Design of Detectors and Decoders for MIMO Wireless Systems

by

Wei Tang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in The University of Michigan
2019

Doctoral Committee:

Associate Professor Zhengya Zhang, Chair
Professor Michael P. Flynn
Associate Professor Justin C. Kasper
Assistant Professor Hun-Seok Kim

Wei Tang

weitang@umich.edu

ORCID iD: 0000-0001-5204-9728

To my family and friends for their love and support

ACKNOWLEDGEMENTS

I would like to thanks to my advisor, Professor Zhengya Zhang for his valuable support and guidance. He has been give me great advice, encouragement and inspiration over the years of my PhD life; And I would like to thanks to Professor Michael Flynn, Justin Kasper and Hun-Seok Kim for serving on my dissertation committee and providing valuable feedback;

I am lucky to meet and work with my fellow colleagues in my research group. Thanks to Thomas, Chester, Sung-Gun, Teyuh, Jie-Fang and Shiming, Phil, Chia-Hsiang, Youn Sung for providing a fun and healthy environment in the office; we have had many inspirational discussions about research problems and nice chat about daily life. Also, I would like to thanks to the collaborators at Lund University, Professor Viktor wall, Professor Liang Liu, Hammenth, Rakesh, and Farrokh for being so kind and helpful when I visiting there.

Last but not least, I want to express my deepest appreciation to my family in Taiwan for their unconditional love and support.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF ABBREVIATIONS

3GPP The 3rd Generation Partnership Project

AMM Approximated Moment-Matching

ASIC Application-Specific Integrated Circuit

AWGN Additive White Gaussian Noise

BER Bit Error Rate

BS Base Station

b-sub Backward-Substitution

CAGR Compounded Annual Growth-Rate

CAM Content Addressable Memory

CHEMP Channel Hardening-Exploiting Message Passing

CN Check Node

CPE Constellation Processing Element

CSI Channel State Information

DSP Digital Signal Processor

DVFS Dynamic Voltage Frequency Scaling

EMS Extended Min-Sum

EPD Expectation Propagation Detection

EVN Elementary Variable Node

FBB Forward Body Biasing

FD-SOI Fully-Depleted Silicon-on-Insulator

FEC Forward Error Correction

FER Frame Error Rate

FIFO First-In First-Out

FLOP Floating Point Operations

FSM Finite State Machine

f-sub Forward-Substitution

GF Galois Field

GSM Global System for Mobile communications

HSPA+ Evoluted High Speed Packet Access

IAI Inter-Antenna Interference

i.i.d. Independent and Identically Distributed

IC Integrated Circuit

IO Input Output

IPE Interference Processing Element

ISI Inter-Symbol Interference

LDPC Low-Density Parity Check

LLR Log-Likelihood Ratio

LLRV Log-Likelihood Ratio Vector

LOS Line-of-Sight

LTE Long Term Evolution

LTE-A LTE-Advanced

LUT Look-Up Table

MAC Multiply-Accumulate

MF Matched Filter

MIMO Multiple-Input Multiple-Output

MISO Multiple-Input Single-Output

ML Maximum Likelihood

MMSE Minimum Mean Square Error

MPD Message-passing Detection

MSE Mean Square Error

NBLDPC Non-Binary Low-Density Parity Check

NLOS Non-Line-of-Sight

NS Neumann Series

OFDM Orthogonal Frequency-Division Multiplexing

PAM Pulse-Amplitude Modulation

PAR Peak-to-Average Ratio

PE Processing Element

PER Packet Error Rate

PIC Parallel Interference Cancellation

PLL Phase Locked Loop

QAM Quadrature Amplitude Modulation

RBB Reverse Body Biasing

RF Radio-Frequency

RNG Random Number Generator

RTL Register-Transfer Level

RX Receive

SD Sphere Decoder

SER Symbol Error Rate

SIC Successive Interference Cancellation

SIMO Single-Input Multiple-Output

SINR Signal-to-Interference and Noise Ratio

SISO Single-Input Single-Output

SNDR Signal-to-Noise and Distortion Ratio

SNR Signal-to-Noise Ratio

SQNR Signal-to-Quantization-Noise Ratio

TX Transmit

UMTS Universal Mobile Telecommunications Service

UTBB Ultra Thin Body BOX

VN Variable Node

WLAN Wireless Local Area Network

ZF Zero-Forcing

ABSTRACT

Multiple-Input-Multiple-Output (MIMO) technology makes use of multiple transmit and receive antennas to improve the spectral efficiency and reliability by spatial diversity and multiplexing. However, MIMO systems require complicated baseband detector designs to cancel the Inter-Antenna Interference (IAI). This work develops high-performance and energy efficient MIMO detectors using state-of-the-art iterative detection and decoding, message-passing detection and expectation-propagation detection approaches.

Iterative Detection and Decoding, or IDD, is an iterative receiver design that improves the error rate performance and relaxes both the detector and decoder designs by iterating the soft decisions between the detector and the decoder. Through iterative interference cancellation and error correction, the Signal-to-Interference-and-Noise Ratio (SINR) can be substantially improved. In this dissertation, a 2.4 mm^2 4×4 MIMO IDD ASIC incorporating Minimum Mean Squared Error (MMSE) detector and Non-Binary Low-Density Parity Check (NBLDPC) decoder is demonstrated. The IDD chip exploits an efficient nonbinary interface between the detector and the decoder to achieve a 1.02 Gb/s throughput, 20 pJ/b energy efficiency and superior detector performance compared to previous detector designs.

The upcoming 5G wireless communication relies on scaling up the numbers of antennas at the base station, using a new technology known as massive MIMO. Massive MIMO often refers to a multiple-user wireless communication system, where the base station is equipped with hundreds of antennas and serves tens of single-antenna user

terminals. The large number of antennas entails a high complexity in baseband digital signal processing. To lower the complexity, previously demonstrated massive MIMO systems deployed linear detectors. Despite its simplicity, a linear detector requires expensive matrix inversion operations, the cost of which can become excessive in a massive MIMO system.

In a rich scattering environment, a massive MIMO channel can be modeled as an independently and identically distributed (i.i.d.) Rayleigh fading channel. This favorable property allows us to explore approximate detection algorithms to greatly reduce the computational complexity while still maintaining close-to-optimal BER-SNR performance. This idea is realized and demonstrated in a 0.58 mm^2 128 × 32 low complexity Message-Passing Detector (MPD) for a massive MIMO base station. With a symbol hardening technique, the complexity of the MPD is reduced by more than 60 %. The detector is implemented in a pipelined block-parallel architecture using a layered-grouped schedule to accelerate convergence, enabling an average throughput of 2.76 Gb/s at 221 mW. The chip incorporates adaptive precision control and clock gating to improve energy efficiency to 80 pJ/b.

Practical massive MIMO channels for mobile applications are fast varying. From the channel measurement in practical environments conducted by Lund University, it is discovered that the massive MIMO channel could vary and degrade. A highly correlated channel is observed when mobile users are closely deployed in an environment where the line-of-sight (LOS) signal propagation paths dominate. In such a condition, a conventional linear MMSE detector is unable to satisfy the BER-SNR requirement. To address this challenge, a 2.0 mm^2 iterative expectation-propagation detector (EPD) is presented for a 128 × 16 massive MIMO system supporting up to 256-QAM modulation. Tested with measured channel data, the detector achieves 4.3 dB processing gain over state-of-the-art massive MIMO detectors, enabling 2.7 times reduction in transmit power for battery-powered mobile terminals. The EPD

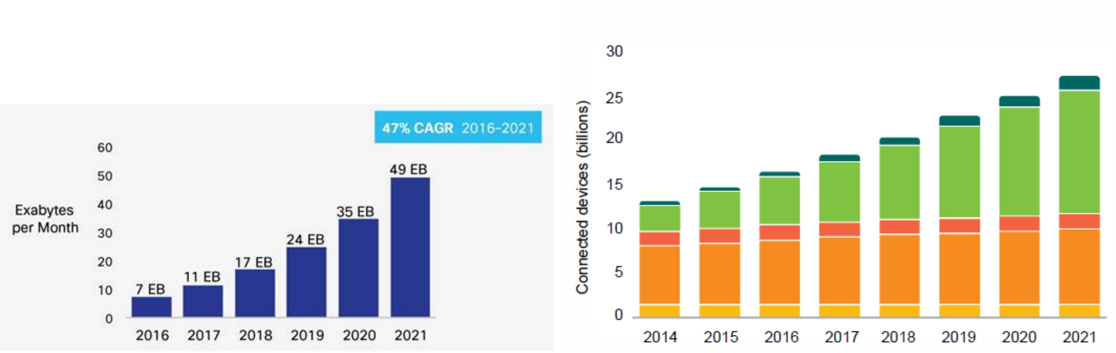
chip uses link-adaptive processing to meet a variety of practical channel conditions with scalable energy consumption. The design is realized in a condensed systolic array architecture and an approximate moment-matching circuitry to reach 1.8 Gb/s at 70.6 pJ/b. The performance and energy efficiency can be tuned over a wide range by the UTBB-FDSOI body bias.

This dissertation studies the NBLDPC decoder and MIMO detector designs for both small-scale and large-scale MIMO system. The cross-domain optimizations from system and algorithm to architecture and circuits led to a highly efficient and adaptive detector and decoder designs to meet the demanding performance requirements of future mobile communication systems.

CHAPTER I

Introduction

The global data traffic is predicted to have a Compounded Annual Growth-Rate (CAGR) of 47 % with 25 billion IoE connected devices by 2021 [2, 3, 4] as shown in Fig. 1.1. It is envisioned that the upcoming 5G wireless communication will meet the traffic demand at a 100 times higher energy efficiency, 10 times higher spectral efficiency and reliability compared to the current 4G standard. Small-scale MIMO and massive MIMO, exploiting the use of multiple numbers of antennas, is a key enabling technology.



(a) Prediction from Cisco.[2]

(b) Prediction from Ericsson.[4]

Figure 1.1: Global mobile data traffic and connected devices growth trend.

1.1 MIMO Technology

Since throughput = spectral efficiency (bits/s/Hz) × bandwidth (Hz), one way of increasing data throughput is to increase the number of elements along the spatial dimension by placing multiple antennas at both the transmitting and receiving side of the communication link. This MIMO technology has many advantages: it increases the data throughput [5] and the link reliability by exploiting the link diversity [6] in rich scattering environments.

Based on the configurations of the number of transmit (TX) antennas and the number of receive (RX) antennas , MIMO systems can be classified as single-input single-output (SISO), single-input multiple-output (SIMO), multiple-input single-output (MISO), and multiple-input multiple-output (MIMO) systems, as shown in Fig. 1.2.

The data rates of wireless communication standards from the second generation (2G) to the fifth generation (5G) are shown in Fig.1.3. MIMO technology has been introduced starting from 3G to satisfy the ever increasing data rate requirement. Recent

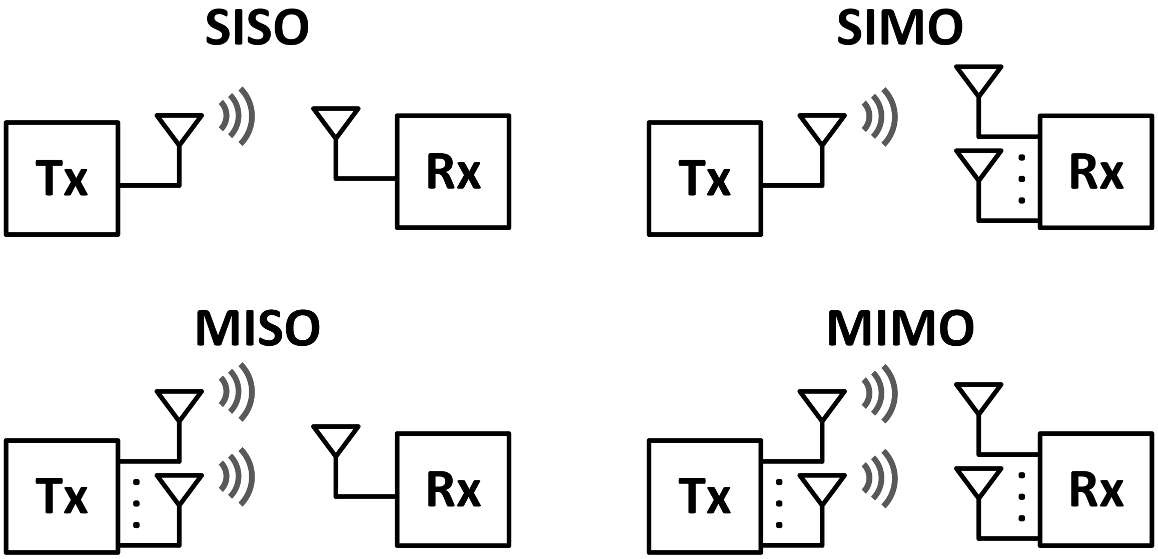


Figure 1.2: Illustration of four configurations of multiple antenna systems: SISO, SIMO, MISO and MIMO.

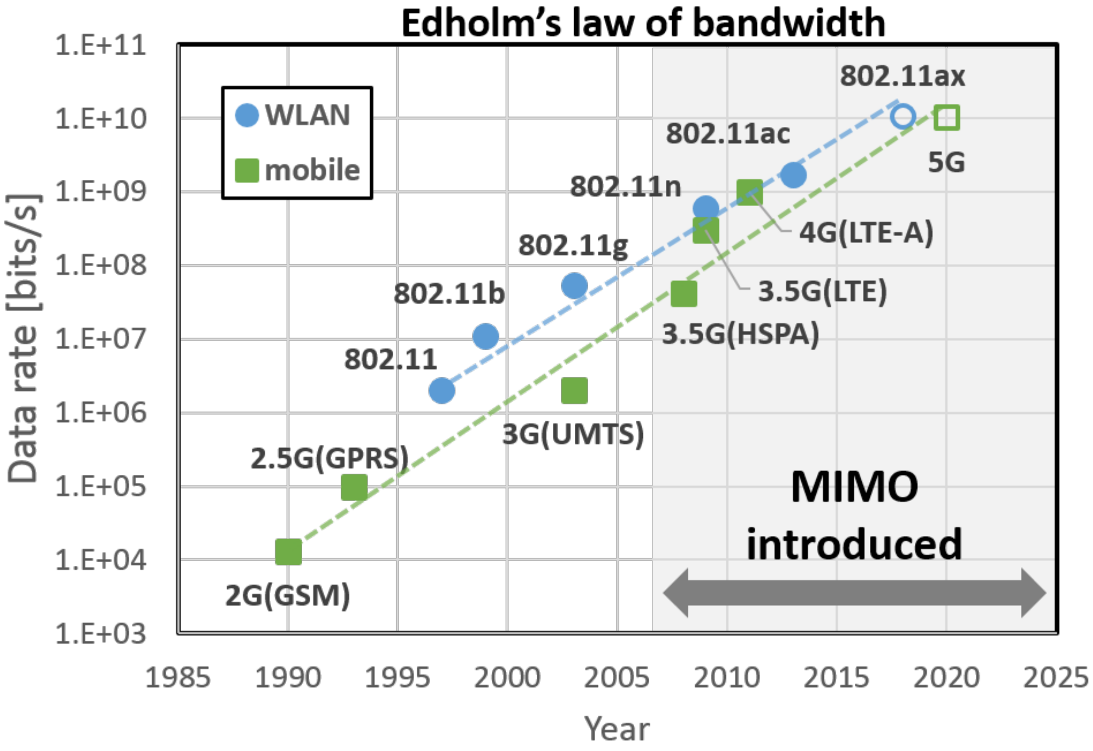


Figure 1.3: The data rate of wireless communication standards.

wireless communication standards such as IEEE 802.11n, 802.11ac, Evoluted High Speed Packet Access (HSPA+) [7], 3GPP Long-Term Evolution (LTE) Advanced Release 10/11 [8], mobile communication, and WiMAX all adopt MIMO communication to increase spectral efficiency and data rate. For example, IEEE 802.11n allows for up to a 4 × 4 antenna configuration (4 transmit and 4 receive antennas); IEEE 802.11ac calls for up to a 8 × 4 antenna configuration and 3GPP LTE Advanced release 10 [8] specifies up to an 8 × 8 antenna configuration. The enhancement in spectral efficiency and higher data rate comes with a significant computational cost: workload profiling indicates that MIMO detection at the receiver can consume a large number of computing cycles in the physical layer baseband processing [9].

The increasing data rates motivate further exploitation of a large number of antennas. As pointed out in [10], the effects of the additive receive noise, the small-scale fading, as well as the inter-user interference could be averaged out by increasing the number of antennas at the base station. Scaling up MIMO provides higher degrees of freedom in the spatial domain to increase the overall spectral efficiency and reliability. A massive MIMO is a system equipped with a larger number of antennas (in hundreds) at the base station. Such a massive MIMO system typically serves multiple user terminals (in tens) using the same time-frequency resources.

Another important distinction of a massive MIMO system is that it exploits the multi-path components are exploited in massive MIMO to improve the signal strength at the receiver. This is possible due to the high spatial resolution and array gains in massive MIMO with a large number of antennas at the base station. Moreover, massive MIMO allows processing efforts to be concentrated at the base station side, while battery-operated mobile terminals can have low hardware cost and power consumption.

1.2 MIMO Signal Detection

One of the most complex and energy consuming blocks in a MIMO system is the signal detection at the receiver. This is because transmitting multiple streams simultaneously through a wireless channel would result in the crosstalk of the signals at the receiver. Therefore, signal processing needs to be performed to separate the data streams, which is generally termed MIMO detection.

Consider a MIMO system as shown in Fig. 1.4 where N_t transmit antennas communicate to a receiver with N_r antennas and each antenna transmits an M-Quadrature Amplitude Modulation (QAM) symbol at each channel use. The system model is given in (1.1), where $\mathbf{x}^c = [x_1^c, x_2^c, ..., x_{N_t}^c]^\mathsf{T}$ is the transmitted vector of QAM symbols, $x_i^c \in \tilde{\mathcal{A}}$ is represented by a complex value, and $|\tilde{\mathcal{A}}| = M$. $\mathbf{H}^c$ is a $N_r \times N_t$ complex matrix representing a memoryless flat-fading complex MIMO channel. For an i.i.d. Rayleigh channel, each coefficient of $\mathbf{H}^c$ is drawn according to a complex zero-mean unit-variance Gaussian distribution. Assume the perfect channel state in-

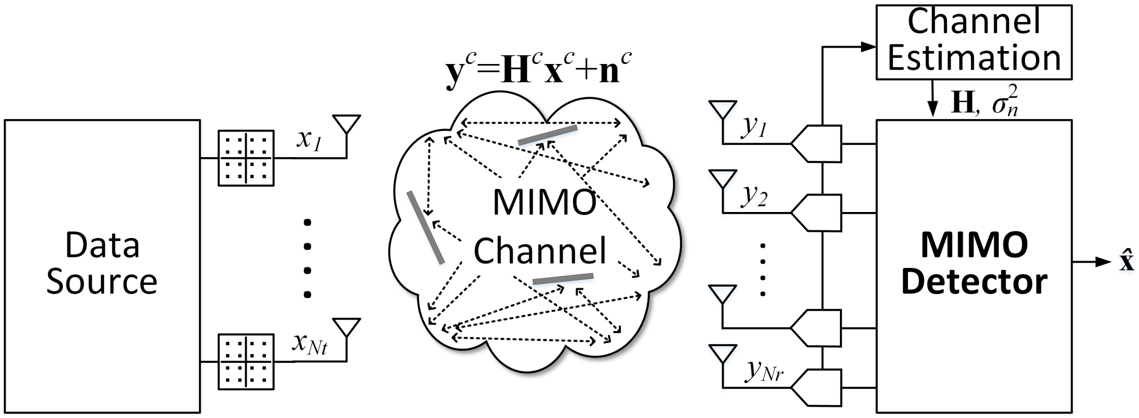


Figure 1.4: Illustration of an $N_t \times N_r$ multi-user MIMO system.

formation (CSI) at the receiver and $\mathbf{H}^c$ is known at the receiver. The channel output is $\mathbf{y}^c \in \mathbb{C}^{N_r}$, where

$$\mathbf{y}^c = \mathbf{H}^c\mathbf{x}^c + \mathbf{n}^c \tag{1.1}$$

and $\mathbf{n}^c \in \mathbb{C}^{N_r}$ is an additive white circular-symmetric complex Gaussian noise vector with independent zero-mean components and the N_0-variance. According to this model, the Signal-to-Noise Ratio (SNR) is defined as:

$$\text{SNR(dB)} = 10\log_{10}(N_t E_s/N_0) \tag{1.2}$$

where E_s is the constellation average energy in Joules.

The complex values in (1.1) can be written in the real domain as:

$$\begin{bmatrix}\Re(\mathbf{y}^c)\\ \Im(\mathbf{y}^c)\end{bmatrix} = \begin{bmatrix}\Re(\mathbf{H}^c) & -\Im(\mathbf{H}^c)\\ \Im(\mathbf{H}^c) & \Re(\mathbf{H}^c)\end{bmatrix}\begin{bmatrix}\Re(\mathbf{x}^c)\\ \Im(\mathbf{x}^c)\end{bmatrix} + \begin{bmatrix}\Re(\mathbf{n}^c)\\ \Im(\mathbf{n}^c)\end{bmatrix}$$

$$\Rightarrow \mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1.3}$$

where $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts, respectively. And note that the real and imaginary part of a QAM symbol are represented by the two underlying Pulse Amplitude Modulation (PAM) symbols from alphabet $\mathcal{A}$, i.e. $\mathbf{x} \in \mathcal{A}^{2N_t}$, where

$|\mathcal{A}| = \sqrt{M}$.

1.2.1 Maximum Likelihood Detection

Upon observing $\mathbf{y}$, the Maximum Likelihood (ML) detection enumerates all possible combinations of the transmitted constellation symbols and find the best (most-likely) combination as the estimate. The posterior of the estimation is:

$$P(\mathbf{x} \mid \mathbf{y}, \mathbf{H}) \propto P(\mathbf{y} \mid \mathbf{x}, \mathbf{H}) P(\mathbf{x}). \tag{1.4}$$

Here each of the prior is uniformly and discretely distributed over the QAM constellation, thus the overall prior $P(\mathbf{x})$ spans a space of $\mathcal{A}^{2N_t}$. The maximum likelihood estimates can be formulated as follows:

$$\mathbf{x}^{ML} = \underset{\mathbf{x} \in \mathcal{A}^{2N_t}}{\arg\max}\, P(\mathbf{y} \mid \mathbf{x}, \mathbf{H}). \tag{1.5}$$

Maximum likelihood detection can achieve the optimal error rate performance, however, the complexity of such an enumeration procedure increases exponentially with the number of transmit antennas (N_t) and the QAM order (M). Many researchers have worked on efficient and reduced search space for MIMO detection, such as sphere decoding [11, 12, 13, 14] and Tabu search [15, 16], to manage the implementation cost in hardware. However, these search-based detections still scale poorly, especially in a large-scale MIMO system using more than 8 antennas.

1.2.2 Minimum Mean Equared Error (MMSE) Detection

A low-complexity linear detection, such as Minimum Mean Squared Error (MMSE) or Zero-Forcing (ZF), has sub-optimal performance in terms of error rate. To avoid exhaustive enumeration, an MMSE detection approximates the prior in (1.4) by a continuous Gaussian distribution with zero-mean and a variance equal to the symbol energy E_s:

$$P(\mathbf{x} \mid \mathbf{y}, \mathbf{H}) = \mathcal{N}(\mathbf{y}; \mathbf{H}\mathbf{x}, N_0\mathbf{I}) \prod \mathcal{N}(x_i; 0, E_s). \tag{1.6}$$

With the continuous Gaussian approximation of the prior, the estimates having the minimum mean squared error can now be calculated as:

$$\begin{aligned} \mathbf{x}^{MMSE} &= (\mathbf{H}^H\mathbf{H} + N_0/E_s\mathbf{I})^{-1}(\mathbf{H}^H\mathbf{y}) \\ &= \mathbf{A}^{-1}\,\mathbf{y}^{MF}, \end{aligned} \tag{1.7}$$

where we define the MMSE filtering matrix $\mathbf{A}^{-1}$ and matched-filtered received signals $\mathbf{y}^{MF}$. The result of MMSE detection are "soft" symbols, i.e. the i-th estimate is a Gaussian distribution with a mean of x_i^{MMSE} and a variance of A_{ii}^{-1}.

There is an inherent trade-off between accuracy and complexity when implementing a MIMO detector. Our design goal is to achieve an acceptable accuracy with as low complexity as possible. Different methods will be explored in this dissertation: in Chapter II, an iterative detection and decoder (IDD) technique is investigated to improve detection and decoding accuracy while simplifying both detector and decoder designs; in Chapter III, a message-passing detection (MPD) is designed to take advantage of an i.i.d. Rayleigh massive MIMO channel to reduce complexity; and in Chapter IV, a versatile expectation-propagation detection (EPD) is adopted to ensure detection accuracy in highly correlated massive MIMO channels with adaptive power consumption.

1.3 Dissertation Outline

The MIMO signal detector is one of the most energy- and latency-critical parts in the MIMO baseband digital processing. The complexity of the detector grows with the scale of a MIMO system. This dissertation focuses on efficient MIMO detector designs for both small-scale and large-scale MIMO systems.

In Chapter II, an MMSE-NBLDPC IDD design for a 4×4 MIMO system is presented. The IDD technology enables competitive BER-SNR performance with a low-complexity MMSE detector and a nonbinary LDPC decoder.

In Chapter III, a low-complexity message-passing detector (MPD) design for a 128×32 massive MIMO system is presented. With the i.i.d. Rayleigh channel assumption in a massive MIMIO system, the proposed high-throughput MPD design can be substantially simplified to achieve high energy efficiency and the optimal detection performance.

In Chapter IV, a link-adaptive close-to-optimal iterative expectation-propagation detector (EPD) for practical massive MIMO systems is presented. The design is able to adapt to different practical channel scenarios from the worst-case highly correlated channel to the best-case i.i.d. channel.

In Chapter V, we conclude this dissertation and provide outlooks for future research in massive MIMO processor design.

CHAPTER II

MMSE-NBLDPC Iterative Detector and Decoder

The latest MIMO wireless systems have adopted iterative detection and decoding (IDD) to reduce the signal-to-noise ratio (SNR) required for a reliable transmission. An IDD system consists of a soft-in soft-out (SISO) detector to cancel interference, and a SISO forward error correction (FEC) decoder to remove errors. The two blocks exchange soft information to improve the frame error rate (FER) iteratively.

State-of-the-art IDDs based on sphere decoding (SD) and binary low-density parity-check (LDPC) FEC have been demonstrated in [17], [18] for up to 4x4 64-QAM MIMO system, achieving up to 396Mb/s in detection throughput [17] and 586Mb/s in decoding throughput [18]. As antenna configuration continue to scale beyond 4×4 and modulation order increases above 64-QAM, the complexity of a SISO SD detector is expected to grow exponentially, making it impractical. A SISO minimum mean square error (MMSE) detector [1] features a lower complexity and a higher throughput than a SISO SD detector. An MMSE detector can be more easily scaled to support a large antenna configuration and a high order modulation. The drawback of an MMSE detector is its lower detection performance (measured in error rate). However, an IDD system potentially overcomes this weakness by iteration.

Recent IDD designs have used binary low-density parity-check (LDPC) codes for FEC [17, 18, 19]. However, binary LDPC codes are not matched to high-order modulation and a loss is expected. Compared to binary LDPC codes, nonbinary LDPC (NBLDPC) codes defined over Galois field (GF) outperform binary LDPC codes of comparable block length in coding gain [20]. Even at a moderate block length, an NBLDPC code offers a superior coding gain, and the coding gain improves with a larger GF size.

Despite the good coding gain, the decoding of NBLDPC codes over a large GF(q) requires intensive computation. To reduce the decoding complexity, Declercq and Voicila [21, 22] proposed the truncated extended min-sum (EMS) decoding algorithm, using only n_m, where $n_m \ll q$, most reliable entries in a q-element log likelihood ratio vector (LLRV) as the message being passed between check nodes (CNs) and variable nodes (VNs) in decoding. The truncation reduces the elementary operation of decoding from $O(q^2)$ to $O(qn_m)$, sacrificing only marginal bit error rate (BER). Using the truncated EMS algorithm, the recent work by Park [23] demonstrated a Gb/s NBLDPC decoder. Used in an IDD system, an NBLDPC decoder is expected to enhance the detection-decoding performance [24].

In this work, we present the first MMSE-NBLDPC IDD implementation and the design optimizations. We match the GF size of the NBLDPC code with the QAM constellation size, thereby improving the performance and simplifying the detector-decoder interface. The superb error-correcting capability provided by the NBLDPC code allows us to apply a much simplified version of the truncated EMS decoding using only the top dozen entries out of a 256-entry LLRV to simplify the decoder implementation. Both the detector and the decoder designs are optimized through algorithm, architecture, and circuit techniques to achieve higher throughput and lower power consumption compared to prior art.

The rest of the paper is organized as follows. In Section 2.1, we present the background of MMSE detection algorithm and the truncated EMS decoding algorithm. Our unique nonbinary interface design is described in Section 2.2. The circuit, ar-

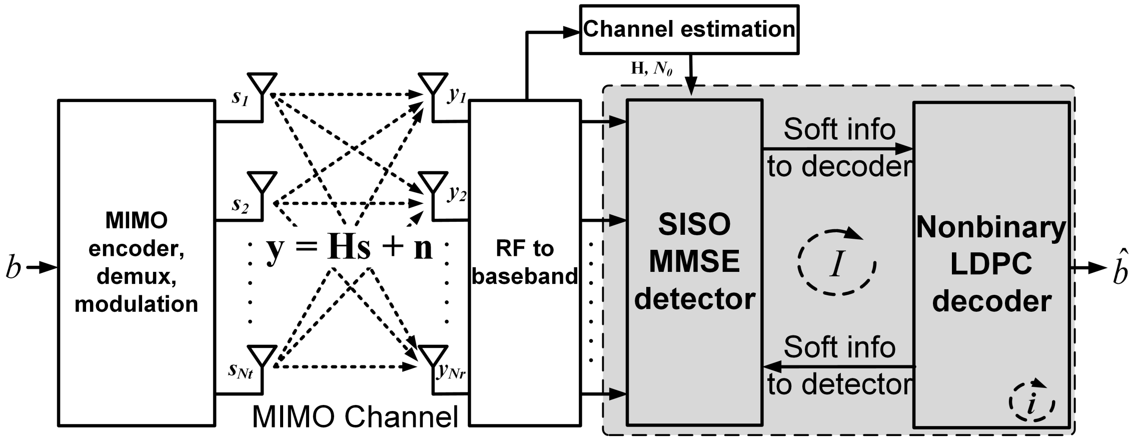


Figure 2.1: Illustration of an $N_r \times N_t$ IDD MIMO system with a soft-input-soft-output (SISO) MMSE detector and an onbinary LDPC decoder.

chitecture, and algorithm co-optimization for the MMSE detector and the NBLDPC decoder are described in Section 2.3 and 2.4, respectively. Section 2.5 provides silicon measurement results and conclusions are drawn in Section 2.7.

2.1 Background

The block diagram of an IDD MIMO system is shown in Fig. 2.1. At the MIMO transmitter, the source bits $\mathbf{b}$ are encoded by an FEC encoder into a codeword. The codeword is mapped to QAM symbols $\mathbf{s}$ and subsequently sent over N_t parallel transmit antennas. The signals travel through a wireless channel that introduces fading, interference and noise. At the MIMO receiver, N_r antennas pick up the corrupted signals that are converted to the received symbols $\mathbf{y}$. Mathematically, $\mathbf{y}$ is given by

$$\mathbf{y} = \mathbf{Hs} + \mathbf{n}, \tag{2.1}$$

where $\mathbf{y} \in \mathbb{C}^{Nr\times 1}$, channel matrix $\mathbf{H} \in \mathbb{C}^{Nr\times Nt}$, transmitted QAM symbols $\mathbf{s} \in \mathbb{C}^{Nt\times 1}$, and the noise $\mathbf{n}$ is modeled as an additive white circular-symmetric complex Gaussian vector with independent zero-mean components and N_0-variance

A SISO MMSE detector performs MMSE filtering to cancel the interference on the received data from other antennas. The detector outputs soft symbols and their variances, which represent the estimated symbols in a signal constellation and the likelihoods of the symbols. The soft symbols and variances are converted to a-prior log-likelihood ratio (LLR) for each bit to be used in a SISO FEC decoder. A SISO FEC decoder performs error correction, and it outputs a-posteriori LLRs. The posterior LLRs are converted to soft symbols and variances, and fed back to the SISO detector to start the next IDD iteration. IDD iterations improve the quality of detection and decoding. A successful convergence is indicated by the convergence of soft symbols and narrowing of variances. In the following subsections, we provide a brief background on MMSE detection and NBLDPC decoding.

2.1.1 MMSE-PIC Detection

In this work, we use the MMSE parallel interference cancellation (MMSE-PIC) algorithm based on [1] as described below. In an IDD system, step 1 (pre-processing) is only done in the first iteration.

Step 1. Pre-processing: Compute the Gram matrix $\mathbf{G} = \mathbf{H}^{\mathsf{H}}\mathbf{H}$, and perform match filtering $\mathbf{y}^{MF} = \mathbf{H}^{\mathsf{H}}\mathbf{y}$. The Gram matrix and match filtering are done once in the first IDD iteration, and reused in subsequent iterations.

Step 2. Initialization: Compute soft symbol s_t and its variance σ_t^2, $t = 1...N_t$, based on the assumption of Gaussian distribution; Compute MMSE filter matrix $\mathbf{A}$:

$$\mathbf{A} = \mathbf{G}\boldsymbol{\Lambda} + N_0\mathbf{I}, \tag{2.2}$$

where $\boldsymbol{\Lambda} = \text{diag}(\sigma_1^2, ..., \sigma_{N_t}^2)$, and N_0 is the estimated channel noise variance.

Step 3. Matrix inversion: Compute $\mathbf{A}^{-1}$ by performing lower-upper decomposition (LUD) to get $\mathbf{A} = \mathbf{LU}$, followed by forward substitution to get $\mathbf{L}^{-1}$ and backward substitatuion to get $\mathbf{A}^{-1}$.

Step 4. Interference cancellation:

$$\mathbf{y}_t^{IC} = \mathbf{y}^{MF} - \sum_{j \neq t} \mathbf{g}_j s_j,\ t = 1...N_t, \tag{2.3}$$

where $\mathbf{g}_j$ is the j-th column of $\mathbf{G}$.

Step 5. MMSE filtering:

$$\hat{\mathbf{s}}_t = \mu_t^{-1} \mathbf{a}_t^H \mathbf{y}_t^{IC},\ t = 1...N_t, \tag{2.4}$$

where $\mathbf{a}_t^H$ denotes the t-th row of $\mathbf{A}^{-1}$, and the bias term:

$$\mu_t = \mathbf{a}_t^H \mathbf{g}_t. \tag{2.5}$$

The SNR of MMSE-PIC is calculated as:

$$\rho_t = \frac{1}{\hat{\sigma}_t^2} = \frac{\mu_t}{1 - \sigma_t^2 \times \mu_t},\ t = 1...N_t. \tag{2.6}$$

The outputs of the detector are the soft symbols $\hat{s}_t$ and their variances $\hat{\sigma}_t^2$, $t = 1...N_t$. In an IDD system, the detector output symbols and variances are converted to LLRs for FEC decoding.

2.1.2 Extended Min-Sum (EMS) Nonbinary LDPC Decoding

An $n \times m$ regular-(d_v, d_c) NBLDPC code over GF(q) can be depicted in a bipartite graph that consists of n VNs and m CNs. Each VN connected to d_c CNs and each CN connected to d_v VNs. $\mathcal{N}(j)$ is the set of CNs connected to VN j, and $\mathcal{M}(i)$ is the set of VNs connected to CN i. The weight of the connection between VN j and CN i is $\alpha_{j,i}$, $\alpha_{j,i} \in \text{GF}(q)$.

An NBLDPC code is decoded by passing messages between VNs and CNs. A message passed between a VN and a CN is a vector of 2^q LLRs, called an LLR vector (LLRV), containing one LLR per GF(2^q) symbol. The EMS decoding algorithm keeps only n_m, $n_m \ll 2^q$, most reliable LLRs, and uses a GF index vector (GFIV) to keep track of the n_m symbols. The LLRs in an LLRV are sorted and normalized: the LLR value of the most reliable GF symbol is set to 0 and the remaining LLR values are normalized to it. The steps of EMS decoding is described below. In iterative decoding, the first step (initialization) is done only in the first iteration.

Step 1. Initialization: each VN j withprior LLRV L_j.

Step 2. VN to CN propagation: Each VN sends a V2C message to each of the d_v connected CNs. The message from VN j to CN i is denoted $\mathbf{u}_{j,i}$, $i \in \mathcal{N}(j)$. The GFIV of each message is GF multiplied by $\alpha_{j,i}$, a permutation operation.

Step 3. CN processing: Each CN receives d_c V2C messages and computes C2V messages using (2.7). The message from CN i to VN j is denoted $\mathbf{v}_{i,j}$.

$$\mathbf{v}_{i,j} = \sum_{k \in \mathcal{M}(i), k \neq j} \mathbf{u}_{k,i}, \tag{2.7}$$

where $\mathcal{M}(i)$ is the set of VNs connected to CN i, and Σ is performed using the forward-backward algorithm [25].

Step 4. CN to VN propagation: Each CN sends C2V messages to the d_c connected VNs. The GFIV of each C2V message is GF divided by $\alpha_{j,i}$, an inverse permutation.

Step 5. VN processing: Each VN receives d_v C2V messages and computes V2C messages

$\mathbf{u}_{j,i}$ and posterior LLRVs $\mathbf{z}_j$ using (2.8).

$$\begin{aligned} \mathbf{u}_{j,i} &= \mathbf{L}_j + \sum_{k \in \mathcal{N}(j), k \neq i} \mathbf{v}_{k,j}, \\ \mathbf{z}_j &= \mathbf{L}_j + \sum_{k \in \mathcal{N}(j)} \mathbf{v}_{k,j}, \end{aligned} \tag{2.8}$$

where $\mathcal{N}(j)$ is the set of CNs connected to VN j, and Σ is performed using elementary step [22].

In an IDD system, the decoder output LLRs are converted to soft symbols s and variances σ^2 for detection.

2.2 Detector-Decoder Interface and Optimization

The MMSE detector processes soft constellation symbols, while an NBLDPC decoder processes LLRs of GF symbols. Translations between soft symbols and LLRVs are required to implement an IDD system. Without loss of generality, we assume a 2^b-QAM constellation that is widely used in wireless communication systems. In this work, We propose to map the n-bit QAM symbol to a GF(2^b) symbol to enable the direct and simplified translations between soft symbols and LLRVs without any information loss. Matching the constellation and GF size provides the highest performance [26].

2.2.1 Converting Soft Symbols to LLRV

Standard Conversion. A soft symbol $\hat{s}$ in a 2^b-QAM constellation is first converted to bit LLR one by one for $i = 1, 2, ..., b$, then combine them into LLRV. This bit-by-bit conversion is performed in the following 2 steps:

Step 1. Convert $\hat{s}$ to n bit-LLRs [27] as:

$$L_i = \ln \frac{P(b_i = 1 \mid \hat{s})}{P(b_i = 0 \mid \hat{s})} = \frac{1}{\sigma^2} \left(\min_{a \in S_i^1} |\hat{s} - a|^2 - \min_{a \in S_i^0} |\hat{s} - a|^2 \right), \tag{2.9}$$

where b_i refers to bit i of the n-bit symbol, and S_i^1 and S_i^0 refer to the set of constellation points with their bit i being 1 and 0, respectively. The symbol LLRs in LLRV used in NBLDPC is calculated by adding the bit LLRs together as in (2.10) [24].

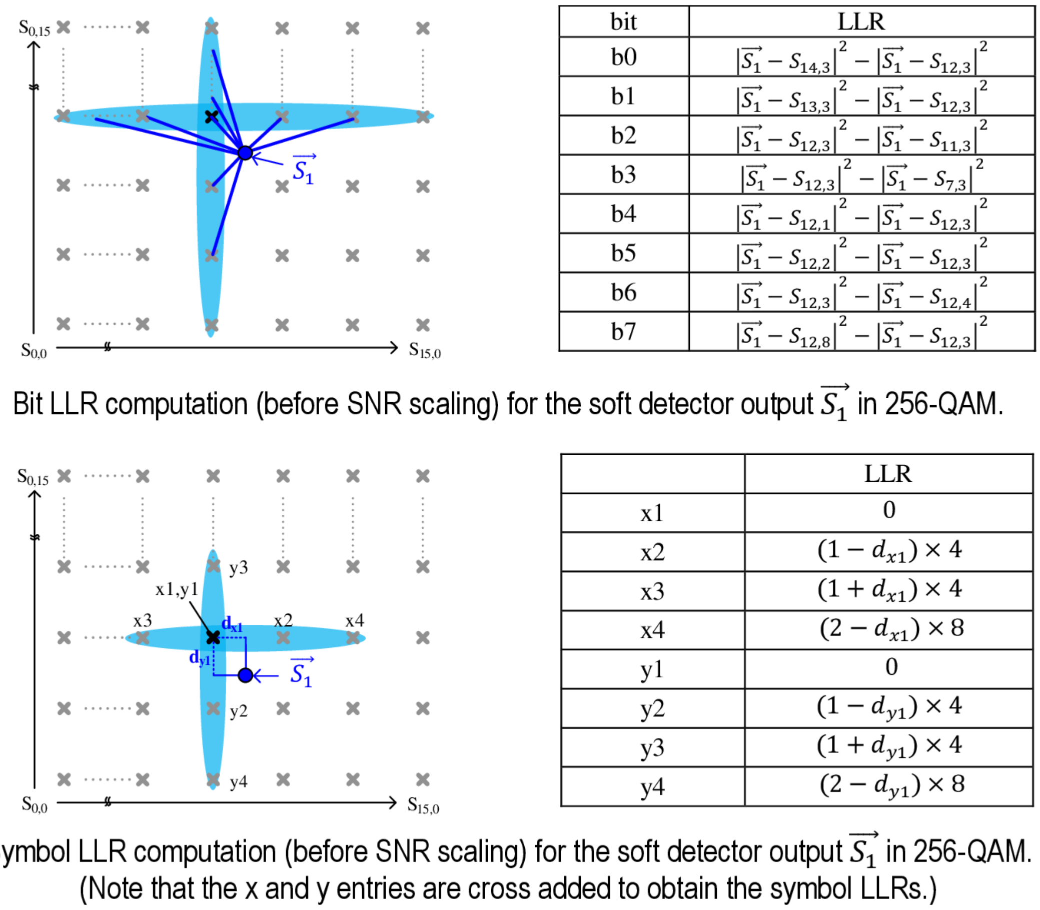
Step 2. Assemble the bit-LLRs to symbol-LLRs [24] from n_m nearest neighbors, and construct LLRV:

$$L_K = \ln \left(\prod_{i=1}^{b} \frac{P(K_i = 1 \mid \hat{s})}{P(K_i = 0 \mid \hat{s})} \right) = \sum_{i=1}^{b} (2K_i - 1) L_i, K_i \in \{0, 1\}, \tag{2.10}$$

where K is a GF(2^b) symbol, and K_i is the i-th bit of K's binary representation.

This bit-by-bit conversion requires searching of constellation points to find the nearest neighbors of $\hat{s}$ for each bit. In a high-order constellation, such as 256-QAM, the search and Eucleadian distance calculation can be expensive. The search can be done on the real and imaginary axis of a 2^b-QAM constellation to narrow the search space from 2^b candidates to $2^{b/2}$ candidates, as highlighted in Fig. 2.2(a). The computational complexity of the bit-by-bit conversion is listed in Table 2.1. Note that the search and bit LLR computation in the table can be further simplified if a certain mapping system is used, e.g. Gray mapping [28].

Even with all these simplifications, converting a soft symbol to b bit LLRs requires $2b$ multiply-adds (MACs), and converting b bit LLRs to a symbol LLR requires n adds. An LLRV for the truncated EMS decoding consists of n_m symbol LLRs, so bn_m adds are required to assemble an LLRV. In the end, the n_m symbol LLRs in the LLRV are sorted and normalized. The computational complexity of the bit-by-bit conversion is



bit	LLR
b0	$\lvert\vec{S_1} - S_{14,3}\rvert^2 - \lvert\vec{S_1} - S_{12,3}\rvert^2$
b1	$\lvert\vec{S_1} - S_{13,3}\rvert^2 - \lvert\vec{S_1} - S_{12,3}\rvert^2$
b2	$\lvert\vec{S_1} - S_{12,3}\rvert^2 - \lvert\vec{S_1} - S_{11,3}\rvert^2$
b3	$\lvert\vec{S_1} - S_{12,3}\rvert^2 - \lvert\vec{S_1} - S_{7,3}\rvert^2$
b4	$\lvert\vec{S_1} - S_{12,1}\rvert^2 - \lvert\vec{S_1} - S_{12,3}\rvert^2$
b5	$\lvert\vec{S_1} - S_{12,2}\rvert^2 - \lvert\vec{S_1} - S_{12,3}\rvert^2$
b6	$\lvert\vec{S_1} - S_{12,3}\rvert^2 - \lvert\vec{S_1} - S_{12,4}\rvert^2$
b7	$\lvert\vec{S_1} - S_{12,8}\rvert^2 - \lvert\vec{S_1} - S_{12,3}\rvert^2$

Bit LLR computation (before SNR scaling) for the soft detector output $\vec{S_1}$ in 256-QAM.

	LLR
x1	0
x2	$(1 - d_{x1}) \times 4$
x3	$(1 + d_{x1}) \times 4$
x4	$(2 - d_{x1}) \times 8$
y1	0
y2	$(1 - d_{y1}) \times 4$
y3	$(1 + d_{y1}) \times 4$
y4	$(2 - d_{y1}) \times 8$

Symbol LLR computation (before SNR scaling) for the soft detector output $\vec{S_1}$ in 256-QAM.
(Note that the x and y entries are cross added to obtain the symbol LLRs.)

Figure 2.2: SISO Detector input soft symbols and variances in (a) binary scheme, and (b) proposed nonbinary scheme.

Table 2.1: Comparsion of Bit-by-Bit Conversion and Direct Conversion from Soft Symbol to LLRV

	Standard Conversion	Direct Conversion
Soft symbol to LLRs	$2b + 2$ adds, $b + 2$ mults bit-LLRs)	$2\lceil\sqrt{n_m}\rceil$ adds (symbol-LLRs)
LLRV construct	$b \times n_m$ adds	n_m adds
Total	$b + 2$ mults, $b(n_m + 2) + 2$ adds	$2\lceil\sqrt{n_m}\rceil + n_m$ adds

summarized in Table 2.1.

Direct Conversion. Instead of the bit-by-bit conversion, we propose a new direct conversion of a soft symbol to an n_m-element LLRV for NBLDPC decoding. First, we take advantage of the fact that LLRV is sorted and normalized to the most reliable symbol, denoted T in Fig. 2.2(b), to calculate normalized symbol LLR directly using (2.11).

$$L_K = \ln \frac{P(s = K \mid \hat{s})}{P(s = T \mid \hat{s})} = \frac{1}{\sigma^2} \left(|\hat{s} - K|^2 - |\hat{s} - T|^2 \right), \tag{2.11}$$

where T is the symbol is the most reliable symbol and it represents the constellation point, that is closest to the soft symbol $\hat{s}$. Note that based on (2.11), $L_T = 0$, a result of normalization. For a QAM constellation, (2.11) can be computed by taking the real and imaginary part independently, which are then sumed together.

As an example shown in Fig. 2.2(b), the distance between the soft symbol $\hat{s}$ and its nearest constellation point T of coordinate (x_1, y_1) is d. The projection of d on the real axis (x-axis) and imaginary axis (y-axis) are d_x and d_y. Without loss of generality, assume the constellation points are spaced by 2. It follows that the distance from $\hat{s}$ to its second nearest constellation point along the x-axis, x_2, is $2 - d_x$ and to the second nearest constellation point along the y-axis, y_2, is $2 - d_y$. The real and imaginary part of the LLR can be computed as follows.

$$\begin{aligned} L_{x2} &= \frac{1}{\sigma^2} \left(|2 - d_x|^2 - |d_x|^2 \right) = \frac{4}{\sigma^2}(1 - d_x) \\ L_{y2} &= \frac{1}{\sigma^2} \left(|2 - d_y|^2 - |d_y|^2 \right) = \frac{4}{\sigma^2}(1 - d_y). \end{aligned} \tag{2.12}$$

Notice that the square terms are canceled out, and the calculation requires only L1 distance without multipliers. The technique is illustrated in Fig. 2.2(b). The calculation of (2.12) can be further simplified using bit invert, shift and add.

In this work, we adopt the trucated EMS decoding of NBLDPC code using n_m the most reliable symbols. As such, the conversion requires the distances from the soft symbol to approximately the $\lceil \sqrt{n_m} \rceil$ nearest constellation points along x-axis and y-axis. The n_m symbol LLRs are obtained by cross-adding the distances to the nearest neighbors along x-axis and y-axis, and no additional sorting stage is needed.

The computational complexity of the direct conversion method is summarized in Table 2.1 for comparison with the conventional bit-by-bit method.

For a large constellation the complexity of the direct conversion method is significantly lower than the bit-by-bit conversion method. For example, for QAM(256) ($b = 8$), GF(256) NBLDPC code, and decoding using $n_m = 16$, the bit-by-bit conversion requires 16 MACs, 128 adds, and 16 sorts; whereas the direct conversion requires only 24 adds.

2.2.2 Converting LLRV to Soft Symbols

Standard Conversion. The LLRV from NBLDPC decoder is converted to a soft symbol and its variance in 2 steps:

Step 1. Symbol LLR L_α is converted to the probability of constellation point $P(s)$:

$$P(s) = \frac{\exp(-0.5L_a)}{\sum_{a\in \mathrm{GF}(2^b)} \exp(-0.5L_a)}, \tag{2.13}$$

where the GF(2^n) symbol α is mapped to the matching constellation symbol s, $s \in 2^b$-QAM. This is often implemented using table lookup.

Step 2. The soft symbol is computed as the weighted sum of constellation points:

$$\hat{s} = \sum_{a\in 2^b\text{-QAM}} aP(a), \tag{2.14}$$

$$\sigma^2 = \sum_{a\in 2^b\text{-QAM}} (a-\hat{s})^2 P(a).$$

The standard conversion requires 2^b table lookups in step 1 and $2^b + 2^{b+1}$ MACs in step 2. The complexity scales quadratically with the constellation size b, and the conversion complexity becomes prohibitive for high-order constellation, such as QAM(256).

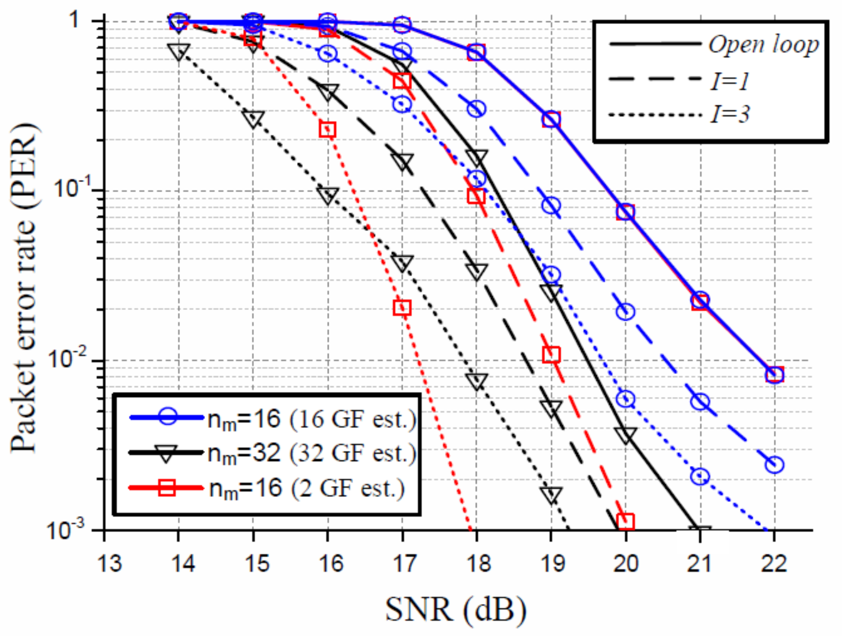


Figure 2.3: PER comparison among three setups: (1) 16 GF elements for decoding and 16 GF elements for soft symbol estimation, (2) 32 GF elements for decoding and 32 GF elements soft symbol estimation, and (3) 16 GF elements for decoding and only top 2 GF elements for soft symbol estimation.

Using the EMS decoding of NBLDPC code, an LLRV consists of n_m most likely symbol-LLRs. The complexity of the conversion can be reduced by only account for the probablities of the n_m nearest constellation points using equations similar to (2.13) and (2.14).

Approximate Conversion. To further simplify the conversion, we apply an approximation by choosing only the top two most likely symbol-LLRs. By removing the less likely symbols that tend to be less accurate due to numerical saturation in NBLDPC decoding and the likely duplication of symbols in the decoder output, the simplification can even improve the error-rate performance as shown in Fig. 2.3.

Suppose the two most likely symbols are s_0 and s_1. Step 1 of the conversion is reduced to the following.

$$P_{s_0} = \frac{\exp(0.5L_{s_0})}{\exp(0.5L_{s_0}) + \exp(-0.5L_{s_0})}, \tag{2.15}$$
$$P_{s_1} = 1 - P_{s_0} \approx \neg P_{s_0}.$$

The underlying assumption of the approximation is that the top two constellation symbols dominate. Thus the probability of the second most likely symbol s_1 is $P_{s_1} = 1 - P_{s_0}$. Since $P_{s_1} \leq 0.5$, $P_{s_1} \approx \neg P_{s_1}$, where $\neg$ is the bit-wise inversion. With only two nearest constellation points to consider, the soft symbol and variance calculation is reduced to (2.16).

$$\begin{aligned} \hat{s} &= P_{s_0}s_0 + P_{s_1}s_1, \\ \sigma^2 &= P_{s_0}(s_0 - \hat{s})^2 + P_{s_1}(s_1 - \hat{s})^2, \\ &= P_{s_0}P_{s_1}(s_0 - s_1)^2. \end{aligned} \tag{2.16}$$

The approximate direction conversion underestimates the variance, especially when SNR is low. Therefore an offset term is added as correction.

$$\sigma^2 = P_{s_0}P_{s_1}(s_0 - s_1)^2 + P_{s_1}q, \tag{2.17}$$

where q is an offset term that depends on the size of the constellation.

Table 2.2: Comparsion of Bit-by-Bit Conversion and Approximate Direct Conversion from LLRV to Soft Symbol

	Standard Conversion	Approximate Conversion
LLR → prob.	n_m table lookups	1 table lookup, 1 add
Soft symbol & variance compute	$3n_m$ mults, $2n_m$ adds	5 mults, 2 adds
Total	n_m table lookups, $3n_m$ mults, $2n_m$ adds	1 table lookup, 5 mults, 3 adds

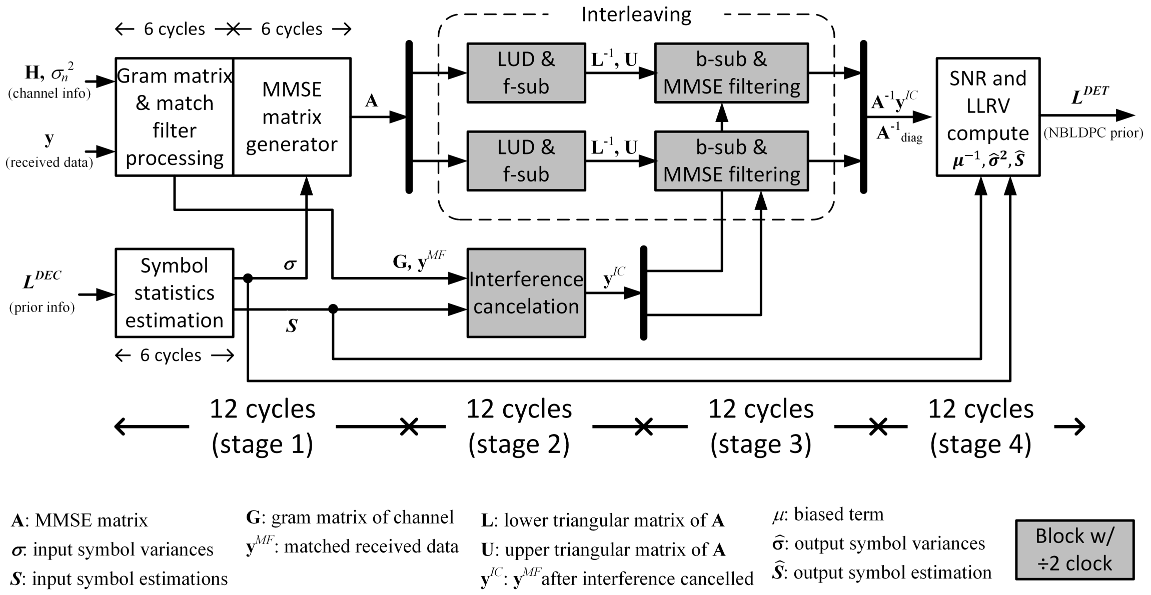


Figure 2.4: Design of the MMSE detector in 4 task-based coarse pipeline stages. Stage 2 and 3 operate at a 2× slower clock frequency, and the remaining stages operate at the base clock frequency.

The complexity of the approximate conversion is fixed regardless of the constellation size or choice of n_m as shown in Table 2.2. Using the approximate conversion, a GF(256) LLRV to a 256-QAM soft symbol and variance conversion requires only 1 table lookup, 5 multiplies and 3 adds, a significant simplification over the conventional method that requires 16 table lookups ($n_m = 16$), 48 multiplies, and 32 adds.

2.3 SISO MMSE-PIC Detector

The MMSE detector design is comprised of 4 coarse pipeline stages as depicted in Fig. 2.4. Channel information and prior input symbol LLRs from the decoder are processed in the first stage to generate the MMSE matrix **A**. The matrix is then inverted using LU decomposition (LUD) for MMSE filtering in the second and third stage, while interference cancellation is done in parallel. The final stage computes SNR and symbol LLRs as the input to the NBLDPC decoder.

The LUD in the second stage contains the critical paths and requires a long latency, causing unbalanced pipeline stages and a throughput bottleneck. As the Newton-Raphson reciprocal unit dictates the inner loop of the LUD, we reformulate the reciprocal computation in a parallel structure to shorten the second stage from 18 cycles [1, 29] to 12 cycles.

To loosen the timing constraint on the long critical paths on the complex multipliers in the second and third stage, we create a 2× slow clock domain for the two stages to allow the gates to be downsized, and recoup the throughput by interleaving between two copies of the datapaths without stalling the pipeline. After gate downsizing, the duplication costs only 24 % additional area over the baseline, but the throughput is increased by 38 %.

In the final stage, we use an algorithmic property to simplify the SNR computation from 4 complex multiplications and additions to 2 real multiplications and 1 addition of shorter bit widths, reducing the area of the final stage by 50 % and power by 46 %. The final MMSE detector in 65nm CMOS achieves a throughput of 1.38 Gb/s.

2.3.1 Tandem Scheduling

Matrix inversion is done in three substeps: LUD, forward substitution (f-sub) and backward substitution (b-sub). A $N_t \times N_t$ matrix $\mathbf{A}$ is first decomposed to a lower and an upper triangular matrix, $\mathbf{L}$ and $\mathbf{U}$, using LUD. $\mathbf{L}^{-1}$ is then found by solving for $\mathbf{X}$ in $\mathbf{LX} = \mathbf{I}$ using f-sub. Finally, $\mathbf{A}^{-1}$ is found by solving for $\mathbf{Y}$ in $\mathbf{UY} = \mathbf{L}^{-1}$ using b-sub.

LUD follows Gaussian elimination which operates from the top row to the bottom row of matrix $\mathbf{A}$, obtaining $\mathbf{L}$ from the left to the right column, and $\mathbf{U}$ from the top to the bottom row. In each step, LUD uses a reciprocal unit to compute the inverse of the diagonal element of $\mathbf{U}$. Assume $\mathbf{A}$ is 4×4 and suppose the reciprocal unit takes n_r cycles, a multiply-add (MAC) takes 1 cycle, and 16 parallel real-valued MACs are allocated, the critical path of LUD can be packed in $4n_r + 6$ cycles. Under this critical

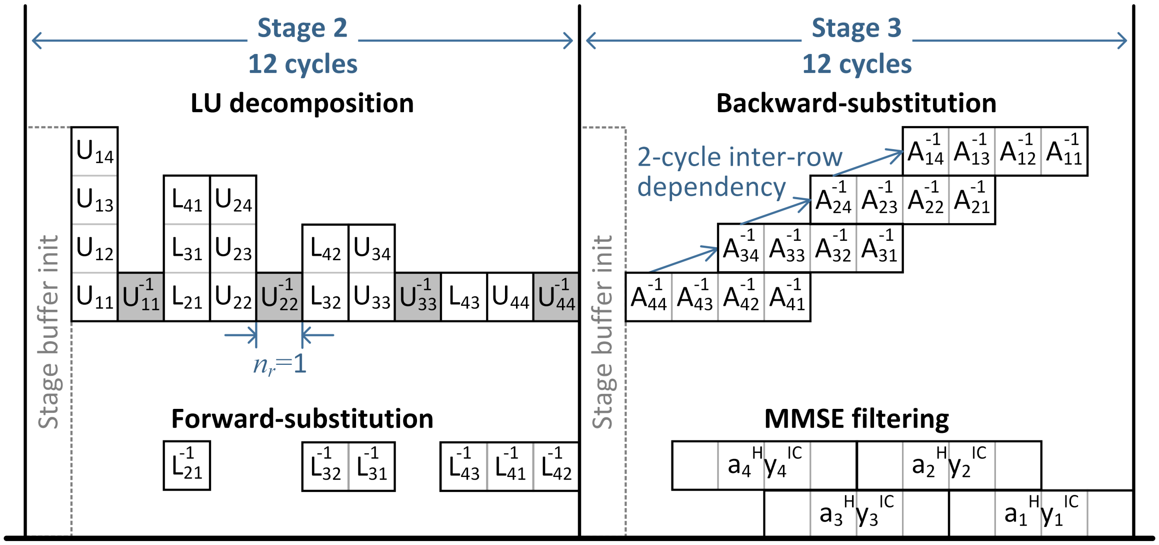


Figure 2.5: Tandem scheduling of matrix Inversion and MMSE filtering.

path, f-sub can be performed in tandem with LUD to hide its latency. As soon as the first element of $\mathbf{L}$ is available, f-sub can start. In this way, f-sub and LUD complete at the same time, as shown in Fig. 2.5. Once the last row of $\mathbf{L}^{-1}$ is found and buffered, b-sub starts from the bottom row to the top row of $\mathbf{L}^{-1}$ to compute $\mathbf{A}^{-1}$.

MMSE filtering is done by vector inner products: $\mathbf{a}_t^H\mathbf{y}_t^{IC}$, $t = 1 \ldots N_t$. Recall that $\mathbf{a}_t^H$ denotes the t-th row of $\mathbf{A}^{-1}$ and $\mathbf{y}_t^{IC}$ is the output of the interference cancellation step. We propose the tandem scheduling of b-sub and MMSE filtering. As soon as an element of $\mathbf{A}^{-1}$ is available, the corresponding product with $\mathbf{y}_t^{IC}$ can be performed. In this way, MMSE completes in only 1 cycle after f-sub is done.

With tandem scheduling, the matrix inversion and MMSE filtering is reduced from 3 coarse pipeline stages [1] to 2 stages, as shown in Fig. 2.5. Tandem scheduling also cuts the number of boundary registers between stages by 85 %, as the output from the previous step is immediately consumed by the subsequent step.

2.3.2 Dual-Lookup Reciprocal Unit

Reciprocal is in the critical path of matrix inversion and dominates the latency. A popular reciprocal design is based on the Newton-Raphson division algorithm.

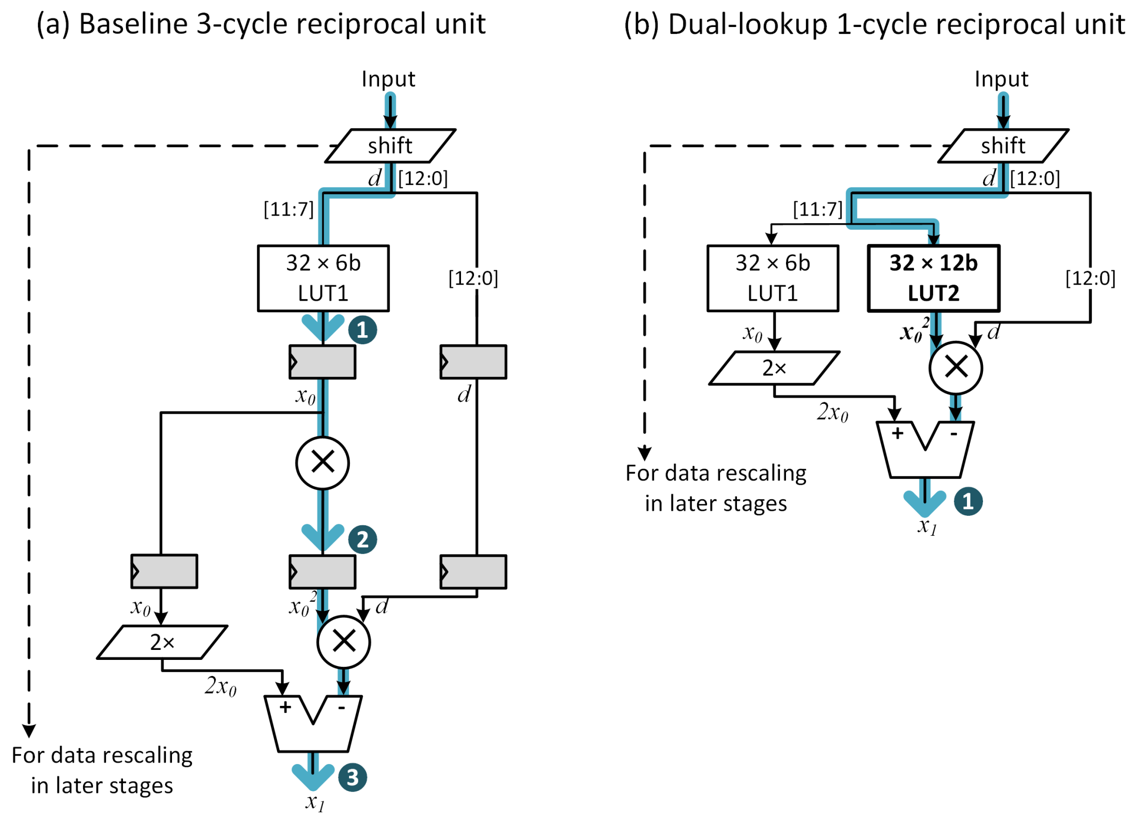


Figure 2.6: Reciprocal unit design: (a) baseline 3-cycle design from [1] and (b) dual-lookup single-cycle design.

Suppose we need to find $x = \frac{1}{d}$, the problem can be formulated as finding the root of $f(x) = \frac{1}{x} - d = 0$. Applying the Newton-Raphson method, the root can be found by iteration with an initial estimate x_0:

$$x_{i+1} = x_i - f(x_i)/f'(x_i) = 2x_i - dx_i^2. \tag{2.18}$$

A baseline reciprocal unit is shown in Fig. 2.6(a) [1]. The initial estimate x_0 is retrieved from a lookup table (LUT). To reduce the LUT size, only the MSB bits are used to address the LUT. Two multiplies are needed to compute dx_i^2, which is then subtracted from $2x_i$ to compute the reciprocal. A better approximation can be obtained by iterations. For a MMSE detector, it was shown that a 32×6b LUT and one iteration (x_1) are sufficient [1]. The baseline design is naturally divided into 3 pipeline stages, costing 3 cycles.

In the baseline design, the latency of the reciprocal unit is dominated by the two multiplies. To reduce latency, we design a dual-lookup reciprocal unit as shown in Fig. 2.6(b) using two LUTs, a 32×6b LUT for retrieving x_0 and a 32×12b LUT for obtaining x_0^2. The addition of the 32×12b LUT allows the redesigned reciprocal unit to have only 1 cycle delay, which translates to 6-cycle latency reduction of the matrix inversion stage. To make the best use of the limited LUT size, we apply dynamic scaling of the matrix $\mathbf{A}$ based on symbol variance, such that the input to the reciprocal unit falls in the range of $[1, 2)$. The designed reciprocal unit provides an average error of 0.00044 and a maximum error of 0.0017, which are sufficient for an MMSE detector [29].

2.3.3 Interleaving Architecture

The MMSE detector is pipelined to 4 stages as shown in Fig. 2.4. Initialization is in stage 1. The tandem scheduling of LUD and f-sub allow them to be grouped in stage 2, and the tandem scheduling of b-sub and MMSE filtering allows them to be grouped in stage 3. Due to the lack of data dependence, interference cancellation is also done in stage 2. Post-processing is done in stage 4.

Despite the optimizations done in stage 2 and 3 of the pipeline, the long critical paths in the multipliers present a tight timing constraint. To loosen the constraint, we use a simple clock divider to create a 2× slow clock domain for stage 2 and 3 to allow the gates to be downsized, and maintain the throughput across the two stages by duplicating the datapaths and interleaving between the two copies as shown in Fig. 2.4. After gate downsizing, the duplication costs only 24 % additional area over the baseline as depicted in Fig. 2.7, but the throughput is increased by 38 % thanks to a higher clock frequency. The downsized gates also reduce the load capacitance,

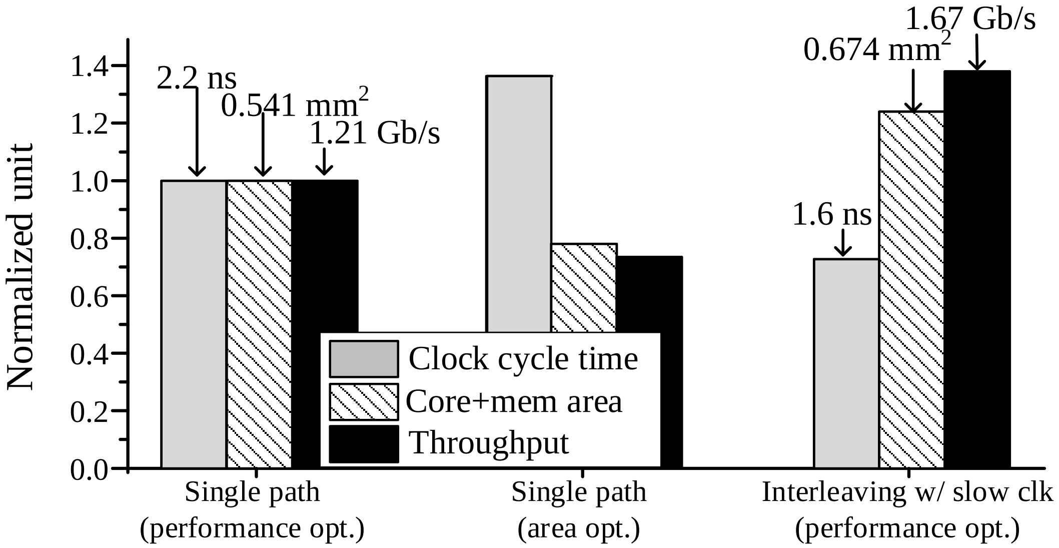


Figure 2.7: Comparison between single-path area optimization (middle) and dual-path interleaving with slow clock (right).

and thus improving the energy efficiency.

2.4 SISO Nonbinary LDPC Decoder

In this work, we implement an NBLDPC decoder for a (52, 26) regular-(2, 4) NBLDPC code over GF(256) with a binary block length of 416 bits. The NBLDPC decoder adopts the truncated extended min-sum (EMS) algorithm using the most $n_m = 12$ reliable GF elements over GF(256) to reduce complexity while still maintaining a good coding performance. In a conventional design [23], a CN has to be idle n_m clock cycles for the associated VNs to complete a GF vector and vice versa due to the serial nature of GF processing. This results in a latency and throughput bottleneck for the NBLDPC decoder design.

We apply a data forwarding technique to provide GF element output directly from VN to CN to eliminate pipeline stalls, and reallocate the memory in VN to enable the forwarding from CN to VN in a similar manner. This forwarding scheme not only shortens the decoding iteration from 36 cycles to 25 cycles, but also eliminates the CN-to-VN message storage. The proposed NBLDPC decoder in 65 nm CMOS achieves a short latency per decoding iteration of 81.4 ns, resulting a throughput of 1.02 Gb/s with 5 decoding iterations.

2.4.1 High-throughput Fully Parallel Architecture

To support a high throughput over 1 Gb/s the proposed decoder is fully parallelized with 52 variable nodes (VN) and 26 check nodes (CN), as shown in Fig. 2.8. The fully parallel architecture also simplifies the control and message scheduling. However due to the serial nature of GF processing in both VN and CN, the data dependencies cause inefficient pipeline stalls in the fully architecture [23]. In return, this inefficiency imposes large area, power and long decoding latency. Therefore we opt for optimized architecture and pipeline schedule to minimize overhead without degrading throughput. After the MMSE detection, the sorted LLRVs with $N_m = 12$ elements for each one of 52 variable nodes are load into the prior memory in the decoder. The decoder then initiates the truncated EMS decoding process for a given number of iterations. The VNs passes the V2C message to the associated CNs through a routing network where the GF symbols are permuted and the LLR values are normalized. The GF permutation is determined by the weights on the NBLDPC Tanner graph, therefore it can be implemented with simple logics. The LLR normalization subtracts the smallest LLR value from the LLR values of the V2C messages to reduce the dynamic range. The CN performs bubble-check algorithm [30] to generate C2V messages and feed the results back to the associated VNs. The VNs calculate the llrv of posterior and generate the extrinsic V2C messages for the next iteration. The decoding procedure stops when the maximal iteration is reached.

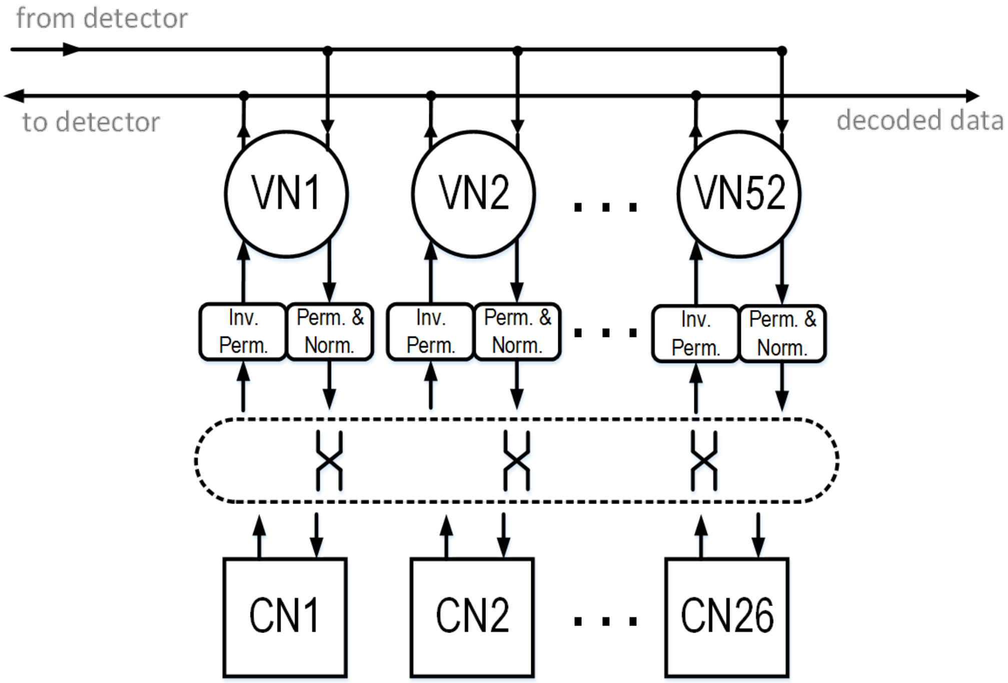


Figure 2.8: Fully parallel architecture of Nonbinary LDPC instantiating 52 variable nodes and 26 check nodes.

2.4.2 Skimmed Check Node with Data Forwarding

A Check node performs GF parity check of $d_c = 4$ input V2C messages and generates the output C2V feeding back to the associated variable nodes. The CN processing is done in an efficient forward-backward manner by using 6 elementary check nodes (ECNs) to perform bubble check algorithm and 6 memories to store the input V2C messages and the intermediate results. An ECN finds $n_m = 12$ most reliable GF-LLR results among all the parity combinations of two input sorted LLRVs. The ECN contains a GF/LLR adder, a candidate selector and an insertion sorter. For each cycle, the candidate selector picks up candidates from the memories and the resulting combination, i.e. GF/LLR summation is inserted into the sorter pushing the current most reliable result out of the sorter. According to the order in the sorters, the candidate selector is able to pick the next candidates. The ECN process takes $n_m = 12$ cycles to generate the $n_m = 12$ C2V messages sequentially. In conventional

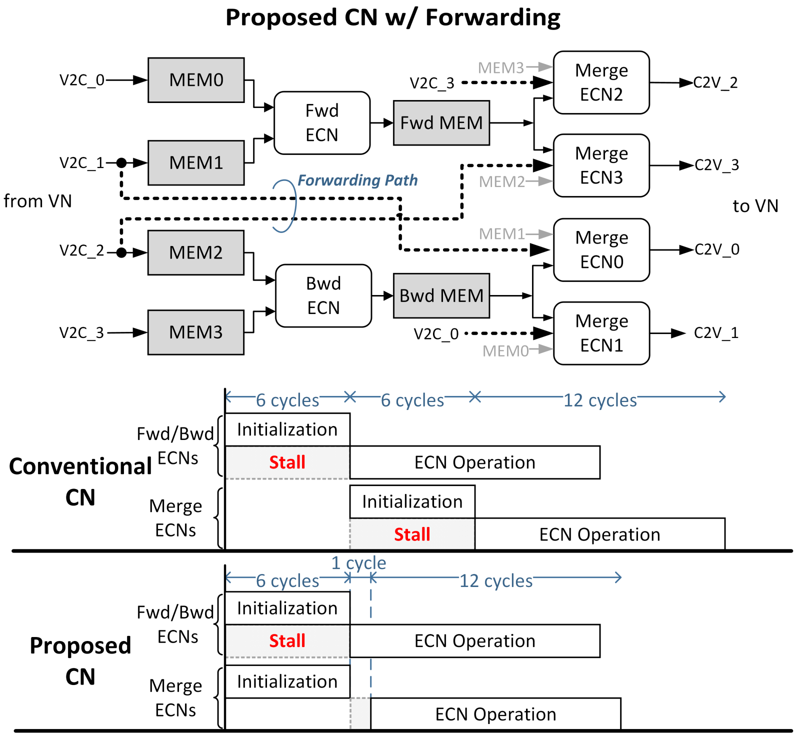


Figure 2.9: Proposed CN design with V2C forwarding; the pipeline schedules of conventional and proposed CN design.

CN design, the ECNs initialize the sorters by the data stored in the memories therefore they cannot start until the V2C elements or intermediate (forward/backward) results are loaded into the memories. This dependency causes $n_b = 6$ cycles of pipeline stalling, where n_b is the sorter size. To hide this dependency, we propose an input forwarding scheme to initialize the sorters as soon as the results are resolved as shown in Fig. 2.9. With the observation that the first(smallest) LLR is always normalized to zero, we are able to forward the data to eliminate the pipeline stalls without lengthening the critical path.

Another inefficiency of conventional design is that the sorter requires large storage to buffer the candidate combination information containing GF indexes, memory indexes and LLR values during the CN processing. We observe that the GF indexes are unused during sorting and candidate selection. Since the sorter is implemented with shift registers, the redundant GF indexes not only waste the storage but also waste switching power shifting all the unused data. This inefficiency is prominent when GF size or sorter size n_b is large. Therefore we skim the buffers in the sorter by eliminating its GF storage. With retiming of the ECN the output GF elements can be retrieved. By doing this, the buffer size of the sorter is reduced by 36% resulting 20 % area reduction and 12 % power reduction of CN based on synthesis results.

2.4.3 Variable Node with Data Reallocation and Forwarding

A variable node computes $d_c = 2$ V2C messages and the posterior (2.8) using the prior message and $d_c = 2$ input C2V messages. The conventional VN implementation in [23] is shown in Fig. 2.10. It requires one prior memory, two C2V Content Addressable Memories (CAMs), two V2C Elementary Variable Nodes (EVNs) and one posterior EVN. The prior memory is loaded with the prior LLRV generated from MMSE detector outputs in the initialization stage. Then the VN operation is carried out in the following three steps.

Step 1. C2V Storing:

At the beginning of VN processing, the two C2V LLRV messages are streamed and stored into two C2V CAMs where the LLR values are indexed by their GF symbols.

Step 2. GF Matching:

After C2V storing is completed, $n_m = 12$ GF symbols in the prior memory are sent to both of the C2V CAMs one by one. The C2V CAM outputs the LLR whose GF index matches with it's input GF symbol of prior. If there is no match, the CAM will outputs its largest LLR value, i.e. $C2V[12]^{(LLR)}$. The CAM output then is added with corresponding prior LLR and buffered in the parallel sorter in the EVNs.

Step 3. V2C Processing:

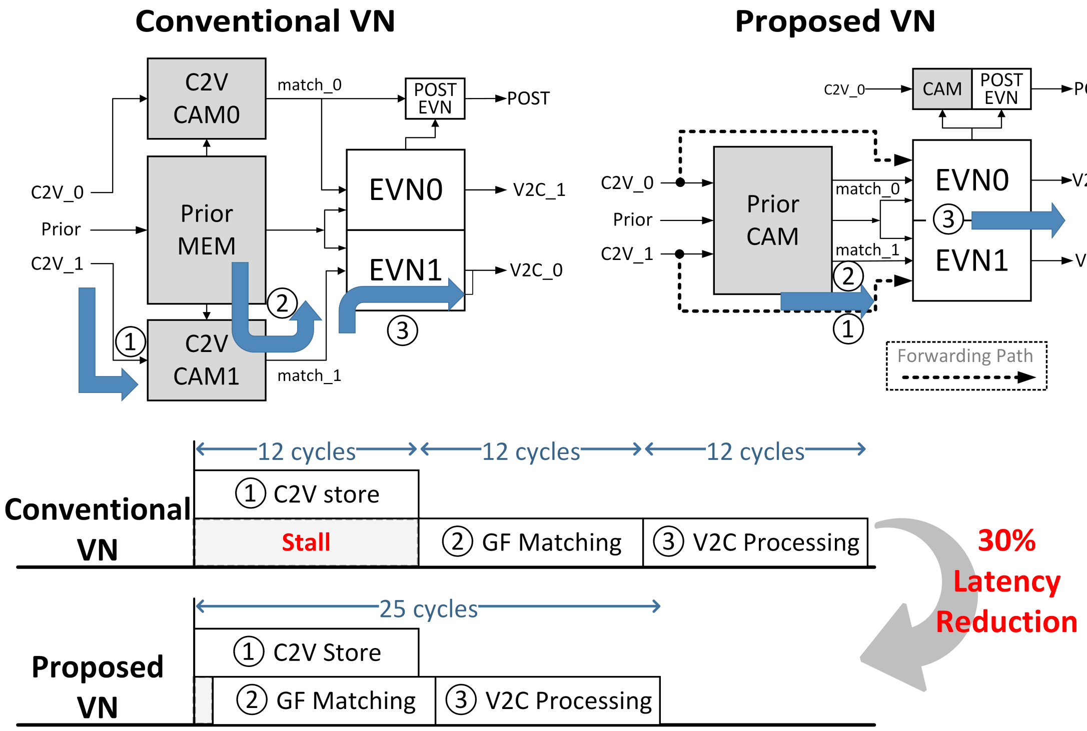


Figure 2.10: Data flow and operation latency of the conventional and proposed VN designs.

In the third step, each LLR of the C2V is added with the largest LLR value of the prior, i.e. $prior[12]^{(LLR)}$, and the result is inserted into the sorter in the EVN. The sorter outputs the final V2C LLRV message containing the first $N_m = 12$ most reliable GF-LLR pairs. The V2C message is then fed back to the associated CNs.

In the conventional design, the VN cannot start to process until the C2V messages are completely loaded into the two CAMs, introducing $n_m = 12$ cycles of pipeline stalling. To reduce the long latency of VN processing, we reallocate the C2V and prior storage allowing the data forwarding from CN to VN, as shown in Fig. 2.10 . This leverages the fact that the prior is updated only at the beginning of the decoding while the C2V is updated for every decoding iteration. Instead of using prior to look up a match in the C2V CAMs, we use the incoming C2V to look up a match in the prior CAM. By doing so, only one prior CAM is required and the matching operation can be done in the fly as C2V message streaming into the VN. Since the two C2V CAMs are eliminated, not only the $n_m = 12$ cycles of stalling for V2C processing are eliminated but also the area and power associated with the C2V CAMs are saved. As for the posterior, we are able to use a small CAM and a simplified EVN since only the first two GF-LLR pairs are required to be fed back to the SISO MMSE detector.

2.5 Low Power Techniques–Clock Gating Exploiting Regular Memory Access

A total of 70.9 kb registers are used for buffering data in and between stages of the detector and the decoder. Registers are used in place of memory arrays to support high access bandwidth and the flexibility of placing small memory blocks. Registers are power hungry, but we recognize a power reduction opportunity as most of the registers used in our design are infrequently updated due to the coarse pipelining, e.g., 1 update every 12 cycles for the 7.6 kb stage boundary registers in the detector, and 1 update every 25 cycles for the 26.2kb CN buffer registers in the decoder. We exploit the access pattern to reduce power by enabling clock gating of the registers when they are idling, saving the detector power and the decoder power by 53 % and 61 %, respectively.

To lower the power consumption, automatic clock gating is applied to stage boundary and buffer registers to save 53 % power of the detector. A total of 9.1 kb registers are used for buffering data in and between stages of the detector. Registers are used in place of memory arrays to support high access bandwidth and the flexibility of placing small memory blocks. Registers are power hungry, but we recognize a power reduction opportunity as most of the registers used in our design are infrequently updated due to the coarse pipelining, e.g., 1 update every 12 cycles for the 7.6 kb stage

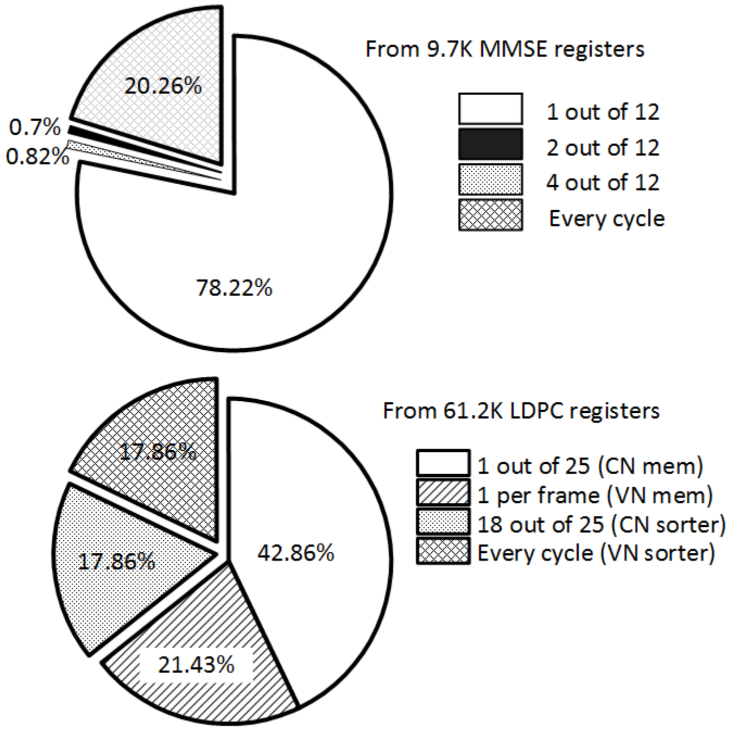


Figure 2.11: Power breakdown and the activities of registers.

boundary registers in the detector. We exploit the access pattern to reduce power by enabling clock gating of the registers when they are idling, saving 53 % power in total.

2.6 Chip Measurement Results

The fabricated test chip is fully functional. The die photo is shown in Fig. 2.12. The MMSE detector core and the NBLDPC decoder core occupy 0.7 mm^2 and 1.7 mm^2, respectively. At room temperature and 1.0 V supply, the MMSE detector runs at a maximum frequency of 517 MHz for a throughput of 1.38 Gb/s, the highest reported throughput of a SISO MMSE detector [1]. At room temperature and 1.0 V supply, the decoder runs at 307 MHz for a throughput of 1.02 Gb/s (5 iterations). The NBLDPC decoder consumes 20.1 pJ/b/iteration, the lowest reported energy of an NBLDPC decoder [23], and it matches the efficiency of the binary LDPC decoder used in IDD

[18]. Although our NBLDPC code is about half the size of [23], the one order of magnitude improvement in energy [23] is significant.

The energy efficiency can be further improved by voltage and frequency scaling, as shown in Fig. 2.13. At 500 mV supply, the MMSE detector and the NBLDPC decoder consume 9.7 pJ/b and 6.9pJ/b/iteration, respectively, for throughputs above 200 Mb/s.

Fig. 2.14 shows the bit error rate and frame error rate curves of proposed IDD system. The testing channel model is a 4×4 TGn Type C channel. As more IDD iterations are done, the error rates are improved more than 3dB from open loop ($I = 0$) to 3 IDD iterations ($I = 3$). The performance gain is traded off by the latency of receiver processing.

Our work is compared with state-of-the-art MIMO detector and decoder designs in Table 2.3.The MMSE detector consumes 19.2 pJ/b, an order of magnitude lower than previous SISO detector designs [17, 18, 1], demonstrating the advantage of the optimized MMSE detection for IDD.

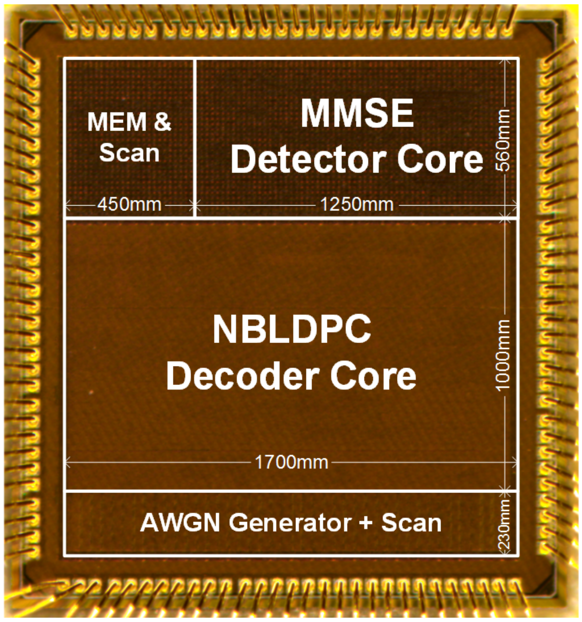


Figure 2.12: Chip die photo.

Table 2.3: Comparison Table of State-of-the-Art MIMO Detectors and LDPC Decoders

Detector	Noethen [17]	Borlenghi [18]	Winter [19]	Studer [1]	This work
IDD design	Yes	Yes	No	yes	yes
Algorithm	SD SISO	SD SISO	SD SO	MMSE SISO	MMSE NB-SISO
MIMO system	≤4 × 4	≤4 × 4	≤4 × 4	4 × 4	4 × 4
Modulation	≤64	≤64	≤64	≤64	256
Technology [nm]	65	65	65	90	65
Core area [mm^2]	-	2.78	0.31	1.5	0.7
Preprocessing area [kGE]	383[a]	-[b]	-[b]	410	347[c]
Detection area [kGE]		872	215		
Frequency [MHz]	445	135	333	568	517
Power [mW]	87	-	38	189	26.5
Throughput [Mb/s]	396	194	296-807	757	1379
Area efficiency [Mb/s/kGE]	1.03	0.22	1.37-3.75	1.85	3.68
Energy efficiency [pJ/b]	220	920	48	250	19.2

Decoder	Noethen [17]	Borlenghi [18]	Winter [19]	Park [23]	This work	
IDD design	yes	yes	no	no	yes	
Code	LDPC	LDPC	LDPC	NBLDPC GF(64)	NBLDPC GF(256)	
Block length	768	1944	768	960	416	
Technology [nm]	65	65	65	65	65	
Core area [mm^2]	-	0.78	3.6	7.04	1.7	
Decoding area [kGE]	-	-	-	2780	935	
Frequency [MHz]	500	299	267	700	307	
Power [mW]	-	-	367	3866	103	
Iterations	10	10	10	10 to 30[d]	5	10
Throughput [Mb/s]	155	586	235.2	1150	1024	512
Area efficiency [Mb/s/mm^2]	100.92	751	65.33	163	602	301
Energy efficiency [pJ/b/iter]	232	21	170	277	20.1	

[a] : memory for data exchange is included.

[b] : data pre-processing block (QRD) is not included.

[c] : total area is 264 kGE if no interleaving processing.

[d] : with early termination.

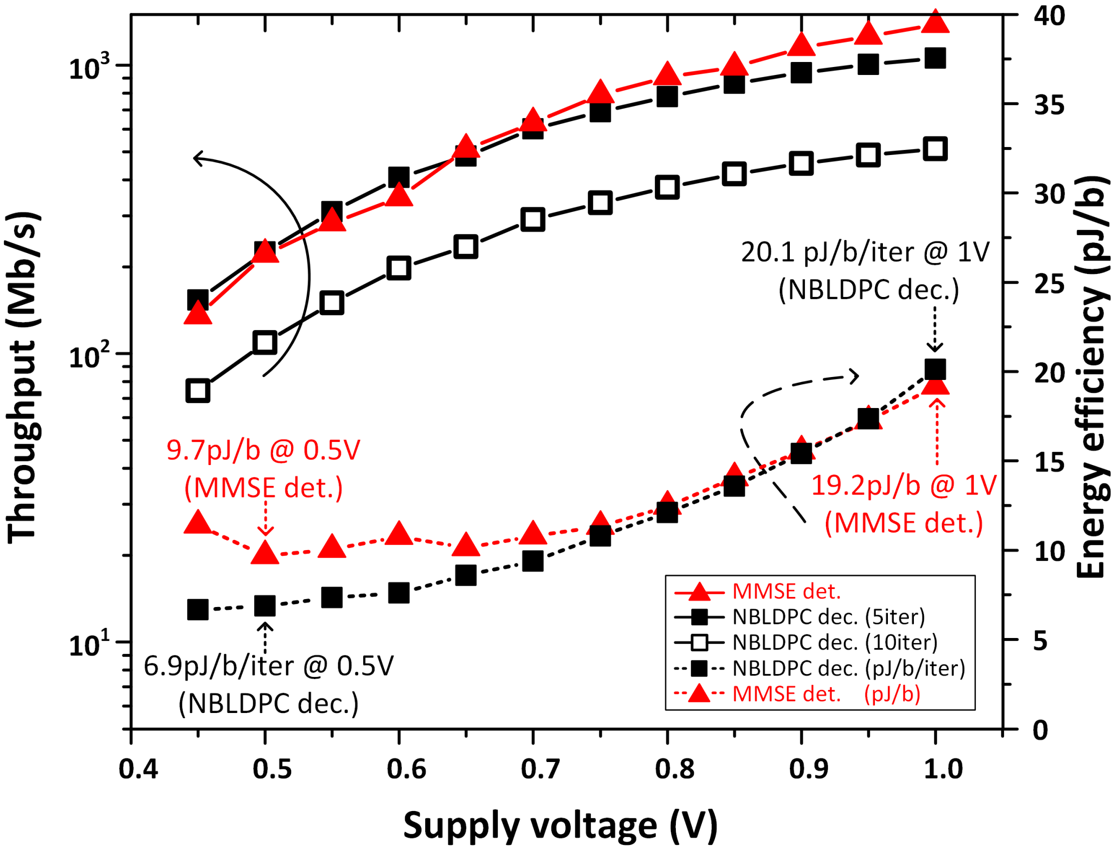


Figure 2.13: Measured throughput and energy efficiency with voltage scaling.

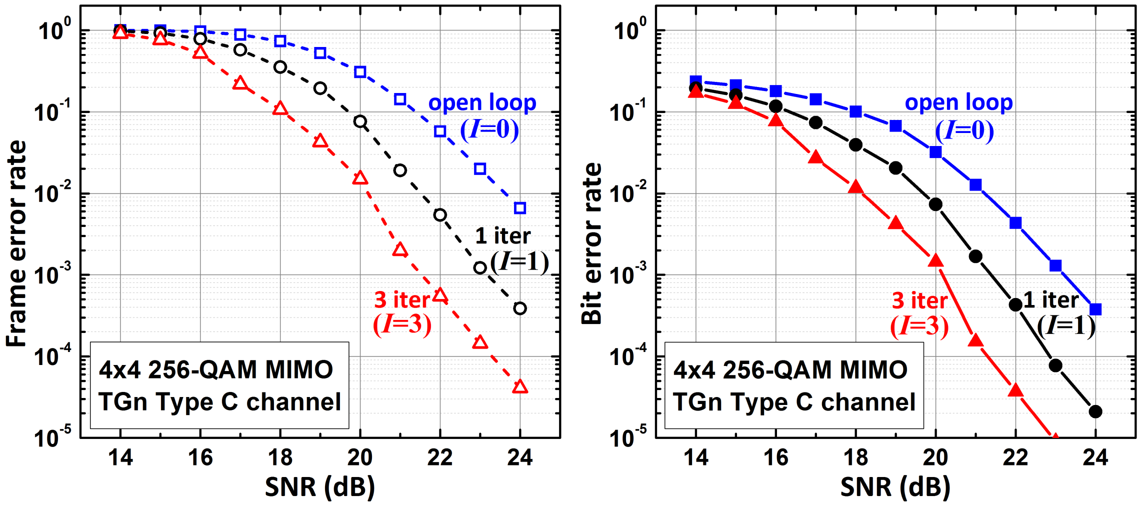


Figure 2.14: The bit error rate and frame error rate versus SNR of the proposed MMSE-NBLDPC IDD system.

2.7 Summary

We demonstrate an MMSE-NBLDPC iterative detector-decoder for a 4 × 4 256-QAM MIMO system to achieve an excellent error rate that improves with iterations.

To minimize latency over the iterative loop and improve throughput, the MMSE detector is divided into 4 task-based coarse pipeline stages so that all stages can operate in parallel. Both the number of stages and the stage latency of the detector are minimized, and the long critical paths are interleaved and placed in a slow clock domain to support a high data rate in a cost-effective way. The resulting MMSE detector achieves an 82 % higher throughput, and almost 3.5 times the throughput of the latest SD detector.

The NBLDPC decoder is implemented using 78 processing nodes to enable fully parallel message passing. Serial Galois field (GF) processing is pipelined using a data forwarding technique to cut the decoding latency by 30 % over the latest design [23]. The detector and decoder exchange symbol log-likelihood ratios (LLR) that are efficiently computed based on the L1 distance to the nearest neighbors in the QAM constellation.

To lower the power consumption, automatic clock gating is applied to stage boundary and buffer registers to save 53 % of the detector power and 61 % of the decoder power. The results are demonstrated in a 65nm MMSE-NBLDPC iterative detector-decoder test chip that achieves 1.38 Gb/s detection and 1.02 Gb/s decoding (5 iterations), consuming 26.5 mW and 103 mW, respectively.

CHAPTER III

Low-complexity Message-Passing Massive MIMO Detector

The upcoming fifth generation (5G) wireless communication systems will significantly increase the network capacity and coverage, and improve the spectral and energy efficiency. Large-scale MIMO, or massive MIMO, has been identified as a key disruptive technology for 5G [31, 32, 33]. Large-scale MIMO is a multi-user MIMO technique that relies on a large number, e.g., hundreds, of base station antennas to serve a multiplicity of, e.g., tens, of autonomous single-antenna users in each time-frequency resource [34]. The large number of antennas provide a high spatial multiplexing gain for an increased capacity; and the radiated energy can be focused to the intended receivers for an improved energy efficiency.

The optimal maximum likelihood detector searches all the possible points of user-symbol space incurring prohibitive complexity. Even limited-search detectors, like sphere decoder [13] and Tabu Search [16] are still considered impractical due to their high complexities that increase exponentially with the number of antennas and the QAM size. A lower complexity minimum mean square error (MMSE) detector has demonstrated good performance for massive MIMO [31, 32, 33, 35], especially when the loading ratio is large, i.e. the number of base station antennas is much greater than the number of users. However, the complexity of matrix inversion still grows cubically with the number of user terminals, i.e. $O(N_t^3)$, where large matrix inversions, i.e. size of 8×8 or greater introduce large area and latency overheads. Prior work has shown that for a massive MIMO system of a large loading ratio with i.i.d. Rayleigh fading channel, an approximated or implicit matrix inversion is sufficient for achieving near-optimal BER-SNR performances. An approximated or implicit matrix inversion even reduces the implementation complexity to $O(N_t^2)$. For example, Neumann series based detector [36], when using 3 terms of approximation, can have low implementation complexity with a detection performance slightly worse than an MMSE detector. Recently iterative message-passing detectors (MPD) have emerged, such as channel hardening-exploiting message passing (CHEMP) [37]. A CHEMP detector does not require matrix inversion, and it provides a better BER than an MMSE detector under an independent and identically distributed (i.i.d.) Rayleigh fading channel.

In this chapter, we present a low complexity message-passing detector for a 256-QAM massive MIMO uplink system serving 32 users ($N_t = 32$). The rest of the chapter is organized as follow. In Section 3.1, we give the background of the message-passing detection algorithm and analyze its complexity. We then present the algorithm-architecture co-optimization for efficient hardware implementation in Sections 3.2 and 3.3. In Section 3.4, we elaborate on the proposed low-power techniques to improve the energy efficiency of the MPD detector. Section 3.5 provides the silicon measurement results, and Section 3.6 concludes this work.

3.1 Message-Passing Detection Algorithm

In a massive MIMO uplink system, the base station is equipped with N_r antennas serving N_t single-antenna users at the same time and frequency resources. The

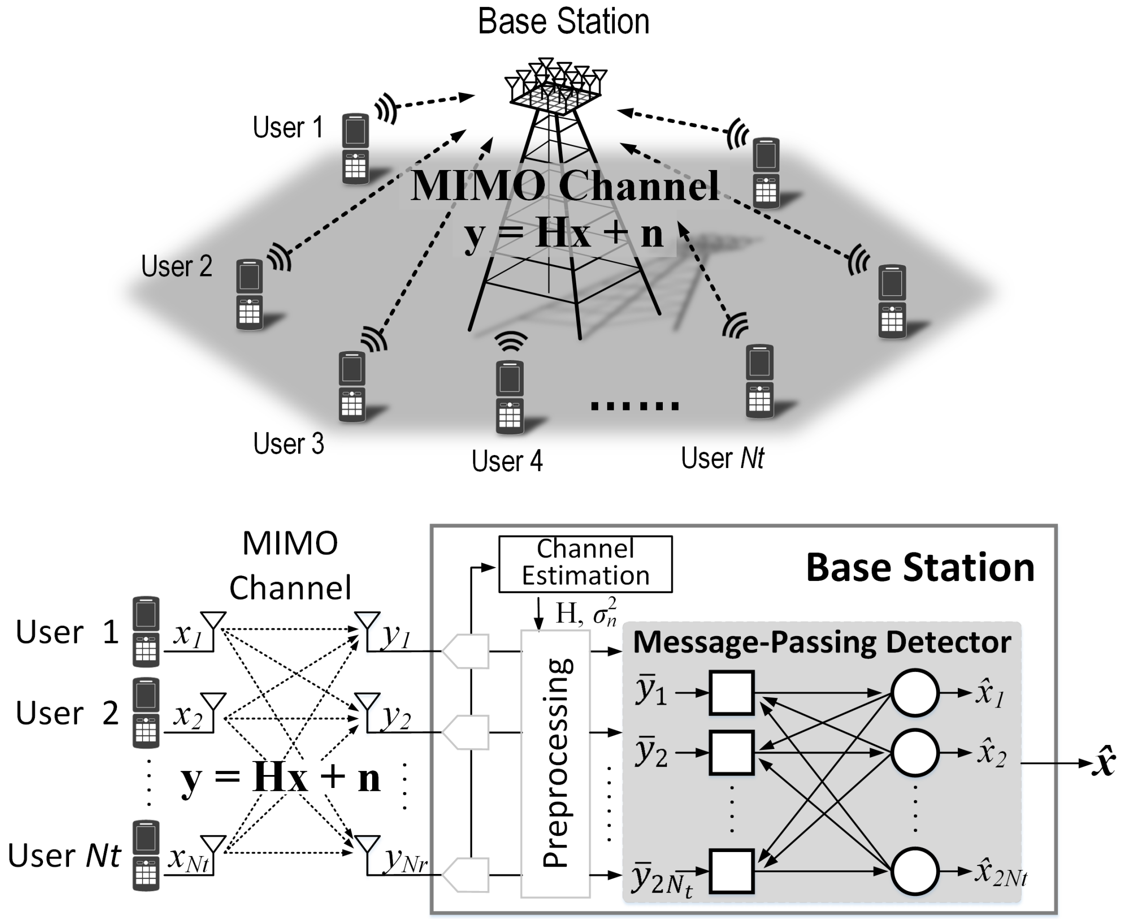


Figure 3.1: Illustration of an uplink large-scale MIMO system of N_t single-antenna users and N_r antennas at base station; and a top-level block diagram of a MPD detector.

received per-tone signal vector $\mathbf{y}^c$ can be modeled as:

$$\mathbf{y}^c = \mathbf{H}^c \mathbf{x}^c + \mathbf{n}^c \tag{3.1}$$

Here the received symbols are $\mathbf{y}^c \in \mathbb{C}^{N_r}$, and the transmitted user M-QAM symbols are $\mathbf{x}^c \in \tilde{\mathcal{A}}^{N_t}$, where $\tilde{\mathcal{A}}$ is the QAM alphabet. The channel matrix $\mathbf{H}^c \in \mathbb{C}^{N_r \times Nt}$ is assumed to be an i.i.d. Rayleigh fading channel, and $\mathbf{n}^c$ is an additive white circular-symmetric complex Gaussian noise vector $\mathbf{n}^c = [n_1^c, n_2^c, .., n_{N_r}^c]^\mathsf{T}$ with independent zero-mean components and N_0-variance. The complex values in the equation can be written in the real domain as:

$$\begin{aligned}\begin{bmatrix}\Re(\mathbf{y}^c)\\ \Im(\mathbf{y}^c)\end{bmatrix} &= \begin{bmatrix}\Re(\mathbf{H}^c) & -\Im(\mathbf{H}^c)\\ \Im(\mathbf{H}^c) & \Re(\mathbf{H}^c)\end{bmatrix}\begin{bmatrix}\Re(\mathbf{x}^c)\\ \Im(\mathbf{x}^c)\end{bmatrix} + \begin{bmatrix}\Re(\mathbf{n}^c)\\ \Im(\mathbf{n}^c)\end{bmatrix}\\ &\Rightarrow \mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \end{aligned} \tag{3.2}$$

where $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts, respectively. The real and imaginary part of a QAM symbol are represented by the two underlying Pulse Amplitude Modulation (PAM) symbols from alphabet $\mathcal{A}$, i.e. $\mathbf{x} \in \mathcal{A}^{2N_t}$, where $|\mathcal{A}| = \sqrt{M}$.

After matched filtering (MF) on received $\mathbf{y}$ and normalizing by N_r, we get

$$\begin{aligned}\frac{\mathbf{H}^{\mathsf{T}}\mathbf{y}}{N_r} &= \frac{\mathbf{H}^{\mathsf{T}}\mathbf{H}\mathbf{x}}{N_r} + \frac{\mathbf{H}^{\mathsf{T}}\mathbf{n}}{N_r}\\ \Rightarrow \overline{\mathbf{y}}^{MF} &= \overline{\mathbf{G}}\mathbf{x} + \overline{\mathbf{n}}, \end{aligned} \tag{3.3}$$

Based on (3.3), the message-passing detector estimates the mean and variance of $\mathbf{x}$ by iteratively performing interference cancellation and constellation matching.

3.1.1 Interference-Plus-Noise Approximation and Cancellation

From the i-th user's perspective, the matched-filtered channel observation $\overline{y}_i^{MF}$ is the intended symbol x_i coupled with interference from the other users plus noise. The equation (3.3) can be rewritten as:

$$\overline{y}_i^{MF} = \overline{G}_{ii}x_i + \sum_{j=1,\, j\neq i}^{2N_t} \overline{G}_{ij}x_j + \overline{n}_i, \tag{3.4}$$

where the non-bold symbols represent the elements of the vector (or matrix) indexed by subscripts, e.g. $\overline{G}_{ij}$ is the element in the i-th row and j-th column of $\overline{\mathbf{G}}$.

Let w_i represents the interference-plus-noise of user i:

$$w_i = \sum_{j=1,\, j\neq i}^{2N_t} \overline{G}_{ij} x_j + \overline{n}_i. \tag{3.5}$$

We approximate w_i as a Gaussian random variable due to the large number of N_t in massive MIMO and the central limit theorem, i.e., $w_i \sim \mathcal{N}(\mu_{w_i}, \sigma^2_{w_i})$. The mean and variance are calculated as:

$$\begin{aligned} \mu_{w_i} &= \sum_{j=1,\, j\neq i}^{2N_t} \overline{G}_{ij} \mathbb{E}[x_j] + 0, \\ \sigma^2_{w_i} &= \sum_{j=1,\, j\neq i}^{2N_t} \overline{G}^2_{ij} \mathrm{Var}[x_j] + \sigma^2_{\overline{n}} \end{aligned} \tag{3.6}$$

Here $\mathbb{E}[x_j]$ and $\mathrm{Var}[x_j]$ are the mean and variance of the symbol estimate x_j. By canceling this interference-plus-noise w_i from the observation $\overline{y}_i^{MF}$, the intended symbol x_i can be estimated also as a Gaussian random variable whose mean and variance are calculated as follows:

$$\begin{aligned} \mu_{x_i} &= (\overline{y}_i^{MF} - \mu_{w_i})/\overline{G}_{ii}, \\ \sigma^2_{x_i} &= \sigma^2_{w_i}/\overline{G}^2_{ii}. \end{aligned} \tag{3.7}$$

3.1.2 Constellation Matching

The symbol estimates from interference cancellation are refined in this step by considering the constellation information. With the Gaussian approximation in (3.7), the posterior probability of each constellation point is calculated as:

$$P(x_i = s \mid \overline{y}_i^{MF}, \overline{\mathbf{G}}) \propto \exp(\frac{-1}{2\sigma^2_{x_i}}(s - \mu_{x_i})^2) \quad \forall s \in \mathcal{A}, \tag{3.8}$$

where s is a symbol in the PAM constellation $\mathcal{A}$. And (3.8) can be viewed as taking $|\mathcal{A}| = \sqrt{M}$ discrete constellation samples from the continuous Gaussian estimation of the symbol x_i. Given the probabilities of constellation points, the mean and variance of the symbol x_i are refined as:

$$\begin{aligned} \mathbb{E}[x_i] &= \sum_{\forall s \in \mathcal{A}} sP(x_i = s), \\ \mathrm{Var}[x_i] &= \sum_{\forall s \in \mathcal{A}} s^2 P(x_i = s) - \mathbb{E}[x_i]^2. \end{aligned} \tag{3.9}$$

Here we drop the condition terms $\overline{y}_i^{MF}$ and $\overline{\mathbf{G}}$ of $P(x_i = s)$ for convenience. This update process can be viewed as a re-match of the first and second moments (mean and variance) of the symbol estimate. The updated mean and variance will be used to calculate the interference-plus-noise (3.6) for the next message-passing detection iteration. The iteration stops when all the symbol estimates converge or a certain number of iterations is reached.

3.2 Complexity Reduction and Convergence Speedup

In every MPD iteration, the mean and the variance calculation in (3.6) for each symbol requires $6(N_t - 1)$ multiply-accumulates (MAC); and constellation matching requires $\sqrt{M}$ Gaussian evaluations (which can be implemented by table lookup) in (3.8) and $3\sqrt{M} + 1$ MACs in (3.9). The hardware implementation cost grows with the number of users N_t and the order of modulation M.

To reduce the hardware complexity, we simplify the original message-passing detection by a symbol hardening technique. In addition, we adopt the layered scheduling to roughly double the convergence speed. The proposed schemes are elaborated in the following subsections.

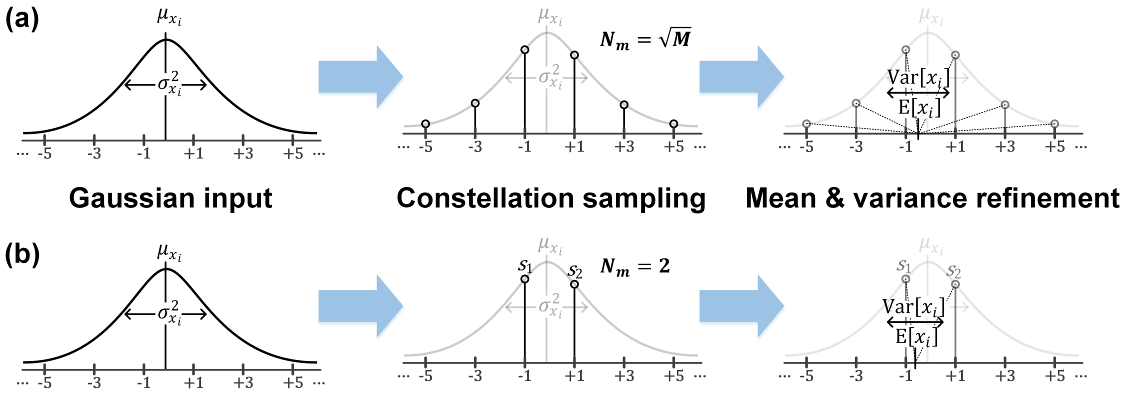


Figure 3.2: (a) Full constellation matching ($N_m = \sqrt{M}$) and (b) constellation matching using 2 nearest neighbors ($N_m = 2$).

3.2.1 Low-Complexity Symbol Hardening

The constellation matching step in the original MPD requires exhaustive computation to obtain the probabilities of all constellation points, which dominates the implementation cost, especially when the constellation size is large. We propose a symbol hardening technique based on the nearest neighbor approximation.

3.2.1.1 Nearest-Neighbor Symbols Approximation

We first approximate constellation matching by using only N_m (out of $\sqrt{M}$) most likely constellation points in $\mathcal{A}$. For example, if $N_m = 2$, we will choose two most likely constellation points s_1 and s_2 (which are also the two nearest neighbors of μ_{x_i} as shown in the Fig. 3.2(b)) to approximate the original constellation matching in (3.9) as:

$$\begin{aligned} \mathbb{E}[x_i] &\approx s_1 P(x_i = s_1) + s_2 P(x_i = s_2), \\ \mathrm{Var}[x_i] &\approx s_1^2 P(x_i = s_1) + s_2^2 P(x_i = s_2) - \mathbb{E}[x_i]^2. \end{aligned} \tag{3.10}$$

Since s_1 and s_2 are two neighboring constellation points that are spaced apart by 2 (say $s_2 = s_1 + 2$), and $P(x_i = s_1) + P(x_i = s_2) = 1$, the equations can be further simplified as:

$$\begin{aligned}\mathbb{E}[x_i] &\approx s_1 + 2P(x_i = s_2),\\ \mathrm{Var}[x_i] &\approx 4P(x_i = s_2)(1 - P(x_i = s_2)).\end{aligned} \tag{3.11}$$

The simplification requires only one costly Gaussian evaluation and one MAC operation by a factor of 8 from the baseline (3.9) for the 256-QAM symbol estimation.

3.2.1.2 Symbol Hardening

When an i.i.d. channel is assumed for a massive MIMO system, the variance of the symbol estimates converges early and at a fast pace, shown in Fig. 3.3. This allows for the aggressive choice of $N_m = 1$. This nearest-neighbor approximation not only eliminates the variance calculation but also simplifies the mean calculation further to

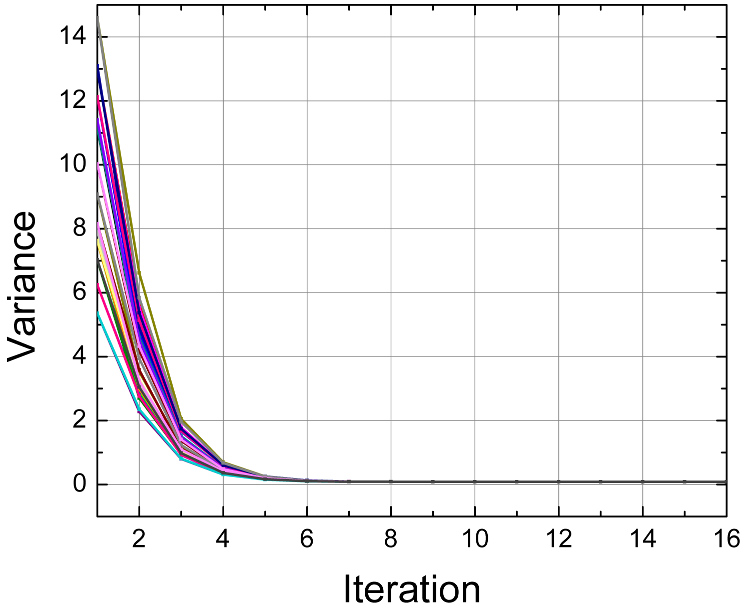


Figure 3.3: Convergence of interference canceled symbol estimate variance. Here 64 lines represent the value of $\sigma_{x_i}^2$ for $i = 1, ..., 64$.

Table 3.1: Computational Complexity Comparison of Original, Two-Neighbor Approximation and Symbol-hardening MPD

	Original MPD ($N_m = \sqrt{M}$)	Two-neighbor approx. MPD ($N_m = 2$)	Symbol-hardening MPD ($N_m = 1$)
Interference cancellation	$6(N_t - 1)$ MACs	$6(N_t - 1)$ MACs	$2(N_t - 1)$ MACs
Constellation matching	$3\sqrt{M} + 1$ MACs $\sqrt{M}$ Gaussian lookups	1 MAC 1 Gaussian lookup	
Total	$6N_t + 3\sqrt{M} - 5$ MACs $\sqrt{M}$ Gaussian lookups	$6N_t - 5$ MACs 1 Gaussian lookups	$2N_t - 2$ MACs

The complexity number is evaluated for one symbol in each iteration. The overall MPD complexity is the total number $\times 2N_t \times N_{iter}$, where N_{iter} is the number of iterations.

one hard decision making:

$$\begin{aligned} \mathbb{E}[x_i] &\approx \mathsf{hard}(\mu_{x_i}) = 2\left[(\mu_{x_i} - 1)/2\right] + 1, \\ \mathrm{Var}[x_i] &\approx 0. \end{aligned} \tag{3.12}$$

Here notation [.] represents nearest integer function which can be implemented by bit slicing, requiring no multipliers compared to original calculations (3.8) and (3.9). That is, the 5-bit harden symbol output for each of the real and imaginary parts of 256-QAM is obtained from the 8-bit soft symbol estimate x_i truncating the 3 LSB bits (represent $2^0, 2^{-1}, 2^{-2}$). Moreover, symbol hardening eliminates all variance calculations, saving half of the MACs in both interference cancellation and constellation matching with a negligible impact on SNR.

The computational complexity of the original MPD, the two nearest-neighbor approximated MPD, and the proposed symbol-hardening MPD are summarized in Table 3.1. In all, the approximation cuts 11072 MACs and 1024 Gaussian lookups while sacrificing SNR by less than 0.1 dB for a $N_r = 128 \times N_t = 32$ 256-QAM MIMO system.

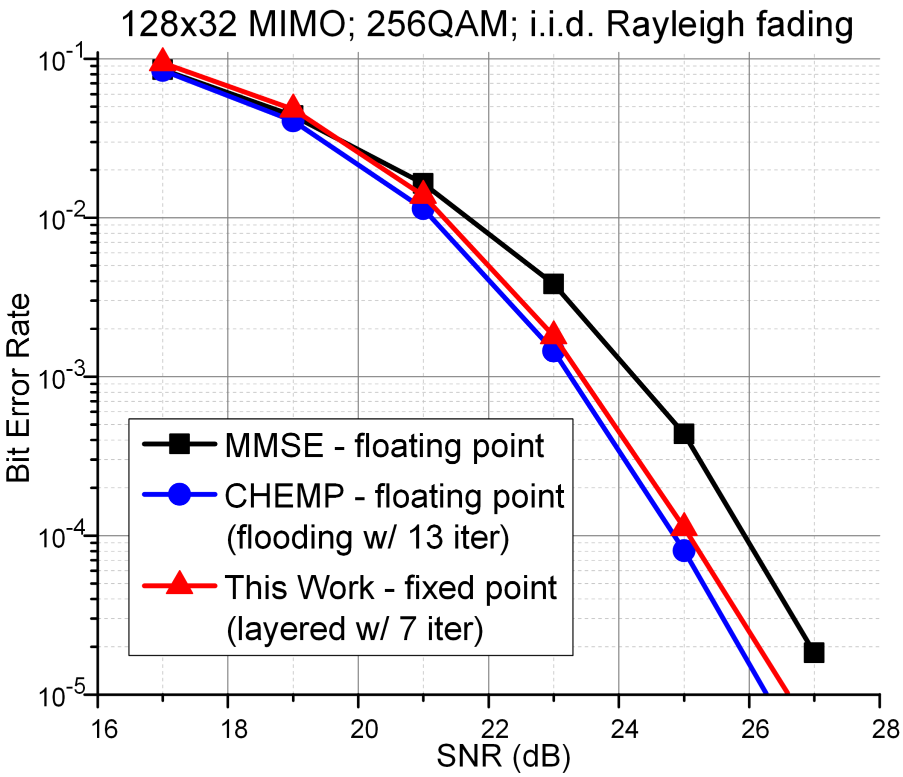


Figure 3.4: Bit error rate (BER) performance of 128 × 32 uplink MMSE, CHEMP and proposed detections.

3.2.2 Flooding and Layered Schedules

Inspired by the message-passing architectures for LDPC decoding, we explore both flooding and layered schedules in the proposed message-passing detection for efficient hardware implementations.

Flooding scheduling. With a flooding scheduling, the messages containing means and variances of all the symbol estimates are passed in parallel for each iteration. This scheduling requires the most hardware resources.

Layered scheduling. The layered scheduling divides all the messages into N_{layer} layers. Each layer processing updates the partial interference contributed from the symbols in the layer. Right after a layer is done, the updated symbol estimates are forwarded to the next layer. As an added benefit, this intra-iteration forwarding speeds up convergence by nearly 2 times compared to the flooding schedule. As shown in Fig. 3.4, the proposed message-passing detector with layered scheduling ($N_{layer} = 4$) requires only 7 iterations to achieve similar BER-SNR performance as the flooding scheduling running 13 iterations. The proposed low-complexity layered message-passing detection is summarized in Algorithm 1.

Algorithm 1 The Proposed 4-Layered Hard-Symbol Message-Passing Detection

```
Inputs:
    $\overline{\mathbf{G}} = \mathbf{H}^{\mathsf{H}}\mathbf{H}/N_r = [\overline{\mathbf{g}}_1, ..., \overline{\mathbf{g}}_{2N_t}]$, where $\overline{\mathbf{g}}_i$ is the $i$-th column of $\overline{\mathbf{G}}$
    $\overline{\mathbf{y}}^{MF} = \mathbf{H}^{\mathsf{H}}\mathbf{y}/N_r$
Initialization:
    All $\mathbf{x} \leftarrow 0$
    Partial interference $\mathbf{W}_l \leftarrow 0$ for $l = 1 : 4$
Iteration:
while iteration count $iter <$ max iterations $iter_{max}$ do
    for layer $l = 1 : 4$ do
        // Partial interference update
        $\mathbf{W}_l \leftarrow \sum_{\forall j \in \mathcal{L}} \overline{\mathbf{g}}_j x_j$, with layer indices $\mathcal{L} = \{4(l-1)+1 : 4l\}$
        // Interference cancellation
        $x_i^{soft} \leftarrow (\overline{y}_i^{MF} - \sum_{k=1}^{4} \mathbf{W}_k + \overline{\mathbf{g}}_i x_i)/\overline{G}_{i,i} \quad \forall i = 1 : 2N_t$
        // Symbol hardening
        $\mathbf{x} \leftarrow \mathsf{hard}(\mathbf{x}^{soft})$
        // Early termination
        if all $\mathbf{x}$ converges then
            Terminate the iteration.
    iteration count $iter \leftarrow iter + 1$
```

3.3 Architectural Optimization

A message-passing detector requires two sets of processing elements (PE) that iterate between each other, a set of N_t constellation PEs (CPE) to update the real and imaginary parts of a user estimate by considering the $\sqrt{M}$ constellation points, and a set of N_t Interference PEs (IPE) to compute and cancel the interference-plus-noise for each user. Each message passing between PEs contains 2 symbol estimates, the real and imaginary parts of a user QAM symbol. In this section, we explore three

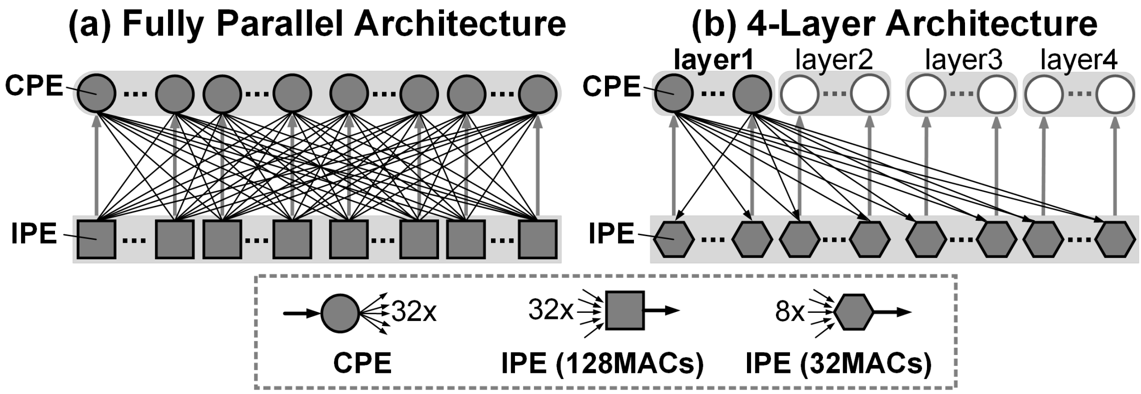


Figure 3.5: Architectural optimization from (a) fully parallel architecture using a flooding schedule to (b) block parallel architecture using a layered schedule.

architecture designs for a massive MIMO uplink detector supporting $N_r = 128$ base station antennas and $N_t = 32$ single-antenna users.

3.3.1 Fully Parallel Architecture

The detector can be simply mapped to a fully parallel architecture with 32 Interference PEs and 32 constellation PEs (Fig. 3.5(a)) to support flooding scheduling. All 64 PEs are able to send messages to each other at the same time. This architecture requires nearly 4K MACs and 10K interconnects between the PEs. Despite the high throughput, the fully parallel architecture is dominated by global wiring, resulting in a large silicon area, low clock frequency, and high power.

3.3.2 4-Layer Architecture

The 4-layer architecture is shown in Fig. 3.5(b). It divides the use of all 32 user estimates into 4 layers (i.e. user $4(l-1)+1$ to user $4l$ belong to layer l, for $l = 1, ..., 4$), thus the 4-layer architecture uses only 1/4 as many MACs in each Interference PE. The interference cancellation is done layer by layer. For example, in processing layer 1, all 32 Interference PEs compute the partial interference contributed from the 8

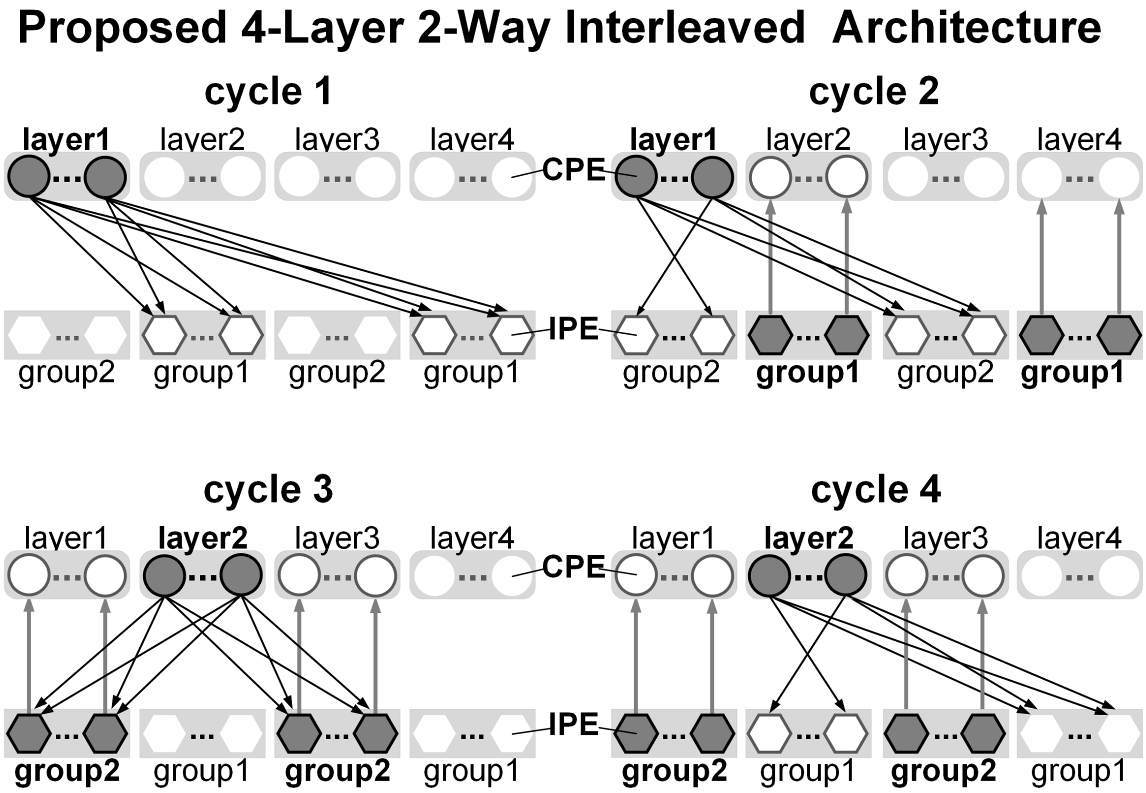


Figure 3.6: The proposed 4-layer 2-way interleaved architecture and the data flow of first 4 cycles.

users in the layer 1, and combine with the other partial interference calculated and stored in previous processing of layer 2, 3, and 4. The combined interference is used for cancellation and the results are sent to CPEs to perform constellation matching. After a layer is done, the updated symbol estimates are forwarded to the next layer. As an added benefit, this intra-iteration forwarding speeds up convergence by nearly 2 times compared to the flooding schedule used in the fully parallel architecture.

3.3.3 4-Layer Architecture with 2-way Interleaving

To reduce the area and power further, we halve the number of Interference PEs and time-multiplex their use into 2 groups with a 2-stage pipeline (shown in Fig. 3.6). In each stage, a group of CPEs and a group of IPEs are able to active at the same

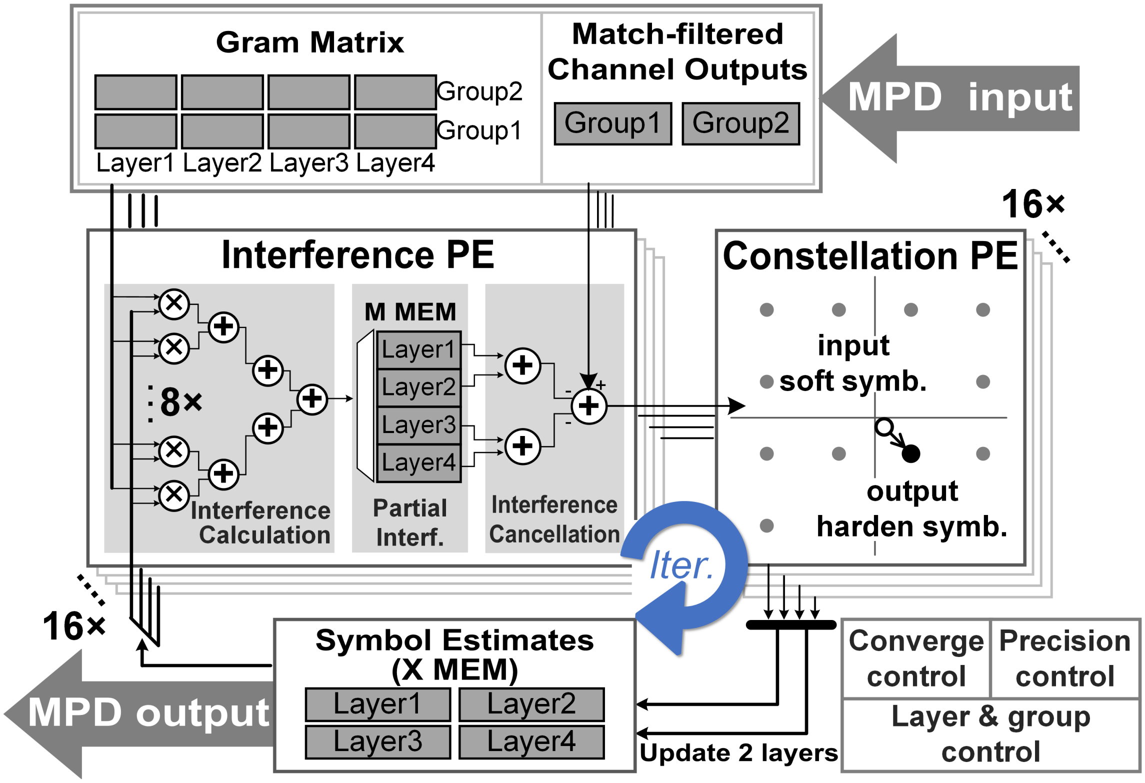


Figure 3.7: Detailed block diagram of the proposed message-passing detector using 16 Interference PEs and 16 constellation PEs.

time without violating data dependency between consecutive layers. The layer-group schedule is interleaved to avoid pipeline stalls.

The detailed block diagram of the proposed 4-layer 2-way interleaved architecture is shown in Fig. 3.7. The design is consist of 16 Interference PEs, 16 Constellation PEs, a symbol estimate memory (X MEM), an input memory, and a group of controllers. The input memory stores the preprocessed Gram matrix $\overline{\mathbf{G}}$ and match-filtered channel outputs $\overline{\mathbf{y}}^{MF}$. The controllers are in charge of convergence detection and early termination, dynamic precision control, and layer-group scheduling. Based on auto-place-and-route results shown in Fig. 3.12, the overall architectural optimization from (a) to (c) reduces area and power by 4.24 times and 2.83 times respectively at a cost of 2.41 times lower throughput. The throughput can be further recovered by 1.5 times by enabling early termination, which will be mentioned in the next section.

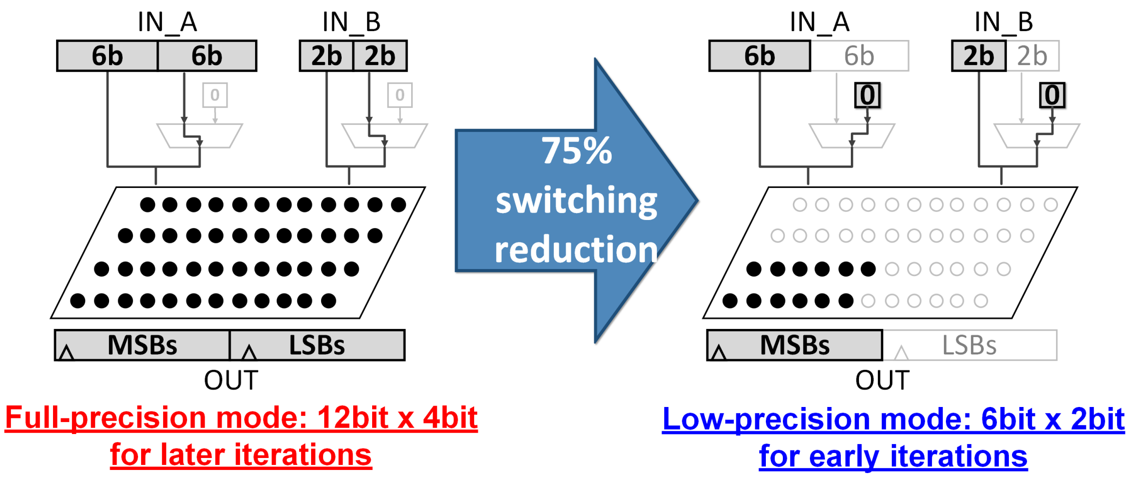


Figure 3.8: Multiplier with full precision mode (left) and low precision mode (right).

3.4 Low-Power Techniques

We apply adaptive dynamic precision control, clock gating and early termination to our proposed 4-layer 2-way interleaved architecture for further improving the energy efficiency. The three low-power techniques are elaborated in the following subsections.

3.4.1 Adaptive Dynamic Precision Control

The datapath power of the MPD architecture is dominated by the multipliers. To save significant dynamic power, we exploit the convergence behavior of the MPD to adapt the multiplier precision dynamically.

3.4.1.1 Convergence Behavior

In early iterations, the user symbol estimates are noisy and unstable so the detector makes coarse symbol estimates. That is, the symbol updates often result in large jumps from between constellation points. With more iterations, the symbol estimates tend to be more accurate and the detector fine-tunes the estimates. The symbol updates eventuality become small movements and convergence is reached.

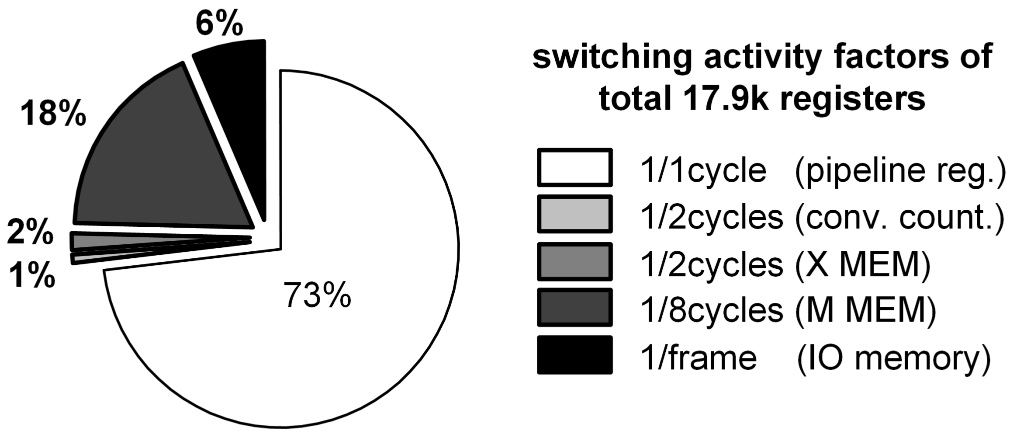


Figure 3.9: Power breakdown and the activities of registers.

3.4.1.2 Multiplier Design with Full/Low-Precision Mode

Based on the convergence behavior, the MPD detector requires only low-precision multiplications for coarse symbol estimates in the early iterations and high-precision multiplications to fine tune the results in the late iterations. Therefore, we design the multipliers in the Interference PEs to support two precision modes as shown in Fig 3.8. In particular, a 12b×4b full-precision multiplier is designed to support a 6b×2b low-precision mode with the LSBs disabled, saving 75 % of the switching activity and reducing dynamic power. The precision mode can be dynamically tuned to achieve lower power consumption to meet different SNR and BER requirements.

3.4.2 Clock Gating Exploiting Regular Memory Access

Registers are used on-chip as data memory to support the wide access required by the architecture. The memory access is deterministic and regular as shown in Fig. 3.9, e.g., the 3Kb partial interference memory (M MEM) is only updated once every 8 cycles, and the 512b symbol estimate memory (X MEM) is updated once every 2 cycles. Therefore we implement clock gating to turn off the clock input when the memory is not updated to save dynamic power.

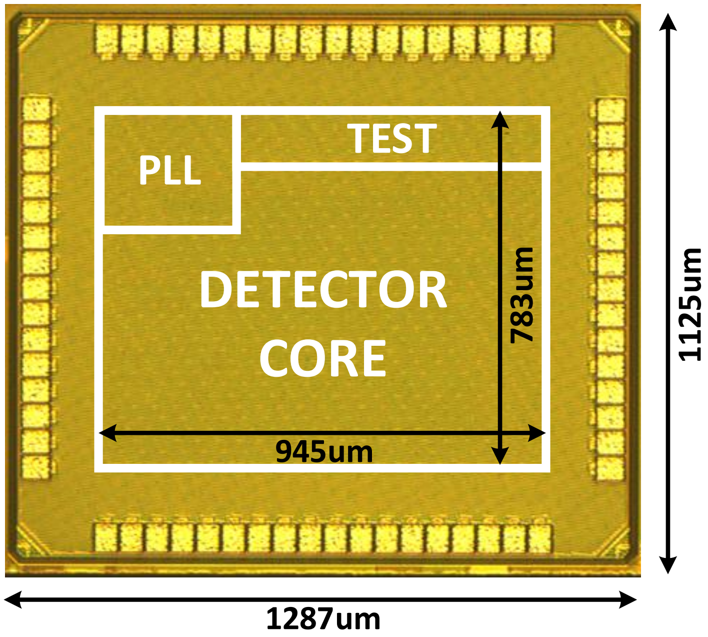


Figure 3.10: The proposed MPD Chip photo containing a PLL, testing block and detector core.

3.4.3 Early Termination

In this work, we consider that the convergence is reached when the current symbol estimate matches the estimate from the previous cycle. The MPD iteration is early terminated when all the symbol estimates reach convergence. A convergence checker is implemented and a global Finite State Machine (FSM) early-termination controller monitors the convergence of each symbol.

3.5 Chip Measurement Results and Discussion

The die photo of the massive MIMO detector design is shown in Fig. 3.10. The test chip is fabricated in a 40 nm CMOS technology including a 0.58 mm^2 detector core, a PLL to generate the clock, a test memory to store test vectors, and scan chains for input and output. The chip is measured to run at a maximum frequency of 433 MHz at the nominal supply voltage of 0.9 V in room temperature, dissipating 238.7 mW.

3.5.1 Voltage Scaling

Fig. 3.11 shows the average throughput (Gb/s) and energy efficiency (pJ/b) with voltage-frequency scaling from the nominal VDD of 0.9 V to the lowest VDD of 0.55 V. With early termination enabled on-chip, detection converges in 5.7, 5.2, and 4.9 iterations on average at 23 dB, 25 dB, and 27 dB SNR, enabling a throughput up to 2.82 Gb/s and an energy efficiency up to 84.2 pJ/b.

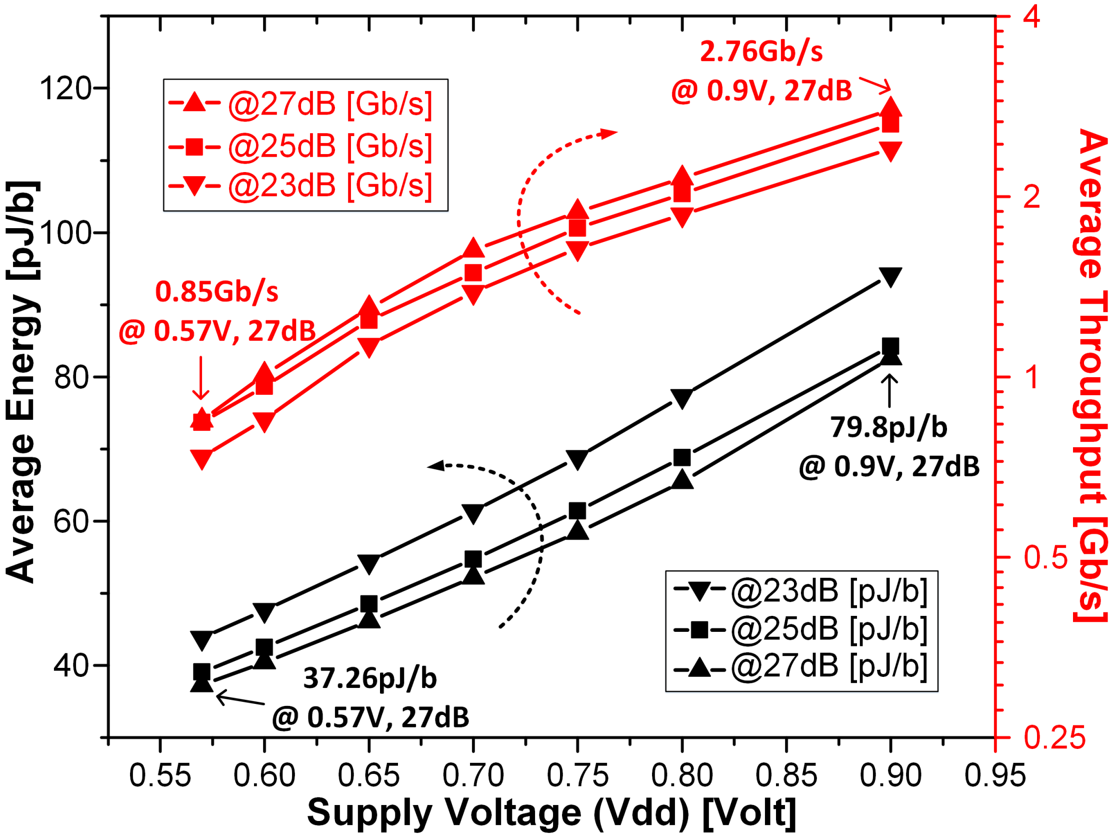


Figure 3.11: Measured average throughput (red) and energy efficiency (black) with voltage scaling at different SNRs.

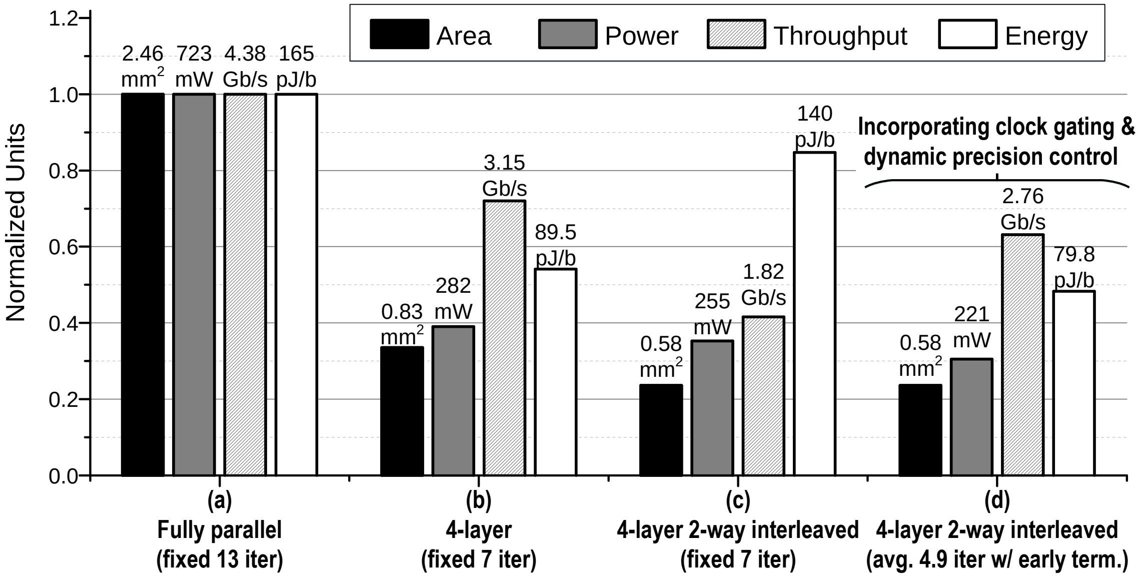


Figure 3.12: Measured throughput, power, and energy consumption improvement by proposed dynamic precision control, clock gating and early termination.

3.5.2 Results of Architectural Optimizations

Fig. 3.12 shows the MPD chip area, power, throughput, and energy compared to the architectures mentioned in Section 3.3. Compared to the baseline fully parallel architecture (a), the proposed 4-layer 2-way architecture (c) reduces area and power by 76 % and 65 % respectively while sacrificing 69 % of throughput loss, based on the place-and-route results. By adopting early termination, the MPD detector chip can recover the throughput loss to 37 % with average 4.9 detection iterations. In addition, the clock gating and the dynamic precision control allows the chip to reduce power dissipation by 70 % and energy per bit by 52 % over the baseline fully parallel architecture.

3.5.3 Comparison

We compare the MPD chip with state-of-the-art MIMO detector designs in Table 3.2. Note that all the previous approaches, including sphere decoding [18, 17, 19] and MMSE [38], incur much higher implementation costs in a large-scale MIMO

Table 3.2: Comparison Table of State-of-the-Art MIMO Detector Designs

Detector	Borlenghi [18]	Noethen [17]	Winter [19]	Chen [38]	This Work
Alogorithm	SD[a]	SD[a]	SD[a]	MMSE	MPD[b]
MIMO size	MIMO	MIMO	MIMO	MIMO	massive MIMO
$(N_r \times N_t)$	$\leq 4 \times 4$	$\leq 4 \times 4$	$\leq 4 \times 4$	4×4	128×32
Modulation	$\leq$64	$\leq$64	$\leq$64	256	256
Technology [nm]	65	65	65	65	40
Core area [mm^2]	2.78	-	0.31	0.7	0.58
Gate counts [kGE]	872	383	215	347	1022
Frequency [MHz]	135	445	333	517	433
Power [mW]	-	87	38	26.5	238.7
Throughput [Gb/s]	0.194	0.396	0.296-0.807	1.379	2.76 [c]
Area efficiency [Mb/s/mm^2]	0.07	-	0.96-2.6	1.97	4.76
Energy efficiency [pJ/b]	920	220	48	19.2	79.8
Energy efficiency [pJ/b/TX antenna]	230	55	12	4.8	2.49

[a] sphere decoding.

[b] message-passing detection.

[c] early termination with average 4.92 iterations and minimal 3.25 iterations at SNR=27dB.

system. This chip is the first silicon demonstration of a detector that supports a large-scale multi-user MIMO configuration, with 128 base station antennas and 32 single-antenna users. Despite the large-scale MIMO processing, the throughput of this chip still surpasses all the previous designs, and the energy efficiency remains competitive.

3.6 Summary

We demonstrate a 0.58 mm^2 128×32 256-QAM massive MIMO uplink detector based on message-passing detection. With the proposed symbol hardening technique, the complexity is reduced by more than 60 %. The detector implements a pipelined block parallel architecture using a layered-grouped schedule to accelerate convergence, enabling a throughput of 2.76 Gb/s (running an average 4.92 iterations with early termination) at 221 mW. The chip incorporates adaptive precision control and clock gating to improve energy efficiency by up to 43 %.

CHAPTER IV

Link-adaptive Expectation-Passing Massive MIMO Detector

In this chapter, we present a 2.0 mm^2 128×16 massive MIMO detector IC that provides 21 dB array gain and 16 times multiplexing gain at the system level. The detector implements iterative expectation-propagation detection (EPD) for up to 256-QAM modulation. Tested with measured channel data [39] , the detector achieves 4.3 dB processing gain over state-of-the-art massive MIMO detectors [40, 41], enabling 2.7 times reduction in transmit power for battery-powered mobile terminals. The EPD chip uses link-adaptive processing to meet a variety of practical channel conditions with scalable energy consumption. The design is realized in a condensed systolic array architecture and an approximate moment-matching circuitry to reach 1.8 Gb/s at 70.6 pJ/b. The performance and energy efficiency can be tuned over a wide range by UTBB-FDSOI body bias.

4.1 Channel Conditions and Link Adaptive Detection in Practical Massive MIMO Systems

Real-time detection for massive MIMO is compute-intensive and power-hungry due to large matrix dimensions and fast varying channels. Our MPD detector and

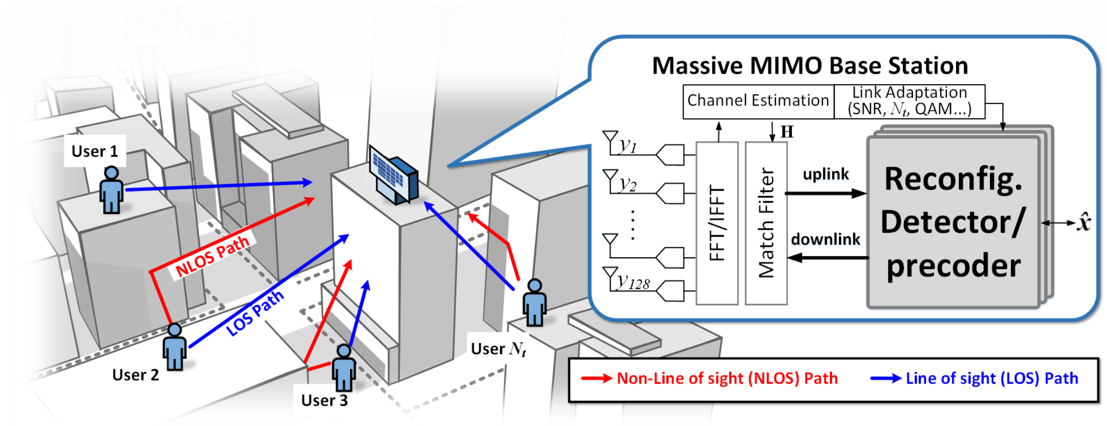


Figure 4.1: Illustration of a multi-user massive MIMO system.

other previous works [40, 41] demonstrated low-complexity massive MIMO detectors based on independent and identically distributed (i.i.d.) channel assumption in massive MIMO. However, the i.i.d. channel assumption does not always hold and these simplified detectors suffer from significant performance losses when tested in measured massive MIMO channels, especially in cases of high user load.

Fig. 4.2 shows the condition numbers of measured LOS (blue), NLOS (red) and ideal iid channels (green) with 100 subcarriers. The shade of each channel represents the range, the dashed line represents the standard deviation, and the solid line represents the means of the condition numbers. In the measured LOS channels where the correlated links are dominant, the channel matrices $\mathbf{H}$ are ill-conditioned with large condition numbers around 30. This makes the linear detection algorithms unstable and sensitive to errors, resulting in more than 6 dB of SNR loss against the optimal ML detection; whereas in the ideal i.i.d. channel, which assumes every link is perfectly independent to each other, the condition numbers are relatively small. A well-conditioned channel enables the use of linear MMSE detections to achieve nearly optimal performance.

In designing a practical massive MIMO detector, we select EPD that performs iterative interference cancellation [42] to offer near-optimal performance even in unfa-

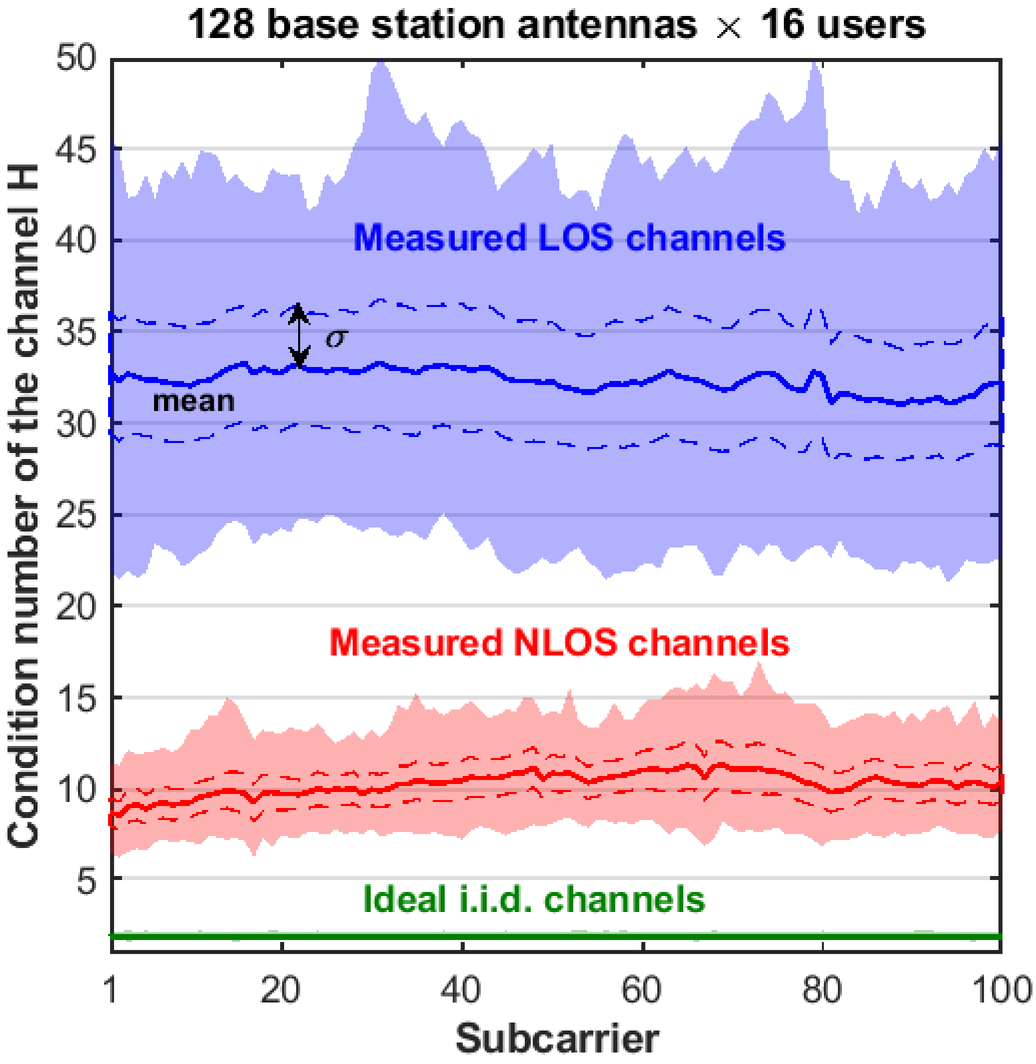


Figure 4.2: The condition numbers of measured LOS (blue), NLOS (red) and ideal iid channels with 100 subcarriers. The shade of each channel represent the range, the dashed line is the standard deviations, and the solid line is the means of the condition numbers.

vorable channel conditions. The complexity of EPD is limited to $O(K^3)$ per iteration, where N_t is the number of users. By iterative processing, an EPD adapts the processing effort to the channel, so as to achieve the required BER at the lowest energy. The EPD design incorporates explicit matrix inversion, so it could be reused for both uplink and downlink processing. Evaluated using measured massive MIMO channels, the EPD outperforms a linear MMSE detector by 0.7 dB, 4.3 dB, and 3.5 dB in i.i.d.,

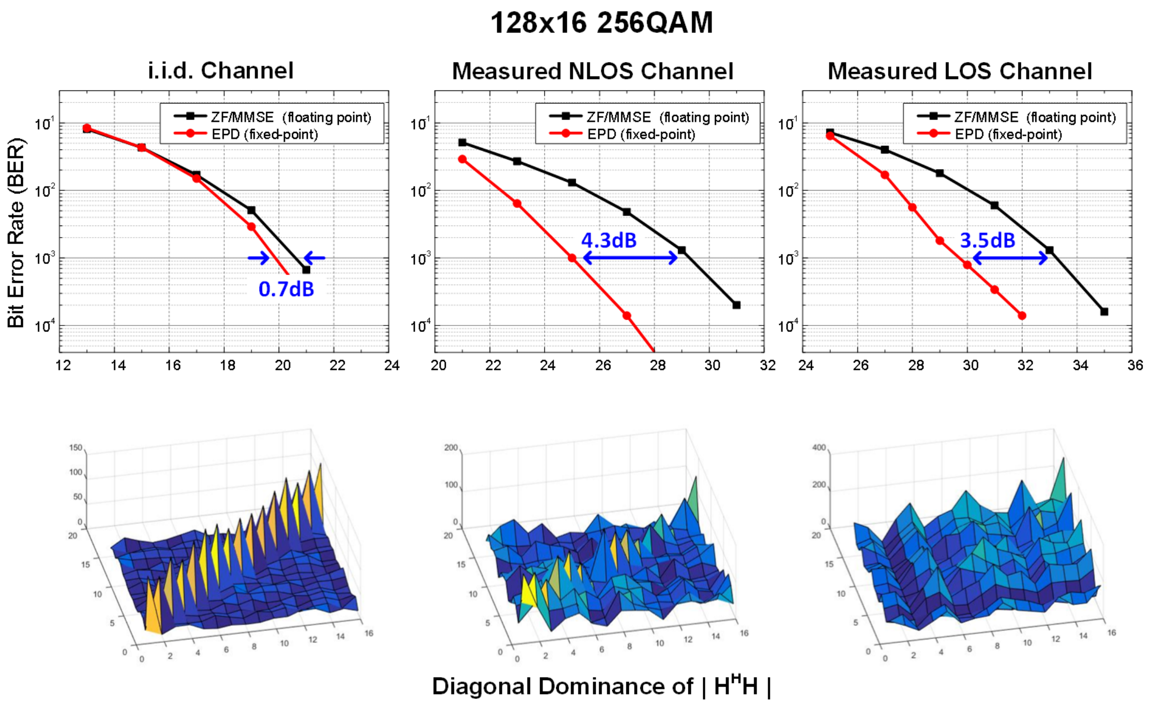


Figure 4.3: Top: BER of MMSE detector (black line) and EPD (read line) under different channels. Bottom: the absolute values of Gram matrices $|\mathbf{H}^{\mathsf{H}}\mathbf{H}|$.

non-line-of-sight (NLOS), and line-of-sight (LOS) conditions, respectively, as shown in Fig. 4.3.

4.2 Expectation Propagation Detection (EPD)

As mentioned in Section 3.1, a massive MIMO uplink systemwith N_r base station antennas serving N_t single-antenna users in each channel (subcarrier) use can be modeled in real-domain as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{4.1}$$

Here the received symbols $\mathbf{y} = [y_1, y_2, ..., y_{2N_r}]^{\mathsf{T}} \in \mathbb{R}^{2Nr \times 1}$, and transmitted user symbols $\mathbf{x} = [x_1, x_2, ..., x_{2N_t}]^{\mathsf{T}} \in \mathbb{R}^{2Nt \times 1}$. The channel matrix $\mathbf{H} \in \mathbb{R}^{2Nr \times 2Nt}$. And $\mathbf{n}$ is an additive white Gaussian noise vector $\mathbf{n} = [n_1, n_2, .., n_{N_r}]^{\mathsf{T}}$ with independent zero-mean components and N_0-variance. Note x_k and $x_{(k+N_t)}$ are the two Pulse Amplitude

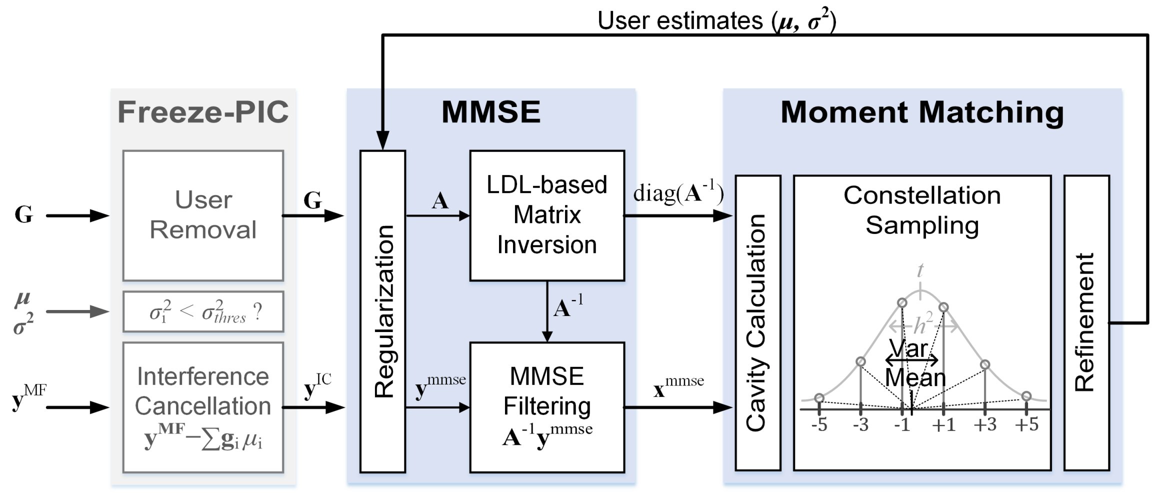


Figure 4.4: Block diagram of EPD. The blocks in gray shade perform dynamic dimension reduction and parallel interference cancellation (detailed in 4.3).

Modulation (PAM) symbols representing the real and imaginary part of an M-QAM symbol, i.e. $x_k + ix_{(k+N_t)}$, of the user k. The PAM alphabet is denoted by $\mathcal{A}$ and $|\mathcal{A}| = \sqrt{M}$.

According to the Bayes' theorem, the posterior probability of the user symbol is:

$$\begin{aligned} P(\mathbf{x} \mid \mathbf{y}) &\propto P(\mathbf{y} \mid \mathbf{x}) P(\mathbf{x}) \\ &\propto \mathcal{N}(\mathbf{y}; \mathbf{H}\mathbf{x}, N_0\mathbf{I}) \prod_{i=1}^{2N_t} P(x_i), \end{aligned} \tag{4.2}$$

where the prior probability of the user symbol $P(x_i)$ is a discrete uniform distribution over the PAM constellation $\mathcal{A}$:

$$P(x_i) = \frac{1}{\sqrt{M}} \mathbb{I}_{x_i \in \mathcal{A}} = \begin{cases} 1/\sqrt{M} & x_i \in \mathcal{A}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

In practice, it is infeasible to enumerate all possible combinations of user symbols in massive MIMO systems with a high QAM order.

Similar to an MMSE detector, an EPD detector first approximates each prior by a Gaussian distribution $\hat{P}(x_i)$ with mean μ_i and variance σ_i^2. Then the posterior is estimated to be:

$$\begin{aligned}\hat{P}(\mathbf{x} \mid \mathbf{y}) &= \mathcal{N}(\mathbf{y}; \mathbf{H}\mathbf{x}, N_0\mathbf{I}) \prod_{i=1}^{2N_t} \hat{P}(x_i) \\ &= \mathcal{N}(\mathbf{y}; \mathbf{H}\mathbf{x}, N_0\mathbf{I}) \prod_{i=1}^{2N_t} \mathcal{N}(x_i; \mu_i, \sigma_i^2). \qquad (4.4)\end{aligned}$$

This approximated posterior $\hat{P}(\mathbf{x} \mid \mathbf{y})$ in all can be modeled as a Gaussian distribution, which is tractable compared to the true posterior $P(\mathbf{x} \mid \mathbf{y})$. An EPD detector minimizes the divergence between the approximated posterior $\hat{P}(\mathbf{x} \mid \mathbf{y})$ and the true posterior $P(\mathbf{x} \mid \mathbf{y})$ by iteratively refining the mean (μ_i) and variance (σ_i^2) of the Gaussian-approximated prior. The EPD iteration is performed in 4 steps: 1) initialize the approximated prior as $\mathcal{N}(0, E_s)$, where E_s represents the symbol energy; 2) calculate the approximated posterior $\hat{P}(\mathbf{x} \mid \mathbf{y})$ using MMSE estimation; 3) refine the mean and variance of each prior by moment-matching; 4) repeat step 2 and 3 until convergence or a certain iteration limit. The step 2 and 3 are elaborated in the following subsections.

4.2.1 Posterior Calculation with MMSE Estimation

With Gaussian priors $\hat{P}(x_i)$, the approximated posteriors $\hat{P}(\mathbf{x} \mid \mathbf{y})$ in (4.4) are calculated by MMSE estimation steps as described below.

Step 1. Pre-processing: Compute Gram matrix $\mathbf{G} = \mathbf{H}^{\mathsf{H}}\mathbf{H}$, and perform match filtering $\mathbf{y}^{MF} = \mathbf{H}^{\mathsf{H}}\mathbf{y}$. Note that, this pre-processing is only done in the first iteration of the EPD detection and the Gram matrix $\mathbf{G}$ can be shared by multiple uplink data within the channel coherence time.

Step 2. Regularization: With the mean (μ_i) and variance (σ_i^2) of the Gaussian prior, regularize $\mathbf{G}$ and $\mathbf{y}^{MF}$ as:

$$\mathbf{A} = \mathbf{G} + N_0\mathbf{\Lambda}, \tag{4.5}$$

$$\mathbf{y}^{MMSE} = \mathbf{y}^{MF} + N_0\boldsymbol{\lambda}, \tag{4.6}$$

where regularization matrix $\mathbf{\Lambda} = \text{diag}(1/\sigma_1^2, ..., 1/\sigma_{N_t}^2)$ and regularization vector $\boldsymbol{\lambda} = [\mu_1/\sigma_1^2, ..., \mu_{N_t}/\sigma_{N_t}^2]^\mathsf{T}$.

Step 3. Matrix inversion: Compute MMSE filter matrix $\mathbf{A}^{-1}$ by performing LDL decomposition to get $\mathbf{A} = \mathbf{LDL^T}$, followed by forward substitution to get $\mathbf{L}^{-1}$, and backward substitution to get $\mathbf{A}^{-1}$.

Step 4. MMSE filtering: Apply MMSE filter matrix $\mathbf{A}^{-1}$ on $\mathbf{y}^{MMSE}$ to obtain the mean of the posterior $\hat{P}(\mathbf{x} \mid \mathbf{y})$ and extract the diagonal terms of $\mathbf{A}^{-1}$ to obtain the variance:

$$\begin{aligned} \mathbb{E}[\mathbf{x} \mid \mathbf{y}] &= \mathbf{A}^{-1}\mathbf{y}^{MMSE}, \\ \text{Var}[\mathbf{x} \mid \mathbf{y}] &= \text{diag}(\mathbf{A}^{-1}). \end{aligned} \tag{4.7}$$

The outputs of the MMSE estimation are the first and second moment, i.e. mean and variance, of the approximated posterior $\hat{P}(\mathbf{x} \mid \mathbf{y})$ which will be used to refine the mean and variance of the prior estimate for the next EPD iteration.

4.2.2 Gaussian Prior Refinement with Moment-Matching

To reduce the divergence from the true posterior $P(\mathbf{x} \mid \mathbf{y})$, an EPD detector performs the following steps:

Step 1. Restore one of the Gaussian-approximated prior factors in (4.4) to the true uniformly distributed discrete prior $\mathbb{I}_{x_i \in \mathcal{A}}$:

$$\begin{aligned}\tilde{P}(x_i \mid \mathbf{y}) &\propto \mathcal{N}(\mathbf{y}; \mathbf{H}\mathbf{x}, N_0 \mathbf{I}) \prod_{j \neq i} \hat{P}(x_j)\ \mathbb{I}_{x_i \in \mathcal{A}} \\ &\propto \hat{P}(x_i \mid \mathbf{y}) \ / \ \hat{P}(x_i) \times \mathbb{I}_{x_i \in \mathcal{A}} \\ &\propto \hat{P}_{\setminus i}(x_i \mid \mathbf{y}) \times \mathbb{I}_{x_i \in \mathcal{A}} \qquad\qquad \text{for } i = 1, ..., 2N_t. \qquad (4.8)\end{aligned}$$

Here $\hat{P}_{\setminus i}(x_i \mid \mathbf{y}) \sim \mathcal{N}(t_i, h_i^2)$ is defined as the cavity probability [42], whose mean t_i and variance h_i^2 are computed as:

$$\begin{aligned}h_i^2 &= (1/\mathrm{Var}[x_i \mid \mathbf{y}] - 1/\sigma_i^2)^{-1}, \\ t_i &= h_i^2 \times (\mathbb{E}[x_i \mid \mathbf{y}]/\mathrm{Var}[x_i \mid \mathbf{y}] - \mu_i/\sigma_i^2). \qquad (4.9)\end{aligned}$$

And by multiplying $\mathbb{I}_{x_i \in \mathcal{A}}$, the posterior $\tilde{P}(x_i \mid \mathbf{y})$ is obtained by taking discrete samples of constellation points on the continuous cavity probability $\hat{P}_{\setminus i}(x_i \mid \mathbf{y})$ as:

$$\tilde{P}(x_i = s \mid \mathbf{y}) \propto \exp\left(\frac{-(s - t_i)^2}{2h_i^2}\right) \qquad s \in \mathcal{A}, \qquad (4.10)$$

whose mean and variance are denoted as $\mathbb{E}[x_i]$ and $\mathrm{Var}[x_i]$.

Step 2. Refine the Gaussian-approximated prior, as $\hat{P}(x_i)^{(new)}$, to match the first and second moments (mean and variance) between $\tilde{P}(x_i \mid \mathbf{y})$ and $\hat{P}(x_i \mid \mathbf{y})^{(new)}$, where

$$\begin{aligned}\hat{P}(x_i \mid \mathbf{y})^{(new)} &\propto \mathcal{N}(\mathbf{y}; \mathbf{H}\mathbf{x}, N_0 \mathbf{I}) \prod_{j \neq i} \hat{P}(x_j)\ \hat{P}(x_i)^{(new)} \\ &\propto \hat{P}_{\setminus i}(x_i \mid \mathbf{y}) \times \hat{P}(x_i)^{(new)}. \qquad (4.11)\end{aligned}$$

The refined mean $\mu_i^{(new)}$ and variance $\sigma_i^{2(new)}$ of the prior $\hat{P}(x_i)^{(new)}$ are calculated as follows:

$$\mu_i^{(new)} = h_i^2 \times (\mathbb{E}[x_i]/\mathrm{Var}[x_i] - t_i/h_i^2),$$
$$\sigma_i^{2(new)} = (1/\mathrm{Var}[x_i] - 1/h_i^2)^{-1}. \tag{4.12}$$

Finally, the prior update is smoothened with an update rate β for next EPD iteration:

$$\mu_i = \beta\mu_i^{(new)} + (1-\beta)\mu_i,$$
$$\sigma_i^2 = \beta\sigma_i^{2(new)} + (1-\beta)\sigma_i^2. \tag{4.13}$$

In the next EPD iteration, this updated mean and variance of the prior is used to perform MMSE estimation mentioned in subsection 4.2.1 . The EPD iteration stops when the prior converges or a certain iteration limit is reached. Then the EPD detector outputs the hard decisions of the last symbol estimates $\boldsymbol{\mu} = [\mu_1, ..., \mu_{2N_t}]^\mathsf{T}$:

$$\mathbf{x}^{EPD} = \mathsf{hard}(\boldsymbol{\mu}) = 2\left[(\boldsymbol{\mu} - 1)/2\right] + 1. \tag{4.14}$$

Here, the notation [.] represents nearest integer function which can be implemented by bit slicing.

4.3 Link Adaption in EPD for Energy Efficiency

The iterative nature of the EPD algorithm offers an additional layer of link adaptation, in addition to modulation and code scheme, to deal with the varying channels in the real mobile user deployment. That is, in a favorable propagation environment, where highly independent links are dominant, EPD may need only one iteration to obtain reliable estimates. In this case, EPD has the same BER-SNR performance as a simple MMSE detector can achieve. On the other hand, when the channel degrades to a more correlated one, EPD is able to adapt to the channel by using more iterations to cancel the interference at the cost of longer latency and higher processing effort. Besides controlling of iteration number, we design within-iteration adaptations to improve energy efficiency.

4.3.1 Dynamic Dimension Reduction

Within the EPD iteration, the computation is dominated by the matrix inversion, whose complexity scales cubically with the matrix size $2N_t$. With different user deployments, the lightly correlated symbol estimates tend to converge much faster than the others. Hence we set a variance threshold σ^2_{thres} and consider a symbol estimate to be converged if it has a small variance, i.e. $\sigma_i^2 < \sigma^2_{thres}$. Then, we "freeze" the set of converged symbol estimates ($\mathcal{M} = \{m \mid \sigma_m^2 < \sigma^2_{thres},\ m = 1, ..., 2N_t\}$) and remove them from later processing, reducing the computational workloads. We refer to $\mathcal{M}$, the set of frozen user indices, as the "frozen list".

1. **Frozen symbol removal**: remove the columns and the rows associated with the frozen symbols in the original Gram matrix as the dimension-reduced Gram matrix $\mathbf{G}_{\mathcal{K}}$. Where $\mathcal{K} = \{1, 2, ..., 2N_t\} \backslash \mathcal{M}$ is the set of unfrozen symbol indices, referred to as "free list".

   This reduces the dimension of the matrix for later MMSE processing. Thus the computation requirement of EPD is significantly reduced after each iteration. In our implementation, the Gram matrix buffer is designed to be flexible to dynamically disable multiple columns and rows.

2. **Interference cancellation**: cancel the interference contributed by the con-

verged ("frozen") symbols:

$$\mathbf{y}^{IC} = \mathbf{y}^{MF} - \sum_{i \in \mathcal{M}} \mu_i \mathbf{g}_i, \tag{4.15}$$

where $\mathbf{g}_i$ is the i-th column of $\mathbf{G}$. The updated $\mathbf{y}^{IC}$ then goes through MMSE processing (regularization and filtering) mentioned before:

$$y_i^{MMSE} = y_i^{IC} + N_0 \left(\frac{\mu_i}{\sigma_i^2} \right) \qquad i \in \mathcal{K}. \tag{4.16}$$

4.3.2 Early Termination

At the end of each EPD iteration, the convergence controller checks if all the users are frozen. If so, all the user estimates are considered to be converged and the controller early terminates the EPD iteration to save unnecessary processing.

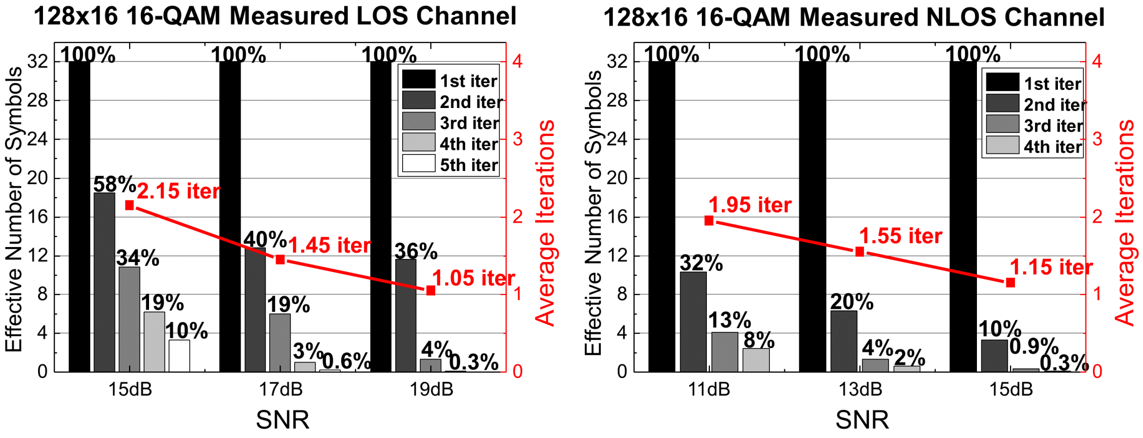


Figure 4.5: The effective number of symbols and iterations reduced by dimension reduction and early termination under different SNRs in LOS channel (left) and NLOS channel (right).

4.3.3 Evaluation

To evaluate the gain of the dynamic dimension reduction and the early termination techniques, we perform Monte Carlo simulations using different massive MIMO

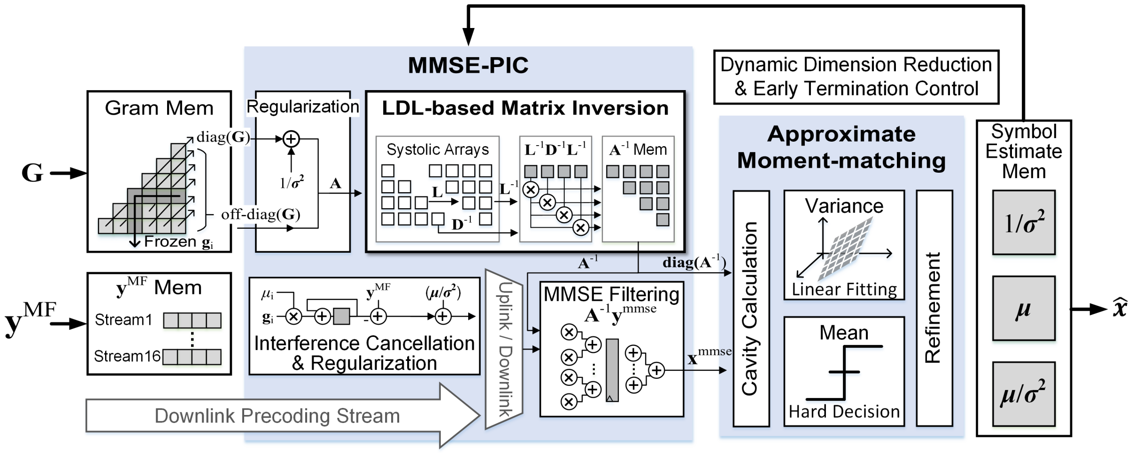


Figure 4.6: Link-adaptive EPD architecture.

channels measured by LuMaMi testbed, as shown in Fig. 4.5. Here, the results are averaged and we set the threshold variance $\sigma^2_{thres} = 0.02$. On average, the effective number of symbols, and thus the matrix size, can be reduced by more than 50 % after the first iteration in the most challenging LOS channels and more than 70 % in the less-correlated NLOS channels. The EPD iteration can be early terminated within 2 iterations on average. As can be seen in the figure, the EPD with the proposed adaptation scheme employs just-enough processing efforts to meet the target BER.

4.4 EPD Architecture

The overall EPD architecture design is shown in Fig. 4.6. The Gram and $\mathbf{y}^{MF}$ memory buffer incoming channel and 16 match-filtered uplink streams. The memory supports flexible access patterns required for reconfiguration. The MMSE-PIC filter cancels the inter-user interference from the uplink user data. Inside the MMSE-PIC, the Gram matrix is regularized by current estimated variance as in (4.5); then the regularized matrix $\mathbf{A}$ is fed in an LDL decomposition based matrix inversion unit constructed by condensed systolic arrays to calculate $\mathbf{A}^{-1}$. Meantime, the interference contributed by the converged symbol estimates is canceled from the match-filtered $\mathbf{y}^{MF}$ as in (4.15) followed by the regularization as in (4.16). The resulting $\mathbf{y}^{MMSE}$ is filtered by the MMSE filtering matrix $\mathbf{A}^{-1}$ to obtain the MMSE estimation.

The moment-matching unit refines the symbol estimates by incorporating constellation information. The detection control unit dynamically adjusts the per-iteration processing effort and detects early convergence. Updated symbol estimates from the moment-matching unit are buffered in the symbol estimate memory and fed back to the MMSE-PIC filter for iterative refinement. Note that for the downlink data streams, our architecture can also support MIMO precoding by sharing most of the processing and memory units.

In the following subsections, we will elaborate on the implementations of the 2 dominant processing units in the EPD architecture.

4.4.1 Matrix Inversion Architecture

One of the most compute-intensive and accuracy-critical parts of the EPD is the matrix inversion block in the MMSE filter. The matrix inversion architecture is shown in Fig. 4.7 with 4 blocks: First, matrix $\mathbf{A}$ is decomposed to a lower triangular matrix $\mathbf{L}$ and diagonal matrix $\mathbf{D}$ by a systolic array (block 1). The elements of $\mathbf{L}$ and $\mathbf{D}$ are calculated as follows:

$$D_{jj} = A_{jj} - \sum_{k=1}^{j-1} L_{jk} L_{jk}^{*} D_{kk}, \tag{4.17}$$

$$L_{ij} = D_{jj}^{-1} \left(A_{ij} - \sum_{k=i}^{j-1} L_{ik} L_{jk}^{*} D_{kk} \right) \qquad i > j. \tag{4.18}$$

Second, $\mathbf{L}^{-1}$ is computed by another systolic array (block 2); Third, the partial products of $\mathbf{L}^{-1}$ and $\mathbf{D}^{-1}$ in (4.19) are calculated (block 3). Finally, the partial products are accumulated and the inverted matrix $\mathbf{A}^{-1}$ is stored (block 4).

$$A_{ij}^{-1} = \sum_{k=1}^{2N_t} L_{ki}^{-1} D_{kk}^{-1} L_{kj}^{-1} \qquad i \leq j. \tag{4.19}$$

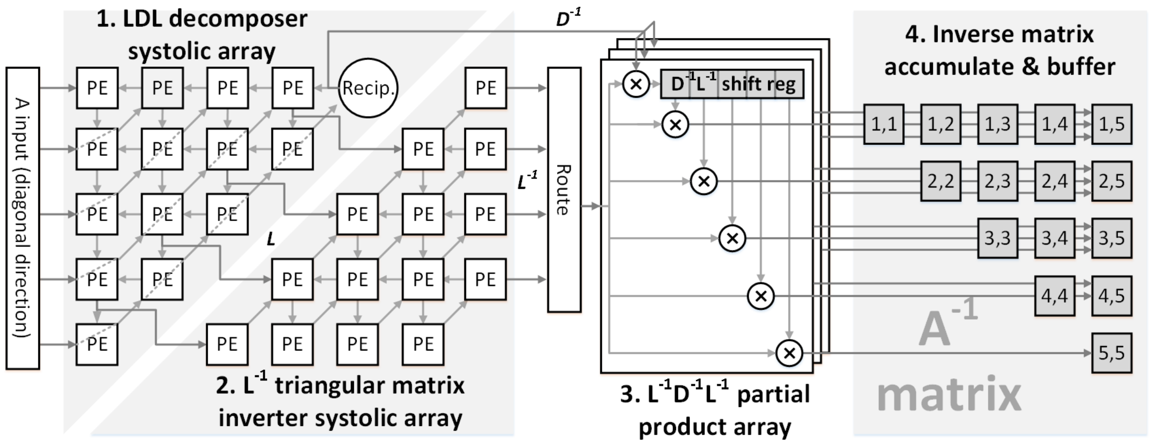


Figure 4.7: Example of a 5 × 5 matrix inversion micro architecture based on systolic arrays.

4.4.1.1 Condensed LDL and L^{-1} Systolic Array

A systolic array is often used to implement the LDL decomposition to realize highly accurate matrix inversion. The systolic array architecture features a regular architecture with an efficient routing and simple control. However, the hardware utilization of a systolic array architecture is only 33.3 % [43] due to the need for zero-padding inputs.

In this work, we implement a condensed LDL systolic array, which merges underutilized PE circuitry to improve the hardware utilization to 90 % for a 16 × 16 array, while reducing the interconnect overhead by more than 70 %. As shown in Fig. 4.8, a PE in a regular systolic array performs division (PE0), multiplication (PE1) or MAC (PE2 and PE3) operations and passes its output to the neighboring PEs. In our condensed systolic array, every three PEs in a row are merged. The merging shortens data movements in the systolic array. Rather than passing data along with many stages of unused operations in a systolic array, our condensed array limits data movements using holding buffers to maximize data reuse. The data reuse is especially advantageous in our design, as it requires a relatively long 28-bit data bit width to support a wide range of channel conditions.

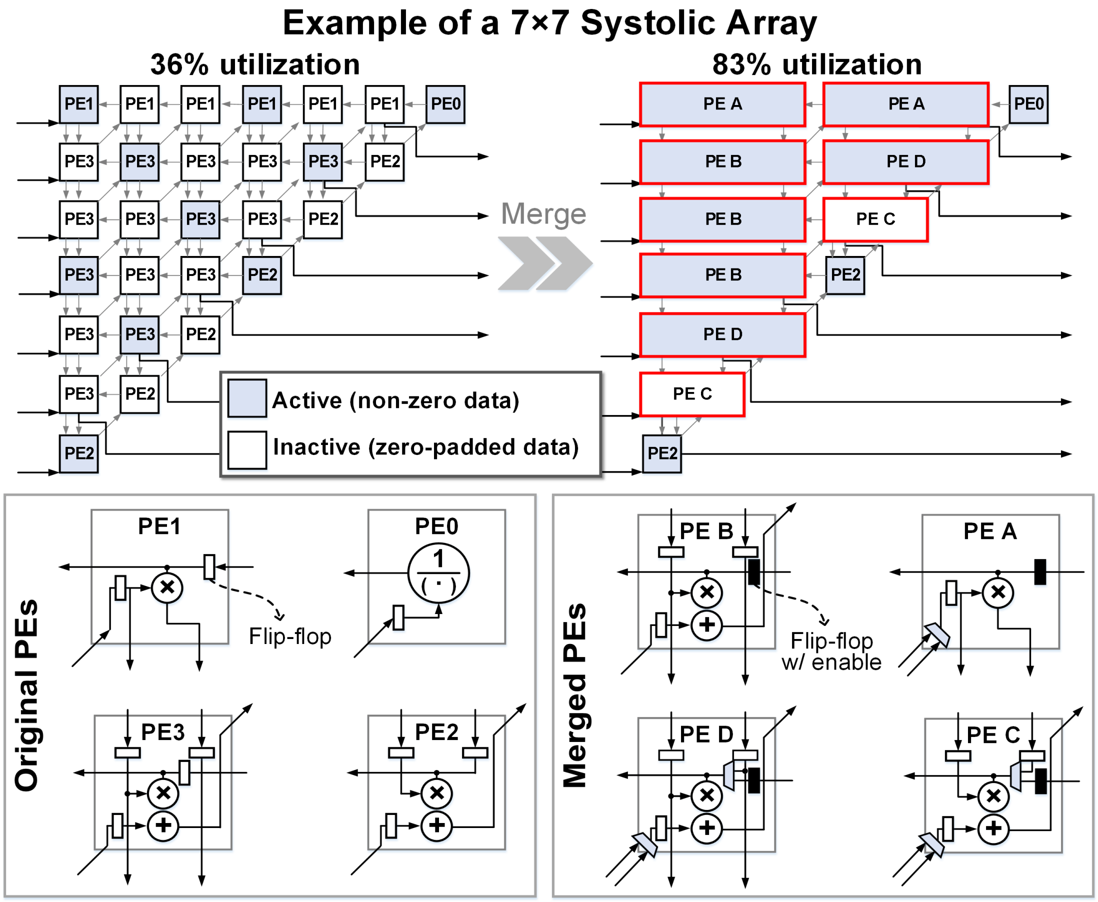


Figure 4.8: Condensed LDL systolic array with enhanced utilization and merged PE designs.

The condensed array architecture reduces silicon area by 62 % compared to the regular systolic array. Moreover, the condensed array shortens data movement delay and provides a larger fraction of a clock period to data processing. The same design method is applied to condense the $\mathbf{L}^{-1}$ systolic array to reduce the area cost and improve its PE utilization.

4.4.1.2 Parallel Partial Product Unit

As block 2 streams out the elements of a row of $\mathbf{L}^{-1}$ (L_{ki}^{-1} for $i = 1...k-1$) one by one, a parallel partial product unit (block3) consumes the data stream right away to compute the partial product terms in (4.19). A set of partial products is represented by $\mathbf{P}_k$:

$$\mathbf{P}_k = \{L_{ki}^{-1} D_{kk}^{-1} L_{kj}^{-1} \mid 1 < i \leq j < k\}. \tag{4.20}$$

For the k-th data stream, i.e. $\{L_{ki}^{-1} \mid i = 1...k-1\}$, the input data L_{ki}^{-1} is multiplied by D_{kk}^{-1} inherently generated from the LDL systolic array and the result is pushed into a shift register; in the meantime, the input L_{ki}^{-1} is multiplied by the elements stored in the shift register in parallel to generate the partial products in $\mathbf{P}_k$.

4.4.1.3 Partial Product Accumulation and Inversion Matrix Buffer

The partial products are sent to the MMSE filter matrix buffer and accumulated to complete the summation in (4.19). The accumulation operation is designed to be near the memory to take advantage of the regular data flow and reduced routing overhead. The final inverted matrix $\mathbf{A}^{-1}$ of size $2N_t \times 2N_t$ serves as the MMSE filtering matrix in the first EPD iteration for every uplink stream within the channel coherent time. Thus, after being calculated for the first stream, $\mathbf{A}^{-1}$ is stored and reused for the remaining data streams within the coherence time. The remaining uplink streams can skip the full-dimension matrix inversion in their first iteration, saving large of energy and latency.

4.4.2 Approximate Moment-Matching (AMM)

The moment-matching unit refines current symbol estimates by matching the moments (mean and variance) of the posteriors in (4.8) and (4.11). The computation

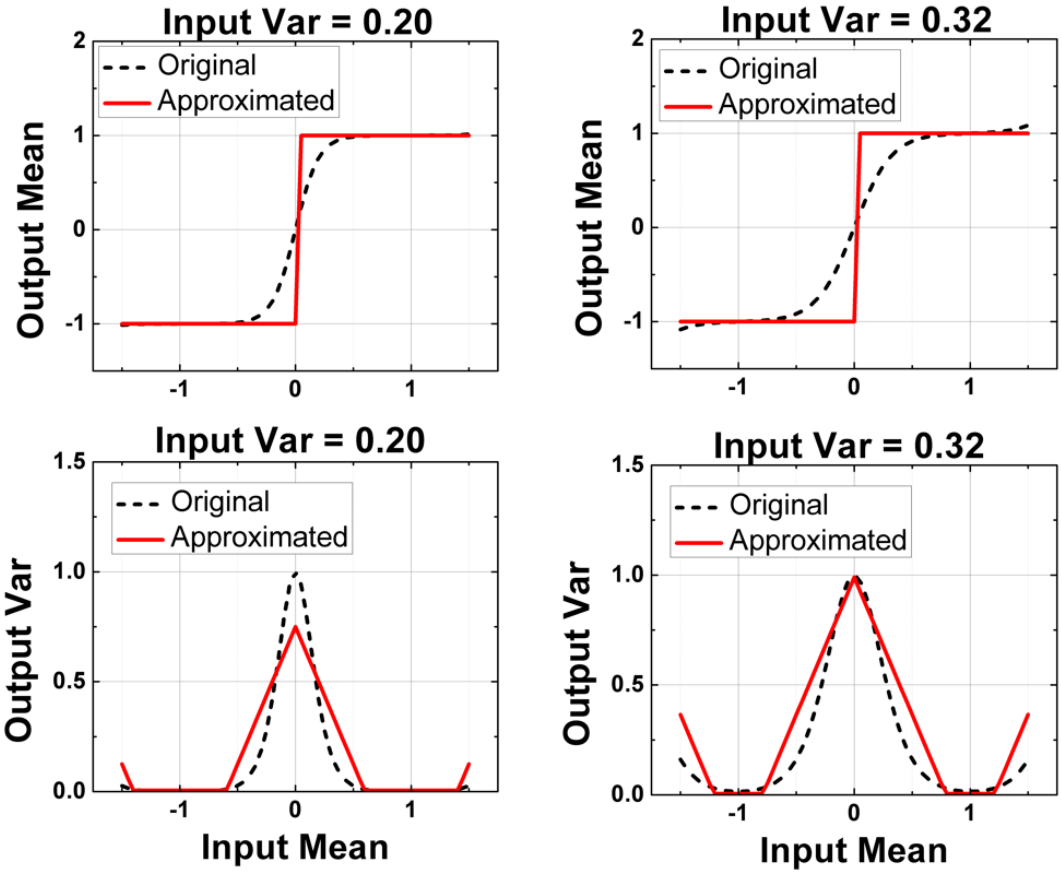


Figure 4.9: The approximations of mean and variance under input variance of 0.2 and 0.32 for example.

requires to calculate the likelihood probability of each constellation point as in (4.10):

$$\mathrm{E}[x_i] = \frac{1}{C} \sum_{s \in \mathcal{A}} s \times \exp\left(\frac{-(s - t_i)^2}{2h_i^2} \right), \tag{4.21}$$

$$\mathrm{Var}[x_i] = \frac{1}{C} \sum_{s \in \mathcal{A}} s^2 \times \exp\left(\frac{-(s - t_i)^2}{2h_i^2} \right) - \mathrm{E}[x_i]^2, \tag{4.22}$$

$$C = \sum_{s \in \mathcal{A}} \exp\left(\frac{-(s - t_i)^2}{2h_i^2} \right). \tag{4.23}$$

The computational complexity is proportional to the product of the modulation size $|\mathcal{A}|$ and the number of simultaneously served users. For a high order modulation and a massive MIMO system, the complexity is prohibitive.

We design an approximate moment-matching (AMM) to make the complexity independent of modulation size by exploiting the symmetry of the QAM constellation in computing the mean and variance estimates. Such an AMM approach is suitable for a flexible detector that supports a wide range of modulation and MIMO sizes. The complexity can be further reduced by a piecewise linear approximation to mean and variance updates:

$$\begin{aligned} \mathrm{E}[x_i] &\approx \mathsf{hard}(t_i) = 2\left[(t_i - 1)/2\right] + 1, \\ \mathrm{Var}[x_i] &\approx P_{00} + P_{10}(1 - |t_i - \mathsf{hard}(t_i)|) + P_{10}h_i, \end{aligned} \tag{4.24}$$

The mean update is reduced to a hard decision of the input soft symbol t_i, where the notation [.] represents nearest integer function which can be implemented by bit slicing; and the variance update is fitted to a first-order polynomial function of the input mean and variance with coefficients P_{00}, P_{01} and P_{10}. The choice of the coefficient values is summarized in Table 4.1 and the corresponding approximation as depicted in Fig. 4.9 sacrifices only 0.5 dB SNR loss based on the simulation.

As shown in Fig. 4.10, compared to a brute-force moment-matching implementation using 2 dividers, 65 MACs, and 16 exponential evaluations, the AMM circuitry uses only 2 MACs. AMM also eliminates costly exponentiation and division, and reduces intermediate bit width requirement. The technique cuts the silicon area of the moment-matching unit by more than 90 %.

Table 4.1: Coefficient Values of the Proposed Approximate Moment Matching (4.24)

Coefficient	Value
P_{00}	0.3501
P_{10}	-1.2500
P_{01}	2.0000

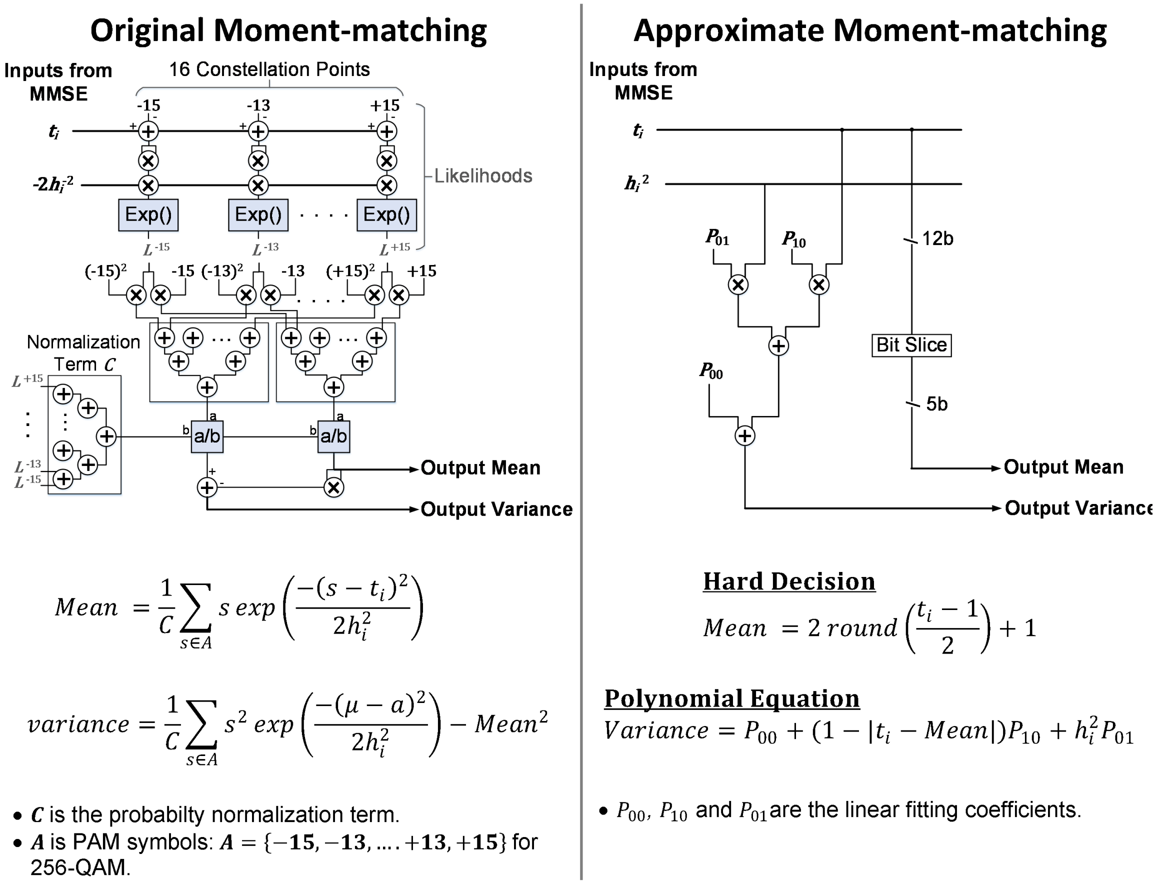

Original Moment-matching
Inputs from MMSE
16 Constellation Points
-15
-13
+15
t_i
$-2h_i^2$
Likelihoods
Exp()
Exp()
Exp()
L^{-15}
L^{-13}
L^{+15}
$(-15)^2$
-15
$(-13)^2$
-13
$(+15)^2$
+15
Normalization Term C
L^{+15}
L^{-13}
L^{-15}
a
b
a/b
a
b
a/b
Output Mean
Output Variance
$Mean = \frac{1}{C}\sum_{s\in A} s\,exp\left(\frac{-(s-t_i)^2}{2h_i^2}\right)$
$variance = \frac{1}{C}\sum_{s\in A} s^2\,exp\left(\frac{-(\mu-a)^2}{2h_i^2}\right) - Mean^2$
C is the probabilty normalization term.
A is PAM symbols: $A = \{-15,-13,\ldots.+13,+15\}$ for 256-QAM.
Approximate Moment-matching
Inputs from MMSE
t_i
h_i^2
P_{01}
P_{10}
12b
Bit Slice
5b
P_{00}
Output Mean
Output Variance
Hard Decision
$Mean = 2\,round\left(\frac{t_i-1}{2}\right)+1$
Polynomial Equation
$Variance = P_{00} + (1-|t_i - Mean|)P_{10} + h_i^2 P_{01}$
P_{00}, P_{10} and P_{01} are the linear fitting coefficients.

Figure 4.10: Circuitry implementations and complexities of the original and approximated moment-matching.

4.5 Chip Measurement Results and Discussion

An EPD test chip is fabricated in ST 28nm UTBB-FDSOI technology, occupying 2.0 mm core area as shown in Fig. 4.11.

4.5.1 Voltage Scaling and Body-biasing

The measurement results at different core voltages and body biasing in room temperature are shown in Fig. 4.12. At a nominal voltage of 1.0 V, the EPD chip runs at 512 MHz, delivering a system throughput of 1.6 Gb/s. By applying forward body biasing of 0.4 V, a maximum working frequency of 569 MHz is achieved, corresponding

Technology	28nm
Core Area	2mm²
Number of BS Antennas (M)	128
Number of Users (K)	≤ 16
QAM size	4, 16, 64, 256
Channel Adaptiveness	i.i.d., NLOS, LOS
System Throughput	1.8Gb/s
Power Consumption	127mW

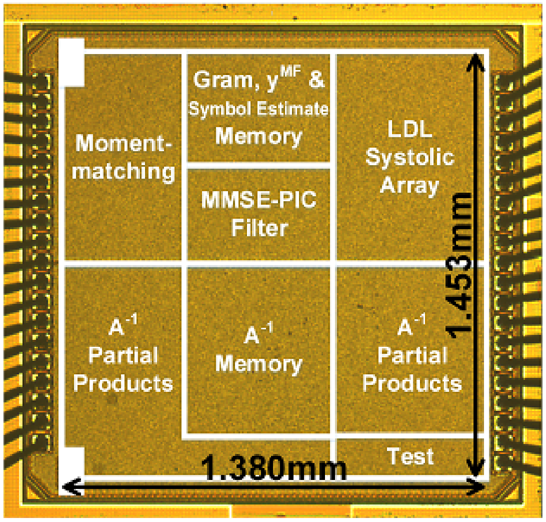


Figure 4.11: Chip features and microphotograph.

to an 11 % boost in detection throughput to 1.8 Gb/s. The corresponding core power consumption is 127 mW, translating to an energy efficiency of 70.6 pJ/b. For a low-power application, reverse body biasing of 0.2 V and voltage scaling of 0.7 V can be applied to reduce the power consumption to 23.4 mW at a throughput of 754 Mb/s.

4.5.2 Comparison

Compared to the prior MIMO detector designs shown in Table. 4.2, our EPD chip provides flexibility in terms of modulation and channel adaptation, supports both uplink and downlink processing, and achieves a high processing gain while maintaining competitive energy and area efficiency. Note that the MPD chip in [40] takes advantage of the assumption of the diagonal dominance in i.i.d. channels using a low-complexity, 13-bit implementation without explicit matrix inversion. However, the MPD encounters an early error floor and fails to provide sufficient processing gain in practical but unfavorable channels such as LOS. In comparison, our EPD chip obtains 4.3 dB processing gain in highly correlated channels, equivalent to a 2.7 times boost in link margin that can be utilized to significantly lower the TX power and

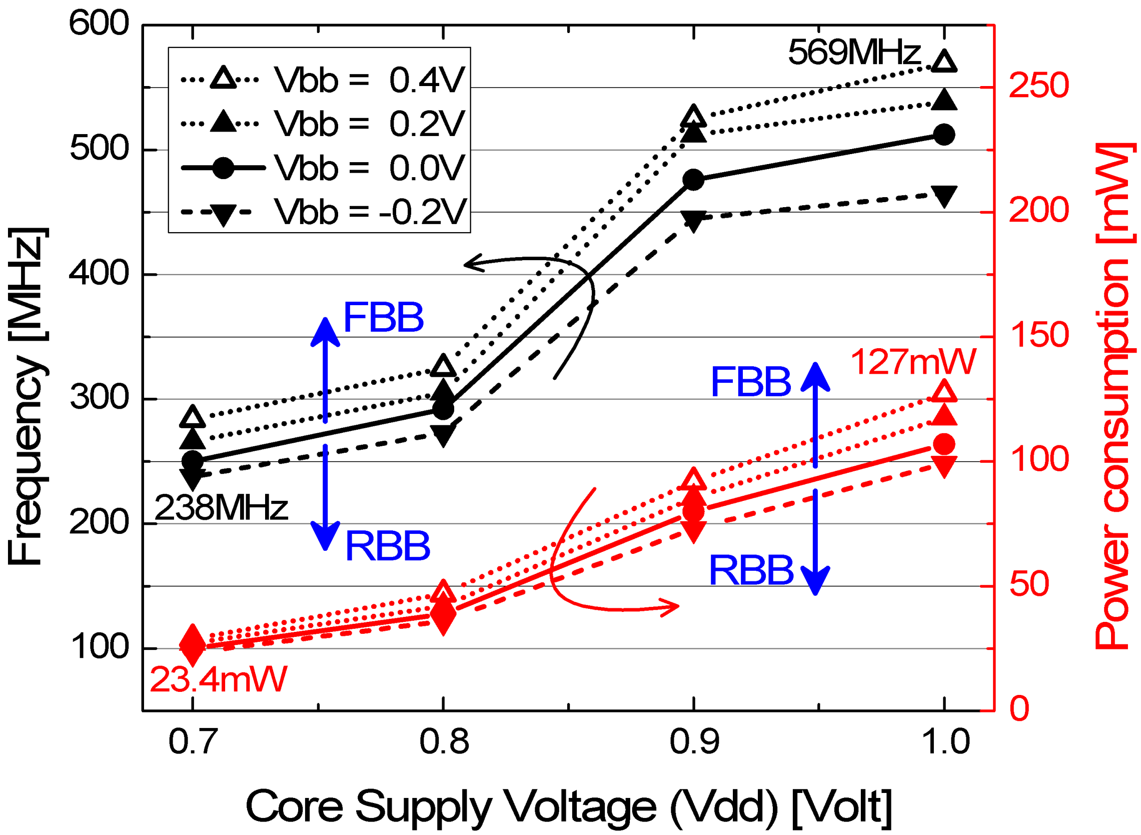


Figure 4.12: Measured frequency and power with different core voltages and body biases in a 128x16 256-QAM LOS channel.

relax the frontend requirements.

4.6 Summary

This work presents a 2.0 mm^2 128 × 16 massive MIMO detector IC that provides 21 dB array gain and 16× multiplexing gain at the system level. The detector implements iterative expectation-propagation detection (EPD) for up to 256-QAM modulation. Tested with measured channel data [39], the detector achieves 4.3dB processing gain over state-of-the-art massive MIMO detectors [40, 41], enabling 2.7× reduction in transmit power for battery-powered mobile terminals. The IC uses link-

Table 4.2: Comparison Table of State-of-the-Art MIMO Detector Designs

		Chen [38]	Tang [40]	Prabhu [41]	This Work
System	Algorithm	MMSE	MPD[a]	MMSE	EPD[b]
	MIMO [$N_r \times N_t$]	4×4	128×32	128×8	128×16
	Modulation	256	256	256	QPSK to 256
	Channel Adaptiveness	no	no	no	yes
	Support Precoding	no	no	yes	yes
	Link Margin Improvement [c]				
	i.i.d. Channel [dB]	0	Error floor [d]	0	0.7
	NLOS Channel [dB]	0	Error floor [d]	0	4.3
	LOS Channel [dB]	0	1.0	0	3.5
Implementation	Technology [nm]	65	40	28	28
	Core Area [mm^2]	0.7	0.58	-	2.0
	Power [mW]	26.5	221	18	127
	Frequency [MHz]	517	425	300	569
	System Throughput[e][Gb/s]	1.38	2.76	0.30	1.80
	Energy Efficiency[f][pJ/b]	307	20	240	70
	Area Efficiency[g][Mb/s/kGE]	0.24	10	0.26	0.5

[a] Message-passing detection.

[b] Expectation-passing detection.

[c] Link margin improvement reflects to SNR gain over MMSE at BER=10^{-3}.

[d] Error floor occurs before BER=10^{-3} in NLOS and LOS channels.

[e] System throughput assumes channel coherence within 7 OFDM symbols and N_t subcarriers.

[f] Energy efficiency is $(Power/Throughput)/(N_t/16)^2$.

[g] Area efficiency is $(Throughput/GateCount) \times (N_t/16)^2$.

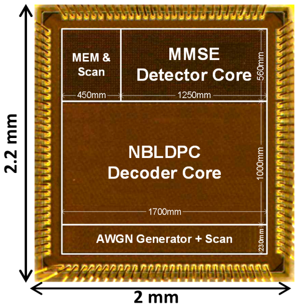
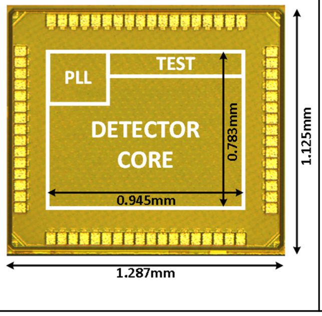
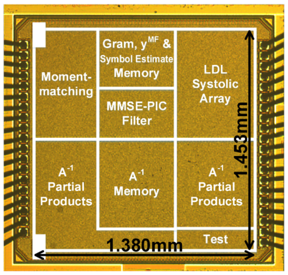
adaptive processing to meet a variety of practical channel conditions with scalable energy consumption. The design is realized in a condensed systolic array architecture and an approximate moment-matching circuitry to reach 1.8 Gb/s at 70.6 pJ/b. The performance and energy efficiency can be tuned over a wide range by UTBB-FDSOI body bias.

CHAPTER V

Conclusion and Outlook

This dissertation presents studies of detector and decoder designs for MIMO systems from small scale (4 × 4) to large scale (128 × 32 and 128 × 16). Table 5.1 and the following subsections summarize the three MIMO detector designs presented in this dissertation.

Table 5.1: Conclusion of the Three MIMO Detector Designs in This Dissertation

	MMSE-NBLDPC Iterative Detector and Decoder	**Message-Passing Detector**	**Expectation-Propagation Detector**
Significance	1st nonbinary IDD 4×4 MIMO	1st Low-complexity Massive MIMO (32 users)	Real-time link-adaptation Massive MIMO (16 users)
Key Novelty	Simplified nonbinary interface	High throughput, low energy	2.7× link margin improvement
Chip Photo	[ISSCC’15] 	[VLSIC’16] 	[ISSCC’18] 

The MMSE-NBLDPC IDD reduces the overall SNR requirement for reliable transmission and simplifies the interface between detector and decoder to reduce the implementation overhead. The test chip is able to achieve more than 1 Gb/s of throughput with a 20 pJ/b of energy efficiency.

The MPD ASIC is the first published massive MIMO detector that can serve up to 32 concurrent users. Taking advantage of the favorable i.i.d. channel property in massive MIMO, the MPD is implemented with very low complexity while achieving near-optimal error rate performance. With the layered scheduling and interleaved architecture, the test chip demonstrates an excellent area efficiency of 4.76 Mb/s/mm^2 and an energy efficiency of 2.49 pJ/b per transmit antenna.

The EPD is able to satisfy SNR requirements in a wide range of real-world massive MIMO channel conditions (tested with measured channels from Lund University). The detector complexity is similar to an MMSE detector, and yet the designed EPD chip can provide over 3 dB of SNR gain over an MMSE detector under the measured channels. Incorporating dynamic dimension reduction, condensed systolic array, and approximate moment-matching, the test chip can adapt to different channel conditions and achieve the lowest processing energy to meet the target error rate requirements.

Based on the research presented by this dissertation, we briefly outline future research topics. In this dissertation, the perfect channel estimation is assumed. To be one step closer towards practice, analysis of the MIMO detection under a certain amount of uncertainty on the CSI should be done in the future.

Another possible direction is to look into the joint design of uplink detection and downlink precoding. Downlink MIMO precoding shares similar processing to the uplink MIMO detection, thus they can share hardware as demonstrated in the EPD work. To fully exploit the degree of freedom offered by massive MIMO, we need to incorporate the concepts of hybrid precoding by co-optimizing precoding in digital and analog domain in the future.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] C. Studer, S. Fateh, and D. Seethaler. Asic implementation of soft-input soft-output mimo detection using mmse parallel interference cancellation. *IEEE Journal of Solid-State Circuits*, 46(7):1754–1765, July 2011.

[2] Cisco visual networking index: Global mobile data traffic forecast update, 20162021 white paper. `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html`, Mar 2017.

[3] Verizon. State of the market: Internet of things 2016. `https://www.verizon.com`, 2016. [Online; accessed 30-December-2017].

[4] Ericsson. Mobility report - on the pulse of the networked society. `https://www.ericsson.com`, 2017.

[5] N. Chiurtu, B. Rimoldi, and E. Telatar. On the capacity of multi-antenna gaussian channels. In *Proceedings. 2001 IEEE International Symposium on Information Theory (IEEE Cat. No.01CH37252)*, pages 53–, 2001.

[6] Tzi-Dar Chiueh, Pei-Yun Tsai, and I-Wei Lai. *Baseband Receiver Design for Wireless MIMO-OFDM Communications*. Wiley-IEEE Press, 2nd edition, 2012.

[7] Erik Dahlman, Stefan Parkvall, Johan Skold, and Per Beming. *3G evolution: HSPA and LTE for mobile broadband*. Academic press, 2010.

[8] 3GPP. 3gpp standards website. *[Online]. Available: `http://www.3gpp.org`.*

[9] F. Sheikh, C. H. Chen, D. Yoon, B. Alexandrov, K. Bowman, A. Chun, H. Alavi, and Z. Zhang. 3.2gbps channel-adaptive configurable mimo detector for multi-mode wireless communication. In *2014 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 1–6, Oct 2014.

[10] T. Marzetta. Noncooperative cellularwireless with unlimited numbers of base station antennas. *IEEE Transactions on Wireless Communications*, 9:35903600, Nov 2010.

[11] B. Hassibi and H. Vikalo. On the sphere-decoding algorithm i. expected complexity. *IEEE Transactions on Signal Processing*, 53(8):2806–2818, Aug 2005.

[12] Zhan Guo and P. Nilsson. Algorithm and implementation of the k-best sphere decoding for mimo detection. *IEEE Journal on Selected Areas in Communications*, 24(3):491–503, March 2006.

[13] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei. Vlsi implementation of mimo detection using the sphere decoding algorithm. *IEEE Journal of Solid-State Circuits*, 40(7):1566–1577, July 2005.

[14] S. Chen, T. Zhang, and Y. Xin. Relaxed$k$-best mimo signal detector design and vlsi implementation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(3):328–337, March 2007.

[15] T. Datta, N. Srinidhi, A. Chockalingam, and B. S. Rajan. Random-restart reactive tabu search algorithm for detection in large-mimo systems. *IEEE Communications Letters*, 14(12):1107–1109, December 2010.

[16] N. Srinidhi, T. Datta, A. Chockalingam, and B. S. Rajan. Layered tabu search algorithm for large-mimo detection and a lower bound on ml performance. *IEEE Transactions on Communications*, 59(11):2955–2963, November 2011.

[17] B. Noethen, O. Arnold, E. P. Adeva, T. Seifert, E. Fischer, S. Kunze, E. Mat, G. Fettweis, H. Eisenreich, G. Ellguth, S. Hartmann, S. Hppner, S. Schiefer, J. U. Schller, S. Scholze, D. Walter, and R. Schffny. 10.7 a 105gops 36mm2 heterogeneous sdr mpsoc with energy-aware dynamic scheduling and iterative detection-decoding for 4g in 65nm cmos. In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 188–189, Feb 2014.

[18] F. Borlenghi, E. M. Witte, G. Ascheid, H. Meyr, and A. Burg. A 2.78 mm2 65 nm cmos gigabit mimo iterative detection and decoding receiver. In *2012 Proceedings of the ESSCIRC (ESSCIRC)*, pages 65–68, Sept 2012.

[19] M. Winter, S. Kunze, E. P. Adeva, B. Mennenga, E. Mats, G. Fettweis, H. Eisenreich, G. Ellguth, S. Hppner, S. Scholze, R. Schffny, and T. Kobori. A 335mb/s 3.9mm2 65nm cmos flexible mimo detection-decoding engine achieving 4g wireless data rates. In *2012 IEEE International Solid-State Circuits Conference*, pages 216–218, Feb 2012.

[20] M. C. Davey and D. MacKay. Low-density parity check codes over gf(q). *IEEE Communications Letters*, 2(6):165–167, June 1998.

[21] D. Declercq and M. Fossorier. Decoding algorithms for nonbinary ldpc codes over gf(q). *IEEE Transactions on Communications*, 55(4):633–643, April 2007.

[22] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard. Low-complexity decoding for non-binary ldpc codes in high order fields. *IEEE Transactions on Communications*, 58(5):1365–1375, May 2010.

[23] Y. S. Park, Y. Tao, and Z. Zhang. A fully parallel nonbinary ldpc decoder with fine-grained dynamic clock gating. *IEEE Journal of Solid-State Circuits*, 50(2):464–475, Feb 2015.

[24] S. Pfletschinger and D. Declercq. Getting closer to mimo capacity with non-binary codes and spatial multiplexing. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pages 1–5, Dec 2010.

[25] Henk Wymeersch, Heidi Steendam, and Marc Moeneclaey. Log-domain decoding of LDPC codes over GF(q). In *2004 IEEE International Conference on Communications*, volume 2, pages 772–776 Vol.2, June 2004.

[26] D Declercq, M Colas, and G Gelle. Regular GF (2q)-LDPC modulations for higher order QAM-AWGN channels. In *International Symposium on Information Theory and its applications ISITA, Parma, Italia*, 2004.

[27] J. Erfanian, S. Pasupathy, and G. Gulak. Reduced complexity symbol detectors with parallel structure for isi channels. *IEEE Transactions on Communications*, 42(234):1661–1671, Feb 1994.

[28] I. B. Collings, M. R. G. Butler, and M. McKay. Low complexity receiver design for mimo bit-interleaved coded modulation. In *Spread Spectrum Techniques and Applications, 2004 IEEE Eighth International Symposium on*, pages 12–16, Aug 2004.

[29] D. Auras, R. Leupers, and G. H. Ascheid. A novel reduced-complexity soft-input soft-output mmse mimo detector: Algorithm and efficient vlsi architecture. In *2014 IEEE International Conference on Communications (ICC)*, pages 4722–4728, June 2014.

[30] E. Boutillon and L. Conde-Canencia. Bubble check: a simplified algorithm for elementary check node processing in extended min-sum non-binary ldpc decoders. *Electronics Letters*, 46(9):633–634, April 2010.

[31] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson. Scaling up mimo: Opportunities and challenges with very large arrays. *IEEE Signal Processing Magazine*, 30(1):40–60, Jan 2013.

[32] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta. Energy and spectral efficiency of very large multiuser mimo systems. *IEEE Transactions on Communications*, 61(4):1436–1449, April 2013.

[33] J. Hoydis, S. ten Brink, and M. Debbah. Massive mimo in the ul/dl of cellular networks: How many antennas do we need? *IEEE Journal on Selected Areas in Communications*, 31(2):160–171, February 2013.

[34] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta. Massive mimo for next generation wireless systems, ieee communications magazine. *IEEE Intl. Solid-State Circuits Conf.*, 52:186195, Feb 2014.

[35] S. Yang and L. Hanzo. Fifty years of mimo detection: The road to large-scale mimos. *IEEE Communications Surveys Tutorials*, 17(4):1941–1988, 2015.

[36] B. Yin, M. Wu, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer. A 3.8gb/s large-scale mimo detector for 3gpp lte-advanced. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3879–3883, May 2014.

[37] T. L. Narasimhan and A. Chockalingam. Channel hardening-exploiting message passing (chemp) receiver in large-scale mimo systems. *IEEE Journal of Selected Topics in Signal Processing*, 8(5):847–860, Oct 2014.

[38] C.-H. Chen, W. Tang, and Z. Zhang. 18.7 a 2.4mm2 130mw mmse-nonbinary-ldpc iterative detector-decoder for 4 x 4 256-qam mimo in 65nm cmos. In *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, pages 1–3, Feb 2015.

[39] S. Malkowsky, J. Vieira, L. Liu, P. Harris, K. Nieman, N. Kundargi, I. C. Wong, F. Tufvesson, V. wall, and O. Edfors. The world's first real-time testbed for massive mimo: Design, implementation, and validation. *IEEE Access*, 5:9073–9088, 2017.

[40] Wei Tang, C. H. Chen, and Zhengya Zhang. A 0.58mm2 2.76gb/s 79.8pj/b 256-qam massive mimo message-passing detector. In *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, pages 1–2, June 2016.

[41] H. Prabhu, J. N. Rodrigues, L. Liu, and O. Edfors. A 60pj/b 300mb/s 128 × 8 massive mimo precoder-detector in 28nm fd-soi. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 60–61, Feb 2017.

[42] J. Cépedes, P. M. Olmos, M. Sánchez-Fernández, and F. Perez-Cruz. Expectation propagation detection for high-order high-dimensional mimo systems. *IEEE Transactions on Communications*, 62(8):2840–2849, Aug 2014.

[43] S. J. Bellis, W. P. Marnane, and P. J. Fish. Alternative systolic array for non-square-root cholesky decomposition. *IEE Proceedings - Computers and Digital Techniques*, 144(2):57–64, Mar 1997.

[44] Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pages 1–3534, Dec 2016.

[45] V. Erceg et al. Tgn channel models. *IEEE 802.11 document 03/940r4*, May 2004.

[46] A. Tomasoni, M. Ferrari, D. Gatti, F. Osnato, and S. Bellini. A low complexity turbo mmse receiver for w-lan mimo systems. In *2006 IEEE International Conference on Communications*, volume 9, pages 4119–4124, June 2006.

[47] S. A. Laraway and B. Farhang-Boroujeny. Implementation of a markov chain monte carlo based multiuser/mimo detector. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(1):246–255, Jan 2009.

[48] Xiaodong Wang and H. V. Poor. Iterative (turbo) soft interference cancellation and decoding for coded cdma. *IEEE Transactions on Communications*, 47(7):1046–1061, Jul 1999.

[49] R. Yazdani and M. Ardakani. Efficient llr calculation for non-binary modulations over fading channels. *IEEE Transactions on Communications*, 59(5):1236–1241, May 2011.

[50] H. Kaul, M. Anders, S. Mathew, S. Hsu, A. Agarwal, F. Sheikh, R. Krishnamurthy, and S. Borkar. A 1.45ghz 52-to-162gflops/w variable-precision floating-point fused multiply-add unit with certainty tracking in 32nm cmos. In *2012 IEEE International Solid-State Circuits Conference*, pages 182–184, Feb 2012.

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

[52] E. Bjrnson, J. Hoydis, M. Kountouris, and M. Debbah. Massive mimo systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits. *IEEE Transactions on Information Theory*, 60(11):7112–7139, Nov 2014.

[53] E. Bjrnson, E. G. Larsson, and T. L. Marzetta. Massive mimo: ten myths and one critical question. *IEEE Communications Magazine*, 54(2):114–123, February 2016.

[54] X. Gao, O. Edfors, F. Rusek, and F. Tufvesson. Massive mimo performance evaluation based on measured propagation data. *IEEE Transactions on Wireless Communications*, 14(7):3899–3911, July 2015.

[55] C. Zhang, L. Liu, D. Markovic, and V. wall. A heterogeneous reconfigurable cell array for mimo signal processing. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62:733742, March 2015.

[56] A. Peled and A. Ruiz. Frequency domain data transmission using reduced computational complexity algorithms. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 5:964967, April 1980.

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

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

[59] X. Chen, G. He, and J. Ma. Vlsi implementation of a high-throughput iterative fixed-complexity sphere decoder. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 60(5):272–276, May 2013.

[60] S. Wu, L. Kuang, Z. Ni, J. Lu, D. Huang, and Q. Guo. Low-complexity iterative detection for large-scale multiuser mimo-ofdm systems using approximate message passing. *IEEE Journal of Selected Topics in Signal Processing*, 8(5):902–915, Oct 2014.

[61] S. Song, B. Zhou, S. Lin, and K. Abdel-Ghaffar. A unified approach to the construction of binary and nonbinary quasi-cyclic ldpc codes based on finite fields. *IEEE Transactions on Communications*, 57(1):84–93, January 2009.

[62] B. Zhou, J. Kang, S. W. Song, S. Lin, K. Abdel-Ghaffar, and M. Xu. Construction of non-binary quasi-cyclic ldpc codes by arrays and array dispersions - [transactions papers]. *IEEE Transactions on Communications*, 57(6):1652–1662, June 2009.

[63] Y. Tao, Y. S. Park, and Z. Zhang. High-throughput architecture and implementation of regular (2, dc) nonbinary ldpc decoders. In *2012 IEEE International Symposium on Circuits and Systems*, pages 2625–2628, May 2012.

[64] X. Zhang, F. Cai, and S. Lin. Low-complexity reliability-based message-passing decoder architectures for non-binary ldpc codes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(11):1938–1950, Nov 2012.

[65] X. Chen and C. L. Wang. High-throughput efficient non-binary ldpc decoder based on the simplified min-sum algorithm. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 59(11):2784–2794, Nov 2012.

[66] Y. L. Ueng, K. H. Liao, H. C. Chou, and C. J. Yang. A high-throughput trellis-based layered decoding architecture for non-binary ldpc codes using max-log-qspa. *IEEE Transactions on Signal Processing*, 61(11):2940–2951, June 2013.

[67] J. Lin and Z. Yan. An efficient fully parallel decoder architecture for nonbinary ldpc codes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(12):2649–2660, Dec 2014.

[68] C. L. Lin, C. L. Chen, H. C. Chang, and C. Y. Lee. Jointly designed nonbinary ldpc convolutional codes and memory-based decoder architecture. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(10):2523–2532, Oct 2015.

[69] Network TSGRANGRA. Evolved universal terrestrial radio access (e-utra); multiplexing and channel coding. *3rd Generation Partnership Project (3GPP), vol. TS*, 36, 2009.

[70] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, volume 2, pages 1064–1070 vol.2, May 1993.

[71] E. Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, July 2009.

[72] B. Tahir, S. Schwarz, and M. Rupp. Ber comparison between convolutional, turbo, ldpc, and polar codes. In *2017 24th International Conference on Telecommunications (ICT)*, pages 1–7, May 2017.

[73] A. Balatsoukas-Stimming, P. Giard, and A. Burg. Comparison of polar decoders with existing low-density parity-check and turbo decoders. In *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–6, March 2017.

[74] E. Ankan, N. ul Hassan, M. Lentmaier, G. Montorsi, and J. Sayir. Challenges and some new directions in channel coding. *Journal of Communications and Networks*, 17(4):328–338, Aug 2015.

[75] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, Aug 2002.

[76] M. O. Damen, H. El Gamal, and G. Caire. On maximum-likelihood detection and the search for the closest lattice point. *IEEE Transactions on Information Theory*, 49(10):2389–2402, Oct 2003.

[77] E. M. Witte, F. Borlenghi, G. Ascheid, R. Leupers, and H. Meyr. A scalable vlsi architecture for soft-input soft-output single tree-search sphere decoding. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 57(9):706–710, Sept 2010.

[78] L. Liu. Energy-efficient soft-input soft-output signal detector for iterative mimo receivers. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(8):2422–2432, Aug 2014.

[79] B. Yin, M. Wu, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer. A 3.8gb/s large-scale mimo detector for 3gpp lte-advanced. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3879–3883, May 2014.

[80] L. Dai, X. Gao, X. Su, S. Han, C. I, and Z. Wang. Low-complexity soft-output signal detection based on gaussseidel method for uplink multiuser large-scale mimo systems. *IEEE Transactions on Vehicular Technology*, 64(10):4839–4845, Oct 2015.

[81] C. Zhang, Z. Wu, C. Studer, Z. Zhang, and X. You. Efficient soft-output gauss-seidel data detector for massive mimo systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pages 1–12, 2018.

[82] X. Gao, L. Dai, Y. Ma, and Z. Wang. Low-complexity near-optimal signal detection for uplink large-scale mimo systems. *Electronics Letters*, 50(18):1326–1328, August 2014.

[83] T. Xie, L. Dai, X. Gao, X. Dai, and Y. Zhao. Low-complexity ssor-based precoding for massive mimo systems. *IEEE Communications Letters*, 20(4):744–747, April 2016.

[84] B. Yin, M. Wu, J. R. Cavallaro, and C. Studer. Conjugate gradient-based soft-output detection and precoding in massive mimo systems. In *2014 IEEE Global Communications Conference*, pages 3696–3701, Dec 2014.

[85] B. Yin, M. Wu, J. R. Cavallaro, and C. Studer. Vlsi design of large-scale soft-output mimo detection using conjugate gradients. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1498–1501, May 2015.