A Thesis for the Degree of Doctor

Architecture Analysis and Design of the Platform Based Wireless Communication Systems

Gyongsu Lee

School of Engineering

Information and Communications University

2005

Architecture Analysis and Design of the Platform Based Wireless Communication Systems

Architecture Analysis and Design of the Platform Based Wireless Communication Systems

Advisor : Professor Sin-Chong Park

by

Gyongsu Lee

School of Engineering

Information and Communications University

A thesis submitted to the faculty of Information and Communications University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

> Daejeon, Korea Jun. 30. 2005 Approved by

Professor Sin-Chong Park Major Advisor

Architecture Analysis and Design of the Platform Based Wireless Communication Systems

Gyongsu Lee

I certify that this work has passed the scholastic standards requested by the Information and Communications University as a thesis for the degree of Doctor of Philosophy

Jun. 30. 2005

Advisory committee	major		
	membe	r Hae-Wook Choi	
	membe	r Hyung-Joun Yoo	
	membe	r Youngnam Han	
	membe	r Giwan Yoon	

Ph.D. Gyongsu Lee

2000060

Architecture Analysis and Design of the Platform Based Wireless Communication Systems

School of Engineering, 2005, 134p. Major Advisor: Prof. Sin-Chong Park. Text in English

ABSTRACT

In this dissertation, the super-iterative LDPC-coded MIMO-OFDM system is analyzed, designed and implemented as a target system. The receiver performance and the low-power designing techniques are summarized and applied. Based on the Monte-Carlo simulation results, the throughput of the target system is evaluated for different system parameters such as the modulation order, number of antennas, and receiver configurations. The receiver configurations including the MIMO detection scheme, LDPC decoding scheme, and number of iterations are analyzed. The simulation results show that the throughput can be enhanced when the modulation order grows with a SNR at the receiver side and a larger number of antennas are utilized. Furthermore, it is found that using iterative detection and decoding with sufficiently large number of iterations, more than 4 iterations, increase the throughput.

The extracted parameters by system simulation are applied to actual hardware design. Moreover, the low-power digital hardware design techniques, such as clock gating, operand isolation, and low-power cell replacement, are analyzed and applied to the designed system. By these techniques, total power reduction of 80 % is achieved.

Contents

PREFA	СЕ	1
1. INTR	ODUCTION	3
1.1	MOTIVATION	
1.2	CONTRIBUTIONS AND ORGANIZATION	5
2. PLAT	FORM BASED WIRELESS SOC DESIGN	6
2.1	INTRODUCTION	6
2.2	CHARACTERISTICS OF WIRELESS SOC SYSTEM	8
2.2.	1 Unidirectional Traffic	
2.2.	2 Data Format Conversions	
2.2.	3 Critical Requirements for Latency and Throughput	9
2.3	LATENCY-THROUGHPUT COMPUTATION	
2.3.	1 For Single Processor	
2.3.	2 For Multi-Processor using Simple Protocol	
2.3.	<i>3</i> For Multi-Processor using Synchronous Protocol	
2.4	DESIGN CONSIDERATIONS	13
2.4.	1 Frequency Setting Flow	
2.4.	2 Dedicated FIFO Channel	14
2.5	EXAMPLE OF DERIVATIVE SYSTEM	16
2.5.	1 802.11a System Overview	
2.5.	2 For Single Processor	
2.5.	<i>3</i> For Multi-Processor using Simple Protocol	
2.5.	4 For Multi-Processor using Synchronous Protocol	
2.6	PLATFORM BOARD DESIGN	
3. OVE	RVIEW OF THE MIMO-WLAN SYSTEMS	24
3.1	INTRODUCTION	24
3.2	PROMISING PROPOSALS FOR THE MIMO-WLAN SYSTEMS	

3.2	.1 TGn Sync	27
3.2	.2 WWiSE (World-Wide Spectrum Efficiency)	27
3.3	MIMO SCHEMES FOR THE MIMO-WLAN SYSTEM	28
3.3	.1 Open-loop MIMO Schemes	
3.3	.2 Closed-loop MIMO Schemes	34
4. SUPI	ER ITERATIVE DECODING BETWEEN MIMO DETECTOR AND	
CHAN	NEL DECODER	
4.1	INTRODUCTION	
4.2	MIMO DETECTION ALGORITHM	
4.3	ITERATIVE DEMODULATION AND DECODING STRUCTURE	
4.4	SYSTEM SIMULATION RESULTS OF CODED MIMO-OFDM	
5. HAR	DWARE BLOCK DESIGN OF LDPC CODED MIMO-OFDM	58
5.1	INTRODUCTION	
5.2	System Requirements	
5.3	HARDWARE COMPONENTS DESIGN	62
5.3	.1 LSD Design	62
5.3	.2 LDPCC Codec Design	72
5.4	ADAPTING HARDWARE OPTIMIZING TECHNIQUES	82
5.5	IMPLEMENTATION ON PLATFORM BOARD	85
6. TRA	DE-OFFS OF PERFORMANCE AND COST	86
6.1	INTRODUCTION	
6.2	HARDWARE RESOURCE ESTIMATION	
6.3	TRADE-OFFS OF THROUGHPUT AND HARDWARE RESOURCE AND POWER	90
7. LOW	POWER DESIGN FOR SOC DIGITAL DESIGN	94
7.1	INTRODUCTION	94
7.2	TECHNIQUES FOR THE POWER CONSTRAINTS	94
7.2	.1 Clock Gating	94

7.2.2	Operand Isolation	
7.2.3	Low-power cell replacement	
7.3 Pc	OWER MODELING AND CALCULATION	
7.3.1	Static and dynamic power dissipations	
7.3.2	Power calculation	
7.4 A	PPLYING LOW-POWER TECHNIQUES	
7.4.1	Low power design for LSD-MIMO block	
7.4.2	Summary of adapting low power technique	116
8. CONCLU	USIONS	117
REFEREN	CES	
ACKNOW	LEDGEMENTS	
CURRICU	LUM VITAE	

Figure Contents

FIGURE 2.1 MULTI-PROCESSOR OPERATING TIMING WITH SIMPLE	
PROTOCOL	12
FIGURE. 2.2 MULTI-PROCESSOR OPERATING TIMING WITH	
SYNCHRONOUS PROTOCOL	13
FIGURE 2.3 DESIGN PROCEDURE FOR FREQUENCY SETTING	15
FIGURE 2.4 MINIMUM APPLICABLE BLOCK SIZE OF EACH BLOCK AND)
ASSIGNING BLOCKS TO 3 PROCESSORS	17
FIGURE 2.5 BLOCK DIAGRAM FOR CANDIDATE SYSTEM WITH 3	
PROCESSING UNIT	22
FIGURE 2.6 BOARD LAYOUT AND ACTUAL PHOTO OF PLATFORM BOA	RD 23
FIGURE 3.1 CHANNEL CAPACITY OF THE MIMO-WLAN SYSTEMS	26
FIGURE 3.2 SPATIAL DIVISION MULTIPLEXING	29
FIGURE 3.3 V-BLAST DETECTION (COMBINED WITH ZF)	
FIGURE 3.4 PERFORMANCE COMPARISON OF SDM TRANSMISSION	
$(N_t = N_r = 2)$	32
FIGURE 3.5 SPACE-TIME BLOCK CODING ($N_t = 4$)	33
FIGURE 3.6 TRANSMIT BEAMFORMING	35
FIGURE 3.7 ILLUSTRATION OF TRANSMIT BEAMFORMING	35
FIGURE 3.8 ILLUSTRATION OF WATER-FILLING PRINCIPLE	36
FIGURE 4.1 MIMO CHANNEL MODEL.	38
FIGURE 4.2 PERFORMANCE COMPARISON FOR MIMO DETECTION	
ALGORITHMS (2 TX ANTENNAS AND QPSK MODULATION)	41
FIGURE 4.3 ITERATIVE DEMODULATION AND DECODING	43
FIGURE 4.4 THE REFERENCE PER CURVES USING LDPCC AND	
CONVOLUTION CODE BY WWISE	49

FIGURE 4.5 THE REFERENCE PER CURVES USING LDPCC BY TGN SYNC 50
FIGURE 4.6 THE PER PERFORMANCE CURVES OF 2X2 QPSK
FIGURE 4.7 THE PER PERFORMANCE CURVES OF 2X2 16QAM52
FIGURE 4.8 THE PER PERFORMANCE CURVES OF 2X2 64QAM53
FIGURE 4.9 THE PER PERFORMANCE CURVES OF 3X3 QPSK53
FIGURE 4.10 THE PER PERFORMANCE CURVES OF 3X3 16QAM54
FIGURE 4.11 THE PER PERFORMANCE CURVES OF 3X3 64QAM54
FIGURE 4.12 THE PER PERFORMANCE CURVES OF 4X4 QPSK55
FIGURE 4.13 THE PER PERFORMANCE CURVES OF 4X4 16QAM55
FIGURE 4.14 THE PER PERFORMANCE CURVES OF 4X4 64QAM56
FIGURE 4.15 THE PER PERFORMANCE COMPARISON OF 2X2 QPSK FOR
SLOW AND FAST BLOCK FADING
FIGURE 4.16 THE PER PERFORMANCE COMPARISON OF 2X2 64QAM FOR
SLOW AND FAST BLOCK FADING
FIGURE 5.1 BLOCK INTERFACE WITH EACH DATA SIZE
FIGURE 5.2 TIMING DIAGRAM FOR EACH BLOCK OPERATION61
FIGURE 5.3 RELATIONSHIP BETWEEN COMPONENTS FOR ITERATIVE
DETECTION AND DECODING
FIGURE 5.4 BLOCK DIAGRAM OF LSD
FIGURE 5.5 ARCHITECTURE OF QR DECOMPOSITION
FIGURE 5.6 OPERATION OF THE VECTORING MODE
FIGURE 5.7 OPERATION OF THE ROTATION MODE
FIGURE 5.8 TREE SEARCHING OF THE SPHERE DECODING FOR 2X2 QPSK 69
FIGURE 5.9 THE CORES OF TREE SEARCHING AND NODE EVALUATION 71
FIGURE 5.10 TIMING DIAGRAM OF SEARCHING OPERATION
FIGURE 5.11 REPRESENTATIONS OF MATRICES

FIGURE 5.12 BLOCK DIAGRAM OF BIT-TO-CHECK CALCULATION UNIT 79
FIGURE 5.13 BLOCK DIAGRAM OF CHECK-TO-BIT CALCULATION
FIGURE 5.14 GRAPHS OF $\tanh(x/2)$ AND $\phi(x)$
FIGURE 5.15 TOP BLOCK DIAGRAM OF LDPCC DECODER
FIGURE 5.16 MEASUREMENT FOR THE 4X4 LSD MIMO BLOCK IN A PLATFORM FPGA
FIGURE 6.1 PACKET ERROR RATE COMPARISON FOR EACH SYSTEM COMBINATION
FIGURE 6.2 REQUIRED SNR AND NUMBER OF GATES FOR EACH SYSTEM COMBINATION
FIGURE 6.3 NORMALIZED COMPARISONS ON THE BASIS OF SYSTEM CONFIGURATION 1
FIGURE 7.1 TRADITIONAL AND GATED CLOCK
FIGURE 7.2 OPERAND ISOLATION
FIGURE 7.3 SAMPLE HIGH FAN-OUT GATE STRUCTURE FOR CALCULATING OPTIMUM TRANSISTOR SIZE
FIGURE 7.4 COMPONENTS OF POWER DISSIPATION (CMOS INVERTER) 102
FIGURE 7.5 LEAKAGE POWER CALCULATION EXAMPLE 103
FIGURE 7.6 TOGGLE REPORT EXAMPLE IN SAIF 105
FIGURE 7.7 CELL LIBRARY EXAMPLE OF SMIC 0.18 106
FIGURE 7.8 POWER DISSIPATION RATIO FOR THE SUB-BLOCKS OF 4X4 LSD-MIMO
FIGURE 7.9 OPERATION OF REGISTER BANK AFTER AND BEFORE ADAPTING GATING CLOCK
FIGURE 7.10 ADAPTING OPERAND ISOLATION TO ZF_EST_4X4 BLOCK 113
FIGURE 7.11 TOTAL POWER DISSIPATION SUMMARY FOR EACH REDUCTION TECHNIQUE

Table Contents

TABLE 2.1 ABBREVIATION FOR NOTATING THE LATENCY AND	
THROUGHPUT	7
TABLE 3.1 THROUGHPUT ENHANCEMENT OF TGN SYNC	27
TABLE 3.2 THROUGHPUT ENHANCEMENT OF WWISE	28
TABLE 4.1 PROPOSING SYSTEM CONFIGURATIONS	51
TABLE 5.1 REQUIRED THROUGHPUT FOR GIVEN SYSTEM	59
TABLE 5.2 RESOURCE ANALYSIS OF BIT NODE TO CHECK NODE	
CALCULATION	83
TABLE 5.3 RESOURCE ANALYSIS OF CHECK NODE TO BIT NODE	
CALCULATION	83
TABLE 5.4 RESOURCE ANALYSIS OF LDPCC DECODING (FOR 1	
ITERATION)	84
TABLE 6.1 RESOURCES AND PERFORMANCE OF FFT	86
TABLE 6.2 RESOURCES AND PERFORMANCE OF LSD	87
TABLE 6.3 RESOURCES AND PERFORMANCE OF 1 ST MIMO	87
TABLE 6.4 RESOURCES AND PERFORMANCE OF 2^{ND} MIMO	88
TABLE 6.5 RESOURCES AND PERFORMANCE OF LDPCC DECODER	88
TABLE 6.6 RESOURCES FOR 2X2 QPSK USING MMSE DETECTION	89
TABLE 6.7 RESOURCES FOR 2X2 QPSK USING LSD DETECTION	89
TABLE 6.8 RESOURCES FOR 2X2 QPSK USING LSD DETECTION AND	
SUPER-ITERATION	89
TABLE 6.9 RESOURCES FOR 2X2 16QAM USING MMSE DETECTION	90
TABLE 6.10 SYSTEM CONFIGURATIONS	91
TABLE 7.1 POWER DISSIPATION RATIO FOR THE SUB-BLOCKS OF 4X4	
LSD-MIMO	107

TABLE 7.2 POWER IMPROVING RATIO AFTER CLOCK GATING
TABLE 7.3 POWER DISSIPATION RATIO AFTER CLOCK GATING AND
ENTIRE ISOLATING OPERANDS
TABLE 7.4 POWER APPROVING RATIO TO CLOCK GATING AFTER
OPERAND ISOLATING
TABLE 7.5 POWER DISSIPATION AFTER CLOCK GATING AND PARTIAL
OPERAND ISOLATION
TABLE 7.6 POWER APPROVING RATIO TO CLOCK GATING AFTER
PARTIALLY OPERAND ISOLATING
TABLE 7.7 POWER DISSIPATION AFTER CELL REPLACEMENT 115
TABLE 7.8 POWER APPROVING RATIO TO PARTIAL OPERAND ISOLATING
AFTER CELL REPLACEMENT

Preface

The research project for the LDPC coded MIMO-WLAN system was supported by Bit Engineering Laboratory at System Integration Technology Institute (SITI) of Information and Communications University (ICU). Also Micro Information and Communication Remote Object-oriented System (MICROS) Research Center at Korea Advanced Institute of Science and Technology (KAIST) and the Communication Research Laboratory at University of Pittsburgh were participated in above project. Especially these two research groups have supported the algorithms for the LDPC-coded MIMO-WLAN system. Bit Engineering Laboratory programmed the physical layer of extracted system to simulate all the parameters we want to check and design. They supported to hardware design of the extracted LDPC-coded MIMO-WLAN system.

Especially my contributions on the project are as follows. I made analytic equations to calculate the latencies and throughputs for the platform using multiple processing units and extracted the system parameters to hardware design with Ph.D students of Bit Engineering Laboratory. I programmed the LDPCC codec software and designed its hardware with Verilog language for the total system. During designing LDPCC hardware, I proposed new decoding equations which can reduce the memory usage. To do a platform-based design, I made a FPGA platform to support 3 processors together with the Virtex 2 Pro 100 FPGA of Xilinx Inc. Finally I concentrated on the low-power optimization of the designed hardware system.

From this dissertation, three papers have been submitted to the *IEICE Transactions on Communications* as followed titles.

- Chapter 2: Gyongsu Lee and Sin-Chong Park, "Evaluating Multi-Processor SoC
 Platform Design Using Dedicated FIFO Channels".
- Chapter 4 and 6: Gyongsu Lee and Sin-Chong Park, "Architecture Analysis of MIMO Detection and Iterative Decoding for Coded MIMO-OFDM System".
- Chapter 5: Gyongsu Lee and Sin-Chong Park, "Implementation of the LDPCC Codec with AWGN Channel in a FPGA".

1. Introduction

1.1 Motivation

Achieving high throughput has been considered as a main target of wireless communications. For instance, current WLAN (IEEE 802.11a) systems provide a throughput as high as 54 Mbps [IEEE99]. In order to overcome multipath fading and thermal noise incurred in the propagation channel, many state-of-the-art technologies such as multi-input multi-output (MIMO) and orthogonal frequency-division multiplexing (OFDM) have been adopted.

The next-generation WLAN, which is currently being standardized by the IEEE 802.11 Task Group N, is targeted for a throughput up to 500 Mbps. The physical layer of the system is characterized by coded MIMO-OFDM transmission, which has been commonly known as a solution to achieving high throughput. In particular, multiple transmit and receive antennas increase the bandwidth efficiency, OFDM facilitated by the fast Fourier Transform (FFT) provides robustness against multipath fading, and LDPC codes significantly reduce the error probability through coding gain. Higher modulation order and higher code rate can also increase the throughput, as in current WLAN systems.

Either reducing the error probability or increasing the transmission rate can increase the throughput of a system. Unfortunately, reduced error probability and higher transmission rate are difficult to achieve simultaneously. For example, if the modulation order is increased, the transmission rate becomes larger at the expense of higher error probability. Recall that a number of system parameters affect the throughput (through the error probability and/or transmission rate). The system parameters include the modulation order, number of antennas, and diverse receiver configurations such as the number of decoding iterations. Thus it is not simple to make a decision on which system parameters leads to higher throughput than the others, even though it is critical to the receiver performance.

In addition to the throughput, the power consumption is also important to the design of wireless systems. From the perspective of implementation, the power consumption should be minimized within the system constraints on throughput and latency. As in the case of throughput, system parameters should be optimized taking into account the power consumption. However, there have not been many researches on the minimization of the power consumption of high-throughput wireless systems.

In this dissertation, the emerging 802.11n system is designed and implemented as a target system, taking into account the throughput and power consumption at the same time. First, the throughput of the target system is evaluated for different system parameters such as the modulation order, number of antennas, and receiver configurations. Secondly, based on the system simulation results, the system in a FPGA board is implemented and the power consumption is reduced. Finally, based on these results, the trade-off between throughput and power consumption is presented for the target system, which gives useful insights into how to adapt the system parameters.

1.2 Contributions and Organization

This dissertation consists of eight chapters and is organized as follows.

Chapter 2 introduces the platform based wireless SoC design methodology. Especially the characteristics of wireless SoC system is review and analytical equations for throughput and latency analysis using multiple processing units are proposed.

Chapter 3 briefly introduces the next-generation WLAN systems. The design goals and two promising proposals, TGnSync and WWiSE, are described, focusing on the physical layer. Particularly, it is shown that MIMO transmission is introduced as the key technologies for achieving extremely-high throughput.

Chapter 4 provides the concept of the super iterative decoding algorithm between MIMO detector and channel decoder. Especially for LDPC coded MIMO-OFDM systems is analyzed and simulated with various system parameters.

Chapter 5 summarizes the system requirement and proposes the architectures for the key components of the system to design in hardware.

Chapter 6 gives an idea for trading-offs of the performance and cost of the wireless SoC system.

Chapter 7 gives a review of power concerning problem for digital circuit design and introduces the techniques to reduce consuming power in a designing system.

Finally, Chapter 8 concludes this dissertation.

2. Platform Based Wireless SoC Design

2.1 Introduction

Platform based designs (PBDs) for SoC can reduce the repetitive design and verification procedures and guarantee the functions of pre-validated intellectual property blocks or cores by using verified platforms which consist of the proved ones from some libraries or cores provided by third-party vendors [MC03][MKTC04][SLD04][LRD01]. Since such cores are already pre-designed and verified, a designer can now concentrate on the overall system rather than the individual components, and also reduce the number of steps required to translate a system-level design into a final product. The PBD is proposed as one solution for important issues like short time-to-market and low cost. The PBD analyzes the common characteristics of derivative systems, sets up the common design methods, designs a platform with those methods, and verifies with that platform. A typical SoC platform consists of a collection of fully programmable processing elements and coarse-grained application-specific coprocessors optimized for specific tasks [MKTC04]. Designing with the platform, the software-based partitioning is a method to implement a hardware and software co-existent system. To give design flexibility and to accept various design of derivative system, as many as blocks are designed using software except some blocks difficult to be met system constraints [MC03][SLD04]. Recently, to implement a multi-processor platform, various bus architectures are considered [RM04], and schedulers for specific systems are evaluated [SLD04].

In this paper, one of architectures for platform based design having multiple programmable processing elements are proposed. Especially, a wireless communication system as a derivative system for the platform-based design is selected and some guidelines for its design are given. The FIFO based communication architecture [RM04] is analyzed as an applicable method to utilize the property of unidirectional traffic which is a major characteristic of wireless communication system. Table 2.1 summarizes the abbreviation for this chapter.

N_P	Number of processing Units (CPU, DSP, hardware unit, etc.)
$C_P^{(i)}$ (cycle)	Required cycles for the i-th processor
$F_P^{(i)}$ (Hz)	Clock frequency of the i-th processor
$T_P^{(i)}$ (sec)	Processing time of the i-th processor
$D_R^{(i)}$ (byte)	Required input data size for the i-th processor
$D_W^{(i)}$ (byte)	Output data size after the processing of the i-th processor
$B_{R(W)}$ (byte)	Bus(or FIFO) data width
F _B (Hz)	Bus(or FIFO) clock frequency
$L_{B,R(W)}$ (cycle)	Bus(or FIFO) access latency or arbitration delay
$ au_{S,R(W)}$ (cycle)	Required cycles for data transfer through the Bus (or FIFO)
$\beta^{(i)}{}_{R(W)}$	Possible burst transfer size of the i-th processor
$O_{R(W)}^{(i)}$ (sec)	Bus occupancy time

TABLE 2.1 ABBREVIATION FOR NOTATING THE LATENCY AND THROUGHPUT

2.2 Characteristics of Wireless SoC System

2.2.1 Unidirectional Traffic

The directions of data between functional blocks of the wireless SoC system are fixed. Considering the scrambling block and the convolution encoding block of 802.11a system as an example, the outputs of the scrambler are used only as inputs to the convolution encoder but no signal from the convolution encoder is transmitted to the scrambler. This feature guarantees to use the dedicated channels for connecting processing elements of wireless SoC platform. In this paper, the FIFO memory blocks are used as dedicated channels [RM04][SLD04].

2.2.2 Data Format Conversions

Data can be represented as various types in a system. As an example, the host transfers 4-byte data. In this case, actual 32-bit stream whose width is one bit and length is 32 should be transmitted. For more efficient handling, the data type of processor or bus width can be considered. This data also can be expanded in some blocks. For instance, output size of the convolution encoder whose coding rate is 1/2 is twice its input size. If the type of input to convolution encoder is an array of char equal to 8-bit, an array of short identical 16-bit is suitable for the type of output. As another example, the one bit for BPSK mode is extended to 16-bit width soft-valued number after processing it in mapper. Therefore, how many computation bits are representing an original information bit should be counted in each sub-block.

Each block of 802.11a system has minimum data length can start processing. Though scrambler or convolution encoder can process for each bit, interleaver can make output only after receiving at least 48 bits and the IFFT can operate with minimum bit-extended soft-valued 48 information bits and 16 inserted pilot bits. If 3 bytes are used as input to scrambler and convolution encoder, they operate 24 times for each information bit. For one interleaving operation, the interleaver must wait for receiving entire 48 symbols from the convolution encoder which uses 24 information bits. If IFFT block has less than 64 soft-valued inputs or 128 bytes for 16 bits extension case for each bit, it can process noting.

Handling the bit-extended bits in the processor has severe overhead for data transferring. Considering IFFT processing of 802.11a system, it needs 64 values for each 64-point. If each point value is 16-bit extended, IFFT block can be operated after receiving 128 bytes or 1024 bits. Even though the BUS width is 32-bit, it should read data 32 times. Therefore assigning blocks to processors should be considered not to be divided at the point of transferring bit-extended information.

Designing each block of wireless communication SoC, assigning suitable data type and minimum block size to handle should be decided and blocks should be assigned not to communicate with bit-extended soft-valued data.

2.2.3 Critical Requirements for Latency and Throughput

Most systems should process more than required throughput adequate to data rate.

For example, the throughput of the multimedia player should be larger than the data rate of the MP3 file played on the player for continuously playing it. But the latency is not so sensitive because user can wait a few second to be played after pressing play button. It can be considered just as one of optimization options.

Wireless system commonly use the ARQ schemes to guarantee the result of transmission on the lossy wireless channel. Strong timing requirements are needed because the receiver should reply about success or failure of receiving data within required time.

During designing the wireless communication SoC system, designer should consider not only the throughput requirement but also the latency.

2.3 Latency-Throughput Computation

2.3.1 For Single Processor

Latency is $O_R + T_P + O_W$ sec and throughput is $D_R / (O_R + T_P + O_W)$ Byte/sec when single processor uses C_P cycles to process D_R bits with F_P Hz clock and needs O_R and O_W to read D_R bits and write D_W bits respectively. Where the bus occupancy time is

$$O_{R(W)}^{(i)} = \left[\frac{D_{R(W)}^{(i)}}{B_{R/(W)}\beta^{(i)}_{R/(W)}}\right] \times \left(\frac{L_{B,R/(W)} + \tau_{S,R/(W)} + (\beta^{(i)}_{R/(W)} - 1)}{F_B}\right)$$
(2.1)

With the notations in abbreviation table, the latency and the throughput can be represented as equations below. The latency is

$$\left\lceil \frac{D_R}{B_R \beta_R} \right\rceil \times \left(\frac{L_{B,R} + \tau_{S,R} + (\beta_R - 1)}{F_B} \right) + \frac{C_P}{F_P} + \left\lceil \frac{D_W}{B_W \beta_W} \right\rceil \times \left(\frac{L_{B,W} + \tau_{S,W} + (\beta_W - 1)}{F_B} \right) \right)$$

and the throughput is

$$\begin{split} D_{R} \div \left\{ \left\lceil \frac{D_{R}}{B_{R}\beta_{R}} \right\rceil \times \left(\frac{L_{B,R} + \tau_{S,R} + (\beta_{R} - 1)}{F_{B}} \right) \right. \\ \left. + \frac{C_{P}}{F_{P}} + \left\lceil \frac{D_{W}}{B_{W}\beta_{W}} \right\rceil \times \left(\frac{L_{B,W} + \tau_{S,W} + (\beta_{W} - 1)}{F_{B}} \right) \right\}. \end{split}$$

2.3.2 For Multi-Processor using Simple Protocol

In this paper, processing with multi-processor is categorized by using simple protocol and synchronous protocol. Where simple protocol means each processor can start next job when it can get the next available data unit and synchronous protocol represents every processor waits end of each activity of other processors. For using simple protocol as show in figure 2.2, the latency is the sum of processing times for each processor and the throughput is determined by the longest processing time. They can be represented in equation (2.2) and (2.3) respectively.

$$\sum_{i=1}^{N_P} \left(O_R^{(i)} + T_P^{(i)} + O_W^{(i)} \right)$$
(2.2)

$$\frac{D_R^{(1)}}{\max_i \left(O_R^{(i)} + T_P^{(i)} + O_W^{(i)} \right)}$$
(2.3)

1	$O_R^{(1)}$	$T_{P}^{(1)}$	$O_{W}^{(1)}$	$O_{\scriptscriptstyle R}^{(1)}$	$T_{\scriptscriptstyle P}^{(1)}$	$O_W^{(1)} = C$	$O_{R}^{(1)}$	$T_p^{(1)}$	$O_{W}^{(1)}$	$O_{R}^{(1)}$	$T_P^{(1)}$	$O_W^{(1)}$	$O_R^{(1)}$	$T_P^{(1)}$	$O_W^{(1)}$ C	$D_{R}^{(1)}$	$T_{P}^{(1)}$	$O_W^{(1)}$
2				$O_{\scriptscriptstyle R}^{(2)}$	$T_p^{(2)}$		$O_W^{(2)}$ $O_W^{(2)}$	(2) R	$T_{P}^{(2)}$		$O_W^{(2)} O_R^{(2)}$)	$T_{P}^{(2)}$	1	$O_{W}^{(2)} = O_{R}^{(2)}$		$T_p^{(2)}$	
3							0	(3) R	$T_{P}^{(3)} = O_{W}^{(3)}$)	C	(3) R	$T_{p}^{(3)}$ 0	$O_{W}^{(3)}$	$O_R^{(3)}$	$T_p^{(3)}$	$O_W^{(3)}$	
4										$O_{\scriptscriptstyle R}^{(4)}$	T_F^{\dagger}	4)	$O^{(4)}_W$	$O_{\scriptscriptstyle R}^{(4)}$	$T_{p}^{(4)}$		$O^{(4)}_{W}$	$O_{\scriptscriptstyle R}^{(4)}$
5														$O_{R}^{(5)}$	$T_P^{(5)} = O_W^{(5)}$]	$O_{h}^{(}$	$\frac{5}{2}$ T_{μ}





Figure 2.1 Multi-processor operating timing with simple protocol

2.3.3 For Multi-Processor using Synchronous Protocol

Though simple protocol doesn't consider the status of following processor, synchronous protocol can start processing each data unit at the end of the longest processing job. As shown in figure 2.2, the throughput of the synchronous protocol is same as the simple protocol's. But the latency is not the sum of each processing time but the longest processing time times the number of processing unit as shown in equation (2.4) below.

$$N_P \times \max_i \left(O_R^{(i)} + T_P^{(i)} + O_W^{(i)} \right).$$
(2.4)

Figure. 2.2 Multi-processor operating timing with synchronous protocol

2.4 Design Considerations

2.4.1 Frequency Setting Flow

First of all, the designer estimates required number of cycles, selects the number of processor, and chooses the platform architecture assuming the given system will be implemented with programmable hardware. During this estimation, peripherals of processors or additional hardware blocks as co-processors can be assigned to some tasks which are difficult to be handled with only software. Assuming every processor uses the same frequency, the frequency should be calculated to meet the requirements for latency and throughput. Even though the assignment of same cycle to each process is ideal, it is so

difficult because of difference of the calculating amount, characteristics of calculation, and data type etc. Therefore, any process should wait until available time for each processor starting next processing because each processor handles different amount of processing with same frequency. The idle times shown in figure 2.1 and 2.2 are these waiting periods. In this case, preventing unnecessarily fast processing or letting down the operating frequencies under satisfying the throughput and latency requirements can reduce power consumption. The flow chart of the setting for frequencies is shown in figure 2.3.

2.4.2 Dedicated FIFO Channel

As mentioned before, the directions of data flows in the wireless communicaiton system are already fixed between sub-blocks. Dedicated FIFO channel is candidate architecture to use the feature of the unidirectional traffic. Because the FIFO is a memory system block which contains dual-port sram and control units, the addresses for writing or reading are internally controlled by the FIFO. The data written to the FIFO can be saved to different FIFO memory spaces just by writing it to the same writing register without considering addresses to be written. Similar to writing process, because the FIFO controls reading addresses internally, reading data from the FIFO can be accomplished without any consideration of reading addresses of sequentially written data in the FIFO. Therefore sequential data transferring via FIFO between processors by fixed direction can be more easily implemented than using ordinary memory whose individual addresses are managed by the processors. Only referencing the status, FIFO can be used for the data transferring process. The preceding processor can run whenever the FIFO has available space regardless of the status of following one.



Figure 2.3 Design procedure for frequency setting

2.5 Example of Derivative System

2.5.1 802.11a System Overview

IEEE802.11a system is considered as an example of derivative system for wireless SoC platform based design [802.11A]. In figure 2.4, each I/O number of sub-block to process an OFDM symbol is represented. According to data rate, from 24-bit to 216-bit data is processed in MAC and scrambler and from 48-symbol to 288-symbol data is generated by convolution encoder corresponding to one OFDM symbol data. These data are extended to 16-bit width soft valued numbers after processing of mapper. Therefore 96-byte data is generated by mapper, 128-byte is formed after pilot insertion, and 160-byte data is transmitted finally by guard inserting block. The 160-byte data of final output is a part of data packet corresponding to an OFDM symbol and a packet consists of maximally 2312-byte length or 711 OFDM symbols. Every OFDM symbol of a packet should be transmitted every 4usec period without discontinuity.

Three processors are assigned to sub-blocks like figure 2.4. Because transferring data can handle multiple information bits in one data unit, this assignment guarantees that each process transfers from 3 to 36 bytes for treating one OFDM symbol. As mentioned before, one data bit is extended multiple bits after mapper, if the block later than mapper is assigned to different processor, data size assigned to it will be larger than 96 bytes. Therefore the entire mapper and later blocks of it are assigned to a co-processor. One processor is assigned to manage data of a co-processor and others are assigned to the

blocks previous mapper.

According to data rate, the system should transmit from 24 to 216 bits in 4usec, or it should meet throughput constraint from 6Mbps to 54Mbps. As the other constraint, latency should be less than the Short Inter-Frame Space (SIFS) of 802.11a system, or 16usec.



Figure 2.4 Minimum Applicable Block Size of Each Block and Assigning Blocks to 3 Processors

2.5.2 For Single Processor

In this section, as an example, simple 6Mbps mode is analyzed and the 54Mbps mode and burst mode transferring will be mentioned later. From equation (2.1), the O_R is obtained. To read 3 bytes per 4usec period, processor can read three times of 1 byte or one time sequential 1 and 2 byte because the data types of processor are one of the powers of 2 bytes. Because the processing should be subdivided according to byte length in the latter case, we design to read three times by each one byte data. It means that $\beta = 1$ and $D_R = 3$, and $O_R = \frac{3}{B} \times \left(\frac{L_B + \tau_S}{F_B}\right)$. Assuming access latency to be one cycle and actual reading cycle to also be one, $L_B + \tau_S$ will be 2 cycles. Where letting the width of the

BUS 1 byte wide which is same as the width of reading data unit, O_R will be $6/F_B$.

Because the data read through the BUS are treated in processor and are transmitted to coprocessor which has final transmitting block, there is no data to be written to BUS or $D_W = 0$. Therefore $O_W = 0$ and latency is $6/F_B + C_P/F_P$. By using same bus frequency and processing frequency, latency is simplified to $(C_P + 6)/F$. The throughput is also calculated as $D_R F / (C_P + 6) = 3F / (C_P + 6)$. If the program can be completed with C_P or 900 cycles by the optimization procedures, the latency and the throughput will be 906/F sec and $3F/906 Byte/\sec = 24F/906 bps$ respectively. These should meet the conditions, $906/F \le 16\mu \sec$ and $24F/906 \ge 6Mbps$. Therefore $F \ge 906/16MHz = 56.6MHz$ and $F \ge 909/24 \times 6MHz = 226.5MHz$ should be satisfied. $F \ge 226.5MHz$ is the condition to meet the latency and the throughput constraints simultaneously with a single processor.

2.5.3 For Multi-Processor using Simple Protocol

If three processors are used to implement the given system, N_p is 3 and the total cycles by them should larger than or equal to the cycle by single processor, or $\sum_{i=1}^{3} C_p^{(i)} \ge C_p = 900$. Total cycles are assigned to the processor 1, 2 and 3 with 250, 300, and 350 cycles respectively. Because data reading cycle of 1st processor is same as that of single processor, $O_R^{(1)}$ is $6/F_B^{(1)}$. To write to FIFO, 2 cycles for checking the status of FIFO and 1 cycle for actual writing are required, or $L_B + \tau_s$ is 3. As in the previous case, letting the FIFO width to be 1 byte, $O_W^{(1)}$ is $9/F_B^{(1)}$. Because the reading cycle of the

 2^{nd} processor from the 1st FIFO is same as the writing cycle of the 1st processor to it, $O_R^{(2)}$ is $9/F_B^{(2)}$. If the width of the 2^{nd} FIFO is set to be 2 bytes as the output data size of 2^{nd} processor is doubled up, $O_W^{(2)}$ can be also $9/F_B^{(2)}$. Similarly, the 3^{rd} processor reads data with $O_R^{(3)} = 9/F_B^{(3)}$ cycles. Because this final processor writes noting to BUS or FIFO as in the case of single processor, $O_W^{(3)}$ is zero. Finally, the latency by equation (2.2) is expressed below equation.

$$\begin{split} &\sum_{i=1}^{N_p} \left(O_R^{(i)} + \frac{C_P^{(i)}}{F_P^{(i)}} + O_W^{(i)} \right) = \sum_{i=1}^3 \left(O_R^{(i)} + \frac{C_P^{(i)}}{F_P^{(i)}} + O_W^{(i)} \right) \\ &= \left(O_R^{(1)} + O_R^{(2)} + O_R^{(3)} \right) + \left(\frac{C_P^{(1)}}{F_P^{(1)}} + \frac{C_P^{(2)}}{F_P^{(2)}} + \frac{C_P^{(3)}}{F_P^{(3)}} \right) \\ &+ \left(O_W^{(1)} + O_W^{(2)} + O_W^{(3)} \right) \\ &= \left(\frac{6}{F_B^{(1)}} + \frac{9}{F_B^{(2)}} + \frac{9}{F_B^{(3)}} \right) + \left(\frac{250}{F_P^{(1)}} + \frac{300}{F_P^{(2)}} + \frac{350}{F_P^{(3)}} \right) \\ &+ \left(\frac{9}{F_B^{(1)}} + \frac{9}{F_B^{(2)}} + 0 \right) \\ &= \left(\frac{15}{F_B^{(1)}} + \frac{18}{F_B^{(2)}} + \frac{9}{F_B^{(3)}} \right) + \left(\frac{250}{F_P^{(1)}} + \frac{300}{F_P^{(2)}} + \frac{350}{F_P^{(3)}} \right) \end{split}$$

Setting bus frequency to the same processing frequency, the latency should be met $\frac{265}{F_p^{(1)}} + \frac{318}{F_p^{(2)}} + \frac{359}{F_p^{(3)}} \le 16u \sec \text{ and the throughput from equation (2.3) should be below}$

condition.

$$\frac{3}{\max_{i}\left(\frac{265}{F_{p}^{(1)}},\frac{318}{F_{p}^{(2)}},\frac{359}{F_{p}^{(3)}}\right)} \ge \frac{6}{8}MByte/\sec$$
(2.5)

If all of $F_P^{(i)}$ are set to F_P , the latency is $\frac{265}{F_P} + \frac{318}{F_P} + \frac{359}{F_P} = \frac{942}{F_P} \le 16u \sec$ or

 $F_p \ge 58.9 MHz$, and the throughput is $F_p \ge 89.8 MHz$ because $3/\max_i \left(\frac{265}{F_p}, \frac{318}{F_p}, \frac{359}{F_p}\right)$ should be grater than or equal to 6/8 MByte/sec. From above

two constraints, $F_P \ge 89.8 MHz$ can be summarized.

2.5.4 For Multi-Processor using Synchronous Protocol

Equation (2.4) represents the latency as the equation below.

$$N_{P} \times \max_{i} \left(O_{R}^{(i)} + \frac{C_{P}^{(i)}}{F_{P}^{(i)}} + O_{W}^{(i)} \right)$$

= $3 \max \left(\frac{6}{F_{B}^{(1)}} + \frac{250}{F_{P}^{(1)}} + \frac{9}{F_{B}^{(1)}}, \frac{9}{F_{B}^{(2)}} + \frac{300}{F_{P}^{(2)}} + \frac{9}{F_{B}^{(2)}}, \frac{9}{F_{B}^{(3)}} + \frac{350}{F_{P}^{(3)}} + 0 \right)$
= $3 \max \left(\frac{15}{F_{B}^{(1)}} + \frac{250}{F_{P}^{(1)}}, \frac{18}{F_{B}^{(2)}} + \frac{300}{F_{P}^{(2)}}, \frac{9}{F_{B}^{(3)}} + \frac{350}{F_{P}^{(3)}} \right)$

If the BUS clock and processing clock are the same, above equation is simplified as below and the throughput is same as equation (2.5).

$$3\max\left(\frac{265}{F_p^{(1)}}, \frac{318}{F_p^{(2)}}, \frac{359}{F_p^{(3)}}\right) \le 16u \sec t$$

If every $F_P^{(i)}$ equals F_P , the latency is $F_P \ge 67.3 MHz$ because of below equation.

$$3\frac{359}{F_P} = \frac{1077}{F_P} \le 16u \sec t$$

And the constraint by throughput is same as in the simple protocol, $F_p \ge 89.8 MHz$. Therefore this system must operate higher than or equal 89.8MHz frequency to meet latency and throughput together.

2.6 Platform Board Design

The platform board that can be contained multi-processors in a FPGA is designed. Figure 2.5 shows the block diagram of a FPGA platform board. Three processors in a FPGA communicate each other through the FIFOs also implemented in same FPGA. The blocks with dashed lines are the components on the board not in a FPGA. Figure 2.6 is the board layout and the actual photo of the designed platform board.



Figure 2.5 Block diagram for candidate system with 3 processing unit


Figure 2.6 Board layout and actual photo of platform board

3. Overview of the MIMO-WLAN Systems

3.1 Introduction

As the integrated circuit (IC) technology is rapidly developing, wireless communication systems have exploited numerous sophisticated algorithms for the purpose of improving system performance. Although there are a number of low-rate wireless systems such as ZigBeeTM, Bluetooth, and RF ID, most of the attention is paid to the design of high-rate wireless communication systems. For example, the WLAN sponsored by IEEE 802.11 committees is currently being developed to further enhance the throughput of the existing systems (e.g., IEEE 802.11a systems). In particular, the standardization activity of IEEE 802.11n, which is recognized as the next-generation WLAN standard, has been evolving to achieve throughput of more than 100 Mbps [TGn04], [WWiSE04]. Since the IEEE 802.11n system is characterized by MIMO schemes, it will be called MIMO-WLAN system throughout this dissertation. In the following, a brief overview of the system will be presented along with promising proposals. The emphasis is put on how to achieve a throughput as high as 500 Mbps. Also, several MIMO schemes will be introduced as a key technique.

According to documents available in IEEE 802.11 Task Group N, the purposes of the standardization can be summarized as follows.

To achieve more than 100 Mbps throughput at the service access point (SAP) of medium access control (MAC) layer To provide backward compatibility with respect to other existing WLAN systems (e.g., IEEE 802.11a system)

The MIMO-WLAN systems are characterized by extremely high bandwidth efficiency (up to 15 bits/s/Hz). While the Multi-Band OFDM Alliance ultra-wide band (MBOA-UWB) systems (the so-called IEEE 802.15.3a systems) simply utilize *ultra-wide* bandwidth, the MIMO-WLAN systems have a strict constraint on the bandwidth, e.g., 20 MHz (or, optionally, 40 MHz). Instead of utilizing sufficiently wide bandwidth, several alternative techniques are under consideration in the MIMO WLAN systems. In particular, several MIMO schemes are quite promising since it inherently enhances the throughput without sacrificing any reliability (which has been commonly encountered in traditional modulation and coding). Let's take a look at the channel capacity (Figure 3.1) achieved by multiple transmit and receive antennas over multipath fading channels. Note that the channel capacity is defined as the upper-bound of achievable throughput over a specific channel. (Here the Model D of TGn Channel Models [Erceg03] is assumed.)



Figure 3.1 Channel capacity of the MIMO-WLAN systems

3.2 Promising Proposals for the MIMO-WLAN Systems

The major candidates of the MIMO-WLAN systems have been proposed by TGn Sync [TGn04] and WWiSE [WWiSE04]. At the first glance of their specification materials, it is easily found that their approaches are quite similar in many aspects. For example, both of them proposed the use of several MIMO schemes along with higher-order modulation (quadrature amplitude modulation), higher-rate channel coding and more spectrally efficient OFDM processing. In the following, how the proposals achieve transmission rate of a few Mbps is analyzed.

3.2.1 TGn Sync

TGn Sync provides up to 630 Mbps when 40 MHz channels are available. The enhancement of transmission rate is obtained as summarized in Table 3.1 [TGn04]. Note that a significant increase, for example, multiplication by an integer, is achievable only through (1) increasing bandwidth or (2) MIMO schemes. In particular, TGn Sync proposed the use of closed-loop MIMO schemes, transmit beamforming (TX-BF) as well as open-loop MIMO schemes such as SDM (Spatial Division Multiplexing) and STBC (Space-Time Block Coding), which will be described later in this chapter.

	802.11a	TGn Sync	Rate
Bandwidth	BW=20MHz	BW=20MHz	1x
		BW=40MHz	2x
Modulation	Up to 64QAM	Up to 64QAM	1x
		Up to 256QAM (ABF-MIMO)	1.33x
Coding	Coding rate	1/2,2/3,3/4	1x
		7/8	1.167x
OFDM	$N_{SD}=48$	$N_{SD} = 48(20MHz), 108(40MHz)$	1x,1.125x
	$T_{GI}/T_{SYM} = 800 \text{ ns}/3200 \text{ ns}$	T_{GI}/T_{SYM} =800ns/3200ns	1x
		T_{GI}/T_{SYM} =400ns/3200ns	1.11x
MIMO	$N_{TX}=I$	$N_{TX}=2$	2x
		$N_{TX}=3,4$	3x or 4x

TABLE 3.1 THROUGHPUT ENHANCEMENT OF TGN SYNC

3.2.2 WWiSE (World-Wide Spectrum Efficiency)

Since the 40 MHz channels may not be available in some countries (such as Japan and Europe), WWiSE excluded the use of 40 MHz channels, as opposed to TGn Sync. Furthermore, WWiSE only considers the use of open-loop MIMO techniques, which does not require any channel state information (CSI) at the transmitter side. Also, WWiSE

provides a combined STBC and SDM. As will be shown later, MIMO schemes do not only increase the transmission rate by spatial multiplexing, but also extends the coverage by spatial diversity.

TABLE 3.2 THROUGHPUT ENHANCEMENT OF WWISE

(Underlined characters represent the parts of TGn Sync that are excluded by WWiSE)

	802.11a	TGn Sync	Rate
Bandwidth	BW=20MHz	BW=20MHz	1x
		BW=40MHz	2x
Modulation	Up to 64QAM	Up to 64QAM	1x
		Up to 256QAM (ABF-MIMO)	1.33x
Coding	Coding rate	<i>1/2,2/3,3/4,<u>5/6</u></i>	<u>1.11x</u>
		7/8	1.167x
OFDM	$N_{SD}=48$	$N_{SD} = 54(20MHz), 108(40MHz)$	<u>1.125x</u> ,1.125x
	T_{GI}/T_{SYM} =800ns/3200ns	T_{GI}/T_{SYM} =800ns/3200ns	1x
		T_{GI}/T_{SYM} =400ns/3200ns	1.11x
MIMO	$N_{TX} = I$	$N_{TX}=2$	2x
		$N_{TX}=3,4$	3x or 4x

3.3 MIMO Schemes for the MIMO-WLAN System

In this section, we introduce the well-known MIMO schemes For simplicity, we categorize the MIMO schemes into open-loop and closed-loop MIMO schemes.

3.3.1 Open-loop MIMO Schemes

Open-loop MIMO schemes do not require the channel state information at the transmitter side. As several researchers have reported in the literature, the open-loop MIMO schemes are characterized by the trade-off between spatial multiplexing and spatial diversity [ZT03]. While the spatial multiplexing enhances the transmission rate by the number of transmit antennas, the spatial diversity mitigates the effect of multipath fading, which leads to the range extension.

3.3.1.1. Spatial Division Multiplexing

As shown in Figure 3.2, SDM transmits multiple independent streams (so-called spatial streams) over multiple antennas. Since the signals simultaneously transmitted by different antennas, x_1 and x_2 , act as interferences to each other, the receiver should remove the inter-antenna interferences in order to recover each transmitted signal as \hat{x}_1 and \hat{x}_2 from the received signals, y_1 and y_2 . Just as in the context of multi-user detection, the maximum likelihood (ML) detection is optimum in that it minimizes the error probability. However, the resulting complexity is of prohibitive complexity, i.e., exponential in the number of transmit antennas, making direct implementation impractical.



Figure 3.2 Spatial division multiplexing

The alternatives to the ML detection include linear detection methods such as zeroforcing (ZF) detection and minimum mean square error (MMSE) detection [GSSSN03]. These linear detection methods can be expressed as follows.

ZF:
$$\underline{\hat{x}} = Q(G_{ZF} \underline{y})$$
 where $G_{ZF} = H^+ = (H^H H)^{-1} H^H$

• MMSE:
$$\underline{\hat{x}} = Q(G_{MMSE} \underline{y})$$
 where $G_{MMSE} = (H^H H + I_{N_t})^{-1} H^H$

Here *H* is the channel matrix, \underline{y} is the received signal vector and $\underline{\hat{x}}$ is the recovered signal vector for the transmitted signal vector \underline{x} .

The main drawback of these suboptimal detection techniques is that the number of transmit antennas should be always kept smaller than that of receive antennas. Furthermore, the use of these linear detection techniques leads to performance degradation. It is widely known that MMSE detection outperforms ZF detection at a low SNR. Nevertheless, the performance of MMSE detection is not satisfactory in that it only achieves a fraction of the maximum achievable diversity gain. (Assuming N_t transmit antennas and N_r receive antennas, the linear detection only achieves diversity gain of $N_r - N_t + 1$ among the maximum achievable diversity gain of N_r .) To further improve the performance, it is highly recommendable to utilize the nulling and cancellation technique combined with ZF detection or MMSE detection [GFVW99]. The flow chart for a specific the nulling and cancellation method, also called the V-BLAST (Vertical Bell Laboratories Layered Space-Time) detection, is shown in figure 3.3. The simulation results in figure 3.4 show that the performance of V-BLAST detection remarkably approaches that of ML detection.



* $k^{(i)}(n)$: stream index of the *n*-th signal

Figure 3.3 V-BLAST detection (combined with ZF)

Figure 3.4 shows comparison of several MIMO detection techniques for SDM transmission. While ML detection provides diversity order of N_r , linear detection (e.g., ZF and MMSE detection) provides a fraction of the maximum achievable diversity gain, $N_r - N_t + 1$, as mentioned earlier. By utilizing V-BLAST detection, more than 3dB gain in a SNR is obtained.



Figure 3.4 Performance comparison of SDM transmission ($N_{\rm r}=N_{\rm r}=2$)

3.3.1.2. Space-Time Block Coding

Instead of using multiple antennas to achieve spatial multiplexing, STBC uses multiple antennas to provide spatial diversity (through space-time coding, without any loss in bandwidth). As mentioned earlier, the use of STBC enables the extension of coverage, by mitigating the effect of multipath fading. The WWiSE proposed a hybrid STBC that is a combination of Alamouti's STBC [Ala98] and SDM [Fos96], as shown in Figure 3.5 [WWISE04].



Figure 3.5 Space-time block coding ($N_{\rm t}=4$)

The main advantage of STBC is that the receiver structure is quite simple due to the orthogonality of Alamouti's STBC. Specifically, the use of Alamouti's STBC eliminates the need for matrix inversion.

Even though there exist four different kinds of STBC [WWiSE04], we only present STBC for four transmit antennas ($N_t = 4$) along with the corresponding detection scheme, as an example. If the received signal is expressed as

$$(\underline{y}_1 \quad \underline{y}_2) = (\underline{h}_1 \quad \underline{h}_2 \quad \underline{h}_3 \quad \underline{h}_4) \begin{pmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \\ x_3 & x_4 \\ -x_4^* & x_3^* \end{pmatrix} + (\underline{w}_1 \quad \underline{w}_2),$$

the resulting ZF detection or MMSE detection can be performed without any matrix inversion, as shown in the following. For detailed information, refer to [WWiSE04] and [RGZV03].

$$\mathbf{ZF:} \quad \underline{\eta} = G_{ZF} \left(\frac{\underline{y}_{1}}{\underline{y}_{2}} \right) \text{ where } G_{ZF} = \frac{1}{c_{1}c_{2} - c_{3}} \left(\begin{array}{cc} c_{2}I_{2} & -H_{1}^{H}H_{2} \\ -H_{2}^{H}H_{1} & c_{1}I_{2} \end{array} \right) \left(\begin{array}{cc} H_{1}^{H}H_{1} \\ H_{2}^{H} \end{array} \right)$$
$$\mathbf{MMSE:} \quad \underline{\eta} = G_{MMSE} \left(\frac{\underline{y}_{1}}{\underline{y}_{2}^{*}} \right)$$
$$\text{where } G_{MMSE} = \frac{1}{d_{1}d_{2} - c_{3}} \left(\begin{array}{cc} d_{2}I_{2} & -H_{1}^{H}H_{2} \\ -H_{2}^{H}H_{1} & d_{1}I_{2} \end{array} \right) \left(\begin{array}{cc} H_{1}^{H}H_{1} \\ H_{2}^{H} \end{array} \right)$$

In order to calculate the nulling matrix G_{ZF} or G_{MMSE} , the following parameters should be calculated. Even though the procedure requires a series of multiplications, it doesn't include any matrix inversion which is quite computationally prohibitive.

$$H_{1} = \begin{pmatrix} \underline{h}_{1} & \underline{h}_{2} \\ \underline{h}_{2}^{*} & -\underline{h}_{1}^{*} \end{pmatrix}, H_{2} = \begin{pmatrix} \underline{h}_{3} & \underline{h}_{4} \\ \underline{h}_{4}^{*} & -\underline{h}_{3}^{*} \end{pmatrix}$$

$$c_{1} = \|\underline{h}_{1}\|^{2} + \|\underline{h}_{2}\|^{2}, c_{2} = \|\underline{h}_{3}\|^{2} + \|\underline{h}_{4}\|^{2}, c_{3} = \|\underline{h}_{1}^{H}\underline{h}_{3} + \underline{h}_{2}^{T}\underline{h}_{4}^{*}\|^{2} + \|\underline{h}_{2}^{H}\underline{h}_{3} - \underline{h}_{1}^{T}\underline{h}_{4}^{*}\|^{2}$$

$$d_{1} = c_{1} + 1, d_{2} = c_{2} + 1$$

3.3.2 Closed-loop MIMO Schemes

If the channel state information (CSI) is perfectly known to the transmitter, it is well known that the optimal transmission of MIMO systems is simply transmit beamforming (TX-BF). Even though the beam pattern can be chosen arbitrarily, the well-established singular value decomposition (SVD) seems to be suitable for the MIMO-WLAN systems.

When transmit beamforming is applied to the MIMO transmission, the processing at the transmitter and receiver sides can be performed as illustrated in Figure 4.7. At the transmitter side, the channel matrix H should be decomposed into singular matrices U and V, and a diagonal matrix H as follows.

$$H = UDV^{H}$$
 where $UU^{H} = I_{N_{RX}}$, $diag_{n}(D) = \sigma_{n}(H)$ and $VV^{H} = I_{N_{TX}}$



Figure 3.6 Transmit beamforming



Figure 3.7 Illustration of transmit beamforming

On the other hand, at the receiver side, the following operation is required.

$$\underline{\hat{x}} = Q(\underline{\eta})$$
 where $H_{eff} = (UD)_{(:,1:N_{SS})}P$ and $\eta_n = \frac{1}{\|H_{eff}(:,n)\|^2}H_{eff}^H(:,n)\underline{y}$

Note that, since transmit beamforming generates the parallel subchannels (without interantenna interferences), the detection doesn't require any complex calculation such as matrix inversion.

The power allocation across multiple spatial streams can be determined by using the well-known convex optimization, which leads to the water-filling principle shown in Figure 3.8.



Figure 3.8 Illustration of water-filling principle

4. Super iterative Decoding between MIMO detector and Channel Decoder

4.1 Introduction

In the past decade, many works on the physical-layer study of multiple-input multiple-output (MIMO) techniques have been done. The BLAST systems [Fos96] have been studied for higher data-rate and the orthogonal space-time block code (STBC) [Ala98] was proposed for transmit antenna diversity. Tarokh proposed the MIMO schemes which does not require channel state information at the transmitter side [TSC98].

In general, MIMO system is combined with OFDM or any coding scheme. Physicallayer design and optimization of low-density parity check (LDPC) coded wireless MIMO orthogonal frequency-division multiplexing (OFDM) communications [LYW04][LWN02][BKA04]. As its receiver structure, they proposed turbo receiver or super iterative receiver, which employs a soft demodulator and a soft channel decoder, for their system. For the commercial MIMO-OFDM systems, the promising proposals of the IEEE 802.11n systems have been proposed TGnSync [TGn04] and WWiSE [WWiSE04]. These use convolutional code as default channel coding algorithm and LDPC code as an option.

The LDPC coded wireless MIMO-OFDM is adapted to the promising proposals of the IEEE 802.11n system to meet the latency and throughput requirements. As receiver structure, the super iterative receiver is used with a soft MIMO demodulator and a soft

LDPC decoder. Under given receiver structure, the methods of the system combination to increase transmitting data are compared.

4.2 MIMO Detection Algorithm

The channel for MIMO is assumed as a narrowband flat fading channel. As shown in Fig.1, the information signals are transmitted through N_t transmitted antennas, affected by **H** channels, received by N_r receiving antennas, and recovered by MIMO detector in receiver. It is assumed that the receiver knows the memoryless channel matrix **H** whose columns are linearly independent. It is an $N_r \times N_t$ *i.i.d.* zero-mean unit variance complex Gaussian matrix. s_i of transmitted signal vector $\mathbf{s} = [s_1, \dots, s_{N_t}]^T$ is chosen from complex constellation C with 2^{M_c} and consists of M_c bits. Receiving symbols are represented as equation (4.1).

$$\mathbf{y} = \sqrt{\frac{P}{N_t}} \mathbf{H} \mathbf{s} + \mathbf{n} = \mathbf{H}_e \mathbf{s} + \mathbf{n}$$
(4.1)



Figure 4.1 MIMO channel model.

In this section, various MIMO detection algorithms such as the maximum-likelihood (ML), the zero-forcing (ZF), the minimum mean square error (MMSE), the V-BLAST, the sphere decoding and list sphere decoding (LSD) are summarized and compared briefly.

The ML method selects signal vector to minimize the distance between the received signals vector and transmitted signal vector with channel effects like equation (4.2).

$$\hat{\mathbf{s}} = \arg\min_{all \, \mathbf{s}} \left\| \mathbf{y} - \mathbf{H}_{e} \mathbf{s} \right\|^{2} \tag{4.2}$$

It needs exhaustive searching according to the size of possible number of input vector, $2^{M_c N_t}$. As examples, if 2 transmitting antennas use QPSK modulated signals which is consists of 2 bits, total 16 candidates are existed to be checked, and if 4 antennas transmit 64QAM modulated signals with 6 bits, the $2^{6\times4} = 16,777,216$ candidates should be compared to get only $M_c = 6$ bits.

The ZF algorithm defines vector \mathbf{z} which is sum of signal vector and noise vector like equation (4.3).

$$\mathbf{z} = \mathbf{H}_e^+ \mathbf{y} = \mathbf{s} + \mathbf{H}_e^+ \mathbf{n} \tag{4.3}$$

Where $\mathbf{H}_{e}^{+} = (\mathbf{H}^{*}\mathbf{H})^{-1}\mathbf{H}^{*}$ is the Moore-Penrose pseudo inverse matrix or the general inverse matrix of **H**. Simple linear algebra, multiplication of received signal vector and inversion of channel matrix estimate transmitted data as equation (4.4).

$$\hat{s}_i = \arg\min \|s_i - z_i\|^2$$
, $i = 1, \cdots, 2^{M_c}$ (4.4)

This algorithm is simple, but has severe degradation of error performance because of neglecting noise.

The MMSE algorithm defines vector \mathbf{z} which is product of vector \mathbf{W} and received vector like equation (4.5) and decodes bits same as equation (4.4).

$$\mathbf{z} = \mathbf{W}\mathbf{y} \tag{4.5}$$

The MMSE criterion is denoted as equation (4.6)

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{W}} \|\mathbf{s} - \mathbf{z}\|^2 = \arg\min_{\mathbf{W}} \|\mathbf{s} - \mathbf{W} \cdot \mathbf{y}\|^2$$
(4.6)

Where **W** is represented $\mathbf{H}_{e}^{H} \cdot \left(\mathbf{H}_{e} \cdot \mathbf{H}_{e}^{H} + N_{0}\mathbf{I}_{N_{r}}\right)^{-1}$ from the condition of $E\left[\left(\mathbf{s} - \mathbf{W} \cdot \mathbf{y}\right)\mathbf{y}^{H}\right] = 0$ [Pro95].

The V-BLAST algorithm is the layered ZF or MMSE algorithm. Ordinary ZF detects every element or symbol of transmitted symbol vector in one process, but the layered ZF algorithm detects only one element or symbol and N_t times detection loops should be executed. Each loop is consists of following steps, the zero forcing for one element of the transmitted symbol vector, the detection of the element from zero forcing, the interference canceling for the next loop, and the updating matrix for the next loop. There are the layered ZF/MMSE algorithms without ordering and with ordering. The former is worse than the later because of error propagating.

The sphere decoding is a sub-optimal algorithm for the ML decoding which should

exhaustively enumerate $2^{M_c N_t}$ candidates. It finds vectors in the sphere of radius r whose center is received vector y, $\|\mathbf{y} - \mathbf{H}_e \mathbf{s}\|^2 < r^2$. It reduces r until only one vector lies in the sphere.

And there is the LSD algorithm which is simple modified algorithm of the sphere decoding algorithm. It does not decrease radius and add all searched points to the list if the list is not full. If the list is full, it compares the searched points in the list to replace the largest radius with new searched small radius. By choosing sufficient radius size or list size, the sphere decoding or the LSD have similar performance of the ML decoding.



Figure 4.2 Performance comparison for MIMO detection algorithms (2 Tx antennas and QPSK modulation)

In figure 4.2, the performance of each MIMO detection algorithm is compared using 2 transmitter antennas whose modulation is QPSK. Although the ML is the most complex than any other algorithms, it has the best performance. At the 10⁻³ BER, the performance of ZF is worse than that of ML about 12dB.

4.3 Iterative Demodulation and Decoding Structure

Let's assume N_t transmit antennas, N_r receive antennas, and N_{sc} sub-carriers. The constellation size of QAM is denoted by M_c . The $N_r \times 1$ received signal at the *t*-th channel use, \underline{y}_t , can be expressed as

$$y_{t} = H_{t}\underline{s}_{t} + \underline{w}_{t}, \quad t = 1, \cdots, T$$

$$(4.7)$$

where H_t denotes the channel matrix at the *t*-th channel use, \underline{s}_t denotes the $N_t \times 1$ transmitted signal at the *t*-th channel use, and \underline{w}_t denotes the $N_r \times 1$ additive noise at the *t*-th channel use. Here notice that the channel use index *t* is related to the tone (or sub-carrier) index $f = 0, \dots, N_{sc} - 1$ as $f = \text{mod}(t, N_{sc})$. In other words, denoting the channel matrix at the *f*-th tone by H_f , it follows that $H_t = H_{\text{mod}(t, N_{sc})}$. Letting $x_{t,i}$ denote the *i*-th coded bit at the *t*-th channel use, it follows that

$$\underline{s}_{t} = map(\{x_{t,i}\}_{i=1}^{N_{t} \log_{2} M_{c}}), \quad t = 1, \cdots, T.$$
(4.8)

where $map(\cdot)$ represents the operation combining both QAM and SDM. Consequently, $N_t \log_2 M_c$ coded bits are transmitted at each channel use. For example, for the 64QAM (4,4) case, 24 coded bits are transmitted at each channel use.

Figure 4.3 shows an iterative demodulation and decoding method under consideration [HS03], [VHK04]. "LDPC DEC" represents the decoding algorithm (message passing algorithm), while "MIMO DEM" represents the demodulation algorithm that will be discussed in detail in the following. Obviously, the role of demodulation algorithm is simply generation of the *a extrinsic L*-value $L_{t,i}^{ext} := \ln(p(\underline{y}_t | x_{t,i} = 1)/p(\underline{y}_t | x_{t,i} = 0))$ using the received signal \underline{y}_t , the channel H_t and the *a priori L*-value $L_{t,i}^{prior} := \ln(p(x_{t,i} = 1)/p(x_{t,i} = 0))$. Using the Bayes' rule, it is shown that

$$L_{t,i}^{ext} = L_{t,i}^{post} - L_{t,i}^{prior}$$

$$\tag{4.9}$$

where $L_{t,i}^{post} := \ln(p(x_{t,i} = 1 | \underline{y}_t) / p(x_{t,i} = 0 | \underline{y}_t))$ represents the *a posteriori L*-value.



Figure 4.3 Iterative demodulation and decoding

Then $L_{t,i}^{post}$ can be simply expressed as

$$L_{t,i}^{post} \coloneqq \ln \frac{p(x_{t,i} = 1, \underline{y}_{t})}{p(x_{t,i} = 0, \underline{y}_{t})}.$$

$$= \ln \frac{\sum_{\underline{x}:x_{i}=1} p(\underline{y}_{t} \mid \underline{x}_{t} = \underline{x}) p(\underline{x}_{t} = \underline{x})}{\sum_{\underline{x}:x_{i}=0} p(\underline{y}_{t} \mid \underline{x}_{t} = \underline{x}) p(\underline{x}_{t} = \underline{x})}$$

$$(4.10)$$

where $\underline{x}_t := \{x_{t,i}\}_{i=1}^{N_t \log_2 M_c}$ and $\underline{x} := \{x_i\}_{i=1}^{N_t \log_2 M_c}$. Here note that $p(\underline{x}_t = \underline{x})$ is highly dependent on the LDPC code. Let us assume that coded bits in \underline{x}_t are independent of each other, in other words,

$$p(\underline{x}_{t} = \underline{x}) = \prod_{i=1}^{N_{t} \log_{2} M_{c}} p(x_{t,i} = x_{i}), \quad \forall \underline{x}, \qquad (4.11)$$

which is often the case in deriving the demodulation algorithm [HS03], [VHK04]. Therefore it follows from (4.9) and (4.10) that

$$L_{t,i}^{ext} = \ln \frac{\sum_{\underline{x}:x_i=1}^{M_c} p(\underline{y}_t \mid \underline{x}_t = \underline{x})^{N_t} \prod_{j=1}^{\log_2 M_c} p(x_{t,j} = x_j)}{\sum_{\underline{x}:x_i=0} p(\underline{y}_t \mid \underline{x}_t = \underline{x})^{N_t} \prod_{j=1}^{\log_2 M_c} p(x_{t,j} = x_j)} - L_{t,i}^{prior}.$$
(4.12)

From (4.12), it is shown that the assumption (4.11) enables the demodulation algorithm to calculate $L_{t,i}^{ext}$ without knowing the correlation among coded bits. Also note that $L_{t,i}^{ext}$ is dependent on the LDPC code only through $L_{t,i}^{prior}$.

Since

$$p(\underline{y}_{t} | \underline{x}_{t} = \underline{x}) = \frac{1}{(\pi N_{0})^{N_{r}}} \exp\left(-\frac{1}{N_{0}} \left\|\underline{y}_{t} - H_{t} \cdot map(\underline{x})\right\|^{2}\right), \quad (4.12)$$

then (4.12) can be expressed as

-

$$L_{t,i}^{ext} = \ln \sum_{\underline{x}:x_{i}=1} \exp\left(-\frac{1}{N_{0}} \left\|\underline{y}_{t} - H_{t} \cdot map(\underline{x})\right\|^{2} + \sum_{j=1}^{N_{t}} \ln p(x_{t,j} = x_{j})\right)$$
$$-\ln \sum_{\underline{x}:x_{i}=0} \exp\left(-\frac{1}{N_{0}} \left\|\underline{y}_{t} - H_{t} \cdot map(\underline{x})\right\|^{2} + \sum_{j=1}^{N_{t}} \ln p(x_{t,j} = x_{j})\right) - L_{t,i}^{prior}.$$
 (4.14)

This calculation is practically impossible, since the total number of metric calculation per coded bit amounts to $M_c^{N_t}$, as shown in (4.14). Using the Max-log approximation [HS03], [VHK04],

$$\ln \sum_{i=1}^{K} \exp(a_i) \approx \max_{1 \le i \le K} (a_i), \quad \forall a_i,$$

then (4.14) can be approximated as

$$L_{t,i}^{ext} = \max_{\underline{x}:x_{i}=1} \left(-\frac{1}{N_{0}} \left\| \underline{y}_{t} - H_{t} \cdot map(\underline{x}) \right\|^{2} + \sum_{j=1}^{N_{t} \log_{2} M_{c}} \ln p(x_{t,j} = x_{j}) \right)$$
$$- \max_{\underline{x}:x_{i}=0} \left(-\frac{1}{N_{0}} \left\| \underline{y}_{t} - H_{t} \cdot map(\underline{x}) \right\|^{2} + \sum_{j=1}^{N_{t} \log_{2} M_{c}} \ln p(x_{t,j} = x_{j}) \right) - L_{t,i}^{prior}$$
(4.15)

Even though the summations in (4.14) are removed, the calculation of (4.15) is still of prohibitive complexity. In the following, several methods to calculate $L_{t,i}^{ext}$ efficiently are introduced.

Recalling that $L_{t,i}^{prior}$ is available from the decoding algorithm rather than $\ln p(x_{t,i} = x_i)$,

 $L_{t,i}^{ext}$ can be expressed as

$$L_{t,i}^{ext} = \max_{\underline{x} \in L_t: x_i = 1} \left(-\frac{1}{N_0} \left\| \underline{y}_t - H_t \cdot map(\underline{x}) \right\|^2 + \sum_{j=1, x_j = 1}^{N_t \log_2 M_c} L_{t,j}^{prior} \right) - \max_{\underline{x} \in L_t: x_i = 0} \left(-\frac{1}{N_0} \left\| \underline{y}_t - H_t \cdot map(\underline{x}) \right\|^2 + \sum_{j=1, x_j = 1}^{N_t \log_2 M_c} L_{t,j}^{prior} \right) - L_{t,i}^{prior} .(4.16)$$

Defining \underline{s} as $\underline{s} = map(\{x_i\}_{i=1}^{N_t \log_2 M_c})$, the list L_t of the N_{cd} points \underline{s} that make $\|\underline{y}_t - H_t \underline{s}\|^2$ smallest contains the maximizer of $-\frac{1}{N_0} \|\underline{y}_t - H_t \cdot map(\underline{x})\|^2 + \sum_{j=1, j \neq i, x_j=1}^{N_t \log_2 M_c} L_{t,j}^{prior}$ with high probability, if N_{cd} is sufficiently large. (Note that the point \underline{s} minimizing $\|\underline{y}_t - H_t \underline{s}\|^2$ ($N_{cd} = 1$) is not necessarily the desired maximizer. This is the main difference between the list sphere decoding [HS03] and the modified sphere decoding, in other words, we search the N_{cd} points \underline{s} that satisfy

$$\left\|\underline{y'}_{t} - H'_{t}\underline{s'}\right\|^{2} < r^{2}$$

$$(4.17)$$

where the real-valued signals $\underline{y'}_t$, H'_t and $\underline{s'}_t$ are defined as

$$\underline{y'}_{t} \coloneqq \left(\operatorname{Re}(\underline{y}_{t}^{T}) \operatorname{Im}(\underline{y}_{t}^{T}) \right)^{T}$$
$$H'_{t} \coloneqq \left(\begin{array}{c} \operatorname{Re}(H) & \operatorname{Im}(H) \\ -\operatorname{Im}(H) & \operatorname{Re}(H) \end{array} \right)$$

$$\underline{s'}_t := \left(\operatorname{Re}\left(\underline{s}_t^T\right) \operatorname{Im}\left(\underline{s}_t^T\right) \right)^T.$$

Assume that H'_t is expressed as $H'_t = QU$, where Q is an orthogonal matrix $(Q^T Q = I_{N_t})$ and U is a upper triangular matrix. Then (5.17) can be expressed as

$$\left\|U\left(\underline{\widetilde{s}}-\underline{s'}\right)\right\|^2 < r'^2 \coloneqq r^2 - \underline{y'}_t^T \left(I - H_t' \left(H_t'^T H_t'\right)^{-1} H_t'^T\right) \underline{y'}_t$$
(4.18)

where $\underline{\tilde{s}} = (H_t^{T} H_t^{T})^{-1} H_t^{T} \underline{y}^{T}$ is a zero-forcing (ZF) estimate of \underline{s}_t^{T} . Recalling that U is upper triangular, (4.17) can be expressed as

$$\sum_{n_1=1}^{2N_t} \left(\sum_{n_2=n_1}^{2N_t} u_{n_1,n_2} \left(\tilde{s}_{n_2} - s'_{n_2} \right) \right)^2 < r'^2,$$

and, from this, $2N_t$ necessary conditions hold true as follows.

This part is exactly the same as the original sphere decoding, except that the N_{cd} points \underline{s}' satisfying $\left\|\underline{y}'_t - H'_t \underline{s}'\right\|^2 < r^2$ should be chosen rather than a single point \underline{s}' minimizing $\left\|\underline{y}'_t - H'_t \underline{s}'\right\|^2$. Note that r should be carefully determined so that the number of the points \underline{s} within the sphere (4.17) is close to N_{cd} . Obviously, if r is chosen too small, only a few points \underline{s} can be found within the sphere. On the other hand, if r is chosen too large, the search may be slowed down.

4.4 System Simulation Results of Coded MIMO-OFDM

As system simulations for the performance evaluation, I use follow system configurations. The packet length is 972 bytes. As channel coding schemes, I use the low density parity check (LDPC) code whose code rate is 1/2 and parity check matrix is the one proposed by WWiSE group and the convolution code whose code rate is 1/2 and constraint length is 7. The number of coded bit of the list sphere detection (LSD) is 128 if $2^{N_T M_c} > 128$ else $2^{N_T M_c}$.

The channel which I used is the frequency selective channel with 3 delay-tapped model defined by 50 ns delay resolution and which has exponential power delay profile. For block fading models, the fading block of the slow model is one packet and of the fast model is an OFDM symbol.

The results from WWiSE group [WWiSE05] and TGn Sync group [TGn04a] are used as references. The WWiSE group used the channel D which is equivalently 3 delay-tapped model without spatial correlation. For the LDPCC decoding, 12 iterations were used and there is no description about the MIMO detection scheme. When using 16QAM and rate half channel coding, the SNR values to get the 10⁻¹ packet error rate (PER) are 18.5dB and 19.7dB with the LDPCC and Convolution code respectively as shown in figures 5.4 [WWiSE05].



Figure 4.4 The reference PER curves using LDPCC and convolution code by WWiSE

The TGn Sync group used the channel D which is equivalently 3 delay-tapped model with spatial correlation. Like WWiSE group, for the LDPCC decoding, 12 iterations were used and there is no description about the MIMO detection scheme. When using the modulation and coding set (MCS) 11 whose modulation is 16QAM and channel coding is rate half LDPCC, the SNR values to get the 10^{-1} PER is 20.0dB as shown in figures 5.5 [TGn04a].



Figure 4.5 The reference PER curves using LDPCC by TGn sync.

In simulations, the MMSE and the LSD are used as the MIMO detection schemes. The channel coding methods are LDPCC and convolution code and the LDPCC decoders are designed with the UMP-BP algorithm and LLR-BP [FMI99][CF02]. As the iterative decoding architectures, sequential decoding and the super-iterative decoding are used. To do same iterations for LDPCC decoding, when the sequential decoding is used, the number of the internal iteration of LDPCC decoder is 4 and when the super-iterative decoding is used, 2 super-iterations with 2 internal iterations of LDPCC decoder. The combinations of my system configurations are summarized in Table 4.1.

No. NAME	Channel	MIMO /	Channel Decoder		
	Encoder	System Dec.			
1. LDPC (UMP) -MMSE-	LDPC	MMSE	UMP LDPC decoder		
Sequential					
2. LDPC (LLR) -MMSE-	LDPC	MMSE	LLR LDPC decoder		
Sequential					
3. LDPC (UMP) - LSD-	LDPC	LSD	UMP LDPC decoder		
Sequential					
4. LDPC (LLR) - LSD-	LDPC	LSD	LLR LDPC decoder		
Sequential					
5. LDPC (UMP) - LSD-	LDPC	LSD	UMP LDPC decoder		
<i>Super iterative</i>					
6. LDPC (LLR) - LSD-	LDPC	LSD	LLR LDPC decoder		
Super iterative					

TABLE 4.1 PROPOSING SYSTEM CONFIGURATIONS

Following figures are the simulation results of PER performance for each system configuration evaluated on the slow block fading. The PER performance curves with 2x2, 3x3 and 4x4 antenna using QPSK modulation, 16QAM and 64QAM are shown in figure 4.6 to 4.14 respectively.



Figure 4.6 The PER performance curves of 2x2 QPSK



Figure 4.7 The PER performance curves of 2x2 16QAM



Figure 4.8 The PER performance curves of 2x2 64QAM



Figure 4.9 The PER performance curves of 3x3 QPSK



Figure 4.10 The PER performance curves of 3x3 16QAM



Figure 4.11 The PER performance curves of 3x3 64QAM



Figure 4.12 The PER performance curves of 4x4 QPSK



Figure 4.13 The PER performance curves of 4x4 16QAM



Figure 4.14 The PER performance curves of 4x4 64QAM

Simulation results based on slow fading channel are compared to based on fast fading environment. Following figures 4.15 and 4.16 are the performance comparisons between slow block fading and fast for 2x2 QPSK and 2x2 64QAM.



Figure 4.15 The PER performance comparison of 2x2 QPSK for slow and fast block fading





5. Hardware Block Design of LDPC Coded MIMO-OFDM

5.1 Introduction

The super iterative LDPC coded MIMO-OFDM system is analyzed in chap.4. The reviewed system is designed in a Verilog language and verified by the RTL simulator and a platform FPGA board. Especially, the new proposing two key blocks, the LSD-MIMO and the LDPC code are analyzed. The hardware architectures and operations are proposed and summarized in this chapter.

5.2 System Requirements

I analyze the system architecture for applying to promising IEEE802.11n proposals [WWiSE04][WWiSE05]. In this analysis, 20MHz bandwidth is used, the Short Inter-Frame Space (SIFS) or 16usec is allowed as decoding latency, up to 4 antennas are applied, modulation schemes are changed from QPSK to 64QAM, and 1944 coded bits are used for LDPCC codeword size. Table 5.1 summarizes the required throughput for each module. Every FFT module should operate in 4usec duration with 64 samples and the outputs of FFT are used for LSD processing also in 4usec slot. Although the throughput of each LSD is 54 vectors per 4usec, the number of coded bits containing in a vector are different for each system combination. Because the size of input codeword for LDPCC decoder is fixed to 1944 bits, the LLR generator and LDPCC decoder should start operation after receiving 1944 coded bits and finish it until receiving next 1944 coded bits.
	FFT	LSD	LLR gen + Decoder
2x2 QPSK	64samples/4us=16Msps	54vectors/4us =13.5Mvec/s	1944coded bits/36us=54Mcb/s
3x3 QPSK	64samples/4us=16Msps	54vectors/4us =13.5Mvec/s	1944coded bit/24us=81Mcb/s
4x4 QPSK	64samples/4us=16Msps	54vectors/4us =13.5Mvec/s	1944coded bit/18us=108Mcb/s
2x2 16QAM	64samples/4us=16Msps	54vectors/4us =13.5Mvec/s	1944coded bit/18us=108Mcb/s
3x3 16QAM	64samples/4us=16Msps	54vectors/4us =13.5Mvec/s	1944coded bit/12us=162 Mcb/s
3x3 16QAM	64samples/4us=16Msps	54vectors/4us =13.5Mvec/s	1944coded bit/9us=216 Mcb/s
2x2 64QAM	64samples/4us=16Msps	54vectors/4us =13.5Mvec/s	1944coded bit/12us=162 Mcb/s
3x3 64QAM	64samples/4us=16Msps	54vectors/4us =13.5Mvec/s	1944coded bit/8us=243Mcb/s
4x4 64QAM	64samples/4us=16Msps	54vectors/4us =13.5Mvec/s	1944coded bit/6us=324Mcb/s

TABLE 5.1 REQUIRED THROUGHPUT FOR GIVEN SYSTEM



Figure 5.1 Block interface with each data size

In figure 5.1, each 16-bit real and imaginary value is received through the antenna and analog to digital converters. The guard remover passes the 64 samples of 32-bit value after removing 16 guard samples out of 80 received samples. The FFTs store every 64 samples in them to calculate and send 54 data of 32-bit to the LSD module. Because the processing time of LSD has some variance, the interfaces between FFT and LSD and between LSD and MIMO block are designed with FIFO blocks. Our LSD searches 16 candidates and every candidate has distance value with 16 bits and symbol value with 24 bits. Our system architecture has two MIMO blocks. The 2nd MIMO is only for the super iterative decoding between the LDPCC decoder and the MIMO block. If the system has not the option for the super iteration, it can be excluded. The LDPCC decoder gets 1944 coded bits represented by 8 bits. The decoding time for one iteration is restricted by about 1 usec, the LDPCC decoder can use 6 iterations internally or externally. The timing relations for each block are represented in figure 5.2. The receiving data for 6 usec or 1 and half slot durations can be processed within 16 usec which is allowable time or a SIFS. This system can meet the latency and throughput requirements.



Figure 5.2 Timing diagram for each block operation

5.3 Hardware Components Design

5.3.1 LSD Design

5.3.1.1. Background of LSD

5.3.1.1.1. Channel Model

In this section, we present a brief review of the list sphere decoding algorithms for MIMO detection. At the figure 4.1 in previous chapter, $\mathbf{s} = \begin{bmatrix} s_1, ..., s_{N_t} \end{bmatrix}^T$ is a $N_t \times 1$ vector of transmitted symbols. s_i is chosen from complex constellation C with 2^{M_c} , $M_c \ge 1$ possible signal points and $E \|s_i\|^2 = E_s / M$.

 $s_i = map(\mathbf{x}^{<m>}), \quad m = 1, ..., N_t$, where $\mathbf{x}^{<m>}$ is a $M_c \times 1$ vector of coded data bits, and M_c is the number of bits per constellation symbol.

y is a $N_r \times 1$ vector of received signals.

H is a $N_r \times N_t$ iid zero-mean unit variance complex Gaussian matrix, known perfectly to the receiver.

n is a vector of independent zero-mean complex Gaussian noise entries with variance σ^2 per real component, and $\sigma^2 = \frac{N_0}{2}$

5.3.1.1.2. MAP Criterion and Iterative Decoding and Detection

In [HB03] and [LYW04], MAP (Maximum a Posteriori) criterion is well described. A posteriori log-likelihood ratio can be expressed as follow,

$$L_{D}\left(x_{k} \mid \mathbf{y}\right) = \ln \frac{\Pr\left[x_{k} = 1 \mid \mathbf{y}\right]}{\Pr\left[x_{k} = 0 \mid \mathbf{y}\right]} = \ln \frac{\Pr\left[x_{k} = 1\right]}{\Pr\left[x_{k} = 0\right]} + \ln \frac{\Pr\left[\mathbf{y} \mid x_{k} = 1\right]}{\Pr\left[\mathbf{y} \mid x_{k} = 0\right]}$$
(5.1)

The first term, $L_A(x_k) = \ln \frac{\Pr[x_k = 1]}{\Pr[x_k = 0]}$, is a priory log-likelihood ratio and the second, $L_E(x_k | \mathbf{y}) = \ln \frac{\Pr[\mathbf{y} | x_k = 1]}{\Pr[\mathbf{y} | x_k = 0]}$, is an extrinsic log-likelihood ratio.

An extrinsic value can developed as follow,

$$L_{E}(x_{k} | \mathbf{y}) = \max_{\mathbf{x} \in X_{k,1}} \left(-\frac{1}{N_{0}} \cdot \|\mathbf{y} - \mathbf{Hs}\|^{2} + \sum_{j \in J_{k,x}} L_{A}(x_{j}) \right)$$

$$- \max_{\mathbf{x} \in X_{k,0}} \left(-\frac{1}{N_{0}} \cdot \|\mathbf{y} - \mathbf{Hs}\|^{2} + \sum_{j \in J_{k,x}} L_{A}(x_{j}) \right) \quad (5.2)$$

where $X_{k,1} = \{\mathbf{x} | x_{k} = 1\}, X_{k,0} = \{\mathbf{x} | x_{k} = 0\}$ and
 $J_{k,x} = \{j | j = 0, ..., N_{t} \cdot M_{c} - 1, j \neq k, x_{j} = 1\}$

In equation (5.2), exhaustive search for whole possible \mathbf{x} is required for finding the maximum value. List sphere decoder searches these vectors only in the list whose number of elements is Ncd.

$$L_{E}\left(x_{k} \mid \mathbf{y}\right) \approx \max_{\mathbf{x} \in L_{k,1}} \left(-\frac{1}{N_{0}} \cdot \left\|\mathbf{y} - \mathbf{Hs}\right\|^{2} + \sum_{j \in J_{k,x}} L_{A}\left(x_{j}\right)\right)$$
$$- \max_{\mathbf{x} \in L_{k,0}} \left(-\frac{1}{N_{0}} \cdot \left\|\mathbf{y} - \mathbf{Hs}\right\|^{2} + \sum_{j \in J_{k,x}} L_{A}\left(x_{j}\right)\right)$$

List sphere decoder can be implemented by a simple modification to the sphere

decoder [FP85][HB][VB99]. The list sphere decoder does not decrease radius and add all searched points to list if the list is not full, or if the list is full, it compares the searched point in the list with the largest radius and replaces this point if the new point has smaller radius.

Figure 5.3 shows the relationship between components for iterative detection and decoding in the receiver.



Figure 5.3 Relationship between components for iterative detection and decoding

LSD generates the list which LLR gen uses for the calculation. LLR gen calculates an extrinsic LLR of each coded bit with priori LLRs from LDPC decoder according to the equation (5.2). The LDPC decoder calculates posteriori LLRs with extrinsic LLRs from LLR-gen.

5.3.1.2. Hardware Architecture of LSD

Figure 5.4 shows the brief block diagram of LSD.



Figure 5.4 Block diagram of LSD

With the channel matrix H and received vector y, LSD generates N_{cd} candidate bit vectors (\mathbf{x}_{cd}) and the distance ($||\mathbf{y}-\mathbf{H}\mathbf{s}_{cd}||^2$) of each bit vector. LSD consists of precomputation block and searcher block. Pre-computation block calculate QR decomposition and the pseudo inverse matrix of the channel matrix, and generate ZF estimator of the received vector. Searcher performs the tree searching and finds the candidate vectors whose distance is smaller than the initial radius.

5.3.1.2.1. QR Decomposition and Pseudo Inverse

QR decomposition is implemented using the scaled standard QR decomposition algorithm with Givens rotation and scaling [Davis03][GBS89]. The systolic architecture is selected for pipelined operation. Figure 5.5 shows the architecture of QR decomposition blocks, where V, R and d mean vectoring mode, rotation mode and delay respectively. The blocks for vectoring mode and rotation mode are shown in figure 5.6 and 5.7 respectively.



Figure 5.5 Architecture of QR decomposition



Figure 5.6 Operation of the vectoring mode





Figure 5.7 Operation of the rotation mode

Using the result of QR decomposition, the pseudo inverse matrix can be calculated easily. The result of scaled QR decomposition is

$$\mathbf{H} = \mathbf{B}^{T} \mathbf{Z}^{-1} \tilde{\mathbf{R}}, \quad \mathbf{B}^{T} = \prod G_{i,j}^{'}, \quad \mathbf{Z}^{-1} = diag \left(1/z_{1}, \cdots, 1/z_{2N_{r}} \right)$$
$$\Rightarrow \mathbf{Q} = \mathbf{B}^{T} \mathbf{Z}^{-1/2}, \quad \mathbf{R} = \mathbf{Z}^{-1/2} \tilde{\mathbf{R}}$$

The pseudo inverse matrix can be re-written as follow [Davis03],

$$\left(\mathbf{H}^{T} \mathbf{H} \right)^{-1} \mathbf{H}^{T} = \left(\left(\mathbf{B}^{T} \mathbf{Z}^{-1} \tilde{\mathbf{R}} \right)^{T} \mathbf{B}^{T} \mathbf{Z}^{-1} \tilde{\mathbf{R}} \right)^{-1} \left(\mathbf{B}^{T} \mathbf{Z}^{-1} \tilde{\mathbf{R}} \right)^{T} = \left(\tilde{\mathbf{R}}^{T} \mathbf{Z}^{-1} \tilde{\mathbf{R}} \right)^{-1} \tilde{\mathbf{R}}^{T} \mathbf{Z}^{-1} \mathbf{B}$$
$$= \tilde{\mathbf{R}}^{-1} \mathbf{B}$$

Using this manipulation, two Multiplication of matrix and an inversion of arbitrary matrix can be reduced a Multiplication of matrix and an inversion of upper triangular matrix.

5.3.1.2.2. Searcher

Search in LSD is same as that of the sphere decoder. Figure 5.8 shows tree searching of the sphere decoder for 2x2 QPSK mode.



Figure 5.8 Tree searching of the sphere decoding for 2x2 QPSK

$$\begin{split} &\sum_{i=1}^{4} r_{ii}^{2} \left| \left(s_{i} - \tilde{s}_{i} \right) + \sum_{j=i+1}^{N,M_{c}} \frac{r_{ij}}{r_{ii}} \left(s_{j} - \tilde{s}_{i} \right) \right|^{2} \le r'^{2} \\ & \Leftrightarrow \left(r_{4,4} \left(s_{4} - \tilde{s}_{4} \right) \right)^{2} < r'^{2} \\ & \left(r_{3,3} \left(s_{3} - \tilde{s}_{3} \right) + r_{3,4} \left(s_{4} - \tilde{s}_{4} \right) \right) + \left(r_{4,4} \left(s_{4} - \tilde{s}_{4} \right) \right)^{2} < r'^{2} \\ & \dots \\ & \left(r_{1,1} \left(s_{1} - \tilde{s}_{1} \right) + r_{1,2} \left(s_{2} - \tilde{s}_{2} \right) + r_{1,3} \left(s_{3} - \tilde{s}_{3} \right) + r_{1,4} \left(s_{4} - \tilde{s}_{4} \right) \right)^{2} \\ & + \dots + \left(r_{3,3} \left(s_{3} - \tilde{s}_{3} \right) + r_{3,4} \left(s_{4} - \tilde{s}_{4} \right) \right) + \left(r_{4,4} \left(s_{4} - \tilde{s}_{4} \right) \right)^{2} + \left(r_{4,4} \left(s_{4} - \tilde{s}_{4} \right) \right)^{2} \le r'^{2} \end{split}$$

At each node, the temporal distance is calculated, compared with the initial radius for decision of further searching.

The cores of tree searching and node evaluation are designed. The core consists of

calculator of the metric,
$$\sum_{i=1}^{N_i M_c} \left| r_{ii} \left(s_i - \tilde{s}_i \right) + \sum_{j=i+1}^{N_i M_c} r_{ij} \left(s_j - \tilde{s}_i \right) \right|^2$$
, and path decision block which

decide the next node for calculator. These two blocks communicate via an instruction which is described in figure 5.9. The instruction has level field, symbol index field and distance field. Level field means the current depth of node in the tree, symbol index field indicates the symbols of current path, and distance field means the previous and current distance of the path which is indicated in the symbol index field. Distance field at the input of the calculator means the previous distance, but it is the current distance at the output of the calculator.



Figure 5.9 The cores of tree searching and node evaluation

Calculator calculates $\left(r_{ii}\left(s_{i}-\tilde{s}_{i}\right)+\sum_{j=i+1}^{N_{i}M_{c}}r_{ij}\left(s_{j}-\tilde{s}_{i}\right)\right)^{2}$ and add it to previous distance.

 r_{ii} , r_{ij} , \hat{s}_i and \hat{s}_j is the constant value from the received vector in this case. s_i , s_j , i and previous distance are extracted from the instruction.

Calculator and path decision block spend one cycle for the operation, respectively, so that a calculator can support two path decision blocks.

Figure 5.10 shows the timing diagram of the operation of three blocks.



Figure 5.10 Timing diagram of searching operation

Path decision block decides the next node according to the distance field. If the distance field is smaller than the initial radius, the next node will be the first child node of the current node, and if the distance field is larger than the radius, the next node will be the neighbor of the mother node of the current node. The distance field is stored for next operation when the distance is smaller than the radius.

5.3.2 LDPCC Codec Design

5.3.2.1. Background of LDPC Codes

5.3.2.1.1. Encoding Algorithm

For encoding, the parity check matrix is converted to a systematic generator matrix by performing the Gaussian Elimination and through the rearrangement of columns. For an (n, j, k) code, at least (j - 1) rows in each matrix of the ensemble are linearly dependent [Gal63]. This means that the code has a slightly higher information rate than the matrix indicates. In our case, the parity check matrix \mathbf{H}_{org} has 972 rows, 1944 columns, j = 4, k = 8.

After performing the Gaussian Elimination, the parity check matrix \mathbf{H}_{sys} is obtained. Let's assume that the \mathbf{H}_{sys} so obtained is in a systematic form and can be represented by $[\mathbf{I}|\mathbf{P}]$, where \mathbf{I} denotes the identity matrix and \mathbf{P} is the parity check matrix. The assumption is made to explain the algorithm in a simplified manner. If one does not get the parity check matrix in this form, all one needs to do is record the columns that are the columns of the Identity matrix.

The Systematic Generator matrix \mathbf{G}_{sys} is represented by $[\mathbf{P}^{T}|\mathbf{I}]$ which will be used for encoding.

The matrices \mathbf{H}_{sys} and \mathbf{G}_{sys} should satisfy the following constraint:

$$\mathbf{H}_{sys}\mathbf{G}^{T}_{sys} = 0.$$
 (5.3)

The codeword can then be generated by applying the modulo-2 addition on the result of the vector matrix multiplication

$$\left(\mathbf{mG}_{sys}\right) \mod 2 = \mathbf{C} \tag{5.4}$$

where **m** is the $[1 \times 972]$ message vector, **G**_{sys} is the $[972 \times 1944]$ systematic generator matrix, and **C** is the $[1944 \times 1]$ codeword.

5.3.2.1.2. Decoding Algorithm

Both soft decision decoding and hard decision decoding can be used for decoding

LDPC codes. The bit flipping algorithm is a kind of hard decoding with low complexity. However, soft decoding archives a much better performance, and iterative decoding based on the belief propagation is a kind of soft decoding. Unfortunately, the hardware implementation of the belief propagation (BP) algorithm is limited by its complexity. Other algorithms with less complexity can be considered in soft decoding. In [FMI99] and [CF02], a reduced complexity iterative decoding algorithm based on BP algorithm is proposed for LDPC codes, and is referred to as the uniformly most powerful (UMP) BPbased algorithm. This algorithm needs only additions and comparisons instead of multiplications, dividers, exponentials, and logarithms or hyperbolic tangents and logarithms.

The notations defined in [CF02] are summarized as below. A log-likely-hood ratio (LLR) passing along the edge connection bit node n and check node m provides information about the hard decision of bit n and the reliability of this decision, according to all the information propagated to bit node n or check node m. Moreover, the following notations associated with a given iteration are defined.

 F_n : The LLR of bit *n* which is derived from the received value y_n . In decoding, $F_n = (4/N_o) y_n$ is initially used.

 $L_{m,n}$: The LLR of bit *n* which is sent from check node *m* to bit node *n*.

 $z_{m,n}$: The LLR of bit *n* which is sent from bit node *n* to check node *m*.

 z_n : The a posteriori LLR of bit n computed at each iteration.

The LLR BP algorithm used is derived as follows.

- Bit node to Check node Calculation and Decision

For each m, n, update z_{mn} by

$$z_{m,n} = F_n + \sum_{m \in M(n) \setminus m} L_{m,n}$$

$$z_n = F_n + \sum_{m \in M(n)} L_{m,n}$$

$$\left(\hat{c}_n = \begin{cases} 1, \ if \ z_n > 0\\ 0, \ if \ z_n < 0 \end{cases} \right)$$
(5.5)

Because the initial condition of $L_{m,n}$ is 0, z_n and $z_{m,n}$ are F_n .

- Check node to Bit node Calculation

For each m, n, update L_{mn} by

$$L_{m,n} = \ln \frac{1 - T_{m,n}}{1 + T_{m,n}}$$
where, $T_{m,n} = \prod_{n \in N(m) \setminus n} \frac{1 - \exp(z_{m,n})}{1 + \exp(z_{m,n})}$
(5.6)

or

$$L_{m,n} = \prod_{n' \in N(m) \setminus n} \operatorname{sgn}\left(z_{m,n'}\right) \times \phi^{-1}\left(\sum_{n' \in N(m) \setminus n} \phi\left(\left|z_{m,n'}\right|\right)\right) \times (-1)^{k}$$
where, $\phi(x) = \phi^{-1}(x) = -\log\left(\tanh\left(x/2\right)\right)$
(5.7)

Equation (5.7) can be represented by UMP BP-based algorithm as below equation (5.8) [CF02].

$$L_{m,n} \approx \ln \frac{1 - T_{m,n}}{1 + T_{m,n}} = (-1)^{\overline{\sigma_m \oplus \sigma_{m,n}}} \min_{n' \in N(m) \setminus n} |z_{m,n'}|$$

$$\begin{pmatrix} \sigma_{m,n} = \begin{cases} 1, if \ z_{m,n} > 0\\ 0, if \ z_{m,n} < 0\\ \sigma_m = \sum_{n \in N(m)} \sigma_{m,n} \mod 2 \end{cases}$$
(5.8)

5.3.2.2. Simplification of Decoding Equations

Even though the low-density characteristics of LDPCC, equations (5.5) to (5.8), use m by n matrices for $z_{m,n}$ and $L_{m,n}$, their dimensions should be reduced to j by m after modification as described in this section. To explain the proposed equations, we offer an example for (6, 2, 3) LDPCC whose **H** matrix is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \,.$$

Instead of **H** matrix, we define m by $k \mathbf{Q}$ matrix which represents the position of ones for each row.

$$\mathbf{Q} = \begin{bmatrix} 0 & 3 & 4 \\ 1 & 2 & 5 \\ 0 & 2 & 5 \\ 1 & 3 & 4 \end{bmatrix}$$

Figure 5.11 shows the meaningful contents of $z_{m',n'}$ and $L_{m',n'}$ for each row. Matrices $z'_{l',n'}$ and $L'_{l',n'}$ are reduced matrices of $z_{m',n'}$ and $L_{m',n'}$ respectively. Where $l' = \lfloor m'/(m/j) \rfloor$ and can be from 0 to *j*.

$$\begin{aligned} z_{m',n'} &= z'_{\lfloor m''(m/j) \rfloor,n'} = z'_{l',n'} \\ &= F_{n'} + \sum_{l' \in \{0,1,\dots,j-1\} \setminus l'} L'_{l',n'} \\ z_{n'} &= F_{n'} + \sum_{l' \in \{0,1,\dots,j-1\}} L'_{l',n'} \\ \left(\hat{c}_{n'} &= \begin{cases} 1, \ if \ z_{n'} > 0 \\ 0, \ if \ z_{n'} < 0 \end{cases} \right) \end{aligned}$$
(5.9)

$$L_{m',n'} = L'_{\lfloor m''(m/j) \rfloor, \mathcal{Q}_{m',p}} = L'_{l', \mathcal{Q}_{m',p}}$$

$$= \prod_{p' \in \{0, 1, \dots, k-1\} \setminus p} \operatorname{sgn} \left(z'_{l', \mathcal{Q}_{m',p'}} \right)$$

$$\times \phi^{-1} \left(\sum_{p' \in \{0, 1, \dots, k-1\} \setminus p} \phi \left(\left| z'_{l', \mathcal{Q}_{m',p'}} \right| \right) \right) \times (-1)^{k}$$
where, $\phi(x) = \phi^{-1}(x) = -\log(\tanh(x/2))$
(5.10)



Figure 5.11 Representations of matrices

As examples, we calculate z_4 , $z_{3,4}$ and $L_{3,4}$ with new proposing equations as below.

$$\begin{split} z_4 &= F_4 + \sum_{\substack{i' \in \{0,1\}}} L'_{i',4} = F_4 + L'_{0,4} + L'_{1,4} \\ z_{3,4} &= z'_{\lfloor 3/(4/2) \rfloor,4} = z'_{1,4} \\ &= F_4 + \sum_{\substack{i' \in \{0,1\} \setminus \{1\}}} L'_{i',4} = F_4 + L'_{0,4} \end{split}$$

$$L_{3,4} = L'_{\lfloor 3/(4/2) \rfloor, Q_{3,2}} = L'_{1,4}$$

= $\prod_{p' \in \{0,1,2\} \setminus 2} \operatorname{sgn}(z'_{1,Q_{1,p'}})$
 $\times \phi^{-1} \left(\sum_{p' \in \{0,1,2\} \setminus 2} \phi(|z'_{1,Q_{1,p'}}|) \right) \times (-1)^3$

5.3.2.3. Hardware Design of LDPCC Decoder

The bit to check calculation unit described by equation (5.9) can be represented as figure 5.12.



Figure 5.12 Block diagram of bit-to-check calculation unit

It uses an adder with (j+1)-input, and j subtractors with 2-input for each n value, and must run n times to finish the entire bit to check operation. That is to say, the calculation for the bit to check needs n additions and nj subtractors and requires n clock duration if nadditions and nj subtractors can be calculated with 1 clock. The check to bit calculation unit described by equation (5.10) can be represented as figure 5.13.



Figure 5.13 Block diagram of check-to-bit calculation

Effective k column-values, $z'_{l,Q_{m,k}}$, out of *n* should be selected for each row *m*. These values are used as inputs of sign calculation block and magnitude calculation block. To get the magnitude, each input enters $\phi(x)$ function. As shown in figure 5.14, $\phi(x) = \phi^{-1}(x) = -\log(\tanh(x/2))$ can be plotted and it is tabularized for hardware design.



Figure 5.14 Graphs of tanh(x/2) and $\phi(x)$

The check node to bit node calculation unit requires $2k \ \phi(x)$ functions, an adder with *k*-input, and *k* subtractors with 2-input.

Figure 5.15 shows a top block diagram of the LDPCC decoder which is designed by the described two basic calculation units.



Figure 5.15 Top block diagram of LDPCC Decoder

5.4 Adapting Hardware Optimizing Techniques

Designed hardware block should be met the system requirements as described before. The first step to refine a block is to analyze it from the architectural level. To demonstrate hardware refinement, the components of LDPCC decoder are analyzed.

To enhance the throughput of this calculation, the structure can be expanded in parallel. The expanded x ($1 \le x \le n$) units calculate n/x times to finish the entire calculation of bit to check. With x times resources, x times faster calculation is acquired and summarized in Table 5.2.

x	1	8	n=1944
j-input adder	1	8	n=1944
2-input subtractor	<i>j</i> =4	8 <i>j</i> =32	1944j=7776
Clock count	n=1944	n/8=243	1
Throughput (at 100MHz)	50Mbps	400Mbps	97.2Gbps
Latency	19.4usec	2.43usec	10nsec

TABLE 5.2 RESOURCE ANALYSIS OF BIT NODE TO CHECK NODE CALCULATION

To complete the check node to bit node calculation, this unit should be run *m* times. Its parallelization is how many units are used for repetitious *m* calculations. Parallelized *y* $(1 \le y \le m)$ units require *y* times resource and l/y times operation time and are summarized in Table 5.3.

У	1	4	<i>m</i> =972
$\phi(x)$	2 <i>k</i> =16	8 <i>k</i> =64	1944k=15552
k-input adder	1	4	<i>m</i> =972
2-input subtractor	<i>k=8</i>	4 <i>k</i> =32	972k=7776
Clock count	<i>m</i> =972	m/4=243	1
Throughput (at 100MHz)	100Mbps	400Mbps	97.2Gbps
Latency	9.72usec	2.43usec	10nsec

TABLE 5.3 RESOURCE ANALYSIS OF CHECK NODE TO BIT NODE CALCULATION

The total resource of the LDPCC decoder is summarized as Table 5.4 from Table 5.2 and 5.3. All resources are to one time iteration of decoding.

х,у	1,1	8,4	n,m=1944,972
$\phi(x)$	2k=16	8k=64	1944k=15552
j-input adder	1	8	n=1944
k-input adder	1	4	<i>m</i> =972
2-input subtractor	j+k=12	<i>8j+4k=64</i>	1944j+972k =15552
Clock count	2916	486	2
Throughput (at 100MHz)	33.3Mbps	200Mbps	48.6Gbps
Latency	29.16usec	4.86usec	20nsec

TABLE 5.4 RESOURCE ANALYSIS OF LDPCC DECODING (FOR 1 ITERATION)

5.5 Implementation on Platform Board

The system is implemented in a FPGA platform board. The recovered data are measured by logic analyzer and compared with golden data. Figure 5.16 shows the one of the simulated test-patterns, the result of the 4x4 LSD MIMO operation.

	G62		tr																
out_valid all	0			0	0	0	0	0	0	0	0		0	0		0	0	0	0
gold_data all		0		0	0	0	0	0	0	0	0		0	0		0	0	0	0
rec_data all	1		0	0	0	0	o	0	0	0	0		0	0		0	0	0	0
rec_llr all	+0000	0			◆ 36	• 36	◆536	◆ 536		◆ 536	◆ 36		• 36					• 36	◆ 36
num_tot_cd all	000																		
compare all								0)										
in_req all	0			0	0	0	0	0	0	0	0		0	0	0	0	0	0	0
	82		I										1		1			I	
out_valid all	0						1										0		
gold_data all	0	1				1		1		0	:	1				0			
rec_data all	0	1				1		1		0	:	1				0			
rec_llr all	-01536	;			◆ 3	6 🔸	36	◆ 1536		◆36						-0:	1536	5	
num_tot_cd all	024									000									
compare all								0											
in_req all	0									0									
Þ																			

Figure 5.16 Measurement for the 4x4 LSD MIMO block in a platform FPGA

6. Trade-offs of Performance and Cost

6.1 Introduction

The same date can be transmitted by several system combinations. The system with the 4x4 antennas and QPSK modulation can handle the 54Mbps data. This amount of data also can be transmitted with the 2x2 antennas and 16QAM modulation. In this chapter, The example system combinations to send same amount of data is compared and summarized. For a given situation, best system combination should be selected and operated.

6.2 Hardware Resource Estimation

For 100MHz system clock, below table 6.1 shows the relation of FFT performance and resource.

	LUT	MUL	Block RAM	Latency	Throughput	
2 Stong	16236	144	0	0.02 4500	100.0 Mana	
5 sieps	24.0%	100.0%	0.0%	0.05 usec	100.0 Msps	
12 Stops	9599	59	0	0.12 4500	100.0 Mana	
12 steps	14.2%	41.0%	0.0%	0.12 usec	100.0 Msps	
18 Stons	6650	12	0	0.18 11500	100.0 Mana	
40 Sieps	9.8%	8.3%	0.0%	0.40 usec	100.0 Msps	
RASDC	2060	8	0	1 38 11500	$100.0 M_{\rm SDS}$	
R4SDC	3.0%	5.6%	0.0%	1.50 usec	100.0 Msps	
R2 Mamory	869	4	0	1.05 usec	32 8 Mana	
R2 Memory	1.3%	2.8%	0.0%	1.95 <i>usec</i>	52.8 Msps	

TABLE 6.1 RESOURCES AND PERFORMANCE OF FFT

As shown below table, LSD block is calculated for using 400MHz system clock.

	Slice	Multiplier	Block RAM	Latency	Throughput
2x2 ODSK	4611	183	0	12	33.33 Mvec/sec
2x2 QI SK	13.6%	127.1%	0.0%	45	8.89 Mvec/sec
And ODSK	16347	355	0	24	16.67 Mvec/sec
4x4 QPSK	48.4%	246.5%	0.0%	102	3.92 Mvec/sec
222 160 MM	4611	183	0	12	33.33 Mvec/sec
2x2 TOQAM	13.6%	127.1%	0.0%	64	6.25 Mvec/sec
1×1 610 1M	16400	275	0	24	16.67 Mvec/sec
4 <i>x</i> 4 04QAM	48.5%	191.0%	0.0%	717	0.56 Mvec/sec

TABLE 6.2 RESOURCES AND PERFORMANCE OF LSD

As shown below table 6.3 and 6.4, the 1st and the 2nd MIMO block are calculated for using 100MHz system clock respectively.

	Slice	Multiplier	Block RAM	Latency	Throughput
$2r^2 OPSK$	558	32	0	2	100.0 Mcb/s
$2\lambda 2 QI SK$	1.7%	22.2%	0.0%		
And ODSV	1116	64	0	2	200.0 Mcb/s
$4\chi 4 QF SK$	3.3%	44.4%	0.0%		
222 160 111	1116	64	0		200.0 Mcb/s
2x2 I 0 QAM	3.3%	44.4%	0.0%		
1×1 6101M	1674	96	0	2	300.0 Mcb/s
414 04QAM	5.0%	66.7%	0.0%		

TABLE 6.3 RESOURCES AND PERFORMANCE OF 1^{ST} MIMO

	Slice	Multiplier	Block RAM	Latency	Throughput
2x2 QPSK	558	32	0	2	200.0 Mcb/s
	1.7%	22.2%	0.0%		
4x4 QPSK	1116	64	0	2	400.0 Mcb/s
	3.3%	44.4%	0.0%		
$2x^2$	1116	64	0		400.0 Mcb/s
Тодам	3.3%	44.4%	0.0%		
4x4	3348	192	0	2	1200.0 Mcb/s
04QAM	9.9%	133.3%	0.0%		

TABLE 6.4 RESOURCES AND PERFORMANCE OF 2ND MIMO

The resources for LDPCC are summarized in Table 6.5.

TABLE 6.5 RESOURCES AND PERFORMANCE OF LDPCC DECODER

	Slice	Multiplier	Block RAM	Latency	Throughput	
Minimum	467	0	28	2916	266 7 Mcb/s	
Resource	1.4%	0.0%	19.4%	2710	200.7 1100/5	
Sha lah	11200	0	128	186	1600 0 Mah/s	
0 <i>0-c</i> , 4 <i>C-0</i>	33.1%	0.0%	88.9%	400	1000.0 MCD/S	

Based on above summarizing tables, (6.1) to (6.5), the required gates for some transmission modes are estimated as below table (6.6) to (6.9) for each system configuration.

	Slice	Mul	Block RAM	Est. gate
FFT	3325	12		67225
MMSE	1564	59		138332
LDPC	11200		128	145600
Total	16089	71	128	351157

TABLE 6.6 RESOURCES FOR 2x2 QPSK USING MMSE DETECTION

TABLE 6.7 RESOURCES FOR 2x2 QPSK USING LSD DETECTION

	Slice	MUL	Block RAM	Est. gate
FFT	3325	12		67225
LSD	4611	183		425943
MIMO	558	32		71254
LDPC	11200		128	145600
Total	29669	263	128	777247

TABLE 6.8 Resources for 2x2 QPSK using LSD detection and Super-iteration

	Slice	Mul	Block RAM	Est. gate
FFT	3325	12	67225	67225
LSD	4611	183		425943
MIMO I	558	32		71254
MIMO II	2224	128		284912
LDPC	11200		128	145600
Total	21918	355	67353	994934

	Slice	Mul	Block RAM	Est. gate
FFT	3325	12		67225
MMSE	1564	59		138332
LDPC	11200		128	145600
Total	16089	71	128	351157

TABLE 6.9 RESOURCES FOR 2X2 16QAM USING MMSE DETECTION

6.3 Trade-offs of Throughput and Hardware Resource and Power

Table 6.10 summarizes the system configurations what to be compared each other. The 2x2 QPSK system is used as a reference system and the data rate of given system is increased by changing number of antenna or using higher modulation scheme. The LDPCC decoder is designed with the uniformly most powerful (UMP) belief propagation (BP) algorithm [CF02]. The code rate of LDPCC is 1/2. Total iteration number of the LDPCC decoder is same as 4 for any system combination. In this comparison, the packet size with 972 bytes is used. The packet error rate for each system combination is compared. In this comparison, the required SNR is the value to get the 10⁻¹ PER. In figure 6.1 showing the PER performances, it can be found that the system configuration 9 requires the minimum SNR about 8.2dB and the required

The SNR of the 4x4 QPSK system is less than that of the 2x2 16QAM system for each detection or decoding method. But it can be said that the 4x4 QPSK systems are more complex than the 2x2 16QAM systems from figure 6.2 because the required gates of the former are more than that of the later. It also verified that the method for increasing data rate is using much more signal power or more hardware resources. In figure 6.3, the computation power for each hardware component, the number of gates, and the transmitting signal power are normalized and compared on the basis of the 2x2 QPSK system which uses MMSE detector and 4 LDPCC iterations. The MMSE detector consumes much transmitting signal power than that of the LSD detector but uses less computation power. These results give the guideline to select system architecture of the coded MIMO-OFDM system.

No.	Num of antenna	Modula tion	Data Rate (Mbps)		Number of iteration	
				MIMO Detection	LDPCC	Super
					internal	iteration
1	2x2	QPSK	27	MMSE	4	-
2	2x2	QPSK	27	LSD – Sequential	4	-
3	2x2	QPSK	27	LSD – Super iteration	2	2
4	2x2	16QAM	54	MMSE	4	-
5	2x2	16QAM	54	LSD – Sequential	4	-
6	2x2	16QAM	54	LSD – Super iteration	2	2
7	4x4	QPSK	54	MMSE	4	-
8	4x4	QPSK	54	LSD – Sequential	4	-
9	4x4	QPSK	54	LSD – Super iteration	2	2

 TABLE 6.10 System configurations



Figure 6.1 Packet error rate comparison for each system combination



Figure 6.2 Required SNR and number of gates for each system combination



Figure 6.3 Normalized comparisons on the basis of system configuration 1

7. Low Power Design for SoC Digital Design

7.1 Introduction

Nowadays low-power design has been of primary importance in many digital systems. Several useful techniques for low power design are summarized in this chapter, which include clock gating, operand isolation, and low power cell replacement. Particularly, these techniques are applied to the design of 4x4 LSD-MIMO systems. In the following two sections, a brief review of theoretical background is given and power calculation methods are summarized. In the last section of this chapter, the application of these techniques to 4x4 LSD-MIMO systems is summarized.

7.2 Techniques for the power constraints

7.2.1 Clock Gating

Because circuits are being developed with controllable clocks the switching of the clock causes many unnecessary switching activities. It implies that these unnecessary switching activities can be removed by disabling clock signals derived from a master clock signal. Obviously, clock gating significantly reduces power dissipation, which can be explained as follows [WPW00].

1) The load on the master clock is reduced and the number of required buffers in the clock tree is decreased. Therefore, the power dissipation of clock tree can be reduced.

2) The flip-flop receiving the derived clock signals are not triggered in idle cycles and
thus dynamic power dissipation can be reduced.

3) The excitation function of the flip flop triggered by the derived clock may be simplified since it has a don't-care condition in the cycle when the flip flop is not triggered by the derived clock.

The clock-gating problem has been studied in [TFS95] [AM94] [BM99]. In [TFS95] the authors presented a technique for saving power in the clock tree by stopping the clock fed into idle modules. However, a number of engineering issues in designing the clock tree were not addressed and, hence, the proposed approach has not been adopted in practice. In [AM94], a precomputation-based technique is used to generate a signal to control the load enable pin of the flip flops in the data path. The control signal is derived by investigating the relationship between the latched input and the primary outputs of the combinational blocks in the data path. The technique is useful only if the outputs of the block can be predicted for certain input assignments. In [BM99], the authors used a latch to gate the clock in control-dominated circuits. The problem is that the additional latch receives the clock's triggering signal, which results in extra power dissipation in the latch itself.



(a) Traditional sequential logic (b) Gated-clock version Figure 7.1 Traditional and Gated clock

A gated-clock circuit is obtained by modifying the architecture depicted in figure 7.1(a). A signal called activation function (F_a) selectively stops the local clock of the circuit, when the machine does not perform state or output transitions. When $F_a = 1$, the clock will be stopped. The sequential circuit with clock-gating logic is shown in figure 7.1(b). The block labeled "L" represents a latch that is transparent when the global clock signal CLK is low. The latch is required to guarantee correct behavior, because F_a may have glitches that must not propagate to the AND gate when the global clock is high. Moreover, notice that the delay of the logic for the computation of F_a may be on the critical path of the circuit, and its effect must be taken into account during timing verification. The activation function is a combinational logic block containing the primary inputs and state signals of the circuit as input variables. No external information is used; the only input data are the gate-level description of the circuit and the probability distributions of the input signals.

7.2.2 Operand Isolation

In the traditional implementation of a design, data-path operators are always operational. They dissipate power even when the output of the operators is not used. With the operand isolation approach, additional logic (AND or OR gates) called isolation logic is inserted along with an activation signal to hold the inputs of the data-path operators stable whenever their output is not used. A simple example of operand isolation is represented in figure 7.2.



Figure 7.2 Operand Isolation

7.2.3 Low-power cell replacement

7.2.3.1. Transistor Sizing

Transistor sizing is a circuit-level, low-power design technique that targets the shortcircuit power by enlarging/reducing the width of the channel of a transistor [BOI96] [YD05]. Changing the channel width of transistors affects both power and delay. The effect of transistor sizing can be seen as trading speed for lower power dissipation. Earlier approaches were based on the assumption that the power dissipation is proportional to the active area, i.e., the area occupied by active devices. Recent studies show that the power dissipation is a convex function to the active area. The short-circuit power dissipation is also represented as

$$P_{SC} = k\mu W\tau f \tag{7.1},$$

where k is a process and a voltage-dependent parameter, μ is the mobility of the carrier responsible for the transition, W is the width of the transistor, τ is the input transition time, and f is the clock frequency [BOI96].



Figure 7.3 Sample high fan-out gate structure for calculating optimum transistor size

Consider the case where a given CMOS gate drives a load of several other CMOS gates, as shown in figure 7.3. Gate g_1 drives $g_2, ..., g_n$. Let $W_1, ..., W_n$ denote the widths of the transistors of the gates $g_1, ..., g_n$, respectively. The input signal to g_1 has transition time τ . The signal transition time for the input to the gates $g_2, ..., g_n$ is τ_1 , which is also the output transition time of g_1 . The total power consumed by the circuit is given by

$$P = V^{2} \sum_{i=1}^{n} C_{L_{i}} f_{i} + c V^{2} W_{1} f_{1} + k \left(W_{1} \mu \tau f_{1} + \sum_{i=2}^{n} W_{i} \mu' \tau_{1} f_{i} \right).$$
(7.2)

By differentiating eq.(7.2) with respect to W_1 , the condition to get the minimum power is obtained by

$$W_{1}^{*} = \frac{\sqrt{\phi \frac{\mu'}{\mu} C_{L1} \left(\sum_{i=2}^{n} W_{i} f_{i}\right)}}{\sqrt{\left(\frac{k_{1}}{k} + \mu\tau\right) f_{1}}}.$$
(7.3)

 W_1^* is the power optimal size for the transistor in g_1 .

7.2.3.2. Threshold Voltage Scaling

The propagation delay of a CMOS inverter is inversely proportional to $V_{dd} - V_{th}$

$$T_g = K \frac{V_{dd}}{\left(V_{dd} - V_{th}\right)^{\alpha}}$$
(7.4)

where K is a proportionality constant given by the corresponding CMOS technology, and α is the power factor in the drain current equation. Obviously, as shown in equation (7.4), the propagation delay becomes smaller as the threshold voltage V_{th} . However, the decrease of V_{th} increases the leakage current as well. This increment of leakage current causes additional leakage power dissipation, which reduces the benefit of power reduction because of decreasing the supply voltage [GGH97]. A high- V_{th} transistor is used to minimize the leakage current, while a low- V_{th} transistor is used to increase the driving current and speed on the critical path [BKKSSB03] [SMFYSCWMMK97] [JKK02].

7.3 Power Modeling and Calculation

7.3.1 Static and dynamic power dissipations

The power dissipated in a circuit falls into two broad categories, one is the static

power and the other is dynamic power. Static power is dissipated by a gate when it is not switching, that is, when it is inactive or static. Static power is dissipated in several ways. Most of the static power results from source-to-drain sub-threshold leakage, which is caused by reduced threshold voltages that prevent the gate from completely turning off. Static power is also dissipated when current leaks between the diffusion layers and the substrate. For this reason, static power is often called leakage power. Dynamic power is the power dissipated when the circuit is active. A circuit is active whenever the voltage on a net and changes due to some stimulus applied to the circuit. Because voltage on an input net can change without necessarily resulting in a logic transition on the output, dynamic power can be dissipated even when an output net doesn't change its logic state. The dynamic power of a circuit is composed of two different kinds of power, switching power and internal power. The switching power of a driving cell is the power dissipated by charging and discharging of the load capacitance at the output of the cell. The total load capacitance at the output of a driving cell amounts to the sum of the net and gate capacitances on the driving output. Since such charging and discharging activities result from logic transitions at the output of the cell, switching power increases as logic transitions increase. Therefore, the switching power of a cell is a function of both the total load capacitance at the cell output and the rate of logic transitions. Internal power is any power dissipated inside the boundary of a cell. During switching activities, a circuit dissipates internal power by the charging or discharging of any existing capacitances

internal to the cell. Internal power includes power dissipated by a momentary short circuit between the P and N transistors of a gate, called short-circuit power. In most simple library cells, internal power is mostly due to the short-circuit power. For more complex cells, charging and discharging activities of internal capacitance may be a dominant source of internal power dissipation. [PCUG04]



Figure 7.4 Components of Power Dissipation (CMOS inverter)

7.3.2 Power calculation

7.3.2.1. Leakage power calculation

Power Compiler analysis computes the total leakage power of a design by summing the leakage power of the design's library cells, as shown in the following equation (7.5):

$$P_{LeakageTotal} = \sum_{All \, cells(i)} P_{CellLeakage_i} \tag{7.5}$$

where $P_{LeakageTotal}$ is the total power dissipation of the design and $P_{CellLeakage_i}$ is the leakage power dissipation of each cell *i*. As an example, the leakage power calculation performed on a NAND gate is described in figure 7.5.





Figure 7.5 Leakage power calculation example

For total power consumption time of 600, if the cell is at the state defined by the condition A&B 33% of the time. For the remaining 67% of the simulation time, the default value is assumed. Hence, the total cell leakage value is: $(.33 \times .021841nW) + (0.67 \times 32.615136pW) = 29.02376pW$.

7.3.2.2. Internal power calculation

A cell's internal power is the sum of the internal power of all of the cell's inputs and outputs as modeled in the technology library. As an example, the cell in figure 7.5 shows how tools calculate the internal power with path-dependent internal power modeling.

$$P_{\text{int}} = \sum_{i=A,B} E_{i \to Z} \times PathWeight_i \times TR_Z$$

$$E_{i \to Z} = f \left[C_{Load}, Trans_i \right]$$

$$\sum_{i=A,B} PathWeight_i = 1$$
(7.6)

where P_{int} is the total internal power of cell, $E_{i\rightarrow Z}$ is the internal energy for output Z as a function of input transitions, output load, TR_Z is the toggle rate of output Z whose unit is transitions per second, $Trans_i$ is the transition time of input *i*, and $PathWeight_i$ is the weighted average transition time for output Z.

7.3.2.3. Switching power calculation

The switching power can be calculated by equation (7.7).

$$P_{C} = \frac{V_{dd}^{2}}{2} \sum_{All \, nets(i)} \left(C_{Load_{i}} \times TR_{i} \right)$$
(7.7)

where C_{Load_i} is the capacitive load of net *i*, the V_{dd} is the supply voltage and TR_i is the toggle rate of net *i* whose unit is transitions per second. As shown in equation (7.7), the switching power is very sensitive to the supply voltage.

7.4 Applying Low-Power Techniques

Some EDA tools support to low-power design. The Power Compiler of Synopsys, the Cubic Power of Samsung, the PowerMill of Synopsys and HSPICE of Avanti calculate the consuming power of designed system. Moreover, Power Compiler of Synopsys supports several low-power techniques of other EDA tools. It is integrated in the Design Compiler or the Physical Compiler of Synopsys. Especially, low-power techniques such as clock gating, operand isolation and leakage and dynamic power optimization techniques are applied. Even though the clock gating and the operand isolation are the logical level optimizations, the leakage and dynamic power optimizations are accomplished by mixing of the high speed libraries and the low power libraries.

7.4.1 Low power design for LSD-MIMO block

For calculating switching power, the supply voltage of equation (7.7) is defined by the operating conditions. For the typical condition, it is specified 1.8V, for the slow condition, it is 1.62V, and for the fast condition, it is 1.98V. The capacitive load of net *i*, C_{Load_i} is the sum of all cell capacitances specified in the library. As an example, a NAND gate cell of the SMIC 0.18 library is defined in figure 7.6. From the cell library, the capacitive load of each net can be calculated. From the Verilog testbench, actual transitions for every port or net is counted and saved into the Switching Activity Information File (SAIF). Where T0, T1 and TX are the durations of time found in logic 0, 1 and unknown 'X' state respectively. TC is the sum of the rise and fall transitions that are captured during monitoring. IG is the number of glitches captured during monitoring. If the net *i* is the n12159 in figure 7.6, TR_i in equation (2.10) is TC/DURATION = 110/20125nsec = 5.466MHz.

(TIMESCALE 1 ns) (DURATION 20125.00)
<i>(</i>
(<i>n121</i> 59
(T0 17285) (T1 2840) (TX 0)
(TC 110) (IG 0)
)
•••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• ••• •••

Figure 7.6 Toggle report example in SAIF

```
cell (NAND2X1) {
  cell footprint : nand2;
  area : 9.979200;
  pin(A) {
    direction : input;
    capacitance : 0.004328;
  }
  pin(B) {
    direction : input;
    capacitance : 0.004047;
  }
  pin(Y) {
    direction : output;
    capacitance : 0.0;
    function : "(!(A B))";
    internal power() {
       related pin : "A";
       rise power(energy template 7x7) {
index 1 ("0.03, 0.1, 0.4, 0.9, 1.5, 2.2, 3");
index 2 ("0.00035, 0.021, 0.0385, 0.084, 0.147, 0.231, 0.3115");
values ()
            "0.012119, 0.012532, 0.012195, 0.010917, 0.008958, 0.006279, 0.003690",
            "0.013226, 0.012123, 0.011786, 0.010607, 0.008728, 0.006098, 0.003534".
            "0.019702, 0.016359, 0.014934, 0.012686, 0.010338, 0.007480, 0.004817",
            "0.031481, 0.025356, 0.022238, 0.017342, 0.013324, 0.009343, 0.006073",
            "0.046024, 0.038405, 0.034076, 0.027486, 0.021596, 0.016037, 0.011788",
            "0.063165, 0.053958, 0.049018, 0.039885, 0.032038, 0.024645, 0.019157",
            "0.082879, 0.072237, 0.066818, 0.055268, 0.045530, 0.036226, 0.029350");
       fall power(energy template 7x7) {
                      -----
max capacitance : 0.311500;
  ł
  cell leakage power: 32.615136;
```

Figure 7.7 Cell library example of SMIC 0.18

In order to calculating internal power, the TR_Z and $PathWeight_i$ of equation (7.6) are acquired from SAIF and $E_{i\rightarrow Z}$ is obtained from cell library. In the part of internal power of figure 7.7, 'index_1' is $Trans_i$, 'index_2' is the capacitive load and 'values' are tabulated contents of $E_{i\rightarrow Z}$ for each index_1 and index_2.

From the equation (7.5), the leakage power is equal to the total sum of every cell leakage power noted in cell library.

By the power compiler, the power dissipation of the LSD-MIMO block is calculated as 13.692mW assuming that the switching power is 2.508mW (18.32%), the internal power is 11.165mW (81.54%), and the leakage power is 19.2uW(0.14%). The total power dissipation ratio for the sub-blocks of LSD-MIMO is summarized in table 7.1 and figure 7.8.

	Switch Power	Int Power	Leak Power	Total Power
LLR_gen	0.132	0.859	5.96E-4	0.992
LSD	0.209	3.899	1.57E-2	4.124
ZF_est_4x4	2.166	6.406	2.97E-3	8.576
Total	2.508	11.165	1.92E-2	13.692

TABLE 7.1 POWER DISSIPATION RATIO FOR THE SUB-BLOCKS OF 4X4 LSD-MIMO



Figure 7.8 Power dissipation ratio for the sub-blocks of 4x4 LSD-MIMO Based on the above results, several efforts to reduce its dissipating power with these techniques are described in the next section.

7.4.1.1. Clock gating

By Power Compiler, the synchronous load-enable register with multiplexer can be adapt to gated clock technique. The RTL code in the following box shows that the DATA_OUT is replaced by the DATA_IN only if rising edge of CLK and EN is asserted. It can be translated as shown in figure 7.9(a) by traditional synthesis tools. The input of register bank is controlled by the MUX to select new input data of DATA_IN or previous DATA_OUT. Even when EN is low, the MUX logic should operate and the clock line of register bank has switching activity.

```
always @ (posedge clk)
if (EN)
DATA_OUT <= DATA_IN;
```

Adapting clock gating techniques, the previous RTL is changed as follows. The enabling signal is only used to make new gated clock and newly made clock is asserted to the register bank.



(b) Adapting gated clock

Figure 7.9 Operation of register bank after and before adapting gating clock

The total number of registers of the 4x4 LSD-MIMO system is 5063. The 4662 registers (92.08%) are converted to gated registers by the 212 clock gating elements.

The relative power dissipation for the registers, $P_{relative, reg}$, after adapting clock gating can be simply denoted as below.

$$P_{relative, reg} = \frac{\sum_{\substack{all \ gating \ element(i)}} \left[R_{eff,i} \times Num_i \right] + Num_{ungated}}{\sum_{\substack{all \ gating \ element(i)}} Num_i + Num_{ungated}}$$
(7.8)

where *i* is the number of gating element and $R_{eff,i}$ is the enabling rate of *i*-th control signal. The relative power dissipation for the total cells, $P_{relative,total}$, after adapting clock gating can be estimated by relatively occupied area of register.

$$P_{relative,total} = \left(1 - \frac{A_{reg}}{A_{total}}\right) + \frac{A_{reg}}{A_{total}} \times P_{relative,reg}$$
(7.9)

In the LLR-generator block, 517 registers out of 566 registers are gated with 33 gating groups. The 33 gating groups divided into 2 groups, the first group has 512 registers with enabling rate 0.177% and the second group has 5 registers with enabling rate 5.651%. The relative power dissipation of gated register is

$$\frac{\left\{ \left(0.00177 \times 512\right) + \left(0.0565 \times 5\right) \right\} + 49}{\left(512 + 5\right) + 49} = \frac{50.186}{566} = 0.0887.$$

Because the total area of LLR-generator is $144482\mu m^2$ and approximated occupied area of the flip-flops is $566 \times 55 = 31130\mu m^2$, about 21.55% of total area can be assumed to be used by flip-flops. The relative power dissipating by clock gating is expected about $(1-0.2155)+0.2155\times0.0887=0.8036$. The power improving ratio after applying clock gating is summarized in table 7.2.

TABLE 7.2 POWER IMPROVING RATIO AFTER CLOCK GATING

	Switch Power	Int Power	Leak Power	Total Power
LLR_gen	-12.88%	37.02%	0.84%	30.34%
LSD	-146.89%	39.63%	0.00%	30.02%
ZF_est_4x4	-0.55%	1.09%	0.00%	0.69%
Total	-13.32%	17.30%	-0.52%	11.66%

The improving ratio for the LLR-generator by clock gating is 30.34% even though about 19.6% reduction is expected.

The ZF_est_4x4 block has 129 registers or 8 16-bit registers and a 1-bit register. These registers occupy the cell area about $129 \times 55 = 7095 \mu m^2$. Because total area of the ZF_est_4x4 block is $583866.3 \mu m^2$, registers are only 1.25% of total area. Therefore utilizing clock gating does not significantly reduce the consuming power for ZF_est_4x4.

7.4.1.2. Operand Isolation

Although operand isolation has much potential to significantly reduce power dissipation, in some cases, power consumption can be increased after isolation logic is applied. If the activation signal of the operator and the data inputs are strongly correlated, the application of operand isolation may increase power dissipation. A strong correlation means that the data inputs toggle only in the active periods. Because the searching units of the LSD are activating all the time, adopting operand isolation uses additional useless gates to make unnecessary activating signals for all operands. As shown in table 7.3 and 2.4, the LSD block after applying operand isolation consumes more power than that of unapplied one.

TABLE 7.3 POWER DISSIPATION RATIO AFTER CLOCK GATING AND ENTIRE ISOLATING OPERANDS

	Switch Power	Int Power	Leak Power	Total Power
LLR_gen	8.43E-02	0.36	6.01E-04	0.445
LSD	2.125	6.246	3.20E-02	8.404
ZF_est_4x4	1.98E-03	1.14E-02	3.00E-03	1.64E-02
Total	2.212	6.621	3.56E-02	8.868

TABLE 7.4 POWER APPROVING RATIO TO CLOCK GATING AFTER OPERAND ISOLATING

	Switch Power	Int Power	Leak Power	Total Power
LLR_gen	43.42%	33.46%	-1.69%	35.60%
LSD	-311.82%	-165.34%	-103.82%	-191.20%
ZF_est_4x4	99.91%	99.82%	-1.01%	99.81%
Total	22.17%	28.29%	-84.46%	26.68%

The radius calculating block, ZF_est_4x4, has multipliers and adders whose outputs are valid for only one clock period. These operations are selected and operated only if *in vld* signal is asserted. The verilog code is designed as below box.

wire signed [31:0] elm1 = pil1*y1 + pil2*y2 + pil3*y3 + pil4*y4; always @(posedge clk) begin if (rst_n == 1'b0) s1_ <= 0; else if (in_vld == 1'b1) s1_ <= elm1[23:8]; end

The code is interpreted as figure 7.10(a).



Therefore, most of operations are just power dissipating meaninglessly. By adopting operand isolation method, the corresponding block reduces the consuming power by about 99.8%.

However, for the LSD block, power dissipation is almost doubled after adopting operand isolation technique. Because most nets are the outputs of clock gated registers, additional use of the scheme induces gates to make useless activating signals. Therefore, the LSD block is replaced what is unapplied operand isolation.

	Switch Power	Int Power	Leak Power	Total Power
LLR_gen	8.43E-02	0.36	6.01E-04	0.445
LSD	0.516	2.354	1.57E-02	2.886
ZF_est_4x4	1.98E-03	1.14E-02	3.00E-03	1.64E-02
Total	6.02E-01	2.7254	1.93E-02	3.3474

TABLE 7.5 POWER DISSIPATION AFTER CLOCK GATING AND PARTIAL OPERAND ISOLATION

TABLE 7.6 POWER APPROVING RATIO TO CLOCK GATING AFTER PARTIALLY OPERAND ISOLATING

	Switch Power	Int Power	Leak Power	Total Power
LLR_gen	43.42%	33.46%	-1.69%	35.60%
LSD	0.00%	0.00%	0.00%	0.00%
$ZF_{est_{4x4}}$	99.91%	99.82%	-1.01%	99.81%
Total	78.81%	70.48%	-0.01%	72.32%

7.4.1.3. Low-power cell replacement

After RTL clock gating and operand isolation, gate-level dynamic power optimization further reduces the dynamic power. Dynamic power optimization is an additional step to the timing optimization and depends on the switching activity. Well-balanced runtime is achieved by replacing slowly operating low-power cell. Also the leakage power optimization can be performed on the non-critical paths. This is also a kind of timing optimization step. Therefore, using multi-threshold voltage cell and small fan-out low power cell are replaced until having same critical path delays. Even though the power reduction ratio is dependent on how many paths are different from critical path, it is said that designers can realize average dynamic power reductions of 10 % and leakage power reduction of 30% compared to designs optimized for timing and area only [PCUG04]. Because the leakage power is less than 0.1% of total power, replacing low power cell with the power constraint gives about 10% reduction of total power dissipation. Table 7.7 and 7.8 summarize the power dissipation after applying power constraint to replace low power consuming cells which are not concerned to critical path. The power reduction of 18.71% better than average power reduction rate, 10% is achieved.

	Switch Power	Int Power	Leak Power	Total Power
LLR_gen	6.46E-02	0.317	5.50E-04	0.382
LSD	0.39	1.92	1.45E-02	2.32
ZF_est_4x4	1.60E-03	1.03E-02	2.80E-03	1.47E-02
Total	4.56E-01	2.247	1.79E-02	2.721

TABLE 7.7 POWER DISSIPATION AFTER CELL REPLACEMENT

TABLE 7.8 POWER APPROVING RATIO TO PARTIAL OPERAND ISOLATING AFTER CELL REPLACEMENT

	Switch Power	Int Power	Leak Power	Total Power
LLR_gen	23.37%	11.94%	8.49%	14.16%
LSD	24.42%	18.44%	7.64%	19.46%
ZF_est_4x4	19.19%	9.65%	6.67%	10.37%
Total	24.25%	17.54%	7.52%	18.71%



7.4.2 Summary of adapting low power technique

Figure 7.11 Total power dissipation summary for each reduction technique

In this chapter, the power dissipation of digital circuit has been reviewed. In particular, several low-power techniques such as clock gating, operand isolation, and low power cell replacement have been summarized and applied to the 4x4 LSD-MIMO system. It was shown that, by adopting all kinds of listed techniques, the power reduction of 80.1% has been achieved. The results are represented in figure 7.11.

8. Conclusions

In this dissertation, the emerging 802.11n system has been designed and implemented as a target system. The novelties of this dissertation are that the receiver performance (throughput) and the low-power designing of whose hardware.

Based on the Monte-Carlo simulation results, the throughput of the target system was evaluated for different system parameters such as the modulation order, number of antennas, and receiver configurations. The receiver configurations include the MIMO detection scheme, LDPC decoding scheme, and number of iterations. The simulation results showed that the throughput can be enhanced when the modulation order grows with a SNR at the receiver side and a larger number of antennas are utilized. Furthermore, it was found that using iterative detection and decoding with sufficiently large number of iterations increase the throughput. Also it was shown that the best choice of the system parameters approaches the capacity limit within a few dB in terms of required SNR.

The parameters extracted by system simulation are applied to my hardware design. Moreover the low-power digital hardware design techniques, such as clock gating, operand isolation, and low-power cell replacing, are analyzed and applied to the designed system. By these techniques, I can realize total power reductions of 80 %.

박사학위논문

플렛폼 기반의 무선통신 시스템 구조 분석 및 설계

공학부 이경수

LDPC 코드와 다중안테나를 이용한 OFDM 이 결합된 시스템에서 수신성능을 향상시키기 위한 Super-iterative 방식이 분석되고, 이 분석을 바탕으로 하드웨어로 설계되고, FPGA 플랫폼에 실장 되어 테스트가 진행되었다. 특히 수신기의 성능에 대한 분석이 몬테까를로 시뮬레이션 방식에 의해서 다양하게 정리되었고, 저전력 설계를 위한 여러 가지 기술들이 적용되었다. 변조 방식, 안테나의 개수, 수신기의 구조 등 여러 시스템 파라미터가 최상의 시스템 성능을 내기 위하여 어떻게 묶여서 돌아갈 수 있는지가 시스템 시뮬레이션에 의해서 분석되었으며, 다중입출력(MIMO: Multi-Input Multi-Output) 수신 방식을 포함하여 LDPCC 의 복호 방법, 반복회수에 대한 내용이 비교분석 되었다. 분석 결과 수신기에서 높은 신호대잡음비(SNR)를 가지는 영역에서는 고도의 변조방식이 사용될 때, 더 많은 안테나를 사용할 때 시스템의 효율이 더 좋아짐을 확인하였다. 특히 충분히 많은 회수로 반복적인 정류 및 복호를 수행할 때 시스템의 수신 효율이 높아짐을 알 수 있다.

이러한 시뮬레이션에 의한 분석을 기반으로 하드웨어로 설계를 진행하였다. 이때 무선시스템에서 보다 낮은 전력을 소모하도록 하기 위하여 클럭게이팅, 피연산자 격리와 같은 설계 기법이 적용되었고, 아울러 임계경로가 아닌 경로에 포함된 소자들은 임계시간을 넘지 않는 범위에서 다소 느리게 동작하지만 전력 소모가 적은 소자로 대체하는 기법을 추가로 적용하였다. 이러한 방법에 의하여 처음 설계되었던 하드웨어 설계로부터 최대 80% 정도의 전력 소모를 획기적으로 줄일 수 있었다.

References

[802.11A] "Specification of IEEE 802.11a WLAN," IEEE, 1999.

- [Ala98] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Select. Areas Commun.*, vol. 16, pp.1451–1458, Oct. 1998.
- [AM94] M. Alidina and J. Monteiro et al., "Precomputation-based sequential logic optimization for low power," *IEEE Trans. VLSI Syst.*, vol. 2, pp.426–436, Dec. 1994.

[ATMEL99] "ASIC design guidelines," ATMEL Application Note, Rev. 1205A-12/99.

- [BKA04] S. ten Brink, G. Kramer, A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. on Commun.*, vol. 52, pp.670-678, Apr. 2004.
- [BKKSSB03] R. Bai, S. Kulkami, W. Kwong, A. Srivastava, D. Sylvester, and D. Blaauw, "An implementation of a 32-bit ARM processor using dual power supplies and dual threshold voltages," *in Proc. IEEE Computer Soc. Annu. Symp. VLSI*, Feb. 2003, pp. 149–154.
- [BM99] L. Benini, G. De Micheli, E. Macii, M. Poncino, and R. Scarsi, "Symbolic Synthesis of Clock-Gating Logic for Power Optimization of Synchronous Controllers," ACM Transactions on Design Automation of Electronic Systems, vol. 4, no. 4, October 1999, pp.351–375.

- [BMC01] M. Bhardwaj, R. Min, A. P. Chandrakasan, "Quantifying and enhancing power awareness of VLSI systems," *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 9, no. 6, pp. 757–772, 2001.
- [BOI96] M. Borah, R. M. Owens, and M. J. Irwin, "Transistor sizing for low power CMOS circuits," *IEEE Trans Computer-Aided design Integr. Circuits Syst.*, vol.15, no.6, pp.665-671, Jun. 1996.
- [CF02] J. Chen, M. P. C. Fossorier, "Near Optimum Universal Belief Propagation Based Decoding of Low-Density Parity Check Codes," *IEEE Trans. on Commun.*, Vol.50, No.3, March 2002.
- [Davis03] Linda M. Davis, "Scaled and Decoupled Cholesky and QR Decompositions with Application to Spherical MIMO Detection," 2003.
- [DY03] J. Di and J. S. Yuan, "High throughput power-aware FIR filter design based on fine-grain pipelining multipliers and adders," in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, Feb. 2003, pp. 260–261.
- [DYH02] J. Di, J. S. Yuan, and M. Hagedorn, "Energy-aware multiplier design in multirail logic," in Proc. 45th IEEE Midwest Symp. VLSI, Tulsa, OK, Aug. 4–7, 2002, pp. II-294–II-297.

[Erceg04] V. Erceg et al., "TGn Channel Models", IEEE802.11-03/940r4, May 2004.

[FMI99] M. P. C. Fossorier, M Mihaljevic, H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. on Commun.*, Vol.47, No.5, May 1999.

- [Fos96] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Wireless Pers. Commun.*, vol. 1, pp. 41–59, 1996.
- [FP85] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length is lattice, including a complexity analysis," *Math. Comput.*, vol. 44, no. 170, pp. 463-471, Apr. 1985.
- [Gal63] R.G.Gallager "Low Density Parity Check Codes", M.I.T. Press, 1963.
- [GBS89] J. Gotze, B. Bruckmeier and U. Schwiegelshohn, "VLSI-Suited Solution of Linear Systems," ISCAS 1989
- [GFVW99] G. J. Foschini, G. D. Golden, R. A. Valenzuela and P. W. Wolniansky, "Detection algorithm and initial laboratory results using the V-BLAST spacetime communication architecture", *Electronic Letters.*, vol. 35, no. 1, pp. 14-15, 1999.
- [GGH97] R. Gonzalez, B. M. Gordon, and M. A. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE J. Solid-State Circuits*, vol. 32, no. 8, pp. 1210–1216, Aug. 1997.
- [GSSSN03] D. Gesbert, M. Shafi, D-S Shiu, P. J. Smith, and A. Naguib, "From theory to practice: an overview of MIMO space-time coded wireless systems", *IEEE JSAC*, vol. 21, pp. 281-302. Apr. 2003.
- [HB] H. Vikalo and B. Hassibi, "On Sphere Decoding Algorith. I. Expected complexity," Downloadable at http://its.caltech.edu/hvikalo/publications.html

- [HB03] Bertrand M. Hochwald and Stephan ten Brink, "Achieving Near-Capacity on a Multiple-Antenna Channel," *IEEE Trans. On Communications*, Vol. 51, No. 3, Mar. 2003.
- [HMK00] Kyung Hyn-min, "Design and implementation of efficient FFT processor for OFDM applications," Master Thesis, *ICU*, 2000.
- [HS03] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel", *IEEE Trans. Commun.*, vol. 51, pp. 389-399, Mar. 2003.
- [HWA94]Razak Hossain, Leszek D. Wronski, and Alexander Albicki, "Low power design using double edge triggered flip-flops," *IEEE Trans. on VLSI Systems*, vol.2, no.2, June 1994.
- [HZA96] R. Hossain, M. Zheng, and A. Albicki, "Reducing power dissipation in CMOS circuits by signal probability based transistor reordering," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 15, no. 3, pp.361-368, Mar. 1996.
- [JKK02] S. Jung, K. Kim, and S. Kang, "Low-swing clock domino logic incorporating dual supply and dual threshold voltages," *in Proc. 39th Design Automation Conf.*, 2002, pp. 467–472.
- [KP00] S. Kim, M. C. Papaefthymiou, "Reconfigurable low energy multiplier for multimedia system design," *in Proc. IEEE Comput. Soc. Workshop VLSI*, 2000, pp.129–134.
- [LRD01] K. Lahiri, A. Raghunathan, S. Dey, "System-Level Performance Analysis for Designing On-Chip Communication Architecture," *IEEE Transactions on*

Computer-Aided Design of Integrated Circuits and Systems, VOL. 20, No. 6, June. 2001.

- [LRD01] K. Lahiri, A. Raghunathan, S. Dey, "System-Level Performance Analysis for Designing On-Chip Communication Architecture," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, VOL. 20, No. 6, June. 2001.
- [LWN02] B. Lu, X. Wang, and K. R. Narayanan, "LDPC-based space-time coded OFDM systems over correlated fading channels," *IEEE Trans. on Commun.*, vol. 50, pp. 74–88, Jan. 2002.
- [LYW04] B. Lu, G. Yue, X. Wang, "Performance analysis and design optimization of LDPC-coded MIMO OFDM system," *IEEE Trans. on Signal Processing*, vol.52, No.2, Feb.2004, pp.348-361.
- [MC03] G. Martin, H. Chang, "Winning the SoC Revolution," *Kluwer Academic Publisher*, 2003.
- [MC03] G. Martin, H. Chang, "Winning the SoC Revolution," *Kluwer Academic Publisher*, 2003.
- [MKTC04] A. Maxiaguine, S. Kunzli, L. Thiele, S. Chakroborty, "Evaluating Schedulers for Multimedia Processing on Buffer-Constrained SoC Platforms," *IEEE Design* & Test of Computers, vol. 21, no. 5, May 2004, pp.368-377.
- [MKTC04] A. Maxiaguine, S. Kunzli, L. Thiele, S. Chakroborty, "Evaluating Schedulers for Multimedia Processing on Buffer-Constrained SoC Platforms," *IEEE Design & Test of Computers*, vol. 21, no. 5, May 2004, pp.368-377.

[PCUG04] "Power compiler user guide", Synopsys Inc., Release V-2004.06, June 2004.

- [Pro95] J.G. Proakis, Chapter 10, Digital Communications, Third Edition, McGraw Hill, New York, 1995.
- [RGZV03] S. Rouquette-Leveil, K. Gosse, X. Zhuang, F. W. Vook, "Spatial division multiplexing of space-time block codes", *in Proc. ICCT2003*, pp. 1343-1347.
- [RM04] K. Ryu, V.J. Mooney III, "Automated Bus Generation for Multiprocessor SoC Design," *IEEE Trans. on CAD of IC and Systems*, vol.23, no.11, Nov. 2004, pp.1531-1549.
- [RM04] K. Ryu, V.J. Mooney III, "Automated Bus Generation for Multiprocessor SoC Design," *IEEE Trans. on CAD of IC and Systems*, vol.23, no.11, Nov. 2004, pp.1531-1549.
- [SLD04] K. Sekar, K. Lahiri, S. Dey, "Configurable Paltforms with Dynamic Platform Management: An Efficient Alternative to Application-Specific System-on-Chips," Preceedings of the 17th International Conference on VLSI Design, 2004.
- [SLD04] K. Sekar, K. Lahiri, S. Dey, "Configurable Paltforms with Dynamic Platform Management: An Efficient Alternative to Application-Specific System-on-Chips," Preceedings of the 17th International Conference on VLSI Design, 2004.
- [SMFYSCWMMK97] K. Suzuki, S. Mita, T. Fujita, F. Yamane, F. Sano, A. Chiba, Y.Watanabe, K. Matsuda, T. Maeda, and T. Kuroda, "A 300 MIPS/W RISC core processor with variable supply-voltage scheme in variable thresholdvoltage CMOS," in Proc. IEEE Custom Integrated Circuits Conf., 1997, pp. 587–590.

- [SMIC18] SMIC 0.18µm Logic18 Process 1.8-Volt SAGE-XTM Standard Cell Library Databook, Artisan, March 2003.
- [STD130] Samsung Electronics Inc., 0.18 Micron STD130 Standard Cell Library
- [STD90] Samsung Electronics Inc., 0.35 Micron STD90 Standard Cell Library
- [TFS95] E. Tellez, A. Farrah, and M. Sarrafzadeh, "Activity-driven clock design for lowpower circuits," *in Proc. IEEE ICCAD*, San Jose, CA, Nov. 1995, pp. 62–65.
- [TGn04] A. Mujtaba et al., "TGn Sync Complete Proposal", *IEEE 802.11-04/888r2*, Sept. 2004.
- [TGn04a] TGn Simulation Reference material, 2004
- [TSC98] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. on Inform. Theory*, vol. 44, pp. 744–765, Mar.1998.
- [V84] M. J. M. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *IEEE J.Solid-State Circuits*, vol. CS-19, no. 4, pp.468-483, Aug.1984.
- [VB99] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp 1639-1642, Jul. 1999.
- [VHK04] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding", *IEEE Trans. Wireless Commun.*, vol. 3, pp. 2299-2311, Nov. 2004.

- [WPW00] Q. Wu, M. Pedram, and X. Wu, "Clock-gating and its application to low power design of sequential circuits," *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.*, vol. 47, no. 103, pp. 415–420, Mar. 2000.
- [WWiSE04] V. K. Jones et al., "WWiSE IEEE 802.11n Proposal," IEEE 802.11-04/0935r3, Sept. 2004.
- [WWiSE05] "WWiSE Proposal Response to Functional Requirements and Comparison Criteria," IEEE 802.11n, Jan., 2005.
- [YC01] Hyeongseok Yu and Jun-Dong Cho, "Low-power design and architecture," *IEEE Potentials*, vol.20, issue 3, pp.18-22, Aug-Sep, 2001.
- [YD05] Jiann S. Y and Jia Di, "Teaching Low-Power Electronic Design in Electrical and Computer Engineering," *IEEE Trans. On Edu.*, Vol.48, No.1, pp.169-182, Feb. 2005.
- [ZT03] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels", *IEEE Trans. Inform. Theory*, vol. 49, pp. 1073-1096, May 2003.

Acknowledgements

"나의 나 된 것은 하나님의 은혜로 된 것이니.. (고전15:10)"

위의 바울의 고백을 떠올리며 오늘의 이 순간이 있기까지 인도해주신 하나 님께 가장 먼저 감사드립니다.

1998년 ICU가 개교하던 당시 1회 입학생으로 석사과정에 입학한 이후 7년 반이라는 시간이 훌쩍 지났습니다. 이 길고 긴 시간 동안 온 정렬을 다하여 때 로는 엄하게, 때로는 자상하게 가르쳐 주시고 이끌어 주신 박신종 지도교수님 께 정말 감사드립니다. 역시 7년 반 동안 지켜 봐주시고, 가르쳐주시고, 특히 이번 논문심사까지 맡아 주신 최해욱 교수님, 유형준 교수님, 한영남 교수님, 윤기완 교수님께 감사드립니다.

대학 때부터 연인으로, 지금은 나의 아내로 항상 내 옆에서 사랑으로 지켜 봐 주고, 기도해주고, 격려해준 혜정이에게 감사와 사랑의 마음을 전합니다. 학 위과정 중이라 제대로 놀아주지도 못한 14개월의 사랑스런 우리 아들 조은이에 게 미안한 마음과 함께 이제 더욱 자랑스러운 아버지의 모습으로 서 있기를 결 심합니다.

나를 낳아 주시고 길러주시고 기도해주시는 부모님과 경욱이 형, 동생 은 실이에게 감사의 마음을 전합니다. 결혼으로 인해 새롭게 생긴 나의 또 다른 가족인 장인어른, 장모님, 두 처형, 두 처남, 조카 화평이의 기도와 사랑, 격려 에 깊이 감사 드립니다.

이 논문을 위한 프로젝트에 함께 참여하여 정말 성심껏 도와준 비트공학 실험실의 박사과정 승범이, 진이, 석사과정 성록이, 상호, 호석이, 형순이 성민 이, 그리고 KAIST에서 이번에 함께 박사를 마무리하게 된 성정이이게 감사의 마음을 전합니다.

비트 공학실험실을 거쳐간 많은 동료, 후배들. 특히 1기로 들어와서 함께 했던 동기 보흠이형, 성환이형, 해식이, 바로 일년 밑의 후배 2기 태규, 성한이, 현호, 정규, 민정이에게도 함께 했던 많은 시간과 추억을 떠올리며 역시나 감사 의 뜻을 전합니다.

항상 나를 기억해주고 아껴주는 수영교회출신 친구 대영이, 혁일이, 병모, 주용이, 선배라고 챙겨주고 함께 놀아주는 후배 성준이, 승철이, 민규, 선부에게 기쁨을 함께 나누며 감사의 뜻을 전합니다.

KAIST에서 함께 대학시절을 보낸 오랜 룸메이트였던 나의 친구 정욱이, KAIST SFC 바로 아래 기수 후배 동민, 성종, 상준, 지혜, 선영이와도 기쁨을 함 께하고 싶습니다.

기도로 항상 지원해 준 한샘교회 가족들, 잠실중앙교회 16기 친구들, 지금 동안교회 신혼교구 식구들 모두에게 감사합니다.

마지막으로 계속적인 관심과 격려와 배려로 박사를 마무리 지을 수 있도록 도와 주신 ㈜에이디칩스 권기홍 사장님을 비롯한 이희 부사장님, 민병권 연구 소장님, 김관영 수석연구원님, 그리고 많은 회사 선후배 동료 여러분께 감사드 립니다.

이제 하나의 과정을 마무리하는 이 시점에서 더욱 성숙된 자로서, 준비된 자로서 곳곳에 좋은 영향력을 끼칠 수 있는 그런 사람이 되고자 다시 한번 결 심합니다. 곁에서 지켜봐주시고, 격려해 주신 모든분들게 감사드립니다. 앞으로 도 지켜봐 주시기 바랍니다.

감사합니다.

Curriculum Vitae

Name: Gyongsu Lee

Date of Birth: June 28, 1974

Sex: Male

Nationality: Republic of Korea

Education

1993.03~1998.02:	Department of Electrical Engineering, Korea Advanced Science and Technology (KAIST), Daejeon, Korea (B.S.)
1998.03~2000.02:	School of Engineering, Information and Communications University (ICU), Daejeon, Korea (M.S.)

Career

1998.03~1998.12:	Internship Engineer, Electronics and Telecommunications
	Research Institute (ETRI), Daejoen, Korea.
1999.01~1999.07:	Internship Engineer, Sysonchip Inc., Daejeon, Korea.

2000.01~Now: Senior Research Engineer, SoC Division, Research and Development Center, Advanced Digital Chips Inc., Seoul, Korea.
Academic Experience

International Journal

- <u>Gyongsu Lee</u> and Sin-Chong Park, "A Turbo Decoder with Reduced Number of Iterations Using Even Parity-Check Bits," IEICE Trans. on Communication, Vol. E85-B, No.6, pp.1195-1197, Jun. 2002.
- 2.Sunghwan Hyun, <u>Gyonsu Lee</u> and Sin-Chong Park, "A New Tight Bound on the Bit Error Probability for Turbo Codes," IEICE Trans. on Communication, Vol. E84-B, No.5, May. 2001.
- 3. <u>Gyongsu Lee</u> and Sin-Chong Park, "Distributed Power Control in a Fading Channel," Electronics Letters, Vol.38, No.13, pp.653-654, Jun. 2002.
- 4. <u>Gvongsu Lee</u> and Sin-Chong Park, "Implementation of the LDPCC Codec with AWGN Channel in a FPGA," IEICE Trans. on Communications, 2005. (submitted)
- 5.<u>Gyongsu Lee</u> and Sin-Chong Park, "Architecture Analysis of MIMO Detection and Iterative Decoding for Coded MIMO-OFDM System," IEICE Trans. on Communications. (submitted)
- 6.Gyongsu Lee and Sin-Chong Park, "Evaluating Multi-Processor SoC Platform Design Using Dedicated FIFO Channels", IEICE Trans. on Communications. (submitted)
- 7.Gvongsu Lee, Sunghwan Hyun, and Sin-Chong Park, "Tight Expurgated Sphere Bound Appling Verdu Theorem for Turbo Codes," IEEE Trans. on Information Theory. (Submitted)

International Conference

- 8. <u>Gyongsu Lee</u> and Sin-Chong Park, "Optimal Regular LDPC Code Structure for IEEE802.11n System," ITC-CSCC 2005, July, 2005.
- 9. <u>Gyongsu Lee</u> and Sin-Chong Park, "Architecture for Multi-Processor SoC Platform using Dedicated Channels," IWSOC 2005, Banff, Canada, July. 2005.
- <u>Gyongsu Lee</u> and Sin-Chong Park, "Bluetooth Security Implementation based on Software Oriented Hardware-Software Partition", ICC 2005, Seoul, Korea, May. 2005.
- <u>Gyongsu Lee</u> and Sin-Chong Park, "Bluetooth Security Design based on Software Oriented Hardware-Software Partition", WWC 2004, San Francisco, USA, May, 2004.
- <u>Gyongsu Lee</u> and Sin-Chong Park, "A Design of Turbo Codec with Reduced Number of Iterations," IEEE VTS Fall VTC 2001.54th, Vol.4, pp.2342-2345, 2001.
- Hyunho Jung, <u>Gyongsu Lee</u>, Sin-Chong Park, "Parity-Check-Bit-inserted Turbo code," IEEE VTS Spring VTC 2001.53rd, Vol.2, pp.1488-1491, 2001.
- <u>Gyongsu Lee</u>, SungHwan Hyun and Sin-Chong Park, "Evaluation about the Turbo codes based on the specification of 3GPP and 3GPP2," Proceeding of IEEE International Conference on 3rd Generation, Vol.1, pp.670-674, San Franscisco, USA, 2000.
- <u>Gyongsu Lee</u>, Sin-Chong Park, "Turbo MAP Decoder Design for IS-2000 System," IEEE Boston Fall VTC 2000, Boston, USA, Vol.1 pp.412-415, 2000.

- <u>Gyongsu Lee</u>, Sin-Chong Park, "Evaluation of the MAP Decoding for the Turbo Codes of IMT-2000," IEEE Boston Fall VTC 2000, Boston, USA, Vol.3 pp.1266-1269, 2000.
- Gyongsu Lee, Sin-Chong Park, "A Searcher for the Synchronization Channel of WCDMA," IEEE Boston Fall VTC 2000, Boston, USA, Vol.3 pp.1364-1370, 2000.
- <u>Gyongsu Lee</u>, Bongsoo Lee and Sin-Chong Park, "Safe and Effective Packet Resource Management by SIR Measurement," WOC 2002 Banff, IASTED, pp.563-565, 2002.
- <u>Gyongsu Lee</u> and Sin-Chong Park, "Effective Radio Resource Management for Circuit and Packet Services using SIR Measurement," ITC-CSCC 2002 Phuket Thailand, pp.1444-1446, 2002.

Domestic Journal (국내 학술논문집)

20. 현성환, <u>이경수</u>, 박신종, "3GPP2에 적용된 터보부호의 성능 분석," 전자공학회, Vol. 36-S No.2, Feb. 2000.

Domestic Conference (국내 학술대회)

- 21. <u>이경수</u>, 박신종, "시변화 채널에서의 전력분산제어," 2001년도 추계
 한국통신학회, 서울, No.1-97, Nov. 2001.
- 22. <u>이경수</u>, 박신종, "SIR 측정에 의한 패킷 자원의 효율적 운용,"
 2001년도 하계 한국통신학회, 제주, Vol.23, No.1, pp.211-214 Jul. 2001.

- 23. <u>이경수</u>, 박신종, "WCDMA 시스템의 프레임 동기획득을 위한
 주동기화채널코드 특성," 2000년도 한국통신학회 하계종합학술회
 논문집,Vol.21, No.2 ,pp.1002~1005, Jul. 2000.
- 24. <u>이경수</u>, 박신종, "IMT-2000의 채널 부호와 요구사항에 대한 검증,"
 한국통신학회 하계종합학술회 논문집, pp.1662~1665, 1999.