SELECT_IR_SCAN;
        else tap_state_reg = TAP_RUN_TEST_IDLE;
        break;
    }
    // write to connectivity signal, to pull out the state
    tap_state_sig.write(tap_state_reg);
  }
}
```

Appendix B

ASIC Implementation in ST-28 nm FD-SOI

This chapter first describes the ST-28 nm FD-SOI technology node employed for the chip fabrication. The details are to highlight some of the key advantages at device level (mentioned in design trade-off Fig. 3.2) which are leveraged in the implementation. This is followed by description of chip implementation and measurement setup.

FD-SOI TECHNOLOGY

Going below and already at 28 nm technology node conventional planar transistors are becoming unfavorable to offer low power and optimal performance. The main reason for not able to meet the performance requirements is the short-channel effects which impacts the speed, and worsens with each subsequent technology nodes. One key technique which has been identified to continue the technology roadmap is fully depleted transistors.

Fully depleted transistors can be either tri-dimensional (e.g., FinFet and Tri-Gate) which wraps around the channel or planar which is a natural evolution of bulk technology. In the planar transistor version, thin film transistors are fabricated in an ultra-thin layer of silicon over a buried oxide, as shown in Fig. B.1. In case of ST-28 nm transistor, a 7 nm thin layer of silicon sits over a 25 nm buried oxide [75]. This process is comparatively simple with respect to tri-dimensional gates. The fully depleted transistors do not require doping or pocket implants in the channel to control the electrostatic and tune the threshold voltage. This provides a fundamental advantage over bulk in terms of variability. The buried oxide further offers a dielectric isolation of the transistor leading to lower leakage.

Although the technology node is 28 nm the actual channel length is 24 nm. The design kit also provides options to use lengths larger than minimum length. This is used to perform trade-off between leakage, performance and area cost. The options are called poly-bias (PB)-4,8,12, wherein the channel

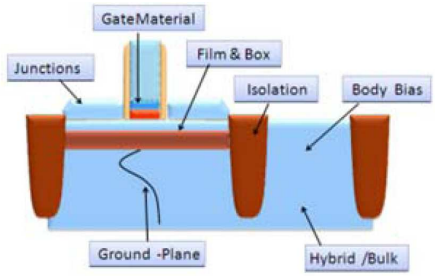


Figure B.1. 2D planar FD-SOI and transistor cross section. © 2013 IEEE. Reprinted, with permission, from [91].

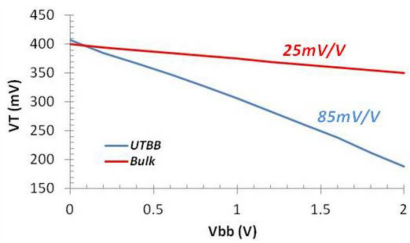


Figure B.2. Body-bias efficiency, © 2013 IEEE. Reprinted, with permission, from [91].

length is increased by lengths 4,8, and 12 respectively.

Body-biasing is well known technique enabling to combine high performance and low power for optimum energy efficiency. Unfortunately, in bulk technology the ability to change the threshold voltage (efficiency) is lowering with advanced nodes. In FD-SOI due to better electrostatics, transistors exhibit a higher body-bias efficiency over its bulk counterpart *i.e.*, 85 mV/V compared to 25 mV/V, as shown in Fig. B.2. Furthermore, it is important to note that the bias range in bulk technology is typically limited to ± 300 mV. In FD-SOI transistors body-bias is extended from -3V (RBB) to +3V (FBB). This provides designers a knob for optimization of energy, performance and leakage reduction.

ASIC IMPLEMENTATION DETAILS

The baseband massive MIMO processor is implemented in ST-28 nm technology with PB4 LVT transistors. The digital ASIC flow is a complex engineering task and certain details are purposefully kept out this thesis. Some of the main parts of the implementation e.g., verification, test subsystem, layout and measurement setup are described in this section.

VERIFICATION

Verification is a very important stage in any circuit design, and additional care is required when design is to be fabricated. Verification although may seem trivial sometimes takes up to 70% of design time, making it one of the major bottlenecks for time to market [92]. The ability of SystemC to generate both hardware and system models is exploited to create a custom verification flow. The key idea is to use the same SystemC sub-routines in testbench throughout the design flow as illustrated in Fig. B.3. The testbench is able to interface with the golden algorithm models in Matlab, which makes it easy for algorithm/architecture optimization as well.

Initial stages of design is verified at an software level using standard C compiler (gcc) and debugger (gdb). This leads to fast verification of the behavioral model or loosely timed micro-architecture. Cycle accurate models in SystemC is also verified in similar way with an optional ability to log waveforms. Once the cycle accurate SystemC models are developed hardware is generated by HLS tool (CatapultC®). It guarantees a functionally equivalent generation, nevertheless, it should be verified. This can be performed by running the same testbench and the generated RTL in a HDL simulator like Modelsim. For final signoff it also important to verify the netlist (post layout) with back-annotated timing information. This can also be performed by the same verification framework. This flow which uses a unified language for modeling, verification, and hardware design, leads to lower overall design time.

TEST SUBSYSTEM

Another important aspect of design is to provide a suitable testing framework. A standard approach for debugging and testing complex digital design is to

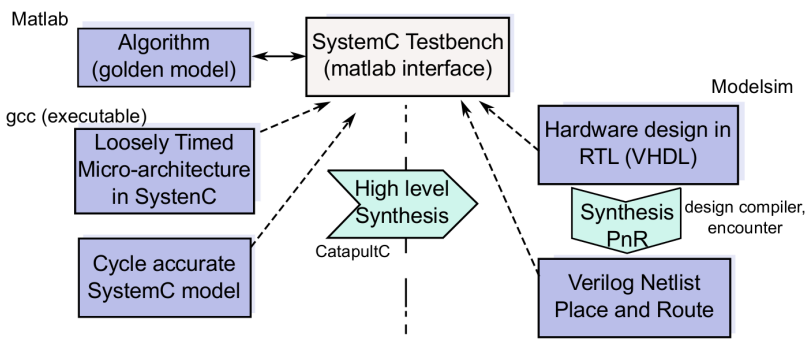


Figure B.3. Verification strategy used for chip development.

use JTAG [54]. In this work a minor modification to standard JTAG interface is performed to ease measurement. The modification is to provide an additional output pin *tdo_valid*, which when high indicates a valid *tdo*. This modification does not deviate from the standard JTAG functionalities.

The JTAG based test subsystem shown in Fig. B.4 mainly consists of the following parts:

- Test Access Port (TAP) is a standard JTAG state-machine.
- IR/DR decoder handles all the muxes and routing of TDI/TDO.
- An input/output buffer module stores matrices required by the design under test. This is used mainly for functionality tests, with the output buffers shifted back to host computer for comparison with golden data.
- BIST is based on Random Number Generator (RNG) generators, and can be used for power measurements.
- The configuration and status registers as the name suggests assist in test setup, design scenarios, triggering design etc.

These modules along with the developed software drivers (JTAG assembler) in Matlab provides for an easy and flexible test framework. This is important for a quick Silicon bring up and testing.

PLACE AND ROUTE

Verified design along with test subsystem and constraints are the primary input for backend flow. The backend flow may look quite obvious, but there are lot of engineering challenges involved. Especially considering constraints like limited area, IO limitations, and meeting timing.

The first step in backend flow is to perform synthesis *i.e.*, converting RTL into gate or cell level netlist. This procedure basically translates the RTL into equivalent gates provided by the technology libraries. Constraints on the timing, area and power dictates the gates which needs to be picked by the synthesis tool (Design Compiler). The netlist after synthesis is imported into place and route tool (Encounter), where floor planning is first performed. This is followed by power planning (rails, rings), clock tree synthesis and signal routing, after which the layout looks like the screenshot shown in Fig. B.5. Timing and signoff checks is performed before importing design into analog layout tool (Virtuso). Design rule checks and other violations are fixed as part of final signoff, before sending for fabrication.

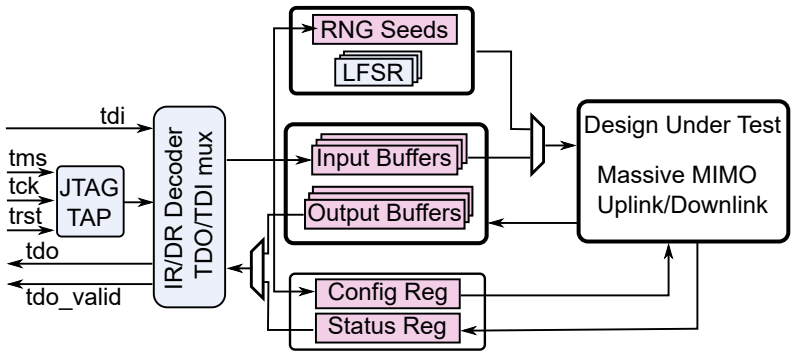


Figure B.4. Test subsystem used for debugging and measurements.

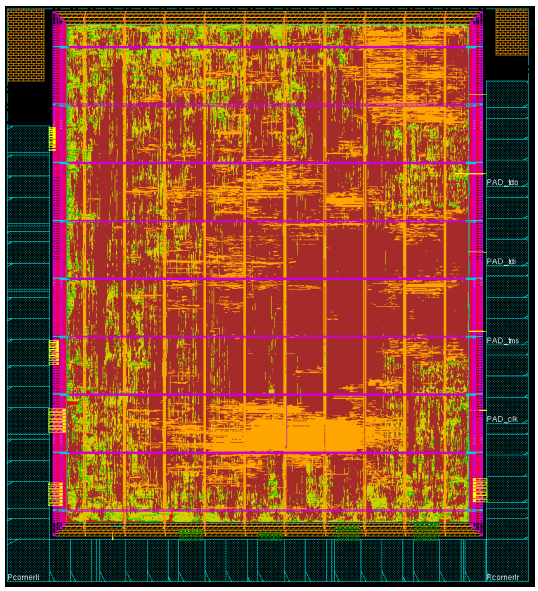


Figure B.5. Place and Route snapshot of the fabricated chip. Four power rings are used for supply and body-bias. IO pads are placed in 3 corners, although not conventional it does not effect functionality or fabrication.

MEASUREMENT SETUP

The fabricated chip was first bonded directly on the Printed Circuit Board (PCB). The PCB was then connected to pattern generator, logic analyzer, voltage sources, and source current measurement instrument. The connections

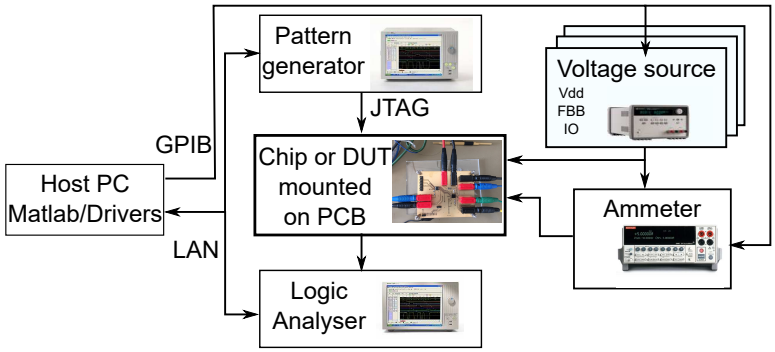


Figure B.6. Lab measurement setup.

from host PC to the logic analyzer was through ethernet, while voltage sources and measuring instruments where connected by a standard GPIB serial protocol. The host PC controlled all the instruments through Matlab, enabling a fully automated setup.

Appendix C

Popular Science Summary

In recent years, the wireless communication industry has seen a tremendous growth, and has had a profound impact on almost every aspect of society. The most noticeable change, although not limited to, is the cellular-phone or smart-phone revolution. Just in over a decade, cellular phones have transformed from an expensive voice-call only device, which few could afford, to a low cost device capable of browsing and streaming live high-definition videos. Subsequently, this has lead to an ever growing demand for high-speed wireless communication systems. The advancements in cellular-phone technology is just a forerunner of a whole new wave of wireless communication applications. There are many other exciting new technologies which are on the horizon, e.g., Internet of everything, smart power grid, smart homes, driverless cars etc. All these technologies have one fundamental thing in common, they typically require a reliable wireless communication link. The current state-of-the-art wireless communication deployment (4G) is already starting to approach the theoretical data speeds (data-rate). To meet the demands for higher data-rate, reliability and number of connected devices, evaluating new ground breaking technologies are becoming crucial. “Massive MIMO" is one such technology capable of meeting the above requirements. It is expected to be a key candidate for the next generation (5G) wireless communication systems.

Theoretical analyses of massive MIMO technology have shown promising performance over 4G, however, it is important to evaluate various design aspects involved in a practical implementation. As in most engineering designs, the typical implementation aspects are cost, design time, power consumption, and performance. Although the cost of integrated-chips have decreased incredibly over the years (making desktop computers, phones, etc. cheaper), it is still quite a relevant design parameter. Furthermore, the new 5G standard

should be low power to adhere to future environmental standards (so called "Green communication" requirements). In case of mobile devices the power consumption needs to be low due to limited battery capacity. Thus, the main implementation challenges are to lower cost and power while simultaneously meeting the computational processing demands.

In this thesis, the focus has been on analyzing the trade-offs involved in implementing digital processing for massive MIMO. Various algorithm optimizations were proposed to lower the overall computational complexity. Hardware and circuit level optimization were performed to improve performance (computational speed) and lower power consumption of the system. As part of the evaluation a massive MIMO chip was fabricated. Measurement results showed that the hardware cost and power consumption were reasonable considering the performance gains, making massive MIMO a promising candidate for next generation wireless communication systems.

Bibliography

[1] Ericsson, “Mobility Report - on the pulse of the networked society,” Jun 2016. "https://www.ericsson.com/ (visited on 8 Oct. 2016)".

[2] Verizon, “State of the Market: Internet of Things 2016,” April 2016. "https://www.verizon.com/ (visited on 8 Oct. 2016)".

[3] FCC USA, “Advanced Wireless Services-3,” Jan 2015. "http://wireless.fcc.gov/auctions/ (visited on 4 Nov. 2016)".

[4] Telecom Regulatory Authority India, “Spectrum Auctions,” 2016. "http://www.trai.gov.in/ (visited on 4 Nov. 2016)".

[5] METIS, “D1.1 Scenarios, requirements and KPIs for 5G mobile and wireless system,” April 2013. "https://www.metis2020.com/documents/deliverables (visited on 6 Nov. 2016)".

[6] J. Malmodin, P. Bergmark, and D. Lundén, “The future carbon footprint of the ICT and E&M sectors,” *Information and Communication Technologies*, 2013. "http://2013.ict4s.org/ (visited on 20 Oct. 2016)".

[7] C. Desset *et al.*, “Flexible power modeling of LTE base stations,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2858–2862, 2012.

[8] B. Noethen *et al.*, “A 105GOPS 36mm^2 heterogeneous SDR MPSoC with energy-aware dynamic scheduling and iterative detection-decoding for 4G in 65nm CMOS,” in *IEEE international Solid-State Circuits Conference (ISSCC) Dig. Tech. Papers*, Feb 2014.

[9] C. H. Chen, W. Tang, and Z. Zhang, "A 2.4mm^2 130mW MMSE-nonbinary-LDPC iterative detector-decoder for 4×4 256-QAM MIMO in 65nm CMOS," in *IEEE international Solid-State Circuits Conference (ISSCC) Dig. Tech. Papers*, Feb 2015.

[10] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, pp. 186–195, February 2014.

[11] C. Zhang, L. Liu, D. Marković, and V. Öwall, "A Heterogeneous Reconfigurable Cell Array for MIMO Signal Processing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, pp. 733–742, March 2015.

[12] J. Gordon, *A Thread Across the Ocean: The Heroic Story of the Transatlantic Cable*. Paw Prints, 2008.

[13] A. M. Nordsveen, "Mobiltelefonens historie i Norge," Nov. 2005. "(in Norwegian). Norsk Telemuseum.".

[14] N. Costa and S. Haykin, *Multiple-Input Multiple-Output Channel Models: Theory and Practice*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control, Wiley, 2010.

[15] A. Peled and A. Ruiz, "Frequency domain data transmission using reduced computational complexity algorithms," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. 964–967, April 1980.

[16] P. Horlin and A. Bourdoux, *Digital Compensation for Analog Front-Ends: A New Approach to Wireless Transceiver Design*. Wiley, 2008.

[17] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge University Press, 2003.

[18] E. Dahlman, S. Parkvall, and J. Sköld, *4G: LTE/LTE-Advanced for Mobile Broadband*. Academic Press, Elsevier/Academic Press, 2011.

[19] IEEE Standard, *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*, 2014. Revision of Std. 802-2001.

[20] T. Marzetta, "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 3590–3600, Nov. 2010.

[21] A. Tulino and S. Verdú, *Random Matrix Theory And Wireless Communications*. Foundations and Trends in communications and information theory, Now, 2004.

[22] E. Björnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: ten myths and one critical question," *IEEE Communications Magazine*, vol. 54, pp. 114–123, February 2016.

[23] X. Gao, O. Edfors, F. Rusek, and F. Tufvesson, "Linear Pre-Coding Performance in Measured Very-Large MIMO channels," in *IEEE Vehicular Technology Conference (VTC Fall)*, pp. 1–5, Sep. 2011.

[24] X. Gao, O. Edfors, F. Rusek, and F. Tufvesson, "Massive MIMO performance evaluation based on measured propagation data," *IEEE Transactions on Wireless Communications*, vol. 14, pp. 3899–3911, Jul. 2015.

[25] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics, 38(8), Aor.*, 1965.

[26] U. Manber and P. Norvig, "The power of the Apollo missions in a single Google search," Aug. 2012. "`https://search.googleblog.com/2012/08/` (visited on 4 Nov. 2016)".

[27] J. Rabaey, *Low Power Design Essentials*. Springer Publishing Company, 1st ed., 2009.

[28] C. Roth, P. Meinerzhagen, C. Studer, and A. Burg, "A 15.8 pJ/bit/iter quasi-cyclic LDPC decoder for IEEE 802.11n in 90 nm CMOS," in *Asian Solid State Circuits Conference (A-SSCC)*, pp. 1–4, Nov 2010.

[29] M. Obashi *et al.*, "A 27MHz 11.1mW MPEG-4 video decoder LSI for mobile application," in *IEEE international Solid-State Circuits Conference (ISSCC) Dig. Tech. Papers*, vol. 2, pp. 294–511, Feb 2002.

[30] K. Seta *et al.*, "50% active-power saving without speed degradation using standby power reduction (SPR) circuit," in *IEEE international Solid-State Circuits Conference (ISSCC) Dig. Tech. Papers*, pp. 318–319, Feb 1995.

[31] J. Tschanz *et al.*, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," in *IEEE international Solid-State Circuits Conference (ISSCC) Dig. Tech. Papers*, vol. 1, pp. 422–478 vol.1, Feb 2002.

[32] S. Payami and F. Tufvesson, "Channel measurements and analysis for very large array systems at 2.6 GHz," in *European Conf on Antennas and Propagation (EUCAP)*, pp. 433–437, Mar. 2012.

[33] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and Spectral Efficiency of Very Large Multiuser MIMO Systems," *IEEE Transactions on Communications*, vol. 61, pp. 1436–1449, April 2013.

[34] J. Vieira *et al.*, "A flexible 100-antenna testbed for Massive MIMO," in *Globecom Workshops*, pp. 287–293, Dec 2014.

[35] F. Rusek, D. Persson, B. Lau, E. Larsson, T. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Processing Magazine*, vol. 30, pp. 40–60, Jan. 2013.

[36] H. Prabhu, J. Rodrigues, O. Edfors, and F. Rusek, "Approximative Matrix Inverse Computations for Very-large MIMO and Applications to Linear Pre-coding Systems," in *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2710–2715, Apr. 2013.

[37] H. Prabhu, O. Edfors, J. Rodrigues, L. Liu, and F. Rusek, "Hardware efficient approximative matrix inversion for linear pre-coding in massive MIMO," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1700–1703, June 2014.

[38] F. Rosário, F. A. Monteiro, and A. Rodrigues, "Fast Matrix Inversion Updates for Massive MIMO Detection and Precoding," *IEEE Signal Processing Letters*, vol. 23, pp. 75–79, Jan. 2016.

[39] M. Wu and *et al.*, "Approximate matrix inversion for high-throughput data detection in the large-scale MIMO uplink," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2013.

[40] B. Kang, J. H. Yoon, and J. Park, "Low complexity massive MIMO detection architecture based on Neumann method," in *International SoC Design Conference (ISOCC)*, pp. 293–294, Nov 2015.

[41] G. W. Stewart, *Matrix Algorithms: Volume 1, Basic Decompositions*. Society for Industrial Mathematics, 1998.

[42] G. Golub and C. Van Loan, *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 1996.

[43] A. Molisch, M. Toeltsch, and S. Vermani, "Iterative Methods for Cancellation of Intercarrier Interference in OFDM Systems," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 2158 –2167, July 2007.

[44] L. Auer, "Acceleration of convergence," *Kalkofen, W (Eds),Numerical Radiative Transfer, Cambridge University Press*, 1987.

[45] S. Malkowsky *et al.*, "Implementation of Low-latency Signal Processing and Data Shuffling for TDD massive MIMO Systems," in *IEEE Workshop on Signal Processing Systems (SIPS)*, Oct. 2016.

[46] P. L. Chiu, L. Z. Huang, L. W. Chai, C. F. Liao, and Y. H. Huang, "A 684Mbps 57mW joint QR decomposition and MIMO processor for 4×4 MIMO-OFDM systems," in *IEEE Asian Solid State Circuits Conference (A-SSCC)*, pp. 309–312, Nov 2011.

[47] R. Gangarajaiah, L. Liu, M. Stala, P. Nilsson, and O. Edfors, "A high-speed QR decomposition processor for carrier-aggregated LTE-A downlink systems," in *European Conference on Circuit Theory and Design (ECCTD)*, pp. 1–4, Sept 2013.

[48] M. Shabany, D. Patel, and P. Gulak, "A Low-Latency Low-Power QR-Decomposition ASIC Implementation in 0.13μm CMOS," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, pp. 327–340, Feb 2013.

[49] Z.-Y. Huang and P.-Y. Tsai, "Efficient Implementation of QR Decomposition for Gigabit MIMO-OFDM Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, pp. 2531–2542, Oct 2011.

[50] I. Koren, *Computer Arithmetic Algorithms, Second Edition: Second Edition*. Ak Peters Series, Peters, 2002.

[51] H. Jagadish and T. Kailath, "A family of new efficient arrays for matrix multiplication," *IEEE Transactions on Computers*, vol. 38, pp. 149–155, Jan 1989.

[52] V. Kumar and Y.-C. Tsai, "On synthesizing optimal family of linear systolic arrays for matrix multiplication," *IEEE Transactions on Computers*, vol. 40, pp. 770–774, Jun 1991.

[53] W. M. Gentleman and H. Kung, "Matrix triangularization by systolic arrays," in *International Society for Optics and Photonics Annual Technical Symposium*, pp. 19–26, 1982.

[54] IEEE Standard, *IEEE Standard Test Access Port and Boundary-Scan Architecture*, 2001. 1149.1-2001.

[55] C. F. Liao, J. Y. Wang, and Y. H. Huang, "A 0.18nJ/Matrix QR decomposition and lattice reduction processor for 8×8 MIMO preprocessing," in *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pp. 161–164, Nov 2013.

[56] K. Mohammed and B. Daneshrad, "A MIMO Decoder Accelerator for Next Generation Wireless Communications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, pp. 1544–1555, Nov 2010.

[57] J. Hennessy, D. Patterson, and D. Goldberg, *Computer Architecture: A Quantitative Approach*. The Morgan Kaufmann Ser. in Computer Architecture and Design Series, Morgan Kaufmann, 2003.

[58] T. Jiang and Y. Wu, "An Overview: Peak-to-Average Power Ratio Reduction Techniques for OFDM Signals," *IEEE Transactions on Broadcasting*, vol. 54, no. 2, pp. 257–268, 2008.

[59] J. Tellado and J. M. Cioffi, "PAR reduction in multicarrier transmission systems," *ANSI Document, T1E1.4 Technical Subcommittee*, vol. 4, pp. 97–367, 1998.

[60] C. Studer and E. Larsson, "PAR-aware large-scale multi-user MIMO-OFDM downlink," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, pp. 303–313, 2013.

[61] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[62] S. Mohammed and E. Larsson, "Constant-envelope multi-user precoding for frequency-selective massive MIMO systems," *IEEE Wireless Communications Letters*, vol. 2, pp. 547–550, Oct. 2013.

[63] J. Tubbax, B. Come, L. Van der Perre, L. Deneire, S. Donnay, and M. Engels, "Compensation of IQ imbalance in OFDM systems," in *IEEE International Conference on Communications (ICC)*, vol. 5, pp. 3403–3407, May 2003.

[64] N. Kolomvakis, M. Matthaiou, J. Li, M. Coldrey, and T. Svensson, "Massive MIMO with IQ imbalance: Performance analysis and compensation," in *IEEE International Conference on Communications (ICC)*, pp. 1703–1709, June 2015.

[65] E. Bjornson, J. Hoydis, M. Kountouris, and M. Debbah, "Massive MIMO Systems With Non-Ideal Hardware: Energy Efficiency, Estimation, and Capacity Limits," *IEEE Transactions on Information Theory*, vol. 60, pp. 7112–7139, Nov. 2014.

[66] A. Molisch, *Wireless Communications*. Wiley - IEEE, Wiley, 2012.

[67] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Transactions on Information Theory*, vol. 45, pp. 1639–1642, Jul 1999.

[68] E. G. Larsson, "MIMO Detection Methods: How They Work," *IEEE Signal Processing Magazine*, vol. 26, pp. 91–95, May 2009.

[69] L. Liu, F. Ye, X. Ma, T. Zhang, and J. Ren, "A 1.1-Gb/s 115-pj/bit configurable MIMO detector using 0.13-μm CMOS technology," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, pp. 701–705, Sep. 2010.

[70] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 491–503, March 2006.

[71] D. Auras, R. Leupers, and G. Ascheid, "Efficient VLSI architectures for matrix inversion in soft-input soft-output MMSE MIMO detectors," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Jun. 2014.

[72] A. Burg, S. Haene, D. Perels, P. Luethi, N. Felber, and W. Fichtner, "Algorithm and VLSI architecture for linear MMSE detection in MIMO-OFDM systems," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2006.

[73] X. Chen, G. He, and J. Ma, "VLSI implementation of a high-throughput iterative fixed-complexity sphere decoder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, pp. 272–276, May 2013.

[74] H. L. Lin, R. C. Chang, and H. L. Chen, "A high-speed SDM-MIMO decoder using efficient candidate searching for wireless communication," *IEEE Transactions on Circuits and Systems II*, vol. 55, Mar. 2008.

[75] F. Arnaud *et al.*, "Switching energy efficiency optimization for advanced CPU thanks to UTBB technology," in *IEEE Electron Devices*, pp. 3.2.1–3.2.4, Dec. 2012.

[76] M. Winter *et al.*, "A 335Mb/s 3.9mm^2 65nm CMOS flexible MIMO detection-decoding engine achieving 4G wireless data rates," in *IEEE international Solid-State Circuits Conference Dig. Tech. Papers*, Feb 2012.

[77] R. Gangarajaiah, H. Prabhu, O. Edfors, and L. Liu, "A Cholesky Decomposition based Massive MIMO Uplink Detector with Adaptive Interpolation," in *ISCAS*, 2017. (accepted).

[78] L. Liu, J. Lofgren, and P. Nilsson, "Area-Efficient Configurable High-Throughput Signal Detector Supporting Multiple MIMO Modes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 9, pp. 2085–2096, 2012.

[79] S. Haene *et al.*, "OFDM channel estimation algorithm and ASIC implementation," in *European Conference on Circuits and Systems for Communications (ECCSC)*, pp. 270–275, July 2008.

[80] S. Gifford, C. Bergstrom, and S. Chuprun, "Adaptive and linear prediction channel tracking algorithms for mobile OFDM-MIMO applications," in *IEEE Military Communications Conference (MILCOM)*, vol. 2, Oct. 2005.

[81] L. Gor and M. Faulkner, "Power Reduction through Upper Triangular Matrix Tracking in QR Detection MIMO Receivers," in *IEEE Vehicular Technology Conference (VTC)*, pp. 1–5, Sept. 2006.

[82] S. Aubert, J. Tournois, and F. Nouvel, "On the implementation of MIMO-OFDM schemes using perturbation of the QR decomposition: Application to 3GPP LTE-A systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3236–3239, May 2011.

[83] 3GPP, "Physical Channels and Modulation, V10.2.0," Jun. 2011.

[84] P.-L. Chiu, L.-Z. Huang, L.-W. Chai, and Y.-H. Huang, "Interpolation-Based QR Decomposition and Channel Estimation Processor for MIMO-OFDM System," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, pp. 1129–1141, May 2011.

[85] P. Luethi, A. Burg, S. Haene, D. Perels, N. Felber, and W. Fichtner, "VLSI Implementation of a High-Speed Iterative Sorted MMSE QR Decomposition," in *ISCAS*, pp. 1421–1424, May 2007.

[86] L. Barbero and J. Thompson, "Fixing the Complexity of the Sphere Decoder for MIMO Detection," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 2131–2142, June 2008.

[87] C. Zhang, H. Prabhu, Y. Liu, L. Liu, O. Edfors, and V. Öwall, "Energy Efficient Group-Sort QRD Processor With On-Line Update for MIMO Channel Pre-Processing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, pp. 1220–1229, May 2015.

[88] C. Zhang, L. Liu, and V. Öwall, "Mapping Channel Estimation and MIMO Detection in LTE-Advanced on a Reconfigurable Cell Array," in *ISCAS*, pp. 1799–1802, May 2012.

[89] M. Fingeroff, *High-level Synthesis: Blue Book*. Xlibris Corporation, 2010.

[90] D. Black and J. Donovan, *SystemC: From the Ground Up*. Springer US, 2005.

[91] P. Magarshack, P. Flatresse, and G. Cesana, "UTBB FD-SOI: A process/design symbiosis for breakthrough energy-efficiency," in *Europe Conference Exhibition in Design, Automation Test*, pp. 952–957, Mar. 2013.

[92] J. Bergeron, *Writing Testbenches: Functional Verification of HDL Models*. Springer US, 2012.