Energy-Efficient VLSI Signal Processing for Multi-Band MIMO Systems

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Electrical Engineering

by

Chia-Hsiang Yang

2010
The dissertation of Chia-Hsiang Yang is approved.

_________________________________________
Miloš D. Ercegovac

_________________________________________
John D. Villasenor

_________________________________________
Kung Yao

_________________________________________
Dejan Marković, Committee Chair

University of California, Los Angeles
2010
To my parents.
# Table of Contents

1 Introduction .................................................. 1
   1.1 MIMO Technology ........................................ 2
   1.2 Energy-Efficient Design ................................. 3
   1.3 System Flexibility Requirements ...................... 4
   1.4 Overview of Previous Work ............................. 5
   1.5 Dissertation Outline ................................. 6

2 MIMO Signal Processing ................................... 8
   2.1 Diversity-Multiplexing Tradeoff ...................... 8
   2.2 MIMO Receiver Design Challenges .................. 9
   2.3 Sphere Decoding Algorithm ........................... 12
   2.4 Extracting Diversity and Spatial-Multiplexing Gains .15
   2.5 Complexity Reduction ................................ 17
   2.6 Channel QR Decomposition ........................... 17
   2.7 Chapter Summary ..................................... 18

3 CAD Methodology and Flow .............................. 19
   3.1 Simulink Design Environment ....................... 19
   3.2 Architecture Optimization Framework ............... 20
   3.3 Hardware Emulation .................................. 22
   3.4 FPGA-Aided Verification .............................. 23
   3.5 Chapter Summary ...................................... 24
# 4 Flexible Sphere Decoder DSP Architecture

## 4.1 Architecture Flexibility Requirements

- **4.1.1 Antenna Array Size**
- **4.1.2 Constellation Size**
- **4.1.3 Tree-Search Scheme**
- **4.1.4 Multiple Sub-Carriers**

## 4.2 Flexible Architecture Design

- **4.2.1 Multiplier Simplification for Scalable Antenna Array Size**
- **4.2.2 Candidate Enumeration for Scalable Modulation**
- **4.2.3 Simplified Candidate-Enumeration Scheme**
- **4.2.4 Bi-Directional Search for Adjustable Tree-Search Direction**
- **4.2.5 Data-Interleaved Processing for Scalable Number of Sub-Carriers**

## 4.3 Hardware Emulation Results

## 4.4 Hardware Complexity Comparison

## 4.5 Chapter Summary

# 5 Multi-Mode Single-Band Sphere Decoder Chip

## 5.1 Design Specifications and Flexibility

## 5.2 Multi-Core Architecture

- **5.2.1 Search Algorithm**
- **5.2.2 System Architecture**
- **5.2.3 Hybrid Memory Structure and Hierarchical Interconnect**
**List of Figures**

1.1 Compute platform choices. .......................... 2

1.2 Area- and energy-efficiency comparison with ISSCC chips. .......... 5

2.1 (a) Fundamental tradeoff between the diversity gain $d$ and spatial-multiplexing gain $r$. (b) Graphical interpretation of the diversity and spatial-multiplexing gains using channel matrix. ............ 10

2.2 Diversity-multiplexing tradeoff in MIMO communications. .......... 10

2.3 A MIMO communication system with multiple transmit and receive antennas. The goal of the MIMO decoder is to approach the ML performance with the lowest computational complexity. ........ 12

2.4 Sphere decoding formulation from exhaustive search to tree search: (a) original exhaustive search, (b) problem re-formulation after matrix transformation, and (c) resulting tree search. .............. 13

3.1 Simulink design environment. ................................ 20

3.2 Design environment for DSP architecture optimization. .......... 21

3.3 Delay parameter characterization. ............................ 22

3.4 BEE2 emulation platform. .................................... 23

3.5 FPGA-aided ASIC verification setup. ............................ 24

4.1 Design challenges and tradeoffs for large antenna size. Impact of antenna array size on (a) area and (b) critical-path delay. .......... 28

4.2 Closest-point selection scheme: (a) exhaustive search architecture, (b) SE enumeration for QAM constellation. ....................... 29
4.3 Basic architecture of (a) depth-first and (b) K-best algorithms. . . 32
4.4 Top-level block diagram of the flexible DSP architecture. . . . . . 34
4.5 Multiplier area reduction: (a) Area reduction due to folded archi-
tecture. (b) Gray-code representation and the simplified multiplier. 36
4.6 Low-complexity realization of the signal decision block. Decision
is made in the original constellation plane scaled by $R_{ii}$. . . . . . 41
4.7 Candidate enumeration scheme. This scheme simplifies the enu-
meration by taking advantage of the geometric relationship. . . . 42
4.8 Candidate enumeration circuit and the control signal transition. . 44
4.9 Circuit implementation of bi-directional search. . . . . . . . . . . 45
4.10 Channel coefficient generation. Only non-zero coefficients are stored,
which saves 48% of memory area. . . . . . . . . . . . . . . . . . . . . . 48
4.11 Summary of area reduction of the flexible DSP architecture. The
final design achieves $20 \times$ reduction in area. . . . . . . . . . . . . . . 49
4.12 Hardware emulation results for several design configurations. . . 50
4.13 Comparison of hardware complexity. . . . . . . . . . . . . . . . 52

5.1 Search scheme of multi-core architecture for $4 \times 4$ QPSK systems.
Search paths are distributed over all PEs to improve search speed. 57
5.2 Tree-partition scheme. All constellation points are sorted for each
decoding stage (antenna). . . . . . . . . . . . . . . . . . . . . . . . . 58
5.3 Candidate enumeration scheme in the 64-QAM constellation plane. 59
5.4 Block diagram of the multi-core architecture. Processing of 16
sub-carriers over 16 PEs is illustrated. . . . . . . . . . . . . . . . . 59
5.5 (a) Hybrid memory structure: $s$ is distributed over all PEs; $R$ and $\tilde{y}$ are shared by all PEs. (b) Hierarchical interconnect network is used to minimize the delay across PEs. 

5.6 Data structure of multi-core sphere decoding. The search path is recorded and can be transferred through the register $s$. 

5.7 Data-interleaved processing for multiple data streams. (a) Data streams are interleaved into each PE. (b) Data-stream interleaving is utilized to extend the processing interval. 

5.8 (a) Multidimensional data access and (b) shared-register architecture with (c) latch-based clock-gating technique. 

5.9 Die photo of the 16-core MIMO sphere decoder. 

5.10 BER performance for several design configurations. (a) $4 \times 4$, 16-QAM mode. (b) $16 \times 16$ mode, 64-QAM and 16-QAM. 

5.11 Measured power and minimum supply voltage versus operating frequency. 

5.12 Area and energy efficiency of dedicated $4 \times 4$ SVD chip and multi-mode sphere decoder. 

6.1 Receiver architecture for the proposed MIMO decoder. 

6.2 (a) Reconfigurable FFT architecture. (b) Detailed circuits of PUs. 

6.3 Iterative decoding scheme between inner and outer decoders. 

6.4 Low-power clock-gated register bank for soft outputs. 

6.5 Choice of the optimal architecture of 2048-point FFT: (a) the level of parallelism and (b) arithmetic factorization. 

6.6 Optimal PU configurations for 128-2048 FFTs.
6.7 Memory implementation for delay-line with length 512 and 1024. . 85
6.8 Die photo of the LTE-compliant sphere decoder chip. . . . . . . . 86
6.9 Power breakdown and measurement results of the LTE-SD chip. . 87
6.10 Area and energy efficiency comparison of SVD chip, SD chip, and
           LTE-SD chip. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
List of Tables

1.1 Summary of state-of-the-art MIMO sphere decoders. .......................... 6

4.1 Comparison of depth-first and K-best algorithms ............................. 31

4.2 Normalized area and delay due to wordlength reduction .................. 38

4.3 Hardware complexity comparison ............................................. 52

5.1 Summary of state-of-the-art MIMO sphere decoders. ........................ 72

6.1 3GPP-LTE downlink design specifications .................................. 76

6.2 Configuration of PUs for different radix representations .................. 80

6.3 Chip summary of the LTE-SD chip ......................................... 87

6.4 Detailed power consumption breakdown ..................................... 88
Acknowledgments

I would like to express my sincere gratitude to my advisor, Professor Dejan Marković. I was very lucky to have such a great advisor. His brilliance and sense of humor have made it enjoyable to work with him. In addition to technical knowledge, his constant drive for perfection and endless enthusiasm have guided me along the way to be a better researcher. This work could not be completed without his guidance and help. I am especially indebted to him for his encouraging support and help on my career plan.

I am thankful to Professor Kung Yao, Professor John Villasenor, and Professor Miloš Ercegovac for being my committee members. Their thoughtful reviews and helpful comments are appreciated. I am thankful to them for their constant encouragement.

I wish to thank my group members: Victoria Wang, Rashmi Nanda, Chengcheng Wang, Vaibhav Karkare, Sarah Gibson, Tsung-Han Yu, Henry Chen, Richard Dorrance, Kevin Dwan, Fengbo Ren, Yuta Toriyama, and Fang-Li Yuan. Several activities organized by them have made the group energetic and coherent. I am also grateful to them for helpful and interesting discussions about research and life. It was joyful to be a part of this group. I am particularly grateful to Victoria Wang and Sarah Gibson for proofreading many of my documents and Tsung-Han Yu for his help on designing the reconfigurable FFT processor.

I would like to thank many good friends around: Chiao-En Chen, Steve Hsu, Limin Lee, Pin-En Su, Ping-Hsuan Hsieh, E-Hung Chen, and ChuanKang Liang for their technical skills and valuable discussions. Szu-Ping Lee was the best outdoor-activities organizer. Yao-Yi Chiang was the best running partner. They
have made my time at UCLA a truly enjoyable and memorable one.

Last, but not least, I would like to thank my family. Their loving support is the greatest driver in my work.

This work was supported by C2S2 (Center for Circuit and System Solutions) and NSF (National Science Foundation). IBM TAPO (Trusted Access Program Office) program and Alchip Technologies, Inc. provided chip fabrication support.
Vita

           National Taiwan University, Taipei, Taiwan.

2002-2004  M.S. Electronics Engineering
           National Taiwan University, Taipei, Taiwan.

2006-2010  Research Assistant, Electrical Engineering Department
           University of California, Los Angeles.

2009-2010  Teaching Assistant, Electrical Engineering Department
           University of California, Los Angeles.

Publications

Decoding,” IEEE Transactions on Circuits & Systems I (TCAS-I), vol. 56, no. 10,

C.-H. Yang, T.-H. Yu, and D. Marković, “A 5.8mW 3GPP-LTE Compliant 8×8
MIMO Sphere Decoder Chip with Soft-Outputs,” in Proc. Int. Symposium on

C.-H. Yang and D. Marković, “A 2.89mW 50GOPS 16×16 16-Core MIMO Sphere
Decoder in 90nm CMOS,” in Proc. European Solid-State Circuits Conference


MIMO communication has recently received significant attention due to its potential to increase link robustness and channel capacity. Complex signal processing and coding schemes are being employed to improve spectral efficiency. Hardware realization of complex MIMO algorithms is quite challenging, especially for large antenna array and constellation sizes. Application-specific design is one option for MIMO signal processing, however the convergence of a variety of applications and standards on a single device argues for a more flexible implementation. From the link perspective, flexible antenna array and constellation sizes are required to maximize the MIMO channel utilization. Another requirement for OFDM-based systems is the support of multiple sub-carriers. These flexibility requirements make the integration of complex algorithms under a fixed power budget very challenging.

To address the complexity, energy efficiency, and flexibility challenges, this dissertation discusses hardware realization of MIMO sphere decoding algorithm. The algorithm can approach maximum likelihood performance with acceptable computational complexity. Existing architectures, K-best and depth-first, are not
scalable for increasing antenna array and constellation sizes. For depth-first architectures, the number of processing cycles is too high; for K-best architectures, the search range is fixed without taking the advantage of tree pruning.

This dissertation proposes a multi-core sphere decoder architecture that can resolve the limitations of the depth-first and K-best search methods. The architecture is based on a scalable sphere decoder DSP that supports antenna arrays from $2 \times 2$ to $16 \times 16$, modulations from BPSK to 64-QAM, and 8 to 128 sub-carriers. With the flexibility of forward-trace and backward-trace for each core, the sphere decoder can speed up the search and improve the BER performance. The multi-core architecture also improves energy efficiency by distributing computations over multiple processing elements.

Two MIMO sphere decoder chips have been designed and implemented to demonstrate several flexibility aspects. A 16-core architecture achieves 17.3 GOPS/mW and 9.45 GOPS/mm$^2$ in a low-$V_T$ 90-nm CMOS. The peak estimated data rate exceeds 1.5 Gbps over a 16 MHz channel. Extending the flexibility to multiple signal bands, the dissertation demonstrates an $8 \times 8$ 3GPP-LTE compliant decoder with a 128-2048 FFT block and soft outputs in 3.35mm$^2$, dissipating 13.83mW in a standard-$V_T$ 65-nm CMOS technology. The chip provides a peak data rate of 960Mbps in the $8 \times 8$, 64-QAM mode over a 20MHz channel. LTE specs are met with 5.8mW power for throughput of 480Mbps. These chips achieve energy efficiency comparable to that of dedicated designs, with only 2× overhead in area efficiency that is required to support multi-mode multi-band operation. The techniques developed in this dissertation enable energy-efficient ASIC implementation of advanced multi-band MIMO signal processing.
CHAPTER 1

Introduction

The advancement of the semiconductor industry has allowed a significant increase in functionality to be integrated on a single chip. This trend allows the integration of complex multidimensional data processing. There are many examples of hardware suited for multidimensional data. In high-performance computing, multi-core CPUs provide higher processing capabilities with lower power consumption. In wireless communications, multi-antenna (also known as MIMO) systems support higher data rate and result in better error performance. In medical electronics, multi-channel probes allows more measurements with finer resolution. Multidimensional data processing is attractive because it can improve system performance by constructively combining the data from multiple dimensions. These multidimensional systems improve performance in various aspects, but the system design is challenging due to the increased complexity.

Conventionally, data processing can be done on several compute platforms: dedicated, programmable, or general-purpose. There exists a tradeoff between power density and flexibility for these platforms, as shown in Fig. 1.1. General-purpose processors have the highest flexibility and therefore can implement any algorithm, but the nature of multiplexing increases the clock speed to improve the computational performance, which increases power density. In contrast, dedicated circuits have the lowest flexibility, but the power density can be minimized for specific applications. The architecture of programmable processors, including
digital signal processing (DSP) processors and Field Programmable Gate-Arrays (FPGAs), is the compromise between general-purpose processors and dedicated circuits. For mobile communication devices, a dedicated design is the preferred option for achieving a longer battery life. However, the rapid evolution of DSP algorithms argues for a more flexible approach. The design goal of this work is to add flexibility on a dedicated platform and further reduce power for MIMO signal processing.

1.1 MIMO Technology

Multi-input multi-output (MIMO) communication has recently received significant attention due to its potential to increase transmission rate and/or signal reliability. Inspired by limited spectrum resources, complex signal processing and coding schemes for multi-antenna technology are employed to improve spectral efficiency. Data streams are transmitted and received through multiple transmit and receive antennas. This allows transmission of independently-faded replicas
from the same signal source as well as independent data streams from multiple sources [1,2]. By constructively combining these signals at the receiver, higher data rates and more reliable communication can be achieved.

MIMO techniques are extensively deployed in several standards, such as IEEE 802.11n, WiMAX and 3GPP-LTE (long-term evolution) cellular standard, ranging from local area networks to cellular systems. Up to four transmit/receive antennas are allowed in existing standards. However, communication systems with more antennas are slowly emerging: systems with $12 \times 12$ antenna arrays have been demonstrated [3]; millimeter-wave systems (operating at 60 GHz and beyond) also provide the possibility of mobile devices with multiple (8+) antennas [4]. For baseband signal processing, the design challenges lie in the MIMO decoder for complex MIMO algorithms. Among several decoding schemes, the sphere decoding algorithm is one of the most promising solutions to achieve maximum likelihood (ML) performance with practical computational complexity [5] for MIMO systems. However, hardware realization of sphere decoding is challenging for large antenna-array systems and/or high-order modulations. Increasing antenna array size dictates quadratic increase in the number of multipliers. Increasing modulation size adds complexity to sorting and comparison circuitry.

1.2 Energy-Efficient Design

Energy efficiency is a key design parameter for battery-based devices. Energy efficiency is the metric to quantifies the energy required to perform an operation [6]. It is defined by

\[
\text{Energy efficiency} = \frac{\# \text{ operations}}{\text{energy}} = \frac{\text{op/s}}{\text{pJ/s}} = \frac{\text{GOPS}}{\text{mW}},
\]

\[ (1.1) \]
where GOPS stands for giga operations per second. In this work, the number of operations is quantified in terms of 12-bit addition. To get an insight into energy efficiency, Eq. 1.1 can be elaborated by

\[
\text{Energy efficiency} = \frac{f \cdot op}{P_{\text{sw}} + P_{\text{leakage}}} = \frac{f \cdot op}{f \cdot C_{\text{sw}} \cdot V_{DD}^2 + P_{\text{leakage}}}. \tag{1.2}
\]

Neglecting the leakage power, energy efficiency can be improved by minimizing the switching capacitance \(C_{\text{sw}}\) and supply voltage \(V_{DD}\) for a fixed \(op\) and \(f\) (operating frequency). Another key design parameter is area efficiency, which is the silicon area required to perform an operation. Sensitivity analysis suggests the supply voltage should be around 0.4 V in a 90-nm CMOS process to achieve the optimized design with balanced sensitivity of design parameters [6]. A MIMO Singular Value Decomposition (SVD) chip [7] is used to demonstrate the achievable energy, area-efficiency, and design complexity through an Area-Energy-Delay optimization methodology, as shown in Fig. 1.2. The dedicated chip supports 4×4 MIMO systems and 16 sub-carriers. It achieves 2.1 GOPS/mW energy-efficiency and 20 GOPS/mm\(^2\) area-efficiency at 0.4 V. Compared with the baseband and multimedia processors published in ISSCC, this chip achieves the highest area and energy efficiency. Using the SVD chip as a starting point, the goal of this work is to add flexibility without compromising energy and area efficiency.

### 1.3 System Flexibility Requirements

There is a growing demand for flexibility to achieve seamless connectivity in devices that can use multiple standards. For example, the flexibility in supporting mobile cellular and wireless network systems allows users to access the available network with the highest signal strength. From the link performance perspective,
flexibility can also improve transmission rate and link reliability. For example, MIMO systems can improve performance by changing coding and modulation schemes, allowing MIMO decoders to dynamically adapt to the channel conditions. To control robustness and data rate simultaneously, flexibility in the antenna array and constellation size is necessary. Another flexibility requirement for Orthogonal Frequency Division Multiplexing (OFDM)-based systems is the support of multiple frequency sub-carriers.

1.4 Overview of Previous Work

Existing work is limited to 4×4 antennas and up to 64-QAM modulation [8–12], as shown in Table 1.1. Previous work is based on a direct-mapped implementation, which results in large hardware complexity and limited performance. For example, increasing the antenna array size to 8×8 was made possible by simplifying the modulation to QPSK [11]. Only one of the search methods, K-best
Table 1.1: Summary of state-of-the-art MIMO sphere decoders.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Array size</th>
<th>Modulation</th>
<th>Search method</th>
<th>$N_{car}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Garrett, JSSC’04 [8]</td>
<td>4×4</td>
<td>16-QAM</td>
<td>Depth-first</td>
<td>1</td>
</tr>
<tr>
<td>Burg JSSC’05 [9]</td>
<td>4×4</td>
<td>16-QAM</td>
<td>K-best</td>
<td>1</td>
</tr>
<tr>
<td>Guo, JSAC’05 [10]</td>
<td>4×4</td>
<td>16-QAM</td>
<td>K-best</td>
<td>1</td>
</tr>
<tr>
<td>Shabany, ISSCC’09 [12]</td>
<td>4×4</td>
<td>64-QAM</td>
<td>K-best</td>
<td>1</td>
</tr>
</tbody>
</table>

or depth-first was chosen for single-carrier systems. The need for a large number of multiplications and inefficient enumeration scheme are the major design challenges in existing designs [8–12].

To address these challenges, algorithm, architecture, and circuit parameters have to be jointly considered in the design process. Additionally, future systems will demand a high degree of flexibility for multiple operation modes. As the system complexity and functionality requirements increase, the future MIMO decoders should be able to support larger antenna array sizes, provide flexibility in a variety of modulation schemes, and support multiple sub-carriers for OFDM-based systems.

1.5 Dissertation Outline

The dissertation is organized as follows. Chapter 2 reviews MIMO decoding algorithms and introduces sphere decoding as a practical solution to approach maximum likelihood detection. Design methodology for power and area minimization is presented in Chapter 3. A flexible sphere decoder DSP kernel is described in Chapter 4. A multi-core architecture for MIMO sphere decoder is proposed to
enhance decoder performance. The implementation of a multi-mode single-band MIMO sphere decoder chip and its measurement results are described in Chapter 5. Extending the flexibility to multiple bands, a multi-mode multi-band sphere decoder compliant with the LTE standard is implemented and tested. The details of key building blocks and measurement results are described in Chapter 6. Chapter 7 concludes this work and discusses future research directions.
CHAPTER 2

MIMO Signal Processing

Multiple antennas provide spatial degree of freedom, in addition to time and frequency. The use of multiple antennas can significantly enhance the system performance [1]. MIMO algorithms greatly differ in their ability to trade off diversity and spatial-multiplexing, and each algorithm presents unique implementation challenge due to its structure and computational complexity. This chapter reviews key MIMO decoding algorithms and discusses their computational complexity and suitability for hardware implementation.

2.1 Diversity-Multiplexing Tradeoff

MIMO technology is used to improve signal reliability and/or transmission rate, which can be quantified through diversity and spatial-multiplexing gains, respectively. Diversity is characterized by the independently-faded paths for one transmitted symbol. The diversity gain $d$ is achieved when the error probability decays as $1/SNR^d$ asymptotically [13]. Lower error probability or higher diversity gain compensates for the path loss, thereby increasing the communication range. The spatial-multiplexing gain is characterized by spatially independent data streams over multiple antennas. The spatial-multiplexing gain is equal to $r$ when channel capacity increases as $r \cdot \log(SNR)$ [13]. Therefore, higher spatial-multiplexing gain supports higher data rate within the same bandwidth (with a vanishingly
Both gains can be improved using a larger antenna array, but for a given antenna array size, there is a fundamental tradeoff between these two gains [13]. For a system with $M$ transmit and $N$ receive antennas, the maximum achievable spatial-multiplexing gain is $\min\{M, N\}$. Given a spatial-multiplexing gain $r$, the achievable maximum diversity gain $d^*$ is $(M - r) \times (N - r)$, as shown in Fig. 2.1(a). The maximum diversity gain $M \times N$ can be achieved by sending the same symbol over all antennas, and the maximum capacity gain can be achieved by transmitting $\min\{M, N\}$ independent symbols simultaneously. The relationship between the optimal diversity gain ($d^*$) and the spatial-multiplexing gain ($r$) can be explained using the channel matrix $H$, as shown in Fig. 2.1(b). Spatial-multiplexing gain $r$ is contributed by the number of independent row/column vectors, i.e., rank, which has the maximum value of $\min\{M, N\}$. Diversity gain is extracted from the remaining $(M - r) \times (N - r)$ paths which are not used for rate increase. Among various MIMO algorithms, Alamouti coding [14] uses data redundancy to increase diversity while compromising spatial-multiplexing gain. In contrast, the Bell Labs Layered Space-Time (BLAST) [15,16] algorithms allocate data streams to spatial sub-channels to maximize spatial-multiplexing gain while sacrificing diversity gain, as shown in Fig. 2.2.

### 2.2 MIMO Receiver Design Challenges

In a MIMO system, the received signal can be modeled as $y = Hs + n$. Vectors $s$ and $n$ represent the transmitted symbols and Additive White Gaussian Noise (AWGN), respectively; matrix $H$ characterizes the MIMO channel whose elements $h_{ij}$ describe the gains of each Tx/Rx antenna pair. Note that the channel model considered here assumes that the channel changes slowly such that
Figure 2.1: (a) Fundamental tradeoff between the diversity gain $d$ and spatial-multiplexing gain $r$. (b) Graphical interpretation of the diversity and spatial-multiplexing gains using channel matrix.

Figure 2.2: Diversity-multiplexing tradeoff in MIMO communications.
the channel gains are constant in the symbol interval and the bandwidth of the
signal is much narrower than the coherence bandwidth, i.e., frequency-flat fading is assumed [17]. In general, the coherence time is longer than the symbol interval, but the frequency-flat fading assumption may not hold for wideband systems. The condition holds for OFDM systems, which transmit signals over several narrow-band channels such that each sub-carrier experiences a frequency-flat fading channel.

MIMO receivers are used to reconstruct the signals corrupted by the multipath MIMO channel, as shown in Fig. 2.3. The channel state information (CSI), i.e., channel matrix $H$, is assumed to be known at the receiver through channel estimation [17]. The ML estimate is optimal in terms of bit error rate (BER) performance for equal priors [18], which is given by

$$\hat{s}_{ML} = \arg \min_{s \in \Lambda} \| y - Hs \|^2,$$

where $\hat{s}_{ML}$ is chosen over the set of possible constellation points $\Lambda$ such that the (square) Euclidean distance is minimized. The goal of MIMO receivers is to achieve ML performance with reduced computational cost. The computational complexity of Eq. 2.1 is exponential, which is infeasible for real-time, large antenna-array systems. There are three major decoding/detection schemes for multi-antenna systems [1, 19]:

- Linear receivers, including zero-forcing (ZF), minimum mean-squared error;

- Nonlinear receivers, such as decision-feedback and ordered successive interference cancellation;

- Sphere decoder.
Figure 2.3: A MIMO communication system with multiple transmit and receive antennas. The goal of the MIMO decoder is to approach the ML performance with the lowest computational complexity.

Among these decoding schemes, the sphere decoder achieves near-ML performance with the lowest computational complexity.

2.3 Sphere Decoding Algorithm

Sphere decoder is a tree-search algorithm. Spatial interference caused by other receive antennas can be reduced by decomposing the channel matrix $H$ as $H = Q \cdot R$, where $Q$ is a unitary matrix and $R$ is an upper-triangular matrix. Therefore, the ML estimate can be written in two forms:

$$\hat{s}_{ML} = \arg \min_{s \in \Lambda} \|R(s - s_{ZF})\|^2$$

(2.2)

and

$$\hat{s}_{ML} = \arg \min_{s \in \Lambda} \|\tilde{y} - Rs\|^2,$$

(2.3)

where $s_{ZF} = (H^H H)^{-1} H^H y$ is the ZF (unconstrained ML) estimate and $\tilde{y} = Q^H y$. The detailed derivation can be found in [20]. Note that the diagonal
Figure 2.4: Sphere decoding formulation from exhaustive search to tree search: (a) original exhaustive search, (b) problem re-formulation after matrix transformation, and (c) resulting tree search.

Elements of $\mathbf{R}$ are real, while the off-diagonal elements are complex. The unique upper-triangular structure of $\mathbf{R}$ allows sequential decoding of symbols, instead of considering all the symbols simultaneously. This is illustrated in Fig. 2.4(b). Taking advantage of monotonically increasing norm accumulation, the search range is bounded within a hyper sphere since the paths with larger Euclidean distance (outside the sphere) can be discarded without losing the ML solution [5].

From the decoding perspective, MIMO sphere decoding can be viewed as a tree-search process, as shown in Fig. 2.4. In the exhaustive search, Fig. 2.4(a), all possible combinations are examined in order to find the optimal solution. In the tree search, Fig. 2.4(b, c), only the highlighted nodes are feasible solutions. The root node is a dummy node, indicating the start of the search process that...
goes from antenna $M$ to antenna 1. The white-filled nodes are outside the search radius and therefore discarded. The whole search space of this tree is the same as that of exhaustive search but the ML decoding solution has to be found by only visiting nodes within a search radius (the sphere). The ML estimate is viewed as the shortest path (with the minimum Euclidean distance, defined in Eq. 2.3) in a tree topology, in which one possible constellation point denotes one node and each row of the $R$ matrix is mapped to each level of the tree whose edges are weighted by channel coefficients. The search radius continues to shrink during the decoding process, in order to narrow the search range, as long as a solution with a smaller Euclidean distance (compared to the current search radius) is found. The choice of the initial search radius is related to the channel SNR [5]. This feature makes sphere decoding achieve ML performance with polynomial complexity $O(M^3)$, on average, instead of exponential complexity ($|\Lambda|^M$), where $|\Lambda|$ is the constellation size and $M$ is the number of antennas. The complexity reduction of sphere decoding could be very significant: $O(10^{25})$ for $16\times16$ antennas and 64-QAM, for example.

The major tree-search algorithms such as depth-first and breadth-first are naturally applicable to sphere decoding [21]. Depth-first explores as far as possible along each branch, but breadth-first searches the neighbors first. K-best approximates a breadth-first search by keeping only K best branches (with the smallest partial Euclidean distance) at each level. The depth-first architecture has longer latency than the K-best architecture because only one node is examined in each processing cycle. ML performance may not be guaranteed in practice due to a finite number of processing cycles and finite buffer size. This problem would be more severe as the size of the search tree grows (due to the increasing antenna size and/or constellation size) and the throughput requirement is also higher, i.e., the processing intervals are shorter. In contrast, K-best has no ability to improve
the error performance even if additional processing cycles were provided. This is because trace-back is not allowed and no search radius is set to confine the search range [22]. In light of these constraints, a multi-core architecture is proposed to extend the number of processing cycles and allow flexibility for trace-back. We start by presenting the processing element, and then describe how to configure many processing elements in a multi-core sphere-decoder architecture.

2.4 Extracting Diversity and Spatial-Multiplexing Gains

With the flexibility in decoding distinct coding and modulation schemes, sphere decoders can effectively explore the entire diversity-multiplexing tradeoff curve. The original data type for sphere decoding is uncoded. By rearranging input data, which changes the effective array size, sphere decoding is capable of decoding space-time codes. This configuration improves the error probability and increases diversity gain. The data rate can be maximized by allocating different modulations over MIMO spatial sub-streams according to channel conditions, thereby increasing spatial-multiplexing gain.

A unified sphere decoder architecture for utilizing diversity and spatial-multiplexing gains along the tradeoff curve is described here. We demonstrate that adding flexibility in varying antenna size and varying modulations are the key features for tuning the operating mode. In order to maximize diversity gain, we have to supply to the receiver multiple independently faded replicas of the same symbol, so that the error probability is reduced [2,13]. The data replicas can be sent in the space and/or time dimensions. Since a unified signal model can be developed for these space-time (ST) coding schemes, the same sphere decoder architecture can be used with some data rearrangement. The sphere decoding algorithm supporting algebraic ST codes [23] and linear dispersion code [24] are reported in prior
work. For these coded data streams, the ML estimate can be written as

$$\hat{s} = \arg \min_{s \in \Lambda} \| y - Bs \|^2,$$

(2.4)

where matrix $B$ depends on code generators and the channel matrix. By interpreting $B$ as $H$ in the original signal model (Eq. 2.1), the sphere decoding algorithm can be applied. Since the matrix dimension is changed due to data rearrangement in the preprocessing stage, the equivalent antenna array size will be changed accordingly. For example, repetition coding by 2 in the space domain for an $8 \times 8$ system will be transformed into data processing in a $4 \times 4$ system (only half of the symbols need to be decoded).

Spatial-multiplexing gain is characterized by increased channel capacity. To maximize the spatial-multiplexing gain, we should either allow the data rate to scale with the SNR or assign different data rates to different spatial sub-streams for a fixed SNR [13]. The modulation scheme should be adaptive according to channel conditions: a larger constellation is applied to sub-streams with higher SNR, and a smaller constellation is applied to sub-streams with lower SNR. In principle, this transmission strategy utilizes water-filling in the space domain.

Therefore, from the transmission performance perspective, additional degrees of freedom are considered in the design in order to take full advantage of the diversity and spatial multiplexing gains available in MIMO wireless channels. Tuning over a range of diversity-multiplexing points is possible by varying the antenna array size and modulation scheme, as in Fig. 2.2. Also, flexibility and scalability are key requirements in the design of multi-mode systems.
2.5 Complexity Reduction

Several effective methods such as detection ordering, candidate enumeration, and search radius setting are applied to improve error performance and/or to reduce the complexity of the basic sphere decoding algorithm [20, 25]. Most of these methods are executed at the preprocessing stage, but candidate enumeration is considered in the tree searcher. For each level, the order of constellation point enumeration is another important factor in improving the search speed. Schnorr-Euchner (SE) enumeration suggests traversing the constellation candidates according to the distance increment in ascending order [9]. The underlying idea of SE enumeration is to speed up the search of the globally optimal solution by taking advantage of locally optimal solutions. Since the candidates for each node have already been sorted, the remaining sibling nodes can be discarded if the node is already outside the search radius. Therefore, the first estimate $\hat{s}_i^{(1)}$ for each row $i$ of $Q^H y - R s$ in Eq. 2.3 is the symbol with minimum distance between $b_i$ and $R_i A$ as in Eq. 2.5, letting $\hat{y} = Q^H y$. Note that $b_i$ is decided by the previously decoded symbols ($s_M$ to $s_{i+1}$), so choosing the closest point for each level does not guarantee finding the solution with the minimum overall Euclidean distance—the ML solution.

$$\hat{y}_i - \sum_{j=i}^{M} R_{ij} s_j = b_i - R_{ii} s_i,$$

where $b_i = \hat{y}_i - \sum_{j=i+1}^{M} R_{ij} s_j$.  

(2.5)

2.6 Channel QR Decomposition

In the block-fading channel, QR decomposition of the MIMO channel is only computed at packet rate rather than at symbol rate. Therefore, QR decomposition is assumed to be computed in DSP/general-purpose processors rather than...
in ASICs. The most commonly used methods for QR decomposition are Grahm-Schmidt decomposition, Householder transformation, and Givens rotations [26]. Several modifications such as division-free and division-reduction methods are proposed to simplify the operation in the original algorithm [27]. For hardware realization, [28] proposed an algorithm suitable for fixed-point implementation and [29] proposed a COordinate Rotation DIgital Computer (CORDIC)-based triangular systolic array architecture to reduce latency.

2.7 Chapter Summary

This chapter introduced MIMO decoding. To achieve ML performance, which is optimal in terms of bit-error performance, the sphere decoding algorithm is adopted due to its computational simplicity. The idea of sphere decoding is to search possible solutions in a tree topology instead of a trellis. By updating the tree-search radius unlikely solutions can be discarded without losing the ML estimate. The flexibility requirements of antenna array size and modulation scheme from the link performance perspective are also discussed. To implement the algorithm in silicon, an integrated computer-aided design (CAD) methodology that includes architecture design, functional verification, chip implementation and testing is needed. The design methodology will be introduced in the next chapter.
CHAPTER 3

CAD Methodology and Flow

An integrated design methodology is adopted in this work to incorporate algorithm, architecture, and circuit implementation in a highly automated environment. Hierarchical design methodology allows us to scale design optimization concepts to very complex designs. A graphical Simulink/Matlab development environment offers bit-true, cycle-by-cycle hardware-equivalent modules for simulation, and then translates this description into logic gates. Due to the limited capacity of single FPGA, BEE2 platform [30, 31] is used to accommodate the whole system and speed up emulation. Chip synthesis, placement, and routing are finished with the aid of automatic commercial back-end tools. The implemented chips are tested with an FPGA-aided verification environment.

3.1 Simulink Design Environment

Simulink/Matlab design environment provides graphical design entry for rapid hardware architecture evaluation [32]. Traditionally, circuit design for communication signal processing is divided into two stages: algorithm design and circuit implementation. Algorithm designers use C/C++ or Matlab for system simulation, and then the designed architecture is implemented by circuit designers using hardware description language (HDL). There are usually several iterations between two design stages to ensure the final design satisfies the specifications. In
this work, Xilinx System Generator (XSG)/SynDSP (now Synphony) block-sets [33, 34] are used to build hardware-equivalent modules, which leverages cycle-accurate software simulation. Fig. 3.1 shows a Simulink model for the MIMO sphere decoding. Area information is extracted by resource estimator (Xilinx) in terms of number of slices in Simulink in the early design stage. Automatically generated HDL codes are exported to design compiler (Synopsys) for a more accurate area and power estimation.

3.2 Architecture Optimization Framework

The integrated Matlab/Simulink environment provides data flow graphical description, which allows for design space exploration [35]. The developed in-house tool optimizes the hardware architecture through several architectural transformations: retiming, scheduling, and unfolding (parallelism) [36]. The optimal design is chosen from possible realizations based on design criteria. In addition, quantization effects due to finite wordlength are considered in the simulation. Data-path wordlengths are optimized with in-house tools [37, 38].
which combines the auto-generated modules and the custom modules is shown in Fig. 3.2. Performance-driven modules are refined to enhance the circuit performance. For example, gated-clock nets are created manually for power saving. This design flow shortens the design time and improves the circuit performance simultaneously.

To operate the circuit at the optimal operating point (around 0.4 V as suggested in [7, 39]), the delay parameters are characterized from the FO4 inverter delay for calibration. Fig. 3.3 shows a typical delay versus supply voltage curve from transistor-level simulation. Since the timing library of standard cells is for standard supply voltage $V_1$ (1.0 V in CMOS 90-nm process), the design is overdesigned by $D_2/D_1$ for chip synthesis. For example, designing for 16 MHz at 0.4 V ("optimal point" in Fig. 3.3) translates to 444 MHz at 1.0 V (reference point) in a low-$V_T$ 90-nm CMOS technology in the SS corner. The maximum clock frequency is designed to be 500 MHz at 1.0 V for process variations tolerance and aggressive voltage scaling.
3.3 Hardware Emulation

The drawback of Simulink-based design flow is its lengthy simulation time, which can be mitigated by FPGA-based hardware emulation [40]. FPGA-based hardware emulation and rapid prototyping have become an attractive solution, which can provide up to 100× faster simulation speed than software simulation [41]. Xilinx University Program (XUP) board (with one Virtex-II Pro 2VP30 FPGA) [42] is used to develop the hardware/software cosimulation environment for small circuits. In this case, the hardware modules built in the Simulink is replaced with the configured FPGA to speed up simulation. Due to the limited capacity of XUP board, BEE2 platform is used for whole system emulation. The BEE2 consists of 5 high-performance Vertex-II 2VP70 Pro FPGAs (~10M equivalent logic gates total). Each FPGA embeds a PowerPC core which minimizes the latency between the microprocessor and reconfigurable logic. Four user FPGAs are used for computation and one for control (control FPGA) [31], as shown in

![Figure 3.3: Delay parameter characterization.](image)
Fig. 3.4. With high speed bandwidth, low latency links, BEE2 provides a virtual single FPGA of five times the capacity [30]. In the BEE2 design flow, hardware-equivalent modules are packaged as hardware processes, which can communicate with software processes for data access and exchange.

3.4 FPGA-Aided Verification

ASIC verification is performed with the use of an IBOB FPGA board [43] for pattern generation and data analysis, as shown in Fig. 3.5. Test vectors are stored on the FPGA, which stimulates the ASIC over two high-speed Z-DOK+ connectors. Although the connector itself supports data rates of up to 6 Gbps, the FPGA I/O limits the date to 330 Mbps. Either an internal or an external clock source can be used to provide flexibility and a wide range of operating frequency. The outputs of the ASIC are captured and fed into block RAMs for analysis. The I/O interface between the client PC and the IBOB board is built on
the BEE Platform Studio environment [44]. The real-time operation, easy FPGA programmability, and low cost make the FPGA approach favorable as compared to traditional pattern generation and logic analysis systems.

3.5 Chapter Summary

CAD tools and hardware infrastructure for system development was described. Matlab/Simulink based graphical design environment provides hardware-equivalent modules for bit-true, cycle-accurate algorithm model. The built-in hardware-cost database allows efficient hardware resource mapping and optimization within the same environment. Wordlength and architectural optimizations are executed to minimize power and area. To accelerate functional verification, an FPGA-based hardware emulation is used. Chip synthesis, placement and routing are performed using commercial back-end tools. The fabricated chip is tested with the aid of FPGA infrastructure due to its real-time operation, easy programmability, and low-cost. A flexible DSP kernel for sphere decoding, which is the key building
block for hardware demonstrations, will be described. Design challenges and proposed solutions will be discussed.
CHAPTER 4

Flexible Sphere Decoder DSP Architecture

To address the limitations faced by prior work (Table 1.1), a flexible DSP architecture for sphere decoding is proposed [45,46]. The proposed architecture provides flexibility and scalability of search method (K-best, depth-first), antenna array size (up to $16 \times 16$), number of sub-carriers (up to 128), and modulation scheme (up to 64-QAM). The multipliers for metric calculation are reused and downsized through hardware sharing and arithmetic simplification to support larger antenna array sizes. Higher-order modulation schemes have been supported via decision plane transformation and simplified metric enumeration. Bi-directional search is supported by reconfigurable registers. By adding appropriate latency for a recursive operation, multiple data streams can be interleaved. The main features of the flexible DSP architecture are as follows:

- Decoding on $R_s$ plane, rather than $s$ plane for hardware simplification;
- Gray code based processing for arithmetic simplification and memory reduction;
- A bi-directional shift delay-line based register bank for choosing channel matrix coefficients;
- A low-complexity implementation of Schnorr-Euchner (SE) enumeration for multiple modulations;
• A shift register chain instead of stack memory for storing temporary decoded symbols.

4.1 Architecture Flexibility Requirements

The architecture requirements are derived by jointly considering tradeoffs at the algorithm, architecture, and circuit layers, with the goal of minimizing chip power and area. Existing architectures [8–12, 47] are evaluated to better define implementation requirements in terms of area and throughput for flexible architecture design. Prior work is extended to include flexibility with respect to a number of parameters: search method, antenna array size, number of sub-carriers, and modulation scheme. These parameters formulate key technical challenges discussed in this section.

4.1.1 Antenna Array Size

For the sphere decoder operating with a large antenna array, the biggest challenge in the implementation is reducing area. Using the number of (complex) multipliers as a first-order area estimate, the number of multipliers needed in the folding and multi-stage architectures are $M$ and $M(M+1)/2$, respectively, where $M$ is the number of transmit antennas. Expanding a $4 \times 4$ system to a $16 \times 16$ system, the relative area increases from 4 to 16 for the folding architecture and from 10 to 136 for the multi-stage architecture. The folding architecture is therefore $2.5 \times$ to $8.5 \times$ more area-efficient compared to the multi-stage architecture, as shown in Fig. 4.1(a). The second design challenge is operating frequency for the folded architecture.

As the array size increases, the number of operands in the Multiply-Accumulate
Figure 4.1: Design challenges and tradeoffs for large antenna size. Impact of antenna array size on (a) area and (b) critical-path delay.

(MAC) operation in the metric calculation unit increases proportionally to the number of antennas. Assuming a tree adder topology, the critical-path delay increases roughly logarithmically with the number of transmit antennas. However, the time required to finish the MAC operation should be scaled down by the number of antennas in order to increase the throughput proportionally to the number of antennas. This timing requirement for a fixed bandwidth is shown in Fig. 4.1(b), assuming the critical-path delay just meets the timing requirement in a 4×4 system. The situation is actually worse when metric enumeration is included in the loop.

4.1.2 Constellation Size

The challenge for the sphere decoder with a large constellation size is that the hardware cost grows quickly as the modulation size increases. Since the admissible constellation points should be enumerated in ascending order according to their distance increments \( |T_i|^2 \) \( (T_i = b_i - R_{ii}s_i) \), the exhaustive search is a
straightforward implementation [9]; it calculates the distance increments of all constellation points and uses a sorting circuit to find the constellation point with the minimum distance, as shown in Fig. 4.2(a). The number of distance calculation units is proportional to the constellation size (64 units are required for 64-QAM, for example).

In the constellation plane, metric enumeration corresponds to finding the points closest to $b_i$ and scaling constellation points $R_{ii}A$ from the closest to the farthest. This is the underlying principle of SE algorithm [5]. The SE enumeration is originally applied to one-dimensional signals, such as a real valued PAM or PSK constellation; therefore it is modified to arrange QAM constellations in $P_Q$ concentric groups to fit the original algorithm [9]. For example, 16-QAM constellation can be expressed as an arrangement of points in 3 concentric circles. Then the problem is reformulated to find the closest point in each subgroup and find the closest point over the subgroups 1 to $P_Q$, as shown in Fig. 4.2(b).

The original algorithm [5] uses phase relationship to find the closest point in a concentric circle. This approach is too complex for hardware implementation due to the calculation of $\cos^{-1}(\cdot)$, so [9] proposed a decision-boundary based method for each concentric subgroup to simplify the SE enumeration. One type of decision
boundary is set by straight lines passing through the origin and the middle point between two adjacent constellation points in a concentric circle, to specify the starting point. Another type of decision boundary is set by straight lines passing through the origin and the middle point between two constellation points around the starting point in a concentric circle, to determine the initial search direction (see [9] for details). However, this simplification is only applicable to small-size constellations such as 16-QAM. Larger constellation sizes are hard to support for several reasons. First, the decision boundary algorithm is quite complex since many multiplications are needed to generate the decision boundaries. Second, the number of subgroups grows quickly, which increases the latency of the min-search circuit. For example, 64-QAM is decomposed into 9 subgroups. Third, the concentric group partition is not scalable as QAM constellation size changes, so it is not suitable for multiple modulations. In [48], a PSK-based enumeration supporting 64-QAM modulation is proposed, but it is still complex because of the embedded sin/cos look-up-tables.

4.1.3 Tree-Search Scheme

Two major types of tree search methods have been reported in previous work: depth-first [8, 9] and K-best [10–12]. The depth-first algorithm starts the search from the root of the tree and explores as far as possible along each branch. Then, it traces back until either a leaf node is found or the node is outside the search radius. The K-best algorithm approximates a breadth-first search by keeping only K branches with the smallest partial Euclidean distance at each level. Main advantages of depth-first are that the ML performance can be achieved, and that radius shrinking can be used for tree pruning. In contrast, the advantages of K-best are its regular datapath and constant throughput. Table 4.1 summarizes the
Table 4.1: Comparison of depth-first and K-best algorithms

<table>
<thead>
<tr>
<th></th>
<th>Area</th>
<th>Throughput</th>
<th>Latency</th>
<th>Radius Shrinking</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth-first</td>
<td>small</td>
<td>variable</td>
<td>long</td>
<td>yes</td>
<td>ML</td>
</tr>
<tr>
<td>K-best</td>
<td>large</td>
<td>constant</td>
<td>short</td>
<td>no</td>
<td>near-ML</td>
</tr>
</tbody>
</table>

architecture comparison in terms of circuit metrics and algorithmic performance.

Given a sufficient number of clock cycles, depth-first ensures the ML performance if the complete solution space is explored. This might not be feasible in practice, however, because of limited buffer size and processing cycles. That is, some termination schemes should be used and thus ML performance is no longer guaranteed. Since the default input is uncoded data, achieving a sub-optimal performance while keeping constant throughput is more important. Then, spacetime codes or error correction-codes can be used to improve the BER performance. The iterative decoding scheme, which combines the MIMO decoder and the error-correction code decoder, was proven to achieve near-capacity performance [5]. The MIMO decoder has to generate the likelihood for each decoded bit in addition to 1/0 output [49][50]. The sphere decoder with such soft outputs is usually called a list sphere decoder [51] since a list of paths should be considered besides the best one (with a minimum Euclidean distance).

In hardware implementation, depth-first is realized in a folding architecture [52] because only one node is visited at a time during the tree search process, as in Fig. 4.3(a). K-best is generally realized as a multi-stage pipeline, as shown in Fig. 4.3(b), since no backward-trace is needed. To process $K$ datapaths at the same time, parallel architecture is applied. If only forward-trace is allowed, the BER performance is limited by the number of parallel processors such as in K-best algorithm. Even though more processing cycles are provided, there is no
Figure 4.3: Basic architecture of (a) depth-first and (b) K-best algorithms.

room to improve the BER performance for the K-best algorithm. Therefore, the capability of supporting forward-trace and backward-trace should be considered in the architecture design.

4.1.4 Multiple Sub-Carriers

OFDM-based systems require the processing of multiple sub-carriers. In a multi-carrier communication system, data streams are transmitted over narrow-band sub-carriers. The data on each sub-carrier suffers only flat fading and thus can be compensated by a one-tap equalizer in the frequency domain. Data-stream interleaving is therefore used to improve area efficiency through logic sharing [7]. By adding appropriate latency for a recursive operation, multiple data streams can be processed in an interleaved way. To accommodate \( m \) streams of data and maintain the overall throughput, one register can be replaced with \( m \) registers (\( m-1 \) pipeline registers are inserted) [49]. The critical-path in the loop is, therefore, shortened due to pipeline insertion and retiming. This technique supports parallel execution of independent data streams with reduced area via logic sharing. By data rearrangement, the technique provides flexibility needed to support a varying number of sub-carriers. Note that data interleaving is also applicable to single carrier systems if multiple data streams are buffered in advance.
4.2 Flexible Architecture Design

To address the design challenges presented in the previous section, a flexible and scalable architecture is constructed to deploy the flexibility requirements. The existing algorithmic expressions are evaluated. Interestingly, a simpler algebraic expression does not necessarily result in area-efficient hardware realization—the specific application and input profile make a simpler hardware implementation possible. A low-complexity multiplier is proposed for metric calculation to reduce area and delay simultaneously. The metric enumeration algorithm is realized using geometric relationships, which simplifies design for a large constellation size. At the circuit layer, the use of bi-directional shift delay lines to choose multiplier coefficients and the clock-gating technique provides a low-area, low-power implementation for input data retrieval. Unlike prior work, architecture flexibility is considered in the design stage [45]. Antenna size, modulation scheme, number of sub-carriers, and search method are designed with flexibility and scalability to cover multiple communication scenarios.

As shown in Fig. 4.4, each flexible DSP architecture consists of two major functional blocks: metric calculation unit (MCU) and metric enumeration unit (MEU). The architecture shown in Fig. 4.4 works with complex numbers. One of the key tradeoffs to consider in the decoder implementation is concurrency versus latency, because the decoding of real and imaginary parts can be done concurrently or sequentially. Joint decoding maps to a parallel architecture with a large area and proportional throughput increase. The sequential approach achieves lower throughput due to time-multiplexing. Joint decoding is more attractive from the BER performance standpoint. It also takes advantage of the speed of logic gates to reduce supply voltage and minimize power. The implementation strategy and structure of building blocks are illustrated next.
The MCU mainly computes $\sum_{j=i+1}^{M} R_{ij}s_j$. Basically, it executes a MAC operation. To accumulate all 16 operands and achieve the highest throughput, there are 16 complex multipliers followed by an adder tree that merges the partial products. It is possible to reduce the number of multipliers in a time-multiplexing manner at the price of lower throughput. For example, 4 complex multipliers can be time-multiplexed by 4 to deploy 16 multipliers, with throughput reduced by 4. Such tuning at the architecture level is used to position the design along the throughput or the power axis, with optimal tuning of variables such as supply voltage.

The MEU enumerates the possible constellation points according to the SE enumeration algorithm. A region-partition method based on Cartesian coordinates is proposed to simplify the algorithm. The constellation plane is partitioned into 64 regions for 64-QAM (8 regions in the real part and 8 regions in the imaginary part). The closest point (with minimum distance) can be decided by the location of $b_i/R_{ii}$ since the real part and the imaginary
part can be decoded separately.

4.2.1 Multiplier Simplification for Scalable Antenna Array Size

From an algorithm perspective, the complexity of sphere decoding is evaluated by the number of nodes visited in the tree search process. However, the supported antenna array size is restricted by the number of physical (complex) multipliers for metric calculation. The number of multipliers is reduced through a folded architecture [52] from $M(M + 1)/2$ to $M$, where $M$ is the number of transmit antennas, as shown in Fig. 4.5(a). An 8.5× reduction in multiplier area is achieved for 16 transmit antennas compared to direct-mapped design.

At the circuit level, the size of multipliers is what determines the area and speed of the sphere decoder, which is generally ignored in the algorithm design stage. A low-complexity multiplier for metric calculation is used to reduce area and delay simultaneously. The cost of the multiply operation is simplified to reduce hardware complexity. The multiplication is required to calculate Euclidean distance, which is mathematically represented by Eq. 2.2 and Eq. 2.3, as adopted in [8, 10, 11, 25] and [9], respectively. At the first glance, Eq. 2.2 has one multiplication while Eq. 2.3 has two. However, careful investigation shows that Eq. 2.3 is a better choice for hardware implementation for at least three reasons:

- $s_{ZF}$ and $Q^H y$ are pre-computed, but the calculation of $s_{ZF}$ is actually more complex than $Q^H y$. The complexity analysis is therefore reduced to comparing vector multiplications: $R(s - s_{ZF})$ and $Rs$.

- To keep the same algorithmic performance, the wordlength of $s_{ZF}$ should be as long as possible. Reducing the wordlength of $s_{ZF}$, therefore, degrades the algorithmic performance of Eq. 2.2. Since the wordlength of $s$ is usually
Figure 4.5: Multiplier area reduction: (a) Area reduction due to folded architecture. (b) Gray-code representation and the simplified multiplier.
much shorter than $s_{ZF}$, separating terms as in Eq. 2.3 results in multipliers with reduced wordlength, thus reducing the area.

- $Rs$ is even less complex than $R(s - s_{ZF})$, since $s$ is drawn from finite alphabets, while $s_{ZF}$ is variable. Therefore, $Rs$ can be implemented by *simplified* multipliers, while $R(s - s_{ZF})$ should be implemented by *full* multipliers.

Without loss of generality, the normalized size of a multiplier can be estimated by the product of wordlengths of the multiplier and the multiplicand. The normalized delay of a multiplier can be estimated by the sum of wordlengths of the multiplier and the multiplicand if an array multiplier is used [53]. The array multiplier approximation works well for first-order comparison purposes. Table 4.2 summarizes the relative area and delay of a multiplier due to wordlength reduction in a 64-QAM system, where wordlength (WL) of $s$ is 3 for real and imaginary parts. Taking a larger wordlength of $s_{ZF}$, 12 bits, for example, Eq. 2.3 dominates in terms of area and delay. If the wordlength of $s_{ZF}$ is reduced to 6 bits, which degrades the algorithmic performance, (2.3) still wins. The area reduction is at least 50% for the case that wordlength of $s$ exceeds 6 bits. The delay reduction also reaches 40% for $R$ with wordlength of 16. The wordlength of $R$ can be reduced to 12 bits through wordlength optimization as long as the quantization noise is masked by the noise contained in the input signal. A $7 \times$ area reduction in multiplier size is achieved relative to the reference design ($16 \times 16$-bit).

Conventionally, an efficient complex multiplication can be realized by three real multiplications and five real additions/subtractions [54]:

\[
(a + jb)(c + jd) = [c \times (a - b) + b \times (c - d)] \\
+ j[d \times (a + b) + b \times (c - d)].
\] (4.1)
Trading one multiplier for three adders is beneficial since the complexity of the multipliers is usually much higher than that of adders for general purpose processors. However, since the multiplication here is small and specific for finite alphabets, the original form is simpler and it can be further simplified by leveraging Gray code.

Gray code is a more compact representation in the constellation plane since only odd numbers are used. Conventionally, the number is transformed to 2’s complement representation for the purpose of arithmetic operations. In the Gray code representation, the corresponding multiplication can be implemented by simple shift, add and invert operations. The code mapping, the associated operations, and the simplified multiplier are shown in Fig. 4.5 (b). The neg operator stands for bit-inversion. A 1-bit carry-in in 2’s complement can be absorbed as a carry-in in the remaining adder stages or simply be discarded as a quantization error on LSB. As long as the quantization noise is masked by the noise contained in the input signal, it has negligible impact on the overall BER performance. Thus, the delay is reduced from one adder to one inverter because the delay of bit-inversion is independent of wordlength.

The shifter has no direct area cost apart from routing, while the cost of inverters and multiplexers is relatively low because they are simple operations. Overall, it is possible to implement one complex multiplier with 6 adders plus few inverters and multiplexers, resulting in a total 38% area reduction compared
to the conventional complex multiplier utilizing the constant operand. Multiplier simplification enables support for larger antenna arrays. An overall $11.3 \times$ area reduction in multiplier size is achieved. Gray coding also reduces 25% registers for $s$ (all from 8-bit to 6-bit), saving 256 DFFs. The area comparison is based on Synopsys synthesis estimates. This implementation does not imply that Gray coding is forced to use in the constellation plane; Gray coding is only used inside the sphere decoder to simplify metric calculation and candidate enumeration. The decoded symbols can be converted into any arithmetic representation at the sphere decoder outputs.

4.2.2 Candidate Enumeration for Scalable Modulation

A scheme proposed by Schnorr and Euchner (SE) suggests traversing the constellation candidates according to the Euclidian distance increment in ascending order to speed up the search process [24, 55]. The underlying idea of SE scheme is to approach the globally optimal solution by using locally optimal solutions. Since the candidates for each node have already been sorted, the remaining sibling nodes can be discarded if the node is already outside the search radius. The first estimate $\hat{s}_i^{(1)}$ for each row $i$ in Eq. 2.3 is decided by the location of $b_i/R_{ii}$, where $b_i = \tilde{y}_i - \sum_{j=i+1}^{M} R_{ij}s_j$. However, $R_{ii}^{-1}$ is usually hard to calculate. A decision-plane transformation scheme is therefore proposed to simplify the decision process.

4.2.2.1 Decision-Plane Transformation

Instead of making a decision based on $b_i/R_{ii}$ in the constellation plane, the decision is made based on $b_i$ in the original plane scaled by $R_{ii}$. The decision boundary $db$ is denoted as $db \in \{-6, -4, -2, 0, 2, 4, 6\}$, so $b_i/R_{ii}$ calculation is simplified to
$db \cdot R_{ii}$. These multiplications can be further simplified as shift, add, and invert operations. Only one adder is needed to implement $db \cdot R_{ii}$ ($6 \cdot R_{ii} = 4 \cdot R_{ii} + 2 \cdot R_{ii}$); others can be implemented by hard-wired shifting and inversion. The proposed low-complexity decision circuit is shown in Fig. 4.6. The negative value can be computed by bit-inversion without the carry-in bit, which saves one adder. One large multiplier ($WL(R_{ii}^{-1}) \times WL(b_i)$) is replaced with a small adder ($WL(R_{ii}^{-1})$) even without calculating $R_{ii}^{-1}$. Decision outputs are mapped to Gray code directly without conversion between 2’s complement and Gray-code representation, saving 25% of registers for $s$ (from 8-bit to 6-bit). An extra symbol remapper is inserted using a simple LUT for varying modulation schemes [46]. Although $R_{ii}$ can be chosen to be always positive to simplify this circuit further, the flexibility of negative values is supported as well in order to relax QR decomposition processing. With the proposed approach, no sorting is needed and it is easy to expand to a large constellation size. Additionally, the use of bit-level arithmetic results in only linear complexity increase as the constellation size grows exponentially. A $4.6 \times$ area and $5 \times$ delay reduction is achieved by our enumeration circuit as compared to the prior work [9].

4.2.3 Simplified Candidate-Enumeration Scheme

The remaining candidates are decided by the distance between $b_i$ and scaled constellation points in ascending order. This two-stage enumeration is similar to the zig-zag enumeration scheme in [56,57], but the zig-zag scheme is only applicable to real-valued cases. The decoded symbol $\hat{s}_i^{(1)}$ is used to enumerate remaining candidates through geometric relationships. The expected complexity of the sphere decoding algorithm is independent of the lattice constellation size; therefore, The adjacent possible constellation points instead of the whole constellation
Figure 4.6: Low-complexity realization of the signal decision block. Decision is made in the original constellation plane scaled by $R_{ii}$. 
Figure 4.7: Candidate enumeration scheme. This scheme simplifies the enumeration by taking advantage of the geometric relationship.

plane are enumerated. This simplification is also reasonable in practice, since finite processing cycles also limit the search range. High points are extracted in the constellation plane as illustrated in Fig. 4.7. Note that this simplified implementation does not necessarily contain the ML solution, but the performance degradation is negligible with proper settings (discussed in Section 5.4). The 8 surrounding points $s^{(2-9)}_i$ include the symbols with the most likely erroneous 1-bit error and 2-bit errors due to the Gray-code constellation, Fig. 4.7(a-b) and Fig. 4.7(c-d). The 2nd closest point for each solution set is decided based on decision boundaries indicated by the dashed lines in Fig. 4.7(a), (c). The remaining points are decided by the search direction, which is specified by other decision boundaries, starting from the 2nd point, as shown in Fig. 4.7(b), (d).

These two decision boundaries are easy to calculate by sign-check and comparison between the real part and the imaginary part of $Z_i = b_i - R_{ii}s^{(1)}_i$. The cross $\times$ in Fig. 4.7 represents the value of $b_i$, and $s^{(1)}_i$ is the closest point decided by the region-partition method. For the 1-bit error subset, the decision boundary
calculated by $Z_i \ (\text{Re}\{Z_i\} > \text{Im}\{Z_i\} \text{ and } \text{Re}\{Z_i\} > -\text{Im}\{Z_i\})$ indicates that the 2nd closest point is the right point, Fig. 4.7(a). The 3rd point is the bottom one because $\text{Im}\{Z_i\}$ is negative, indicating the bottom point is closer than the top one, Fig. 4.7(b). By taking advantage of the decision boundary in Fig. 4.7(a) again, the top point must be closer than the left one; the 4th and 5th points are therefore decided. The search direction takes a zig-zag path shown by the numbers in Fig. 4.7 (a), (b) due to the layout of constellation points. The similar idea is also applied to 2-bit error subset. As a result, these decision boundaries can be used to simplify the SE candidate enumeration significantly.

Fig. 4.8 shows the candidate enumeration circuit and the required control signal transition for our simplified algorithm. The LUT provides the neighboring candidates in Gray code representation used in the constellation plane. For example, given the input 5 (101) to the LUT, H is 7 (100) and L is 3 (111). Notice the LUTs for $\text{Re}\{\hat{s}_i^{(1)}\}$ and $\text{Im}\{\hat{s}_i^{(1)}\}$ are shared for 1-bit and 2-bit error subset. The initial values of control signals A and B (C and D) decide the 2nd point in the 1-bit subset (2-bit subset) and the succeeding transitions enumerate the remaining points. To decide the enumeration sequence by jointly considering these two subsets from the closest to the farthest, a mixed method is adopted: the two solution sets are compared to find the final enumeration sequence with respect to the central point. Only one square operator is needed through time-multiplexing and it is necessary for the Euclidean distance calculation.

4.2.4 Bi-Directional Search for Adjustable Tree-Search Direction

Bi-directional search is critical for BER performance improvement. With only forward search, the BER performance is limited even if more processing cycles are allowed. Bi-directional search is supported by a finite-impulse-response (FIR)-
like architecture, as shown in Fig. 4.9. By observing that the tree searcher traces back by only one level instead of a random jump, a bi-directional shift-register chain is embedded to adjust the search depth. Since the search state is recorded in the shift-registers, no extra memory, such as stack memory [8,9], is needed to keep all the states. Due to the backward-trace requirement, a transposed form FIR architecture is not suitable to reduce the critical-path, but the critical-path is reduced by pipeline insertion along with data-interleaving.

The original critical-path (without pipelining) is the feedback loop starting from shift-register chain bank and back to it, including partial product generation, adder tree, and symbol selection as shown in Fig. 4.4. Pipeline insertion was introduced in [49,51] to support multiple data streams for sphere decoding. Unlike [51], data interleaving is introduced in the loop to reduce the critical-path [49]. To support multiples of 8 sub-carriers, the critical-path is partitioned into 8 pieces.
by inserting 7 pipeline stages, which is higher than 3 stages in [49] due to more operands. Ideally, the critical-path should be equally partitioned to achieve the highest clock frequency, but the circuit granularity limits the balanced pipeline insertion in the design stage. The pipeline depth is around one adder equivalent operator and retiming is utilized to balance delay in the synthesis stage. There are total 130 registers (12-bit to 15-bit) inserted. The estimated maximum clock frequency (synthesis estimate) is 500 MHz, which allows an aggressive voltage-scaling in the ASIC implementation.

The reconfigurable registers support data shift in forward or backward direction. The decoded symbols are recorded in the registers to indicate the search path in the tree topology, as shown in Fig. 2.4(c). The search process starts by forward search and continues until the search path is outside the search radius.
(i.e., partial Euclidean distance is greater than the search radius) or reaches the leaf node. In these cases, the search direction is reversed. The latest decoded symbol is discarded and the next possible constellation point is loaded. Since the state of the registers represents the search state, an external search state can be loaded directly to resume search for any new search trees.

4.2.5 Data-Interleaved Processing for Scalable Number of Sub-Carriers

By applying data-stream interleaving, samples of other independent data streams can be introduced in the loop in place of the repeated values or padding zeros. In the loop, inputs are up-sampled by a factor \( p \), which means that each register in the loop has to be replaced with \( p \) registers (\( p - 1 \) pipeline registers), Fig. 5.7(a). The critical-path in the loop is also shortened due to pipeline insertion and retiming. Technique of interleaving is therefore used to improve area efficiency through logic sharing and to provide flexibility needed to support varying number of data sub-carriers.

The area and power consumption of data access for channel matrix \( \mathbf{R} \) should be minimized as the antenna array size increases. It is not area-efficient to use SRAM because the upper triangular nature of \( \mathbf{R} \) does not fit the rectangular shape well. It is possible to rearrange the elements to fit the rectangular shape, while increasing the complexity of data access. A register bank to store coefficients of \( \mathbf{R} \) in an area-efficient way is shown in Fig. 4.10. The diagonal terms of \( \mathbf{R} \) are real, while the rest are complex numbers. By only storing these non-zero elements, this architecture saves 48% of memory area for 16×16 systems. The memory reduction is estimated using a 1\textsuperscript{st} order approximation in terms of the number of registers saved (from 496 to 256 words). Instead of shifting data of coefficients, a shift delay-line is used to choose the corresponding coefficients. The
column of value 1 in the shift delay-line is activated; shift direction is based on the decision of backward-trace or forward-trace. The coefficients are loaded using a concatenate scan-chain, and the registers for storing coefficients are clock-gated after loading values, eliminating the major dynamic power. A 90% power saving (based on synthesis results) is achieved using this technique compared to shifting data of coefficients like the shift-register chain in Fig. 4.9. This register-bank structure is also flexible for multiple data streams to share the same \( R \) matrix by multiplexing several shift delay-lines.

Fig. 4.11 shows the overall area reduction for the flexible DSP architecture. An overall \( 20 \times \) area reduction is achieved through signal processing and circuit techniques, from the arithmetic level down to the circuit level, which includes architecture folding \( (8.5 \times) \), MEU simplification \( (30\%) \), simplified multiplier \( (20\%) \), memory reduction \( (5\%) \), and wordlength reduction \( (20\%) \).

### 4.3 Hardware Emulation Results

The BER performance of the PE is verified through the hardware emulation environment. In the preliminary experiment, the number of processing cycles is equal to the number of transmit antennas to support the highest throughput. The BER performance of a system with a larger antenna array and repetition coding can easily outperform the ML performance of a system with a smaller antenna array.

Fig. 4.12(a) shows the BER performance of 64-QAM modulation for different antenna array sizes and different repetition coding rates. The repetition coding here refers to sending replicas in space domain to reduce error probability. The performance of \( 4 \times 4, 8 \times 8, \) and \( 16 \times 16 \) is comparable, but the throughput is dif-
Figure 4.10: Channel coefficient generation. Only non-zero coefficients are stored, which saves 48\% of memory area.
Figure 4.11: Summary of area reduction of the flexible DSP architecture. The final design achieves $20 \times$ reduction in area.

To verify the proposed 9-point enumeration scheme, the BER performance of the optimum solutions with different enumeration constraints are drawn in Fig. 4.12(b) for comparison. The optimum solution without the constraint is the ML estimate. The simplified enumeration scheme by searching the neighboring 9 points provides significant improvement compared to the highest-throughput mode. However, the performance degradation between the optimum solution using 9-point enumeration and the ML solution increases in the high SNR regime mainly because of erroneous decoding of the first decoded symbols. To approach the ML performance and avoid error propagation, the ML solution should not be discarded in the early stages. By allocating multiple PEs to search with different
Figure 4.12: Hardware emulation results for several design configurations.
first candidates (16 candidates in this example), the performance degradation can be almost eliminated.

Repetition coding is used to demonstrate the performance improvement by transmitting data replicas. An 8×8 system with repetition coding by 2 has outperformed the 4×4 system with the ML performance by 5 dB. The BER performance gap between this system and ML estimate decreases as the factor of repetition coding increase. For example, only a 0.3 dB difference is observed for the 16×16 system with repetition coding by 4. The data redundancy improves the transmission reliability, which therefore reduces the decoding complexity at the cost of throughput reduction.

4.4 Hardware Complexity Comparison

A comparison of hardware complexity is illustrated in Table 4.3. The estimated silicon area is 0.31 mm² (5.5k FPGA slices [45]) in a standard 90-nm CMOS process from synthesis. To make a fair comparison, the area is normalized by the number of transmit antennas (this is a conservative estimate, because the hardware complexity could grow quadratically with the number of transmit antennas). The data indicates that the proposed architecture is the most area efficient compared to prior work. Furthermore, our design outperforms all previously published designs in terms of supported antenna array size and constellation size, as shown in Fig. 4.13. Unlike previous work, the proposed architecture also supports multiple sub-carriers and search methods. The estimated power consumption is 33 mW at 256 MHz from a supply voltage of 1 V from synthesis results.
Table 4.3: Hardware complexity comparison

<table>
<thead>
<tr>
<th></th>
<th>[8]</th>
<th>[9]</th>
<th>[10]</th>
<th>[11]</th>
<th>[47]</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Area</strong></td>
<td>10 mm(^2)</td>
<td>50k GC</td>
<td>97k GC</td>
<td>500k GC</td>
<td>12.7k GC</td>
<td>*</td>
</tr>
<tr>
<td><strong>Area (norm.)</strong></td>
<td>32.3</td>
<td>2.3</td>
<td>9.2</td>
<td>11.5</td>
<td>4.4</td>
<td>1</td>
</tr>
<tr>
<td><strong>Clock freq. (MHz)</strong></td>
<td>123</td>
<td>71</td>
<td>150</td>
<td>122</td>
<td>100</td>
<td>256</td>
</tr>
</tbody>
</table>

*5.5k FPGA slices, 0.31 mm\(^2\) (90 nm), or 87k gate count (GC)

Figure 4.13: Comparison of hardware complexity.
4.5 Chapter Summary

A flexible DSP architecture for sphere decoding was presented. The proposed architecture supports antenna arrays from $2 \times 2$ to $16 \times 16$, modulations from BPSK to 64-QAM, and 8 to 128 frequency sub-carriers. An architecture with folding and interleaving is used to reduce area and critical-path delay. Varying antenna-array size is supported through hardware reuse. Multiple modulation schemes are realized by metric enumeration which uses a simplified boundary decision algorithm. Compact multiplier realization is achieved using algorithmic transformation and wordlength reduction. Memory size is reduced by using bi-directional shift delay-line based register circuits. Hardware emulations show that the reduced hardware complexity allows for an $8 \times 8$ decoder with repetition coding to outperform, in BER vs. SNR, a $4 \times 4$ ML decoder. The simplified 9-point metric enumeration scheme achieves the ML performance with proper configurations. Built around this DSP kernel, a 16-core MIMO sphere decoder chip is implemented to demonstrate the functionality in silicon. The design details of the 16-core MIMO sphere decoder will be described in the next chapter.
CHAPTER 5

Multi-Mode Single-Band Sphere Decoder Chip

Based on the developed flexible sphere decoder DSP kernel, a 16-core MIMO sphere decoder is proposed to improve search speed, error performance, and power consumption \([22,58]\). Multiple processing elements (PEs) are coordinated for parallel processing to improve computational capability and energy efficiency. As a proof of concept, the proposed multi-core MIMO decoder is designed and implemented in a 90-nm CMOS technology. The chip has flexibility and scalability in antenna array size (up to \(16 \times 16\)), modulation scheme (up to 64-QAM), search method (K-best or depth-first), and number of sub-carriers (up to 128). The proposed multi-core MIMO sphere decoder provides more features than prior work \([8–12]\) as shown in Table 1.1. The decoder chip dissipates only 2.89 mW at 16 MHz operating frequency from a 321 mV supply voltage. The following key features of the chip are highlighted:

- Low-complexity scalable PE (modified from the sphere decoder DSP kernel) for multiple operation modes;

- Multi-core architecture that improves error performance, throughput, and energy efficiency;

- Power and area reduction of the DSP architecture that leverages application-specific data profile;
- Interleaved data processing for multidimensional data access along with clock-gating for power saving.

It will be shown that this chip outperforms the prior state-of-the-art design [12] by 6.6× in energy efficiency and by 8× in bandwidth utilization despite added flexibility for multi-mode operation. Its high performance allows high-speed, reliable communication for next-generation MIMO systems. The ultralow power consumption makes it suitable for mobile devices; the added flexibility makes the integration of a variety of applications and standards on a single DSP chip possible.

5.1 Design Specifications and Flexibility

The sphere decoder is designed to support different system configurations with respect to antenna array sizes, modulation and detection schemes, as well as the number of sub-carriers for OFDM-based systems. The supported number of sub-carriers can be a multiple of 8 through data rearrangement. The main design specification is the throughput constraint for the algorithm. Since a total bandwidth of 16 MHz is assumed, each sub-channel requires 2 MS/s to process the data in the case of 8 sub-carriers. Signal bandwidth of 16MHz is assumed to illustrate design techniques that are generally applicable to any band of interest. The requirement is thus to process 8 parallel streams of data at a 2 MHz rate (1.25 kHz for 128 sub-carriers). The clock specification for the resulting architecture then becomes 256 MHz (16 MHz ×16 antennas). In the 16×16, 64-QAM mode, the system supports computational throughput up to 1.536 Gbps, which ideally means a spectral efficiency of 96 bps/Hz.
5.2 Multi-Core Architecture

Continued scaling of CMOS technology enables smaller devices and faster operation. These benefits can be exploited to increase the number of processors and the number of processing cycles. Based on the scalable PE architecture, a multi-core architecture is proposed to speed up the search and also add flexibility to examine more solutions when additional processing cycles are allowed. Targeting different operating modes, the architecture can be configured to provide either high performance or low power by voltage and frequency scaling.

5.2.1 Search Algorithm

Figure 5.1 illustrates the search scheme for the multi-core architecture. The idea is to search multiple nodes within the search radius simultaneously to speed up the search of admissible solutions, while allowing backward-search to examine more possible solutions. Four PEs are shown in Fig. 5.1 for brevity. Each PE is capable of searching a subset of the whole tree (sub-tree) using the depth-first algorithm. The ML solution lies in the PE with the minimum Euclidean distance. Whenever a better solution is found, the updated Euclidean distance, i.e., the search radius, is broadcasted to all PEs. Once one PE finishes searching the assigned sub-tree, one new sub-tree is assigned to keep it active. The PEs start with different constellation points for antenna $M$, so the search paths are not examined repeatedly. The main advantages of the multi-core architecture as compared to K-best are: 1) backward search capability and 2) search radius shrinking. The multi-core sphere decoder therefore supports backward-search to expand the search range if more processing cycles are allowed. For a given processing time, multiple PEs speed up the search of remaining candidate points, which shrinks the search sphere faster.
According to the SE scheme, the constellation candidates should be traversed from the closest point to the farthest point. A tree-partition scheme that partitions the whole search tree into several sub-trees (sub-tree 1 to sub-tree \(|\Lambda|\) for each antenna) is proposed. The mapping between the constellation plane and the corresponding tree structure is shown in Fig. 5.2. The closer constellation points are more likely to be the better solutions. Therefore, they should be searched first. By taking advantage of constellation points being sorted (from the closest to the farthest), the remaining sibling nodes can be discarded if the node is already outside the search radius. This is because the farther nodes only contribute larger Euclidean distance increments.

Ideally, all constellation points should be sorted. However, the hardware complexity and processing latency of distance sorting for higher-order constellation sizes are not feasible for target application. A simplified candidate-enumeration scheme for single-PE structure is illustrated in Sec. 4.2.3. The enumeration technique is based on the enumeration of the adjacent possible constellation points instead of the whole constellation plane. As a result, the complexity of sphere de-
Figure 5.2: Tree-partition scheme. All constellation points are sorted for each decoding stage (antenna).

coding is independent of the lattice constellation size. This enables the flexibility in multiple constellation sizes. Extending the technique to multiple PEs, multiple PEs (16 in this case) are assigned to process the 16 likely symbols around $\hat{s}_M^{(1)}$ for antenna $M$, as shown in Fig. 5.3. For the remaining antennas, nine likely constellation points are examined. The eight surrounding points include the symbols with the most likely 1-bit and 2-bit errors due to the Gray-code constellation. Simulation results show the BER performance degradation of the proposed candidate-enumeration scheme is negligible (Fig. 4.12). Only sign-check and comparison are needed to enumerate these points from the closest to the farthest for these two subsets. Since no sorting operation is needed for the 1-bit and 2-bit error subsets, hardware area and delay requirements are reduced. This provides efficient sorting for higher-order constellations.
Figure 5.3: Candidate-enumeration scheme in the 64-QAM constellation plane.

Figure 5.4: Block diagram of the multi-core architecture. Processing of 16 sub-carriers over 16 PEs is illustrated.

5.2.2 System Architecture

Figure 5.4 shows top-level block diagram of the multi-core architecture. A hybrid structure of shared memory and distributed memory is adopted for efficient data access. A hierarchical interconnect network is used to minimize the signal processing latency across all PEs. Sub-carriers are interleaved into each PE to utilize hardware sharing. Interleaved signal processing for sub-carriers also extends the processing interval. The extended processing interval makes complex signal processing such as radius checking and updating across PEs possible.
Figure 5.5: (a) Hybrid memory structure: $s$ is distributed over all PEs; $R$ and $\tilde{y}$ are shared by all PEs. (b) Hierarchical interconnect network is used to minimize the delay across PEs.

5.2.3 Hybrid Memory Structure and Hierarchical Interconnect

There are two types of memory structures in multi-core compute platforms: shared and distributed. In shared memory systems, one common memory is physically shared; in distributed memory systems, each processor core has its own memory [59]. Input data $R$ and $\tilde{y}$ only need to be read during the decoding process, thereby they are being shared by all PEs to minimize the use of memory. In contrast, decoded symbol $s$ records the distinct paths for each PE, so the paths are distributed across all PEs to the local cache. The heterogeneous structure is shown in Fig. 5.5(a), where $s$ indicates distributed memory, and $R$, $\tilde{y}$ indicate shared memory. The final decoded sequence is obtained by choosing $s$ with the minimum Euclidean distance among all PEs.

A hierarchical interconnect network connects all PEs to facilitate signal processing across PEs, as shown in Fig. 5.5(b). Controllers for implementing the proposed search algorithm are organized in two hierarchical levels: four local controllers (LCs), each managing a 4-PE cluster, are coordinated by the global...
controller (GC). Local controllers are deployed to coordinate four neighboring PEs and the global controller is used to update the radius and choose the optimal solution. Each controller consists of a comparator for distance comparison and a decision block for radius updating and search-path assignment.

5.2.4 Control Mechanism

Sphere decoding can be classified as a branch-and-bound (B&B) algorithm. Parallelization of B&B problems has been extensively studied by using multiple processors for processing speed-up [59]. A generalized utility for parallel B&B problem has been developed based on four basic rules: branching rule, bounding rule, selection rule, and elimination rule, [60]. The control mechanism of the proposed multi-core sphere decoding algorithm is mapped onto this utility as follows.

- Branching rule: The tree-partition scheme divides a problem into sub-problems. Sub-trees associated to a parent node are assigned to different PEs. The tree structure ensures non-overlapping search for sub-trees.

- Bounding rule: The search radius is set through pre-calculation [5], or it can be calculated after any path is traversed. Afterwards, it keeps updating whenever a shorter path is found.

- Selection rule: The proposed candidate-enumeration scheme decides the priority of the sub-trees. Here, the speedup of finding globally optimal solution is gained by leveraging locally optimal solutions.

- Elimination rule: Whenever all the feasible solutions (within the search radius) of the sub-tree are examined, the sub-tree can be eliminated. Note that a practical realization may argue for termination of the search process due to processing time constraint.
The data structure for the decoding process is shown in Fig. 5.6. One possible search path for PE1 is shown for illustration (shaded squares). The search path is recorded in the register $s$ ($s_{16}$ to $s_1$ in the $16 \times 16$ mode). The search sequence is decided by the proposed candidate-enumeration scheme in Fig. 5.1. Backward search is illustrated in case that search reaches the leaf node. A new search path is assigned by transferred values of register $s$ since the register records the search state for each PE. The decoded symbol $s$ is used to calculate Euclidean distance. Once a smaller Euclidean distance is found, the value is used as the new search radius (radius shrinking) and broadcasted to all PEs. When a PE finishes the search for the assigned sub-tree, one unexamined sub-tree is assigned to it for the upcoming processing cycles from the sub-tree with the minimum distance. The procedure continues until all possible solutions are examined or the search process is terminated.

5.2.5 Interleaved Signal Processing for Multi-Core Architecture

To maximize the use of hardware, signal processing across all PEs should be finished in one clock cycle such that the results can be used in the next cycle. As the number of PEs is increased, it is not possible to finish these operations within one cycle, resulting in processing speed mismatch between the PEs and the interconnect network. It is not feasible to insert the pipeline directly due to the feedback loop inside the PE. The data-interleaving technique is adopted to eliminate the throughput bottleneck. As shown in Fig. 5.7(b), independent data streams $x_i$ ($i = 1, \ldots, p$) are interleaved into each PE. The metric calculation and enumeration are executed in the PE while the radius updating and new-path assignment are conducted across the PEs through the interconnect network. For the baseline operation, five and three cycles are assigned for intra- and inter-PE
Figure 5.6: Data structure of multi-core sphere decoding. The search path is recorded and can be transferred through the register $s$. 
data processing \( p = 8 \), respectively. For OFDM-based systems, the data of different sub-carriers can be mapped to these data streams directly. To support multiples of 8 sub-carriers, the input data streams are buffered in advance for varying number of PEs.

### 5.2.6 Multi-Dimensional Data Access with Clock Gating

As proposed for the single-PE architecture, a delay-line-based register bank is used to store \( R \) coefficients, saving 48% of memory area. Extending to 16 cores, instead of storing 16 copies for 16 cores, only one shared input-register-bank is needed. The input data are accessed using a multi-dimensional shift-register line shown in Fig. 5.8. Eight input data streams are scanned in and stored to match the pipeline depth in all PEs. A token signal is passed to activate the data registers for eight input data streams sequentially. The contents stored in \( R \) and \( \tilde{y} \), Fig. 5.5(a), with the same column are therefore retrieved, while the rest are being clock-gated for power saving. The clock control signals are also used for \( s \) registers, Fig. 5.5(a), so only the activated registers are used, saving multiplexers for feeding decoded symbols into the individual \( s \) register. Latch-based clock gating circuit is used to avoid glitches caused by control-signal transitions. Additionally, the clock-gating scheme avoids the use of shift-registers to avoid excess switching activity for both clock and data. Since only one column of \( R \) and \( \tilde{y} \) needs to be accessed, Ring counter is used to enable clock for the selected column. This results in 90% of memory power reduction.
Figure 5.7: Data-interleaved processing for multiple data streams. (a) Data streams are interleaved into each PE. (b) Data-stream interleaving is utilized to extend the processing interval.
Figure 5.8: (a) Multidimensional data access and (b) shared-register architecture with (c) latch-based clock-gating technique.
5.3 Chip Implementation

Figure 5.9 shows the die photo of the 16-core MIMO sphere decoder chip that supports up to 16×16 antenna array and 64-QAM modulation. 16 cores are placed as 4-PE clusters. Register bank and controller are placed in the central part to facilitate data access across 16 PEs. The chip is fabricated in 1P8M low-\(V_T\) 90-nm CMOS technology. The core area is 5.29 mm\(^2\), an individual PE occupies an area of 0.24 mm\(^2\) including its own power ring. The sixteen PEs are first placed and routed as hard macros, then they are carefully placed to facilitate global clock-tree synthesis and signal routing. The register bank, scheduler, and interconnect network are placed and routed in the remaining area. The total chip area with I/O pads is 8.88 mm\(^2\). The core supply voltage is tunable between 0.2 V-1.2 V, and the supply voltage for I/O pads is 1.2/2.5 V. At the core-I/O boundary, cross-coupled level shifters are inserted to drive output pads for the specified range of supply voltage.

5.4 Measured BER Results

To demonstrate the functionality of the proposed multi-core architecture, BER performance for several configuration modes was recorded. In the testing experiments, the bandwidth was set to 16 MHz, and the number of processing cycles was equal to the number of transmit antennas to achieve the highest throughput. The clock frequency \(f_{clk}\) was 64 MHz and 256 MHz for 4×4 and 16×16 systems, respectively, resulting in throughput of 6\(f_{clk}\) and 4\(f_{clk}\) bps for 64-QAM and 16-QAM modulations. Figure 5.10(a) shows the BER performance of the 4×4 antenna-array system with 16-QAM modulation. Over 7 dB improvement for high SNR regime (BER < 10\(^{-3}\)) is observed by deploying 16 PEs as compared to
the single-PE design. Compared to the ML performance, a 3-5 dB improvement is achieved with 16 PEs when $E_b/N_0$ exceeds 15 dB. Figure 5.10(b) shows the BER performance of the 16×16 system for different constellation sizes and number of PEs. Around 3 dB improvement is achieved using 16 PEs over single-PE architecture ($BER < 10^{-2}$). For a high-SNR regime, a 5 dB improvement is observed for 16-QAM modulation. The BER performance can be further improved if more processing cycles are allowed [46].

5.5 Measured Circuit Performance

Since the chip is designed for several operating modes, the clock frequency $f_{clk}$ is adjustable from 16 to 256 MHz. By reducing $f_{clk}$, the decoder is configured to support either a smaller antenna array or a lower throughput. The supply volt-
Figure 5.10: BER performance for several design configurations. (a) 4×4, 16-QAM mode. (b) 16×16 mode, 64-QAM and 16-QAM.
age is adjusted accordingly to minimize power. Figure 5.11 shows measurement results. Operating at 256 MHz, the power consumption is 275 mW at 0.75 V. At 16 MHz, the chip executes 50 GOPS with the minimum supply voltage of 321 mV. The corresponding power dissipation is 2.89 mW, resulting in a peak energy efficiency of 17.3 GOPS/mW. A 3.4× energy efficiency improvement over the previous estimate [35] is achieved through the shared register bank, extensively deployed gated-clock nets, and aggressive supply voltage scaling. The maximum operating frequency is estimated to be 517 MHz at 1.0 V. Energy per decoded bit (E/bit) is given by Eq. (5.1).

\[
E/\text{bit} = \frac{\text{power}}{f_{\text{clk}} \cdot \text{bit}},
\]

where \(\text{bit}\) represents the number of bits decoded per clock cycle. The chip dissipates 30.1-179 pJ/bit at 0.32 V (16 MHz) and 0.75 V (256 MHz), respectively, with 64-QAM modulation. Six bits are decoded each clock cycle in the high-SNR regime due to complex-valued computation.

### 5.6 Performance Comparison

A comparison of the chip performance with a recently published state-of-the-art chip is listed in Table 5.1. This design supports multiple operation modes with adjustable modulation, array size, number of sub-carriers, and search algorithm. This design outperforms prior work in energy/bit and throughput despite added flexibility for multi-mode operation. The 6.6× higher energy efficiency comes from the design optimization framework, including clock-gating technique and aggressive voltage scaling. The 8× higher spectral efficiency stems from the larger supported antenna array size (4× gain) and operation in the complex-
Figure 5.11: Measured power and minimum supply voltage versus operating frequency.

valued domain (2× gain). The flexibility and energy efficiency are enabled by the low-complexity architecture and joint algorithm-architecture design.

To evaluate the area and energy efficiency of this flexible DSP chip, A dedicated 4×4 SVD (Singular Value Decomposition) chip [61] is used for comparison. The SVD chip implements an adaptive SVD algorithm for 4×4 MIMO systems and is optimized for reduced power and area. Figure 5.12 shows the comparison of the SVD chip and this work. With the aid of a hierarchical optimization methodology, the SVD chip achieves area- and energy-efficiency of up to 20 GOPS/mm² and 2.1 GOPS/mW [7]. The multi-mode sphere decoder achieves 1.4-8.2× higher energy efficiency (2.9-17.3 GOPS/mW) with 2.1× lower area efficiency (9.45 GOPS/mm²). The improved energy efficiency is attributed to optimization across design boundaries and arithmetic simplification. The area efficiency reduction is mainly due to memory overhead for input register bank and interconnect overhead for the multi-core architecture, which provides the flexibility for different search methods.
Table 5.1: Summary of state-of-the-art MIMO sphere decoders.

<table>
<thead>
<tr>
<th></th>
<th>Reference</th>
<th>ISSCC’09 [12]</th>
<th>This work [58]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modulation</strong></td>
<td></td>
<td>64-QAM</td>
<td>BPSK–64-QAM</td>
</tr>
<tr>
<td><strong>Data streams</strong></td>
<td></td>
<td>1</td>
<td>8–128</td>
</tr>
<tr>
<td><strong>Array size</strong></td>
<td></td>
<td>$4 \times$</td>
<td>$2 \times 2$–$16 \times 16$</td>
</tr>
<tr>
<td><strong>Algorithm</strong></td>
<td></td>
<td>K-best</td>
<td>K-best/Depth-first</td>
</tr>
<tr>
<td><strong>Soft/hard decision</strong></td>
<td></td>
<td>Hard</td>
<td>Hard</td>
</tr>
<tr>
<td><strong>Mode (Complex/Real)</strong></td>
<td></td>
<td>R</td>
<td>C</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td></td>
<td>130</td>
<td>90</td>
</tr>
<tr>
<td><strong>Clock freq (MHz)</strong></td>
<td></td>
<td>$282 \left( f_{\text{max}} \right)$</td>
<td>16–256</td>
</tr>
<tr>
<td><strong>Power (mW)</strong></td>
<td></td>
<td>135/1.3V</td>
<td>2.9/0.32V–275/0.75V</td>
</tr>
<tr>
<td><strong>Energy (pJ/bit)</strong></td>
<td></td>
<td>200</td>
<td>30–170</td>
</tr>
<tr>
<td><strong>Bandwidth (MHz)</strong></td>
<td></td>
<td>56 (est.)</td>
<td>16</td>
</tr>
<tr>
<td><strong>Max throughput (bps/Hz)</strong></td>
<td></td>
<td>12</td>
<td>12–96</td>
</tr>
</tbody>
</table>
Figure 5.12: Area and energy efficiency of dedicated 4×4 SVD chip and multi-mode sphere decoder.
5.7 Chapter Summary

A 16-core 16×16 MIMO sphere decoder ASIC is realized in a 5.29 mm² of core area in a 90-nm CMOS technology. Built on the scalable sphere decoder DSP kernel, the chip has flexibility and scalability in antenna array size (2×2 to 16×16), modulation scheme (QPSK to 64-QAM), search direction (forward or backward), and number of sub-carriers (8 to 128). By coordinating multiple PEs, multi-core architecture allows the flexibility in search method (K-best and depth-first). The BER performance is therefore improved since more possible solutions are examined. Energy efficiency is improved by distributed processing with frequency and supply-voltage scaling. Compared to the prior state-of-the-art MIMO sphere decoding chip, this chip has 6.6× higher energy efficiency and 8× better bandwidth efficiency despite added flexibility for multi-mode operation. The chip has up to 8.2× higher energy efficiency compared with the dedicated multi-antenna SVD chip. As the next step, it is interesting to investigate the extension of the flexibility to multiple signal bands. A multi-band MIMO sphere decoder will be described in the next chapter.
CHAPTER 6

Multi-Mode Multi-Band LTE Sphere Decoder Chip

To support OFDM systems, a scalable multi-band MIMO chip for the third-Generation Partnership Project (3GPP) Long-Term Evolution (LTE) standard and beyond is proposed in [62]. The chip implements sphere decoding algorithm with 16-core architecture. The chip is flexible to support multiple configurations: antenna arrays from 2×2 to 8×8, modulations from BPSK to 64-QAM, FFT sizes from 128 to 2048, and hard/soft outputs. In this chip, an 8×8 OFDM-based MIMO sphere decoder with a 128-2048 FFT block and a soft-output generation block is implemented in 3.35 mm², dissipating 13.83 mW in a standard-\( V_T \) 65-nm CMOS technology. Operating at 160 MHz, the chip provides a peak data rate of 960 Mbps in the 8×8, 64-QAM mode over a 20 MHz channel. LTE specs are met with 5.8 mW power for throughput of 480 Mbps. The key features of this decoder chip are described as follows:

- A power-area optimized reconfigurable FFT processor for 128-2048 points;
- A low-power register bank through clock-gating for soft-output generation;
- A simplified pre-processing unit by taking advantage of input data profile;
- Multiple voltage supplies for optimal performance tuning.
Table 6.1: 3GPP-LTE downlink design specifications

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth (MHz)</td>
<td>1.25, 2.5, 5, 10, 15, 20</td>
</tr>
<tr>
<td>FFT size</td>
<td>128, 256, 512, 1024, 1536, 2048</td>
</tr>
<tr>
<td>Antenna array</td>
<td>1×1, 2×2, 3×2, 4×2</td>
</tr>
<tr>
<td>Antenna array</td>
<td>QPSK, 16-QAM, 64-QAM</td>
</tr>
</tbody>
</table>

6.1 LTE Specifications

The 3GPP-LTE is the next generation cellular standard. LTE provides up to 100Mbps and 50Mbps data rate for downlink and uplink transmission [63]. LTE downlink design specs related to this work are listed in Table 6.1. The convergence of a variety of operation modes argues for a more flexible implementation, which makes the integration under a fixed power budget very challenging. Two key technologies used in the LTE standard are OFDM and MIMO. OFDM systems transmit data over several narrow-band sub-carriers. MIMO systems leverage spatial-multiplexing and diversity for improved data rate and range, adding another layer of data dimensionality.

6.2 System Architecture

The receiver architecture considered in this chip is shown in Fig. 6.1. The signals captured by multiple receive antennas are digitized through ADCs and then converted back to frequency domain through FFT. The modulated data streams carried over the narrow-band sub-carriers are constructively combined through the MIMO decoder. Soft outputs are generated for advanced error correction signal processing. A flexible (antenna, modulation, search method) MIMO sphere
decoder is demonstrated in the previous chip for single-band systems. In this chip, the flexibility is extended to multiple bands. A sphere decoder kernel, a reconfigurable FFT block, a soft-output register bank, and a pre-processing unit are integrated into single chip.

### 6.2.1 Reconfigurable FFT

OFDM is one of the modulation/multiplexing techniques for multi-carrier transmission with the minimum frequency spacing to keep the signal orthogonality [19, 64]. Given the $k$th data $X[k]$, the OFDM signal at the transmitter for $N$ data is given by

$$x[n] = \sum_{k=0}^{N-1} X[k]e^{j2\pi nk/N}.$$  

(6.1)

Except for the scaling factor $1/N$, this formulation is actually equal to $N$-point inverse discrete Fourier transform (IDFT), which can be efficiently implemented.
using the *inverse fast Fourier transform* (IFFT) algorithm. At the receiver, the received signal is converted back to $X[k]$ using FFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}. \quad (6.2)$$

Given $N = N_1 \times N_2$, the FFT operation can be represented in a 2-dimensional form [65] by

$$X[k] = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x[n_1 N_2 + n_2] e^{-j2\pi(n_1 N_2 + n_2)(k_1 + k_2 N_1) N_1 N_2} e^{-j2\pi n_2 k_1 N_1} e^{-j2\pi n_2 k_2 N_2}. \quad (6.3)$$

The 2-D decomposition allows to implement a large-size FFT using smaller FFTs. Various FFT sizes can therefore be supported by changing the combination of $N$.

A reconfigurable FFT block is implemented by several small processing units (PUs), as shown in Fig. 6.2. In this example, $N_1 = 256$, $N_2 = 8$ or 6. The PUs are modular to support different radix representations, as shown in Table 6.2, allowing power-area tradeoffs. The FFT block is scalable to support 128 to 2048 points by changing data-path inter-connection. Multi-path single-delay-feedback (SDF) architecture provides high utilization for varying FFT sizes. Unused PUs and delay lines are clock-gated for power saving. Twiddle (TW) factors are generated by trigonometric approximation [19] instead of fetching coefficients from ROMs for area reduction. The TW factors for FFT size 1536 are calculated through hardware reuse, which is infeasible for the ROM-based design.
Figure 6.2: (a) Reconfigurable FFT architecture. (b) Detailed circuits of PUs.
Table 6.2: Configuration of PUs for different radix representations

<table>
<thead>
<tr>
<th>Type</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radix-16</td>
<td>PU1+PU2+PU3+PU4</td>
</tr>
<tr>
<td>Radix-8</td>
<td>PU1+PU2+PU4</td>
</tr>
<tr>
<td>Radix-4</td>
<td>PU1+PU4</td>
</tr>
<tr>
<td>Radix-2</td>
<td>PU4</td>
</tr>
</tbody>
</table>

6.2.2 Soft-Output Generation and Pre-Processing Unit

The chip supports both hard-outputs (1/0) and soft outputs. Soft outputs are calculated by log-likelihood ratio $L(s_{i,b}|y)$ to describe the likelihood of each decoded bit $s_{i,b}$ ($i^{th}$ antenna, $b^{th}$ bit). The log-likelihood ratio (LLR) is defined by

$$L(s_{i,b}|y) = \ln \frac{p[s_{i,b} = 1|y]}{p[s_{i,b} = 0|y]} \approx \min_{s_{i,b}=0} \Lambda(s,y) - \min_{s_{i,b}=1} \Lambda(s,y),$$

where $\Lambda(s,y)$ is proportional to $\|y - Hs\|^2$. Soft outputs provide a 2-3dB improvement in BER performance in conjunction with proper channel coding schemes [5]. A iterative decoding scheme is shown in Fig. 6.3. The BER performance is improved due to the extra LLR information from the MIMO decoder. Fig. 6.4 shows the block diagram of soft-output register bank. Soft outputs are supported through a low-power clock-gated register bank. Operated in the WRITE mode, only registers storing $s_{i,b}$ are updated, while others are clock-gated to minimize power dissipation. In the READ mode, the register contents are scanned-out through a three-dimensional (antenna-bit-data stream (DS)) delay-line based shift-register chain to minimize power dissipation.

The pre-processing unit consists of eight arithmetic units to calculate $\tilde{y} =$
Figure 6.3: Iterative decoding scheme between inner and outer decoders.

Figure 6.4: Low-power clock-gated register bank for soft outputs.
\(Q^H y\) in parallel. The complex multiplier can be simplified since \(Q^H\) is precomputed and updated at a slower packet rate. Let \(y = a + jb\) and \(Q^* = c + jd\), only 3 multipliers and 3 adders are required for a complex multiplication (Eq. 6.5, where \(Q^- = a - b\) and \(Q^+ = a + b\)).

\[
(a + jb)(c + jd) = [c \times (a - b) + b \times (c - d)] + j[d \times (a + b) + b \times (c - d)]
= [cQ^- + b(c - d) + j[dQ^+ + b(c - d)].
\]  

(6.5)

6.3 Power-Area Minimization

Area and energy minimization is achieved by combining signal processing, architecture, and circuit techniques. High energy efficiency is reached through aggressive \(V_{DD}\) scaling. The supply voltages of most blocks are scaled down to around 0.4V to balance energy-delay tradeoff [6, 66]. 0.4V is also close to the optimal operating point for minimal energy dissipation [39]. The design has 5 \(V_{DD}\) domains. The supply voltages are adjusted according to the frequency specs to achieve minimal power consumption. Hard-output MMO sphere decoder kernel from [58] is redesigned to accommodate increased number of sub-carriers as required by LTE channelization.

The FFT block is optimized for minimal power-area product. Given a fixed FFT size, there are several possible architectures regarding the level of parallelism and arithmetic factorization, which are mapped to different configurations of PUs in Fig. 6.2. For example, the optimal architecture of the 2048-point FFT is decided in Fig. 6.5. All possible architectures are enumerated in the power-area space. The architecture with minimal power-area product is chosen as the optimal one. In Fig. 6.5(a), The increasing level of parallelism allows a more aggressive voltage scaling such that the power consumption can be minimized at the cost
of increased area [67]. Although $N_2 = 4$ and $N_2 = 8$ have similar power-area products, $N_2 = 8$ is chosen to address the power issue. In Fig. 6.5(b), by changing the factorization of 256-point FFT, different PU configurations can be explored. The optimal design is A13, which uses two radix-16 ($256 = 16 \times 16$) to connect PU1 to PU4 (Table 6.2). An overall $20 \times$ power-area product reduction is achieved if the architecture with the optimal parallelism and FFT factorization is adopted. Applying the same methodology to other FFT sizes, the optimal configurations are chosen with the minimum power-area product (solid-fill markers in Fig. 6.6).

At the architecture level, the optimal memory partition is also decided according to the power-area product. Fig. 6.7 shows possible memory partition schemes and the optimal partition for length 512 and 1024. At the circuit level, register-file (6T/bitcell) is adopted for the delay line with length larger than 512. Register-file-based design is estimated to consume only 29% power with an 85% silicon area compared to the SRAM-based counterpart. DFF-based delay lines are optimal in terms of power and area for length less than 512. Level-shifters are properly inserted between the low-$V_{DD}$ and high-$V_{DD}$ domains.

### 6.4 Measured Results

The chip diephoto is shown in Fig. 6.8. The die size is $3.16 \times 2.17 \text{ mm}^2$, in which the core size is $2.39 \times 1.40 \text{ mm}^2$. The chip key features and parameters are list in Table 6.3. Chip testing is performed with the use of a custom FPGA board for pattern generation and data analysis. FPGA feeds the test vectors into the chip and captures the outputs through an integrated Simulink/Matlab control interface. Each block was tested separately for individual voltage scaling. The chip dissipates 5.8 mW for the LTE standard, 13.83 mW for full $8 \times 8$ array with soft outputs. The hard-output sphere decoder kernel achieves E/bit of 15 pJ/bit
Figure 6.5: Choice of the optimal architecture of 2048-point FFT: (a) the level of parallelism and (b) arithmetic factorization.
Figure 6.6: Optimal PU configurations for 128-2048 FFTs.

Figure 6.7: Memory implementation for delay-line with length 512 and 1024.
Figure 6.8: Die photo of the LTE-compliant sphere decoder chip.

(18.7 GOPS/mW) and outperforms prior work [12, 58]. The E/bit performance is normalized to a 65-nm technology for a fair comparison. Power breakdown for different operation modes over 20 MHz band in the 64-QAM mode is shown in Fig. 6.9 and detailed in Table. 6.4. The chip fully supports LTE and has added flexibility for systems with larger array size or cooperative MIMO processing on smaller sub-arrays. The cost of flexibility is shown in Fig. 6.10. The area efficiency of this chip is reduced by $2.7 \times$ compared to the SVD chip [61], and $1.3 \times$ compared to the previous sphere decoder chip [58]. The efficiency reduction comes from the control and interconnect overhead of multi-PE, multi-PU architecture. The register-file memory overhead also reduces the area efficiency without contributing computations.
Table 6.3: Chip summary of the LTE-SD chip

<table>
<thead>
<tr>
<th>Technology</th>
<th>1P9M 65nm Std-$V_T$ CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. BW</td>
<td>20 MHz</td>
</tr>
<tr>
<td>FFT size</td>
<td>128, 256, 512, 1024, 1536, 2048</td>
</tr>
<tr>
<td>Modulation</td>
<td>BPSK, QPSK, 16-QAM, 64-QAM</td>
</tr>
<tr>
<td>Outputs</td>
<td>Hard/Soft</td>
</tr>
<tr>
<td>Mode</td>
<td>Complex valued</td>
</tr>
<tr>
<td>Core area</td>
<td>$2.39 \times 1.40 \text{ mm}^2$</td>
</tr>
<tr>
<td>Gate count</td>
<td>2,946K</td>
</tr>
<tr>
<td>Power</td>
<td>13.83 mW (5.8 mW for LTE)</td>
</tr>
<tr>
<td>Energy/bit</td>
<td>15 pJ (21 pJ [58], 100 pJ [12])</td>
</tr>
</tbody>
</table>

Figure 6.9: Power breakdown and measurement results of the LTE-SD chip.
Table 6.4: Detailed power consumption breakdown

<table>
<thead>
<tr>
<th>(V&lt;sub&gt;DD&lt;/sub&gt;, f&lt;sub&gt;clk&lt;/sub&gt;)</th>
<th>8×8 w/ soft outputs</th>
<th>4×4 w/o soft outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT core (parallel×8)</td>
<td>6.20 (0.45V, 20MHz)</td>
<td>2.83 (0.43V, 10MHz)</td>
</tr>
<tr>
<td>Register-file bank (32kb)</td>
<td>2.35 (1V, 40-160MHz)</td>
<td>1.18 (1V, 20-80MHz)</td>
</tr>
<tr>
<td>Hard-output SD kernel (16-core)</td>
<td>0.97 (0.42V, 10MHz)</td>
<td>0.45 (0.36V, 5MHz)</td>
</tr>
<tr>
<td>Pre-processing unit</td>
<td>4.06 (0.82V, 160MHz)</td>
<td>1.34 (0.64V, 80MHz)</td>
</tr>
<tr>
<td>Soft-output bank (parallel×8)</td>
<td>0.25 (0.42V, 20MHz)</td>
<td>NA</td>
</tr>
<tr>
<td>Total power</td>
<td>13.83 mW</td>
<td>5.8 mW</td>
</tr>
</tbody>
</table>

Figure 6.10: Area and energy efficiency comparison of SVD chip, SD chip, and LTE-SD chip.
6.5 Chapter Summary

A 8×8 3GPP-LTE compliant MIMO sphere decoder chip was described. A reconfigurable FFT for 128-2048 points is optimized from algorithm, architecture, down to circuit level. Different levels of parallelism and FFT factorization are evaluated to minimize power and area. The optimal memory partition is decided by the minimal power-area product. A 6T/bitcell register-file is chosen as the memory element due to its lower area and power as compared to SRAM. A low-power clock-gated register bank is used for soft outputs. A simplified multiplier is implemented in the pre-processing unit by leveraging specific input data profile. The chip is flexible to support multiple configurations: antenna arrays 2×2 to 8×8, modulations BPSK to 64-QAM, FFT sizes from 128 to 2048, and hard/soft outputs. The power consumption for the 3GPP-LTE standard is 5.8 mW in 3.35 mm² area in a 65-nm CMOS technology.
CHAPTER 7

Conclusion

This dissertation addresses the complexity, energy efficiency, and flexibility challenges in digital circuit design for MIMO signal processing. Methodology for design optimization is illustrated on a MIMO sphere decoding algorithm. The choice of the optimal design, considering the power-area tradeoff, is highly influenced by the design goal. Equal weight is placed on both power and area in this work. This dissertation presents a flexible DSP kernel, a multi-core architecture, and a reconfigurable FFT processor for MIMO-OFDM signal processing. To improve the flexibility without compromising area or energy efficiency, algorithmic, architectural, and circuit parameters are jointly considered. Leveraging application-specific data profile and using a lower supply voltage and operation frequency, the energy efficiency is significantly improved through distributed processing over multiple processing elements.

An integrated Simulink environment provides rapid hardware resource evaluation for architecture exploration. The power and area information of a design can be extracted efficiently by using automatic logic mapping. Characterization from the basic blocks provides fair estimates for low-voltage system design using normal timing information. The resulting design meets system specs with an improved energy efficiency. As a proof of concept, an energy efficiency of 17.3 GOPS/mW is verified in a 16×16 multi-mode single-band chip. Extending the flexibility for multiple frequency bands, an energy efficiency of 10.2 GOPS/mW
is verified in an 8×8 3GPP-LTE compliant MIMO decoder chip. These efficiency levels match or even exceed those found in dedicated chips.

7.1 Research Contributions

The goal of this research is to minimize the circuit power and area with added flexibility for MIMO signal processing. Starting from a flexible DSP kernel for MIMO sphere decoding, this dissertation presents the multi-core architecture and design techniques for multi-mode operation. Following are major contributions of this research:

- Development of a Gray-code-based multiplier that provides a 38% area reduction compared to the direct-mapped implementation. The hardware is simplified by leveraging algorithmic transformations and wordlength reduction.

- Development of a decoding scheme using a decision-plane transformation. The decoding method reduces the area by 4.6× and the delay by 5× compared to the prior work.

- Development of a simplified enumeration scheme. A low-complexity scheme for candidate enumeration is implemented by taking advantage of geometric relationships and subset partitioning.

- Development of a low-complexity, scalable DSP kernel for MIMO sphere decoding. The scalable processing element (PE) has the flexibility in antenna array size, modulation scheme, and multiple sub-carriers through folded architecture, symbol remapping, and data-interleaved processing. A 20% area reduction is achieved compared to the reference design.
• Development of multi-dimensional data access for a multi-core architecture. Data streams are buffered and fetched through a multi-dimensional shift-register chain and clock-gated registers. This reduces the memory area by 16× and the memory power by 90%.

• Development of a reconfigurable FFT processor to support 128 to 2048 points by changing the data-path inter-connection between processing units. The FFT block is optimized across the algorithm, architecture, and circuit boundaries. It supports various radix configurations and voltage scalings to minimize power-area product.

• Application of aforementioned concepts to a 16×16, multi-mode single-band MIMO decoder. The chip achieves an energy efficiency of 17.3 GOPS/mW and an area efficiency of 9.45 GOPS/mm². It dissipates 2.89 mW from a 321-mV supply voltage.

• Extension of the flexibility to multiple bands and implementation of an 8×8 multi-mode, multi-band MIMO decoder. The chip achieves an energy efficiency of 10.2 GOPS/mW and an area efficiency of 7.1 GOPS/mm². The power consumption is 5.8 mW for the 3GPP-LTE standard.

The minor contributions of this work are:

• Application of the architectural optimization framework to the proposed design. The Simulink design environment, the hardware emulation, and the FPGA-aided verification form a unified framework for architecture evaluation and chip testing.

• Development of a multi-core search algorithm, allowing the flexibility in search method (K-best and depth-first). The BER performance is therefore
improved since more possible solutions are examined.

- Detailing of the multi-core architecture, including the memory structure, the interconnect network, and the control mechanism. The energy efficiency is improved by distributed processing with frequency and supply-voltage scaling.

- Development a low-power clock-gated register bank for soft-output generation. Only active registers are updated, while others are disabled for power saving.

The design methodology and circuit techniques proposed in this dissertation enable energy-efficient ASIC implementation of advanced multi-band MIMO signal processing.

7.2 Future Work

It is interesting to look into adaptive DSP kernels for future radios that include spectrum sensing and optimization across various decoding strategies such as MIMO decoders, space-time decoders, and FEC decoders (e.g., LDPC decoders). Lessons learned from MIMO decoders can be extended to other decoding strategies as well. Another research direction is to build cooperative/distributive MIMO systems by extending the flexibility for multiple users.

At the architecture/circuit implementation level, multi-core architectures are a promising solution for dealing with high computational complexity with acceptable power density. However, the overhead of interconnect dramatically increases with an increasing number of processing cores. Therefore, there is a great need for cost- and energy-efficient interconnect design to make the hardware truly
low-cost, flexible, and energy-efficient. A hierarchical interconnect network is attractive and may be applicable to energy-efficient, reconfigurable computing architectures.
REFERENCES


