THE UNIVERSITY
of EDINBURGH

Energy Efficient Design of an Adaptive Switching Algorithm for the Iterative-MIMO Receiver

Noor Zahrinah Binti Mohd Tadza (Nina)

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
May 2015

Abstract

An efficient design dedicated for iterative-multiple-input multiple-output (MIMO) receiver systems is now imperative in our world since data demands are increasing tremendously in wireless networks. This puts a massive burden on the signal processing power especially in small receiver systems where power sources are often shared or limited. This thesis proposes an attractive solution to both the wireless signal processing and the architectural implementation design sides of the problem. A novel algorithm, dubbed the **Adaptive Switching Algorithm**, is proven to not only save more than a third of the energy consumption in the algorithmic design, but is also able to achieve an energy reduction of more than 50% in terms of processing power when the design is mapped onto state-of-the-art programmable hardware. Simulations are based in Matlab^{TM} using the Monte Carlo approach, where multiple additive white Gaussian noise (AWGN) and Rayleigh fading channels for both fast and slow fading environments were investigated. The software selects the appropriate detection algorithm depending on the current channel conditions. The design for the hardware is based on the latest field programmable gate arrays (FPGA) hardware from Xilinx®, specifically the Virtex-5 and Virtex-7 chipsets. They were chosen during the experimental phase to verify the results in order to examine trends for energy consumption in the proposed algorithm design. Savings come from dynamic allocation of the hardware resources by implementing power minimization techniques depending on the processing requirements of the system. Having demonstrated the feasibility of the algorithm in controlled environments, realistic channel conditions were simulated using spatially correlated MIMO channels to test the algorithm's readiness for real-world deployment. The proposed algorithm is placed in both the MIMO detector and the iterative-decoder blocks of the receiver. When the final full receiver design setup is implemented, it shows that the key to energy saving lies in the fact that both software and hardware components of the **Adaptive Switching Algorithm** adopt adaptivity in the respective designs. The detector saves energy by selecting suitable detection schemes while the decoder provides adaptivity by limiting the number of decoding iterations, both of which are updated in real-time. The overall receiver can achieve more than 70% energy savings in comparison to state-of-the-art iterative-MIMO receivers and thus it can be concluded that this level of 'intelligence' is an important direction towards a more efficient iterative-MIMO receiver designs in the future.

Declaration of Originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the Institute for Digital Communications (IDCOM) of the School of Engineering at The University of Edinburgh.

Noor Zahrinah Binti Mohd Tadza (Nina)
May 2015

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful. Alhamdulillah, all praises to Allah for the strength He gave me and for His blessings that enabled me to complete this thesis.

First and foremost, I would like to express my deepest gratitude and indeed it has been and honour and privilege to work closely with both Professor John Thompson and Dr. Dave Laurenson, without whose guidance and (extreme) patience, I would not have completed this PhD thesis. I have been fortunate to have had the joy of learning from their knowledge, experience and expertise. I am extremely grateful to them for bringing me such an interesting problem to study in this past ~~three~~ four years.

I wish to thank my father, Dr. Mohd Tadza, for his constant words of encouragement that gave me the strength to keep going; my mother, Dr. Noor Hasnah, without whose provision and advice, I would not have been be able to endure this arduous journey; my brothers, Dr. Mohd Yuhyi and (soon to be) Dr. Muhammad Afiq for their relentless moral support and company, that kept my mind off the stress this PhD sometimes created. For their love and support, I will always be thankful. Special appreciation and apologies go to Mr. Pepijn De Cuyper for relentlessly editing the thesis for grammatical errors for which I know were plenty.

To my old friends and the new ones I made along this journey, my colleagues or should I say my comrades at IDCOM, I wish to say "thank you" for enduring together in the same battle. Our shared experiences and difficulties have helped me realize that I am not alone in this crusade. I know that one day, we will look back to these days and smile. Thank you goes to the Ministry of Education (MoE), Malaysia for providing me the scholarship to pursue my doctorate, and to Tun Hussein Onn University (UTHM) for supporting my study leave.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AED	Accumulated Euclidean Distance
APP	*A Posteriori* Probability
ASIC	Application Specific Integrated Circuit
ARQ	Automated Repeat ReQuest
AWGN	Additive White Gaussian Noise
BICM	Bit-Interleaved Coded Modulation
BER	Bit-Error-Rate
BPSK	Binary Phase-Shift Keying
CE	Cross Entropy
CMOS	Complementary Metal-Oxide-Semiconductor
CPU	Central Processing Unit
CRC	Cyclic Redundancy Code
DSP	Digital Signal Processor/Processing
DVFS	Dynamic Voltage and Frequency Scaling
EDA	Electronic Design Automation
ED	Euclidean Distance
EXIT	Extrinsic Information Transfer Chart
FER	Frame-Error-Rate
FFT	Fast Fourier Transform
FLOPS	FLoating-point Operations Per Second
FPGA	Field-Programmable Gate Array
FSD	Fixed Sphere Decoder
GSM	Global System for Mobile
HDL	Hardware Description Language
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
IID	Independent and Identically Distributed
ISE	Integrated Software Environment

LLR	Log-Likelihood Ratio
LORD	Layered ORthogonal Lattice Detector
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
LUT	Look-Up Table
MAP	Maximum *A Posteriori* Probability
ME	Mean-Estimation
MI	Mutual Information
MIMO	Multiple-Input Multiple-Output
MISO	Multiple-Input Single-Output
ML	Maximum-Likelihood
MMSE	Minimum Mean Square Error Estimation
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
OFDM	Orthogonal Frequency-Division Multiplexing
PSK	Phase-Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
SD	Sphere Decoding
SIC	Successive Interference Cancellation
SIMO	Single-Input Multiple-Output
SISO	Single-Input Single-Output
SNR	Signal-to-Noise Ratio
SOCA	Smart Candidate Adding Algorithm
SoC	System-on-Chip
SOVA	Soft-Output Viterbi Algorithm
UMTS	Universal Mobile Telecommunications System
V-BLAST	Vertical-Bell Laboratories Layered Space-Time
VCW	Valid Code Word Checks
VLSI	Very Large Scale Integration Systems
WLAN	Wireless Local Area Network
XPA	Xilinx® Power Analyzer
XPE	Xilinx® Power Estimator
ZF	Zero Forcing

Nomenclature

$\triangleq$	Equals to by definition
$\otimes$	Kronecker product
$(\cdot)^H$	Hermitian transpose operation
$(\cdot)^{\mathrm{T}}$	Conjugate transpose operator
β	Number of clock cycles per detection
γ	Number of output of decoder
δ	Task operation
η	Number of tasks
κ	Number of inputs to decoder
μ	Complex mean
ξ	Number of toggling transistor
ρ	Fading environment
σ^2	Complex AWGN variance
τ	Time
ω	Real value correlation coefficient
Φ	Search sphere radius
φ	Code rate
Π	Interleaver
Ω	Channel correlation index
a	Encoded and interleaved bits
$\hat{\mathbf{a}}$	Demodulated bit vectors
A	Lower limit of integration
b	Number of random interleaved coded bits
$\mathbf{b}$	Number of random interleaved coded vectors
$\mathbb{B}$	Set of bit vectors $\mathbf{b}$
B	Upper limit of integration
$\mathbb{C}$	Circular symmetric complex Gaussian factor
C	Capacitance of hardware chip
$\mathcal{CN}(\mu, \sigma^2)$	Complex Gaussian distribution with mean μ and variance σ^2

d_i	Accumulated Euclidean distance
D_i	Euclidean distance
e	Encoded bits
$E(\cdot)$	Expected value
E	Energy consumption
$\mathbf{E}_s$	Energy consumption per symbol bit
f	Clock frequency of a hardware chip
F	Probability density function
G	$M \times M$ Moore-Penrose pseudoinverse channel matrix of **H**
h	Elements of channel matrix **H**
H	$N \times M$ channel matrix
$\mathbf{H}_w$	$N \times M$ channel matrix unit with $\mu = 0$ and $\sigma^2 = 1$
i	Iteration number for antenna level
I	$N \times N$ identity matrix
$\bar{I}$	$N \times N$ accumulated mutual information
$\Im$	Imaginary part of a complex number
j	Iteration number for nodes considered in search
J	Throughput of an algorithm in bits per second
k	Number of channel realization in an ordered set $\S$
K	Constraint length
K	Number of nodes in K-Best considered
K_e	Number of bits per frame
K_s	Number of symbols per frame
K_u	Number of information bits per frame
$\mathcal{L}$	List of candidates' size in search algorithm
L	Log-likelihood ratio
L_A	*A priori* log-likelihood bit ratio
$\mathbf{L}_A$	*A priori* log-likelihood vector ratio
L_E	Extrinsic log-likelihood bit ratio
$\mathbf{L}_E$	Extrinsic log-likelihood vector ratio
M	Number of transmit antennas
n	Number of nodes in search algorithm
n	Additive white circularly symmetric complex Gaussian noise vector
N	Number of receive antennas

N_0	Noise power spectral density
$\mathcal{O}$	QAM constellation of W points
p_1	Output bit from coder 1
p_2	Output bit from coder 2
P	Power consumption
$Q(\cdot)$	Quantizer argument to the closest point in $\mathcal{O}^M$
$\mathbf{Q}$	Unitary matrix of $M \times N$
$\mathbf{r}$	N-vector of received symbols
r	Receive bit
$\mathbf{R}$	Upper triangular matrix of $M \times N$
R_x	Receiver
$\mathbf{R}_{Rx}$	$N \times N$ receive spatial correlation matrix
$\mathbf{R}_{Tx}$	$M \times M$ transmit spatial correlation matrix
$\Re$	Real part of a complex number
$\S$	Ordered set
s	Transmit bit
S	Total size of search nodes
$\hat{s}$	Transformation of transmit bit
$\mathbf{s}$	M-vector of transmitted symbols
$\hat{\mathbf{s}}$	M-vector of the transformation for the received symbols $\mathbf{r}$
T	Threshold value
T_x	Transmitter
u	Hard data bits
$\hat{\mathbf{u}}$	Detected symbol vectors
$\mathbf{U}$	$M \times M$ upper triangular matrix
V	Voltage of operating hardware chip
W	Number of QAM constellation points
$\mathbf{x}$	General vector description
y	M-vector of decision statistics
$\hat{y}$	M-vector of estimated statistics
$\mathcal{Z}$	Subset of $\mathcal{O}$

Chapter 1
Introduction

Wireless communication has become the fastest growing segment of the communications industry. It has gone through remarkable advancement in the 20^{th} century and along with it, electronic circuit design is also progressing at an exponential rate. Recent innovations in wireless communication technology and computing have led to the current proliferation of devices, each with specific applications, form factor, functionality and battery lifetime. The explosive growth in wireless systems coupled with the proliferation of electronics devices indicate a bright future for wireless networks, both as stand-alone and as a part of a larger networking infrastructure. However, many technical challenges remain in designing robust wireless networks and devices that deliver the performance necessary to support emerging applications. One major challenge materializes in the form of power. With approximately 14 billion electronic devices are connected online; personal ones, such as mobile phones, laptops, set-top boxes, modems, and/or on a larger scale; base stations, wireless hotspots and femtocells, the communication sector has become one power hungry industry. The devices are estimated to waste around US$ 80 billion each year due to inefficient designs. This trend could lead to an estimated loss of around US$ 120 billion by the end of 2020 [1]. Therefore, solutions are sought to overcome the current predicament. This introductory chapter provides a brief review of wireless communications and describes the motivation behind the work that has been undertaken, the technical challenges, and finally the possible contributions this work aims to accomplish.

1.1 Motivation of Work

Due to the large number of devices available, just by reconfiguring the design for each individual device chipsets to be more efficient, would have tremendous impact on the global energy usage. With the adoption of best available technologies, chipsets are able to possess a higher degree of software and hardware flexibility to be more efficient in radio systems. It is said that such devices could perform exactly the same tasks while consuming around 65% less power [1]. Therefore, motivation of this work is to tackle the power consumption problem head on starting from each individual device.

There are two sides to the coin, the **wireless communication** side, which deals with the tremendous data demands, and the other, the **computer architecture** side, where a more efficient implementation is sought for better hardware deployment. On the **wireless communication** side, traffic volume according to regions as depicted in Figure 1.1, taken from the report in [1], shows that data demand is increasing over the years. It is predicted that by the end of 2017, with the fastest growing inclination, the data for Asia Pacific will be more than triple, reaching to about 45 exabyte (EB) in just 5 years. In other regions, demands are also rising year by year. The total world demand for data per year amounts to more than 120 EB per month.

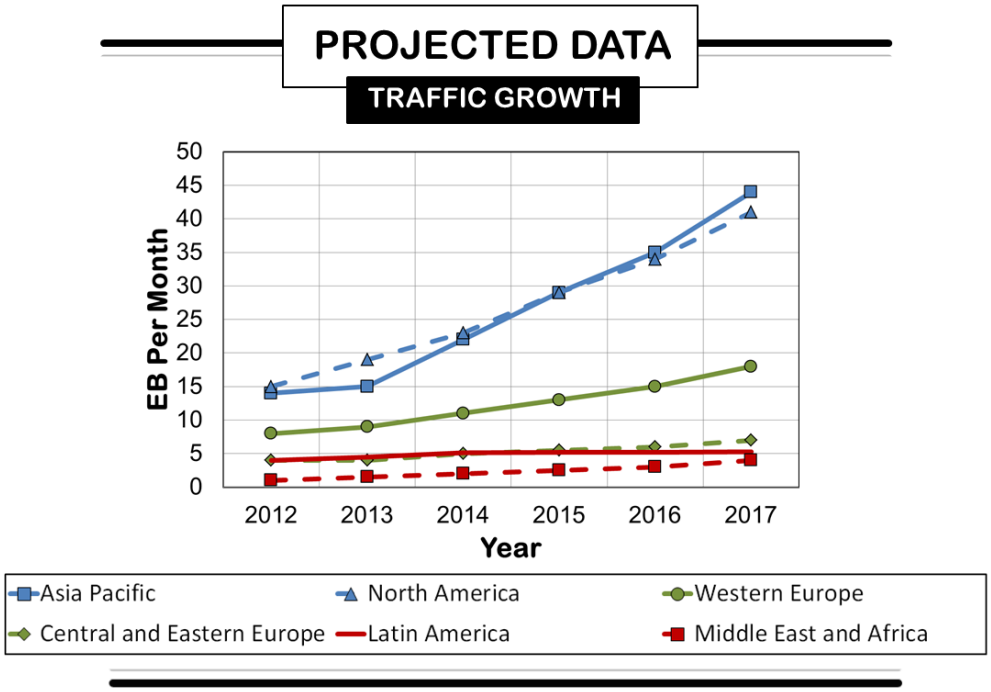

Figure 1.1: *Projected data traffic growth*

In order to cater for this trend in data demand, a significant breakthrough came in the late 1980s when the adaptive use of multiple-input multiple-output (MIMO) antenna systems was proposed. By using multiple antennas at both transmit and receive sides, parallel channels that utilize the same radio spectrum space can be created. MIMO manipulates this to increase the capacity of a channel so more data can be transmitted at one time. While minimizing power usage in these devices in wireless networks is imperative, more priority is given to the receivers since they handle massive computation processing. With billions of devices available, the total power consumption would be massive. Moreover, the receivers are usually limited in power

source where they are operated using a battery, which has a limited lifetime. This brings us to the subject of **computer architecture**. Future wireless receivers aim at supporting a wide variety of wireless communication standards, such as the Long-Term Evolution (LTE), Universal Mobile Telecommunications System (UMTS), wireless local area network (WLAN), and Global System for Mobile (GSM). Key enabling technology for the enormous success of wireless communication is the progress in integrated circuit (IC) technology. It started in the late 1950s with the production of the first metal-oxide-semiconductor field-effect transistors (MOSFET) and with the idea of complementary metal-oxide-semiconductor (CMOS) circuits [2]. IC follows the trend given by Moore's law, which states that the number of transistors in a dense integrated circuit has doubled approximately every two years. Electronic design automation (EDA) software tools help handle larger and faster chips, fabrication technologies for supporting new technology nodes, and verification strategies for the increased circuit complexity. The progress in CMOS IC technology made it possible to pack more and more transistors onto the same area of silicon. This progress allowed to realize increasingly complex functions on a small piece of silicon. With this, the realization of a fast Fourier transform (FFT), a real-time detection and decoding algorithms, or an entire wireless baseband processor on a single chip became feasible.

Figure 1.2 shows the potential energy savings that can be achieved with growing technology in programming and IC circuitry. It depicts the proportion of savings that can be accomplished to compute a given operation, and that the devices of today do not fully reap these benefits in the designs. By the year 2015, just by implementing power minimization techniques to evoke a more efficient hardware design, 70% of potential energy savings can be gained, and this trend continues to rise up to a point where, in 2025, it is predicted that around 87% of energy usage can be conserved if more efficient designs are implemented in these devices. In order to have a more efficient design, flexible software and hardware implementation are needed for the whole receiver. To achieve this flexibility, the processor circuit and signal processing software need to have certain adaptivity whereby they possess a level of 'intelligence'. In principle, this would allow the exchange between transmission standards and algorithms at boot or even dynamically at run-time. This could be in the form of a system that is able to adapt to the detection algorithm on-the-fly to the current operating scenario according to the requests of the system. Current radio communication devices have incorporated digital signal processing (DSP)-based programmability for some receiver blocks. However, many computationally intensive parts still require dedicated hardware for performance and efficiency reasons. This issue is particularly

crucial for MIMO transceivers, where the volume of incoming data is multi-fold, and therefore the energy required to process would be immensely large.

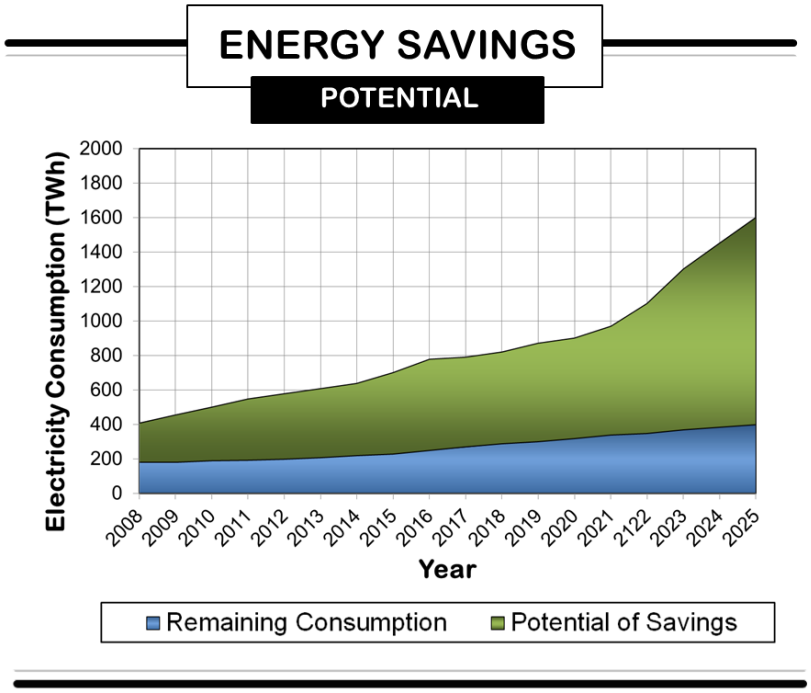

Figure 1.2: *Potential energy savings trend [1]*

This aspect of **computer architecture** and the power management schemes have not been fully exploited. Even though the technology exists, several power minimization techniques are not properly optimized on devices that support MIMO. This thesis therefore proposes a more efficient design for a receiver that rivals the state-of-the-art available in the market today. With the combination of both fields of knowledge, another setback to take into account when designing an efficient hardware capable of transmitting large amounts of data is that when a signal propagates through a wireless channel, it experiences random fluctuations in time if the transmitter or receiver is moving, due to changing reflections and attenuations. Thus, the characteristics of the channel appear to change randomly with time, which makes it difficult to design reliable systems with guaranteed performance. This is imperative to keep in mind in order to confirm the applicability of the new design in realistic situations.

In summary, technological advances in the following areas are needed to overcome the challenges this work aims to tackle:

- Algorithmic design for the MIMO detection and decoding algorithms that support efficiency in implementations.
- Hardware design suitable for low-power handheld computer and communication receiver terminals, which can be implemented on current and future communication systems.
- Measurements and models for wireless indoor and outdoor channels in order to verify the design suited for real-life deployment.

Given these requirements, the work draws from many areas of expertise, which includes the area of communications, signal processing, software and hardware design, and power management schemes. Moreover, given the fundamental limitations of the wireless channels and the explosive demand for its utilization, communication between these interdisciplinary groups is necessary to implement the most rudimentary shell for the thesis work.

1.2 Thesis Contributions

The objective of this work is to design an efficient iterative-MIMO receiver fit for current and upcoming wireless communication standards. The main contributions of this work are distributed in three separate chapters. The chapters integrate into one another to culminate in achieving the main objective of the thesis, which is to design an efficient adaptive algorithm that possesses a level of 'intelligence' for iterative-MIMO receivers. Each stage of the work leads to the next logical progression from experimental to design practicality, as detailed below:

- An Adaptive Switching Algorithm that adapts to real-time channel conditions to minimize the power and energy consumption of iterative-MIMO detection systems is proposed. This is realized in the form of a threshold control unit, which selects the minimum complexity detector capable of meeting the desired bit-error-rate (BER) performance. The adaptive algorithm shows promising BER performance on par with the current available detection schemes with lower resource utilization. An evaluation of the new algorithmic design shows convincing dynamic and static power savings compared to baseline detectors.
- Realistic power and energy saving trends of the Adaptive Switching Algorithm are computed for the chosen hardware circuitry. Detailed power and energy analysis and the

assessment of potential benefits of specific power minimization techniques show more promising results compared to the others. The combination of both the algorithmic design and the hardware design adaptivity results in tremendous gains in the overall proposed design.

- The performance of the Adaptive Switching Algorithm in realistic conditions shows significant power and energy savings with slight BER degradation. The proposed algorithm is suitable to be used as a link between the detector and iterative decoder blocks in the receiver, as a stopping criteria tool to help determine the number of decoding iterations needed per transmission. Hardware design implementation for the proposed algorithm maintains the performance of the Adaptive Switching Algorithm total receiver design in spatially correlated channels with a lower hardware utilization complexity to boot.

1.3 Thesis Outline

The thesis is structured into several chapters covering different stages of the work, following a logical flow of information, starting with the development from theoretical concepts and continuing on with the three main contributions of the research; the proposed Adaptive Switching Algorithm, the design performance of the proposed algorithm on hardware and finally, the performance of the hardware design in realistic channel conditions to test its readiness for real world applicability. The structure of each chapter is described below:

Chapter 2 is divided into two parts, viz. the **wireless communication** and the **computer architecture**. The **wireless communication** part explains the total iterative-MIMO systems and provides additional background on the detecting and decoding techniques. For a reader who is familiar with modern wireless communication systems, this part will serve mainly as a refresher as it introduces the concept of MIMO systems that provides the foundation of the research. The **computer architecture** part presents the different hardware types available and various power minimization techniques labelled as state-of-the-art, each of which promises significant power savings. The combination of the two fields of knowledge provides the comprehensive understanding required as basis for the work described in this thesis.

The proposed novel innovation of the Adaptive Switching Algorithm introduced in **Chapter 3** proves to be suitable for the sole purpose of saving power and energy consumption of the overall receivers in both slow and fast fading environments. The algorithm works by switching between

thresholds pre-calculated between the transmitters and receivers during each transmission in real-time. This novel idea is the first of its kind to produce an 'intelligent' system based on switching from a high to a low complexity detector, exploiting full information of the current channel conditions of a MIMO system. The adaptivity shows that promising savings can be gained in comparison to non-adaptive iterative-MIMO detectors.

Having shown the potential power and energy savings that can be achieved within the receiver design with the proposed algorithmic design of the Adaptive Switching Algorithm, the next stage of work as described in **Chapter 4** extends those findings by incorporating the novel idea of the Adaptive Switching Algorithm onto hardware design, to promote its applicability in implementations as well. With efficient design, the proposed algorithm shows that significant power and energy savings can be gained when different power minimization techniques are utilized. A comprehensive power and energy performance analysis of the Adaptive Switching Algorithm is investigated for the iterative-MIMO systems, with the primary goal of minimizing additional power and energy consumption within the receiver. The work is then extended to examine the potential benefits of several power minimization techniques during the implementation of the Adaptive Switching Algorithm. An in depth investigation shows that power and energy usage can be further optimized when the design for the proposed algorithm is designed on state-of-the-art hardware.

After having demonstrated in the preceding chapters that the Adaptive Switching Algorithm could save significant complexity, power and energy consumption in both algorithmic and hardware design implementation in experimentally controlled conditions, its effectiveness in real-world situations is then verified in **Chapter 5**, whereby the proposed algorithm is executed under spatially correlated channel conditions. The performance of the Adaptive Switching Algorithm in these channel conditions shows that significant energy savings can be gained with slight BER degradation as the correlation between the transmitters and receivers increases. The chapter describes how forwarding the proposed algorithm threshold information to the decoder, which by providing the same necessary information used in the detector as a stopping criteria for the decoder, helps limit the number of iteration(s) required during each transmission. Significant power and energy savings are achieved for the full Adaptive Switching Algorithm receiver in comparison to state-of-the-art hardware, with lower hardware utilization complexity to boot.

The concluding remarks about this work, as presented in **Chapter 6**, enumerates the major

contributions while identifying the novel aspects and improvements in comparison to other research that has been carried out in the same area. Special attention is also paid to the specific areas that could potentially be studied in future work. An appendix that contains a list of publications originating from this work is attached and included as references throughout the thesis.

Chapter 2
Background

2.1 Chapter Contribution

The work described in this thesis revolves around designing an efficient iterative-MIMO receiver that is suitable for state-of-the-art wireless communication standards. This chapter aims to provide comprehensive knowledge in the areas of **wireless communications** for software design and **computer architecture** for the hardware design implementation. The combination of each field of specialization gives the background information required to help the reader in understanding the nature of the work. The chapter begins by introducing the **wireless communication** system under consideration and the blocks within the iterative-MIMO systems i.e. the detector and the decoder. After a brief description regarding each block, the chapter progresses to the other area of specialization, namely the **computer architecture**. Several power minimization techniques in hardware are discussed in detail to shed light on the state-of-the-art methods currently available in the market. The chapter concludes by summarizing the chosen methods in this thesis for detecting and decoding and the reason behind them. It also pinpoints the best power minimization techniques to investigate in this study. Both information will lead to better understanding of the upcoming technical chapters.

2.2 Wireless Communication

Wireless communication is the transfer of information between two or more points that are not connected by an electrical conductor. The most common wireless technologies use radio. Figure 2.1 illustrates the different antenna configurations for wireless communication links. Single-input single-output (SISO), shown in Figure 2.1(a) is effectively a standard radio channel. This type of configuration has one transmitter and one receiver. Due to its simplicity, SISO requires no extra processing for manipulating the diversity that may be used. The disadvantage of SISO is that it is vulnerable to interference and fading. Moreover, the throughput is dependent on the channel bandwidth and the signal-to-noise ratio (SNR), which means it is

bounded by Shannon's law. The single-input multiple-output (SIMO) version is depicted in Figure 2.1(b) and the multiple-input single-output (MISO) is shown in Figure 2.1(c). Due to the usage of multiple antennas, there are several advantages that can be gained when compared to their SISO counterpart. SIMO or MISO is able to increase the receive SNR by coherently combining the wireless signals to achieve **array gain**. Moreover, **diversity gain**, which can be classified as transmit or received diversity, are used to combat fading. The receive diversity does this by enabling the receiver to receive signals from a number of independent channels. Transmit diversity on the other hand, generates redundant data from the multiple transmitters for the one receiver to choose from. This is when the signal is transmitted over multiple (ideally) independent fading paths in time, frequency, or space. This allows the receiver to select the optimum signal to extract the required data. The advantages of using multiple transmitters are that it creates redundancy in coding and moves processing from the receiver to the transmitter. This is highly beneficial for the receiver. The lower processing requirement, which leads to lower power consumption, will have a positive impact on the size needed for multiple antennas, as well as the cost and battery lifetime. In addition, the usage of multiple antennas exploits the spatial dimension to increase the separation between users by directing signal energy towards the intended user. This is **interference reduction**. Lastly, **spatial multiplexing gain** in the multiple antenna setup provides additional data capacity by utilizing the different paths to increase the data throughput capability [3] [4] [5].

By combining the configurations, MIMO may exploit all the advantages provided by the configurations of others [6], from the aforementioned techniques of **array gain**, **diversity gain**, **spatial multiplexing gain** and **interference reduction**. MIMO, as illustrated in Figure 2.1(d), uses multiple antennas at both the transmitters and receivers. It enables a variety of signal paths to carry the data, choosing separate paths for each antenna to enable multiple signal paths to be used. It is found that the signal can take many paths between a transmitter and a receiver. Additionally, by moving the antennas even by a small distance, the paths used by the signal will change. The variety of paths available occurs as a result of the number of objects that appear to the side or even in the direct path between the transmitter and receiver. By using MIMO, these additional paths provide additional robustness to the radio link by improving the SNR, or by increasing the link data capacity. As a result, it is able to considerably increase the capacity of a given channel by increasing the number of receive and transmit antennas. MIMO increases the throughput of the channel linearly with every pair of antennas added to the system. Moreover, as spectral bandwidth is becoming an ever more valuable commodity for radio

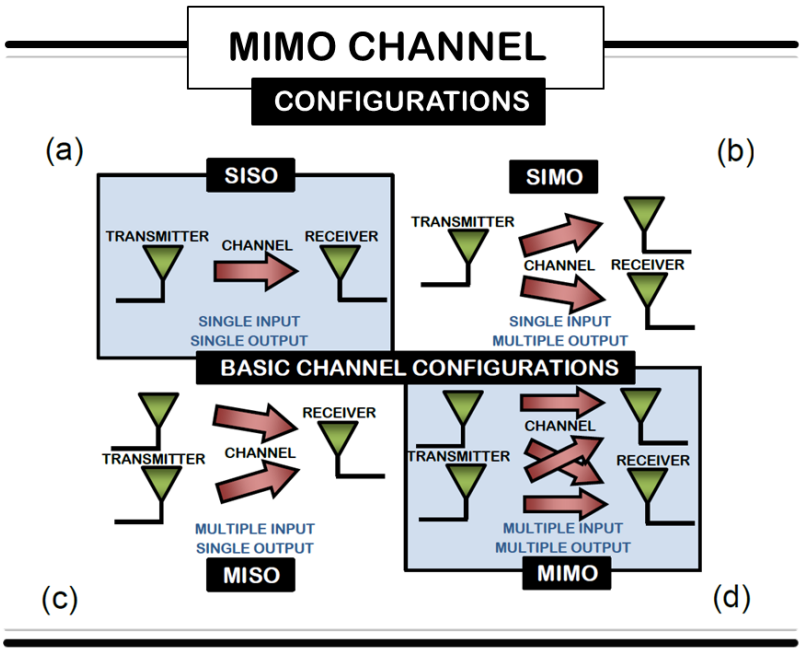

Figure 2.1: *Channel transmission configurations*

communications systems, MIMO is one of the techniques needed to properly exploit available bandwidth more effectively as well. Hence, depending on the purpose of the MIMO system, an appropriate trade-off needs to be found. Due to the increasing demand of data mentioned in the previous chapter, **spatial multiplexing** provides the capacity to cater for this need. The aim of this work is therefore, to find the right trade-off in a system that incorporates **spatial multiplexing**, between the complexity or power consumption and the system performance.

2.2.1 Iterative-MIMO System Architecture

A typical iterative-MIMO architecture is illustrated in Figure 2.2. An in-depth explanation of the full iterative-MIMO system can be found in the next section, however, as an overview, the system can be partitioned into three segments; the **transmitter**, the **channel** and the **receiver**. The **transmitter** is made up of several components. The hard data bits, u, first go through the channel encoder. The channel encoder appends extra data bits to make the data transmission more robust to interferences on the transmission channel. There are many coding schemes available and they can basically be categorized into two major types; linear block codes and convolutional codes. In a typical iterative-MIMO system, the latter is used, specifically the

turbo encoder, where two convolutional codes are used in parallel with some kind of interleaving in between. This gives the encoded e bits, which are interleaved. These are being passed through to the constellation modulator where the bits are mapped onto a digital scheme such as the quadrature amplitude modulation (QAM) or the phase-shift keying (PSK). By representing the transmitted bits a as a complex number and modulating a cosine or sine carrier signal with real ($\Re$) and imaginary ($\Im$) parts respectively, the symbols can be sent with two carriers on the same frequency. Once the symbols are modulated, they are split into several streams depending on the number of transmitters used before being transmitted over a **channel**. The transmission **channel** is essentially a path between two nodes in a network.

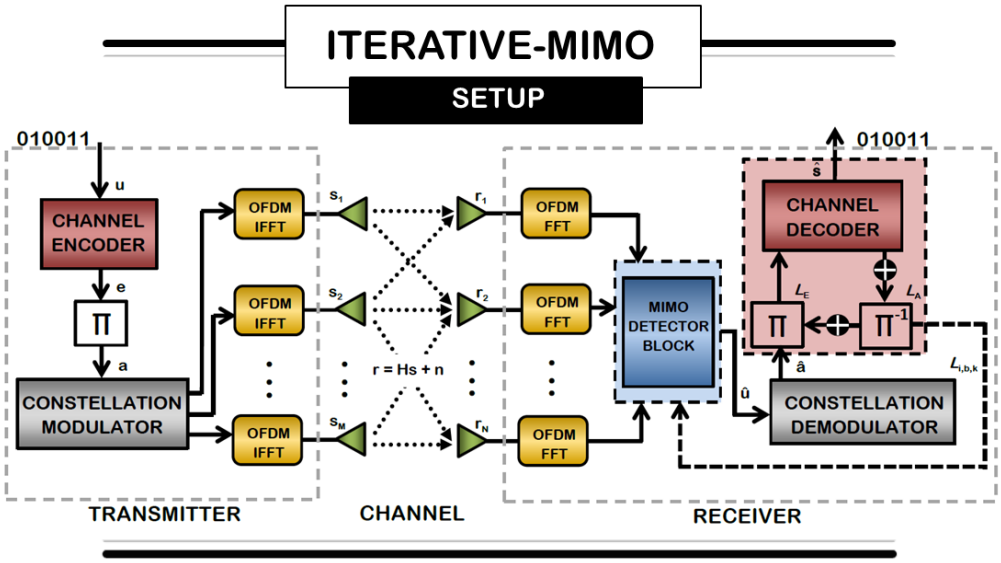

Figure 2.2: *Iterative-MIMO system channel*

Consider a spatial multiplexing MIMO-orthogonal frequency-division multiplexing (OFDM) system with M transmitters, N receivers, and $M \geq N$. The channel can be represented by the matrix described in Equation (2.1).

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{2.1}$$

where the channel matrix $\mathbf{H} \in \mathbb{C}^{M \times N}$ with independent elements $h_{i,j} \sim \mathcal{CN}(\mu, \sigma^2)$, for $1 \leq i \leq M$ and $1 \leq j \leq N$ representing a block fading propagation environment, with $\mu = 0$ and $\sigma^2 = 1$, $\mathbf{s} = (s_1, s_2, \ldots, s_M)^{\mathrm{T}}$ is the transpose vector of the M-dimensional

transmit symbol vector with $E[\mid \mathbf{s}_i \mid^2] = M^{-1}$, $\mathbf{n}$ is the $\mathbb{C}^{N\times 1}$ additive independent and identically distributed (i.i.d.) circular symmetric complex Gaussian noise vector normalized so that its covariance matrix is the identity matrix, i.e. $\mathbf{n} \sim (0, N_0\mathbf{I}_N)$ of $h_{i,j} \sim \mathcal{CN}(0, N_0)$ and $\mathbf{r} = (r_1, r_2, \ldots, r_N)^{\mathrm{T}}$ is the transpose N-vector of received symbols. Throughout this thesis, the SNR is defined as the average SNR per receive antenna according to Equation (2.2).

$$\mathrm{SNR} = \frac{M\mathbf{E}_s}{N_0} \tag{2.2}$$

where $\mathbf{E}_s$ is the energy per transmit symbol s. The received symbols, $\mathbf{r}$, are then processed by the **receiver**. From Figure 2.2, first, the symbols are multiplexed into a single stream before being detected by the MIMO detector to give $\hat{\mathbf{u}}$ bit streams.

In the **receiver**, the detection can be solved in many ways. In order to optimally solve the MIMO detection problem, an exhaustive search for the best solutions can be performed over all signal constellations. The number of possible signal constellations increases exponentially with the number of antennas and the number of bits per modulation symbol. Maximum-Likelihood (ML) detection finds the minimum constellation point in Equation (2.1) within the received symbols. It is given by:

$$\hat{\mathbf{s}}_{ML} = \underset{\mathbf{s}\in\mathcal{O}^M}{\arg\min} \parallel \mathbf{r} - \mathbf{H}\mathbf{s} \parallel^2 \tag{2.3}$$

where $\mathcal{O}$ denotes the constellation size of a specific modulation. The ML detector is optimal and fully exploits all available degree of freedom. Even though ML produces the best BER performance, due to its use of exhaustive search, it can have immense complexity for direct implementation. The complexity grows exponentially with the transmission rate φ, since the detector needs to go through 2^{φ} hypotheses for each received vector. For example, for the case of a 4×4 iterative-MIMO system employing 16-QAM, the detector would need to search a total of $S = 16^4 = 65,536$ candidates in order to find the correct transmitted vector. For 64-QAM, this number rises to more than $S = 64^4 = 16,777,216$. This makes an exhaustive search infeasible for a hardware implementation [7]. As the optimal exhaustive search is far too complex for hardware implementations, many sub-optimal detection algorithms exist with a big range in communications performance and complexity. Several efficient suboptimal detection techniques have therefore been proposed or adapted from the field of multi-user detection.

Even though these techniques are much less computationally demanding than the ML detector, they are often unable to exploit a large part of the available degree of freedom, and thus, their performance tends to be significantly poorer than that of ML detection. However, this trade-off can be made for efficient hardware designs.

Back to Figure 2.2, after the detection, the symbols are then forwarded to the constellation demodulator where the symbols are demapped to get $\hat{\mathbf{a}}$ before going to the turbo decoder, with two constituent decoders working together with deinterleavers in between them. These iterative decoders then produce the hard output for the received symbol bits. Within the **receiver** is where the focus of the work lies. This involves around minimizing power and energy consumption within the iterative-MIMO receiver, particularly, by re-designing the MIMO detector and the iterative decoder parts of the system. The sections below explain different types of detectors and decoders available, and their advantages and disadvantages are highlighted to showcase parts that need to be improved for a better performance in power and/or energy consumption. Finding the right trade-off between communications performance with implementation complexity, and understanding the implications on the whole receiver is one of the major challenges in the design of iterative-MIMO receivers.

2.2.2 MIMO Detectors

MIMO detection algorithms can be seen as a "tree search" problem, as shown in Figure 2.3. This is realized by inverting the channel matrix $\mathbf{H}$ using the $\mathbf{QR}$-decomposition to decompose matrix $\mathbf{H}$ into a unitary matrix $\mathbf{Q}$ of dimension $M \times M$ and an upper-triangular matrix $\mathbf{R}$ of dimension $M \times N$ according to:

$$\mathbf{H} = \mathbf{QR} \tag{2.4}$$

The system model in Equation (2.1) can be left-multiplied by the Hermitian transpose of $\mathbf{Q}$, which is the $\mathbf{Q}^H$, to give:

$$\hat{\mathbf{y}} \triangleq \mathbf{Q}^H \mathbf{r} = \mathbf{Rs} + \mathbf{n} \tag{2.5}$$

When the problem is visualized as a "tree search", the ML detection rule as given in Equation

(2.3) can be approximated as:

$$\hat{\mathbf{s}}_{ML} \approx \underset{\mathbf{s}\in\mathcal{O}^M}{\arg\min} \parallel \hat{\mathbf{y}} - \mathbf{R}\mathbf{s} \parallel^2 \tag{2.6}$$

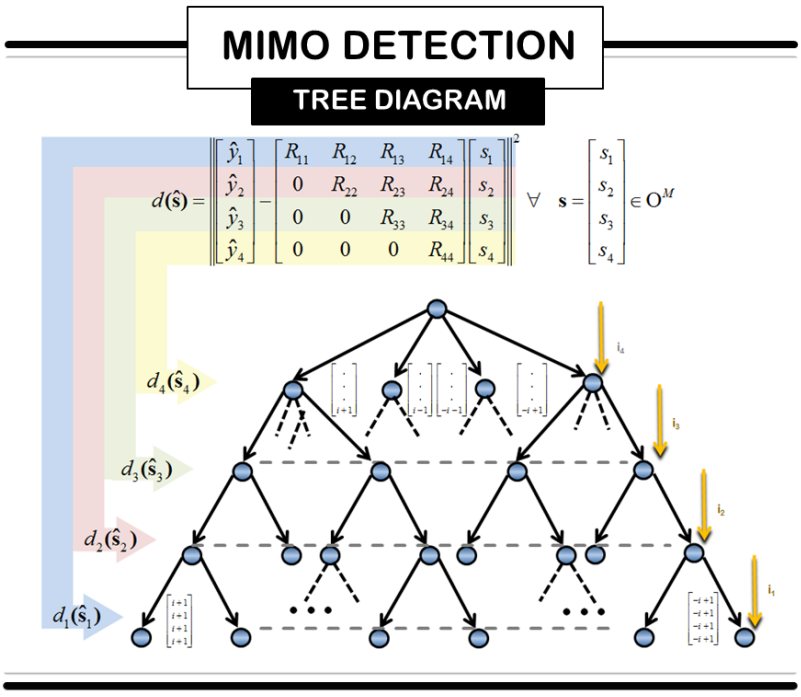

Figure 2.3: *MIMO detection as a tree diagram for* 4*-QAM modulation on a* 4×4 *MIMO system*

Figure 2.3 depicts the search traversing down level i, looking through j nodes until the solution is found, where the $\mathcal{O}$ is the number of constellation points in respective modulation scheme. Since $\mathbf{R}$ is upper-triangular, the minimization in Equation (2.3) corresponds to a "tree search" problem, where the nodes on level i are associated with a partial symbol vector $\mathbf{s} = [s_i, ..., s_M]^{\mathrm{T}}$ and with a corresponding squared partial Euclidean distance (ED), $d_i(\mathbf{s})$. The squared partial ED is given by:

$$d_i(\mathbf{s}_i) = d_{i+1}(\mathbf{s}_{i+1}) + |D_i(\mathbf{s}_i)|^2 \tag{2.7}$$

with $i = M, M-1, ..., 1$. The distance increments $|D_i(\mathbf{s}_i)|^2$ are computed as:

$$|D_i(\mathbf{s}_i)|^2 = |\hat{y}_i - \sum_{j=i}^{M} R_{ij} s_j|^2 \tag{2.8}$$

Therefore, the squared ED for the ML solution is given as:

$$d_{ML} = \min_{\mathbf{s} \in \mathcal{O}^M} (d_1(\mathbf{s}_1)) \tag{2.9}$$

and the ML solution is the associated $\mathbf{s}_1$. With this illustration in mind, the task of a MIMO-detector is to find the vector $\mathbf{s}_1$ that leads to the smallest d_i, i.e. the leaf node with the smallest squared partial ED.

To this end, a vast amount of literature exists that presents algorithms and approximations to process the tree in a clever way in order to find the estimate $\hat{\mathbf{s}}$ with less computational effort than an exhaustive search. The trade-off between the different approaches consists of implementation complexity, BER performance, and throughput.

2.2.3 Hard-Output MIMO Detection

The output of a MIMO detection algorithm is either a hard-output decision (the estimate $\hat{\mathbf{s}}$), or an *a posteriori* probability (APP) for each bit of the transmitted symbol vector. The latter helps further improve the performance of a MIMO detector. This soft-output iterative-MIMO detection algorithms were introduced in [8], and will be described in the next section. A hard-output MIMO detector delivers an estimate $\hat{\mathbf{s}}$ of the transmitted symbol vector $\mathbf{s}$. Starting point is the input-output relation as given in Equation (2.1). Several algorithms exist to obtain the estimate $\hat{\mathbf{s}}$. In general, these are divided into linear detection, successive interference cancellation (SIC) detection, and ML detection methods.

2.2.3.1 Linear Detectors

A linear detector first separates the data streams with a linear filter and then decodes each stream independently. The computational complexity of linear hard-output MIMO detection is small in comparison to other detection schemes. However, the BER performance is significantly worse compared to ML detection. Examples of linear detectors are Zero Forcing (ZF) and minimum

mean square error (MMSE) filters apply an inverse of the channel to the received signal in order to restore the transmitted signal [9]. These linear filters can be implemented at a low complexity, however, their performance is very low as well.

The **ZF** detector inverts the effect of the channel matrix, $\mathbf{H}$. The corresponding channel filter matrix $\mathbf{G}_{ZF}$ is given by Equation (2.10).

$$\mathbf{G}_{ZF} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H \tag{2.10}$$

where $\mathbf{G}_{ZF}$ is the Moore-Penrose pseudoinverse of $\mathbf{H}$. Left-multiplying Equation (2.1) with $\mathbf{G}_{ZF}$ yields the ZF estimate of:

$$\hat{\mathbf{y}}_{ZF} = \mathbf{G}_{ZF}\mathbf{r} = \mathbf{s} + \mathbf{G}_{ZF}\mathbf{n} \tag{2.11}$$

to obtain the symbol-vector estimate $\hat{\mathbf{s}}$, the equalized noise $\mathbf{G}_{ZF}\mathbf{n}$ is ignored and each element of $\hat{\mathbf{y}}_{ZF}$ is mapped to the closest constellation point according to Equation (2.12).

$$\hat{s_i} = [\hat{y_i}]_{\mathcal{O}}, \qquad \text{for} \quad i = 1, ..., M \tag{2.12}$$

The ZF detection removes the co-channel interference and it is the ideal detector when the channel is noiseless, i.e. $\mathbf{n} = 0$. However, in a real system, the noise is enhanced and correlated by $\mathbf{G}_{ZF}$, which is the main reason for the poor BER performance of ZF detection. This phenomenon is known as noise-enhancement [10].

The **MMSE** detector considers the noise power in the interference cancellation and therefore shows a slightly better performance. It reduces the effect of noise-enhancement by minimizing the total error, including the noise term, according to Equation (2.13).

$$\mathbf{G}_{MMSE} = \underset{\mathbf{G}\in\mathbb{C}^{M\times N}}{\arg\min} \parallel \mathbf{G}\mathbf{r} - \mathbf{s} \parallel^2 \tag{2.13}$$

The MMSE estimator matrix $\mathbf{G}_{MMSE}$ can be computed as in [10] to give Equation (2.14).

$$\mathbf{G}_{MMSE} = (\mathbf{H}^H\mathbf{H} + \frac{M}{\mathrm{SNR}}\mathbf{I}_M)^{-1}\mathbf{H}^H \tag{2.14}$$

Left multiplication of Equation (2.1) by $\mathbf{G}_{MMSE}$ yields:

$$\hat{\mathbf{y}}_{MMSE} = \mathbf{G}_{MMSE}\mathbf{r} = \sqrt{\frac{N}{\mathbf{E}_s}}\mathbf{s} + \mathbf{G}_{MMSE}\mathbf{n} \tag{2.15}$$

where the term $\sqrt{\frac{N}{E_s}}$ is the mean (over fading) received energy of the signal transmitted by each antenna, which is the residual noise caused by the co-channel interference. The detection step is carried out, similar to ZF detection, by mapping $\hat{\mathbf{y}}_{MMSE}$ to the closest constellation point analogous to Equation (2.12). The MMSE detector suffers less from the noise-enhancement and therefore achieves the better BER performance in comparison to ZF detection. The computational complexity remains approximately the same as for ZF detection with the exception of the former needing an estimate on the SNR.

2.2.3.2 SIC Detectors

The SIC technique was initially adopted by the Vertical-Bell Laboratories Layered Space-Time (V-BLAST) system [3]. In contrast to the basic ZF and MMSE filters, SIC detects the transmitted streams sequentially. It chooses the substream with largest SNR and removes the interference of each detected stream before continuing the detection process. The performance of the SIC algorithm is generally better than ZF and MMSE filters. The starting point for SIC detection is the $\mathbf{QR}$-decomposition of the system model in Equation (2.5).

The matrix, $\mathbf{R}$, has the property of being upper-triangular and the M^{th} stream can be detected according to:

$$\hat{s}_i = \left[\frac{\hat{y}_M}{R_{M,M}}\right]_{\mathcal{O}} \tag{2.16}$$

The remaining streams are detected according to the following recursion:

$$\hat{s}_i = \left[\frac{1}{R_{i,i}}(\hat{y}_i - \sum_{j=i+1}^{M} R_{ij}\hat{s}_j)\right]_{\mathcal{O}}, \qquad \text{for} \quad i = M-1, ..., 1. \tag{2.17}$$

SIC detection resembles the procedure of ZF detection. However, the streams are processed sequentially, one after another. This allows slicing the estimate $\hat{y}_i$ to $\hat{s}_i$ immediately after its

computation and using the result to cancel out its influence on the subsequent streams. SIC can be visualized as a single tree-traversal from top to bottom always selecting the node with the smallest partial ED. The symbol vector leading to the leaf node is returned as the SIC estimate.

2.2.3.3 ML Detectors

Under the assumption that all transmit symbol vectors are equally likely, ML decoding is the optimum hard-output MIMO detection method in terms of minimizing the symbol BER [10]. The task of an ML detector is to go through all the possible constellation points and level of antennas exhaustively until the minimum node with the smallest ED is found.

A **brute-force ML** detector computes the ED for all possible transmitted vector symbols. The ML solution then corresponds to the vector symbol with the smallest ED. In [11], it was shown that the implementation of the detector is feasible at a throughput of 50 megabit per second (Mbps) for a 4×4 MIMO system with quadrature phase-shift keying (QPSK) modulation, i.e. for $4^4 = 256$ possible vector symbols.

2.2.3.4 Sphere Decoding (SD)

Due to the ML detection problem complexity being extremely high, the brute force manner can also be solved by the sphere decoding (SD) algorithm. SD traverses the tree in a clever way such that the search complexity is significantly reduced by searching over only those lattice points that lie within a hypersphere of radius Φ around the received signal $\mathbf{r}$ [10]. From a "tree search" point-of-view, the ML solution corresponds to the leaf associated with the smallest ED, as shown in Equation (2.9). To find this leaf, SD traverses the tree in a depth-first manner. The hypersphere around $\mathbf{r}$ corresponds to a pruning criterion in Equation (2.18).

$$d_i(\mathbf{s}_i) < \Phi^2 \tag{2.18}$$

Complexity reduction is achieved by pruning those nodes from the tree that violate the sphere constraint. Whenever a node is computed with a partial ED, $d_i(\mathbf{s}_i) \geq \Phi^2$, that branch is pruned and no longer followed. In order to further reduce search complexity, some optimizations on algorithmic level can be applied such as **radius reduction**. The Φ is initialized to $\Phi = \infty$ in order to guarantee to find at least one leaf node. Once the first leaf node is computed, the radius

is updated according to $\Phi \leftarrow d_1(\mathbf{s}_i)$. Now, whenever a new leaf is found that fulfils sphere constraint, Φ is updated again. The reduction of Φ allows for more rigorous tree pruning while still finding the ML solution and therefore leads to a reduced average number of visited nodes. Another technique of reducing complexity is **enumeration**, where each node in the tree has several child-nodes. The processing order of these child-nodes considerably influences search complexity, especially if **radius reduction** is applied. A scheme proposed by Schnorr and Euchner [12] and modified for finite lattices in [13] visits the nodes of the same parent node in ascending order of their partial EDs. SD with Schnorr-Euchner **enumeration** and **radius reduction** is usually denoted as Schnorr-Euchner SD. A drawback of SD is the variable run-time, due to variable search complexity, which renders detection latency unpredictable.

2.2.3.5 Close-to-ML Detection

The variable number of nodes that need to be visited in SD and the still considerable implementation complexity lead to a variety of algorithms that approximate the performance of SD. The price for the reduced implementation complexity or for the constant run-time is slightly worse but still close-to ML BER performance. Therefore, **reduced complexity sphere decoding** aims at decreasing the computational effort to compute a partial ED. To this end, the computation of the squared l^2-norm in Equation (2.7) is approximated by the l^1-norm or the l^∞-norm, respectively [14]. The l^1-norm of a vector $\mathbf{x}$ is defined as:

$$\| \mathbf{x} \|_1 = |\Re(\mathbf{x})| + |\Im(\mathbf{x})| \tag{2.19}$$

and the l^∞-norm of a vector $\mathbf{x}$ is defined as:

$$\| \mathbf{x}_\infty \|_1 = \max\{|\Re(\mathbf{x})|, |\Im(\mathbf{x})|\} \tag{2.20}$$

By application of the l^1-norm, Equation (2.8) becomes:

$$|D_i(\mathbf{s}_i)| = |\Re(D_i(\mathbf{s}_i)| + |\Im(D_i(\mathbf{s}_i)|\} \tag{2.21}$$

and the partial ED in Equation (2.7) can be computed according to:

$$d_i(\mathbf{s}_i) = d_{i+1}(\mathbf{s}_{i+1}) + |D_i(\mathbf{s}_i)| \tag{2.22}$$

With this approximation, the squaring operation in Equation (2.8) is saved, which helps to reduce both delay and circuit area in a potential implementation. For the $l^{\tilde{\infty}}$-norm, the distance increment in Equation (2.8) is computed according to:

$$|D_i(\mathbf{s}_i)| = \max\{|\Re(D_i(\mathbf{s}_i))|, |\Im(D_i(\mathbf{s}_i))| \tag{2.23}$$

and the partial ED in Equation (2.7) becomes:

$$d_i(\mathbf{s}_i = \max(d_{i+1}(\mathbf{s}_{i+1}), |D_i(\mathbf{s}_i)|)) \tag{2.24}$$

In [14], it was shown that the application of the $l^{\tilde{\infty}}$-norm is beneficial in terms of the number of visited nodes as well as in terms of circuit area and clock frequency, while the BER performance is only slightly reduced compared to ML detection performance.

The **K-Best** detector is another algorithm that provides a close-to-ML solution. The K-Best algorithm for MIMO detection was first proposed in 2002 [15]. From a "tree search" point-of-view, it resembles a breadth-first "tree search". On each level of the tree, only the K nodes with the smallest partial EDs are further extended. Compared to SD, the throughput of the K-Best algorithm is constant. However, the BER performance is slightly degraded compared to SD and strongly depends on the chosen K. The K-Best algorithm is well suited for very-large-scale-integration system (VLSI) implementation due to the regular data path and the simple control flow. Architectural transformations like pipelining and resource sharing can easily be applied.

Another algorithm for hard-output MIMO detection is the fixed-throughput **fixed-complexity sphere decoding (FSD)** algorithm [16]. It achieves close-to ML BER performance and, like the K-Best algorithm, it exhibits a constant throughput. The FSD algorithm overcomes the problem of the variable complexity and the sequential behaviour of SD by searching only over a fixed but well-defined number of lattice vectors. A common configuration is to visit all nodes on the top level (i.e., on $i = M$) and only one node per parent node on the lower levels. A decisive factor that significantly contributes to the close-to ML BER performance of FSD is the order in which the streams are processed. The ordering is determined according to the number

of nodes that are visited on the same layer. On the layers where all nodes of a parent node are visited, the stream with the largest noise amplification is chosen; on the other levels, the streams are selected in ascending order of their noise-amplification. In [16], the ordering is called FSD ordering and was obtained via V-BLAST ordering and computed according to [17].

The number of operations, floating-point operations per second (FLOPS) or algebraic operations, required by a detection algorithm is expressed in the "big O" notation. However, its practical meaning may be limited. In particular, for MIMO systems of moderate size, constants and lower order contributions to the computational cost may also be relevant. Matlab^{TM} provides counting of FLOPS. Though this technique is obsolete, it provides a general overview of the complexity of each detection algorithm, where at this stage to be sufficient. Table 2.1 tabulates the FLOPS counts for each detection algorithm using Matlab^{TM} environment running a packet size of $1,024$ utilizing 4-QAM on a 4×4 AWGN channel. SD and K-Best algorithms have variable complexity whereby they are highly dependent on the size of the search radius Φ and the expanded node K. In this case, $\Phi = \infty$ and K is set to be 3.

High Performance			Low Complexity		
Detector	Type	KFLOPS	Detector	Type	kFLOPS
ML	Fixed	28.7	ZF	Fixed	1.7
SD	Variable	24.4	MMSE	Fixed	1.9
K-Best	Variable	21.1	SIC	Fixed	4.2
FSD	Fixed	16.8	V-BLAST/ZF	Fixed	4.8

Table 2.1: *Different algorithm complexity of MIMO detectors measured in kFLOPS*

Figure 2.4 shows the frame-error-rate (FER) curves for the addressed hard-output MIMO detection algorithms. The simulation results are for a 4×4 MIMO-OFDM system with a convolutional code rate of $\varphi = 1/2$. Each OFDM symbol consists of 64 subcarriers using 16-QAM. For the simulation results, perfect channel state information and perfect synchronization are assumed. The simulation results clearly show the large difference between hard-output low complexity linear ZF and MMSE or SIC detection and high performance K-Best and FSD in relation to the ML detection respectively. Since the algorithms of V-BLAST/ZF and FSD show similar inner workings (FSD requires the V-BLAST ordering), in the next chapter, a slightly modified version of the FSD algorithm incorporation with the V-BLAST/ZF, is presented to be the basis of the proposed efficient algorithm.

Better BER performance can be achieved by incorporating the APP in the detection. Figure 2.5

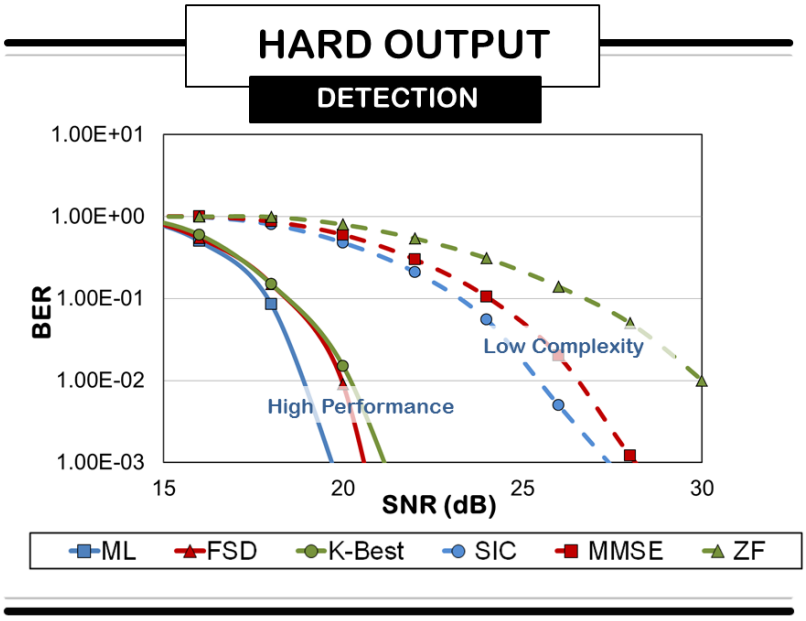

Figure 2.4: *BER performance comparison high performance and low complexity hard decoding* 16*-QAM with convolutional coding of* $\varphi = 1/2$

shows the BER performance for an optimum iterative soft-input soft-output MIMO detector with 4 iterations, for an optimum APP detector, and for an ML hard-output detector [8]. It can be seen that the BER performance for a convolutional coding with code rate of $\varphi = 1/2$ in binary phase-shift keying (BSPK) for additive white Gaussian noise (AWGN) channel shows significant improvement over the hard decoding equivalent. With an iterative-MIMO detector, the best BER performance can be achieved. However, the associated performance gains come at the cost of a substantially increased implementation complexity. This work will utilize the soft-output in the receiver.

2.2.4 Soft-Output MIMO Detection

As already shown in Figure 2.5, better BER performance in a coded MIMO-OFDM system compared to hard-output detection can be achieved by computing the APP for each hard bit, b, that associated to the transmitted symbol vector $\mathbf{s}$. Therefore, the aforementioned detection algorithms have to be adjusted to utilize the given soft-input information. The APPs are usually expressed as log-likelihood ratio (LLR) [18] [19] and are computed according to:

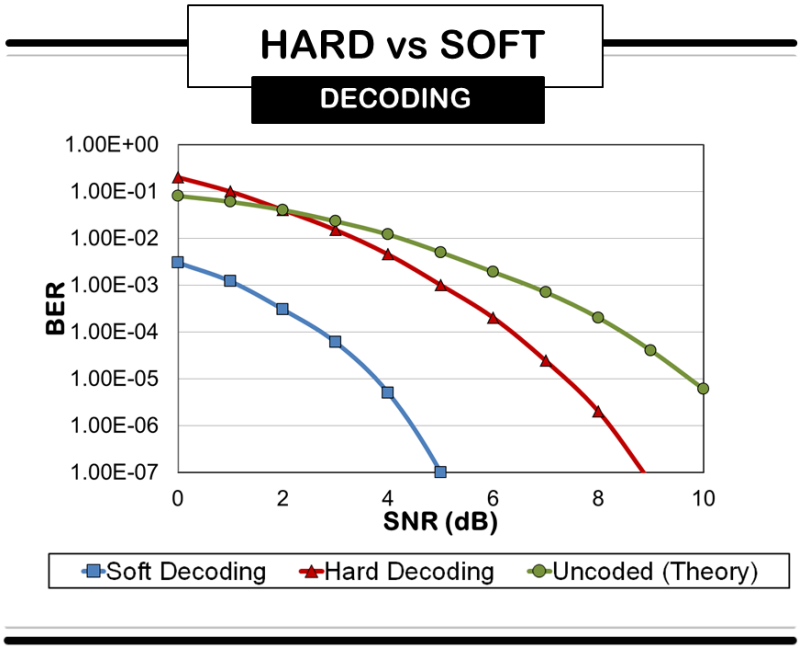

Figure 2.5: *BER performance comparison between hard and soft decoding BPSK with convolutional coding of* $\varphi = 1/2$

$$L_{i,b} \triangleq \frac{\ln P(s_{i,b} = +1|\mathbf{r}, \mathbf{H})}{\ln P(s_{i,b} = -1)|\mathbf{r}, \mathbf{H})} \tag{2.25}$$

for all bits b on level $i = 1, ..., M$. The sign of the LLR value $L_{i,b}$ shows whether bit $s_{i,b}$ is more likely to be $+1$ or 1 and the magnitude of $|L_{i,b}|$ indicates the probability of the estimate. The channel decoder takes advantage of the APPs and improves the estimate on the transmitted bits.

2.2.4.1 Soft-Output ML Detector

In [19], Equation (2.25) can be computed according to:

$$L(i,b) = \frac{\ln \sum_{\mathbf{s} \in \mathcal{Z}_{i,b}^{(+1)}} P_{\mathbf{r}}(\mathbf{r}|\mathbf{s}, \mathbf{H})}{\ln \sum_{\mathbf{s} \in \mathcal{Z}_{i,b}^{(-1)}} P_{\mathbf{r}}(\mathbf{r}|\mathbf{s}, \mathbf{H})} \tag{2.26}$$

under the assumption of equally distributed transmit symbols $\mathbf{s}$. The sets $\mathcal{Z}_{i,b}^{(+1)}$ and $\mathcal{Z}_{i,b}^{(-1)}$ are

subsets of $\mathcal{O}$, where the b^{th} bit of the i^{th} stream is equal to +1 and 1, respectively. By using the well-known Max-log approximation, Equation (2.26) can be simplified to:

$$L(i,b) \approx \frac{1}{\sigma^2}\left(\min_{\mathbf{s}\in\mathcal{Z}_{i,b}^{(-1)}} \| \mathbf{r} - \mathbf{Hs} \|^2 - \min_{\mathbf{s}\in\mathcal{Z}_{i,b}^{(+1)}} \| \mathbf{r} - \mathbf{Hs} \|^2 \right) \tag{2.27}$$

From Equation (2.6), it is obvious that always one of the two minima in Equation (2.27) corresponds to the ML solution. The other minimum in Equation (2.27) must be found by some other means. Note that Equation (2.27) can be transformed by applying the **QR**-decomposition. and then becomes:

$$L(i,b) \approx \frac{1}{\sigma^2}\left(\min_{\mathbf{s}\in\mathcal{Z}_{i,b}^{(-1)}} \| \hat{\mathbf{y}} - \mathbf{Rs} \|^2 - \min_{\mathbf{s}\in\mathcal{Z}_{i,b}^{(+1)}} \| \hat{\mathbf{y}} - \mathbf{Rs} \|^2 \right) \tag{2.28}$$

The APPs according to Equation (2.28) can be computed by soft-output MIMO detection. For example, SD can be used to compute the LLRs in Equation (2.28) to give soft-output FSD.

2.2.4.2 Soft-Output FSD

Soft-output FSD [19] computes Equation (2.28) based on a list $\mathcal{L}$ of candidate symbols. The candidate symbols are obtained by searching the tree according to the hard-output SD algorithm. However, two modifications are necessary. First, **radius reduction** is carried out at a slower rate, which is based on the largest element in $\mathcal{L}$. Second, whenever a leaf is found, its partial ED is written to the list $\mathcal{L}$. If $\mathcal{L}$ is already full and the partial ED associated with the new leaf node is smaller than the largest distance in $\mathcal{L}$, it is replaced. The search complexity strongly depends on the list size where a list size of 1 corresponds to hard-output SD, while larger list sizes are approaching the APP BER performance given in Equation (2.28). In [14], for example, a VLSI implementation results of soft FSD is presented.

2.2.4.3 Linear Soft Output Detector

Linear soft-output MIMO detection is a low complexity method to obtain approximate LLR values. Based on the MMSE solution in Equation (2.13) and by using Equation (2.27) the following approximate LLR values are obtained.

$$L(i,b) \approx \rho_i \left(\min_{\mathbf{s} \in \mathcal{Z}_{i,b}^{(-1)}} \| \hat{y}_i - s \|^2 - \min_{\mathbf{s} \in \mathcal{Z}_{i,b}^{(+1)}} \| \hat{y} - s \|^2 \right) \tag{2.29}$$

where ρ_i denotes the fading environment on the i^{th} stream [5].

$$\rho_i = \frac{1}{M\sigma^2[(\mathbf{H}^H\mathbf{H} + M\sigma^2\mathbf{I}_M)^{-1}]_{i,i}} - 1 \tag{2.30}$$

This can also applied to the ZF detection, with:

$$\rho_i = \frac{1}{M[(\mathbf{H}^H\mathbf{H})^{-1}]_{i,i}} \tag{2.31}$$

2.2.5 Iterative Decoders

Error correction codes provide the capability for bit errors introduced by transmission of a modulated signal through a wireless channel to be either detected or corrected by a decoder in the receiver. In this chapter, codes designed for errors introduced by AWGN channels and by fading channels are described. As shown in Figure 2.5, incorporating an iterative decoding method increases the BER performance. In this work, iterative receivers where MIMO detector and channel decoder exchange reliability information to increase the communications performance is investigated. Fading channel codes are either designed specifically for fading channels or are based on using AWGN channel codes combined with interleaving. The basic idea behind coding and interleaving is to randomize the location of errors that occur in bursts. Since most codes designed for AWGN channels do not work well when there is a long sequence of errors, the interleaver disburses the location of errors occurring in bursts such that just a few simultaneous errors occur, which can typically be corrected by most AWGN codes.

Several iterative coding methods often require increased bandwidth or reduced data rate in exchange for their error correction capabilities. Coupled with block or convolutional interleavers, these coding techniques are extremely powerful codes that exhibit near-capacity performance with reasonable complexity levels. Due to this reason, they are being implemented in current wireless communications. All of these coding techniques, from convolutional codes to turbo codes, provide coding gain at a cost of increased bandwidth or reduced data rate.

2.2.5.1 Convolutional Codes

Where block codes are based on algebraic/combinatorial techniques, convolutional codes are based on construction techniques. Convolutional codes offer an approach to error control coding substantially different from block codes. It can be seen in Figure 2.6, a convolutional encoder encodes the entire data stream into a single codeword and maps the information to code bits by sequentially convolving a sequence of information bits with specific "generator" sequences. The three important information required for this type of coding, the number of inputs κ, the number of outputs γ and the constraint length, K, where it has a memory of $K-1$ elements. In practice, the number of inputs is usually set as 1. The coding rate $\varphi = \frac{\kappa}{\gamma}$ determines the number of data bits per coded bits.

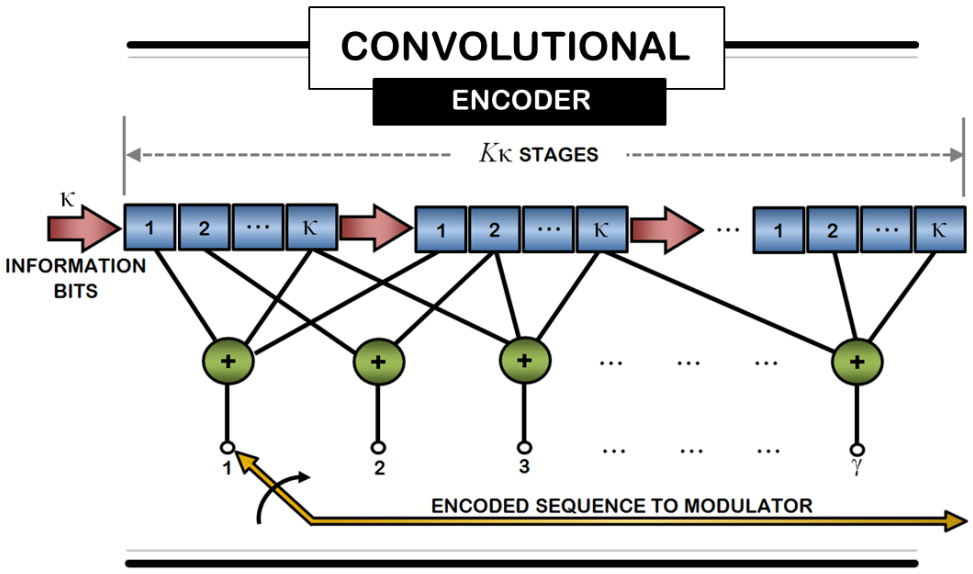

Figure 2.6: *Convolutional encoder*

The performance of a convolutional code depends on the φ and the K, whereby, the longer the K, the more robust the code and the coding gain. Coding gain is the measure in the difference between the SNR levels between the uncoded and coded systems required to reach the same BER level. However, this comes at a price of a more complex decoder and more decoding delay. In addition, smaller coding rate provides a more powerful code due to extra redundancy and it takes less bandwidth efficiency.

2.2.5.2 Turbo Codes

It is theoretically possible to approach the Shannon limit using either a block code with a large enough block length or a convolutional code with a large enough K. However, the processing power required makes this approach impractical. Turbo codes overcome this limitation by using recursive coders and iterative soft decoders. The recursive coder makes convolutional codes with short constraint length appear to be block codes with a large block length, and the iterative soft decoder progressively improves the estimate of the received message. Turbo codes can be generated using specific types of convolutional coding, which is called recursive systematic convolutional (RSC) coders. This work incorporates turbo codes as its error detection and correction method as it is used in the current communication systems.

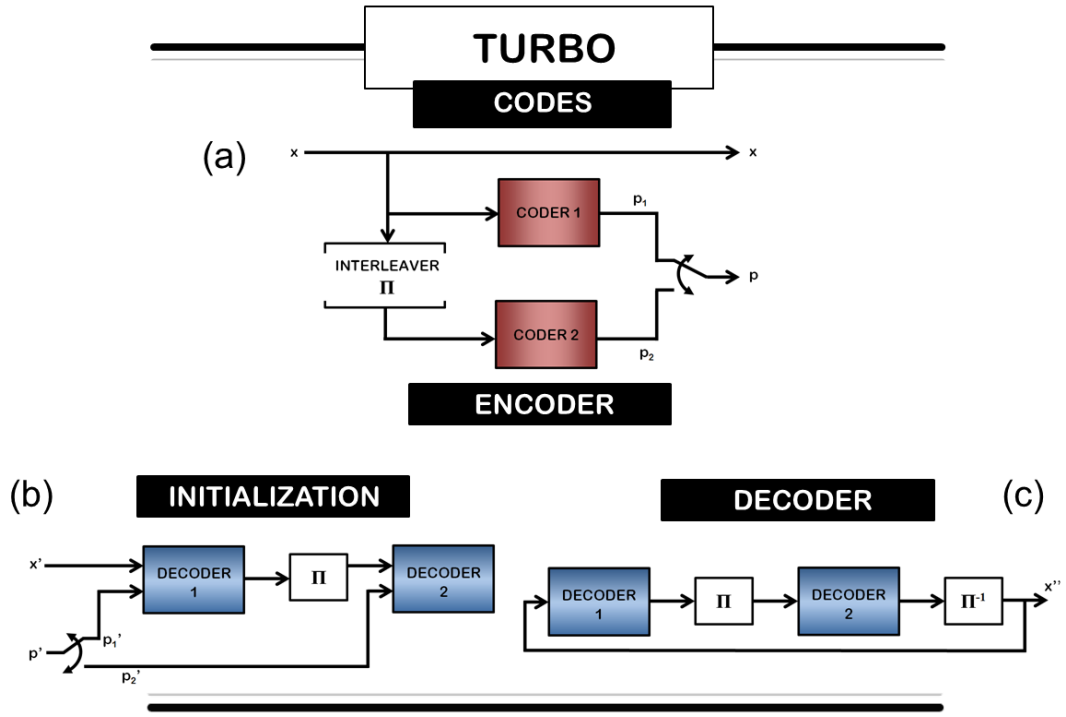

Figure 2.7: *Innerworking of turbo codes for both (a) for encoder, (b) and (c) for decoder*

A turbo code is the parallel concatenation of a number of RSC codes. The input to the second decoder is an interleaved version of the systematic **x**, thus the outputs of coder 1 and coder 2 are time displaced codes generated from the same input sequence. The input sequence is only presented once at the output. The outputs of the two coders may be multiplexed into the stream giving a rate $\varphi = 1/3$ code, or they may be punctured to give a rate $\varphi = 1/2$. This is illustrated

in Figure 2.7(a). The interleaver design has a significant effect on code performance. A low weight code can produce poor BER performance, so it is important that one or both of the coders produce codes with good weight. If an input sequence **x** produces a low weight output from coder 1, then the interleaved version of **x** needs to produce a code of good weight from coder 2. Block interleavers give adequate performance, but pseudorandom interleavers have been shown to give superior performance.

At the receiver, the signal is demodulated with its associated noise and a soft-output provided to the decoder. The soft output might take the form of a quantized value of the decoded bit with its associated noise, or it may be a bit with associated probability. Most often it is the LLR, which is defined as in Equation (2.25).

The LLR is a measure of the probability that given a received soft-input r in **H**, the message bit x associated with a transition in the trellis is 1 or 0. If the events are equiprobable, then the output is 0, but any tendency for x towards 1 or 0 will result in positive or negative values of L. It is simplest to view the decoding process as two stages; initializing the decoder and decoding the sequence. The demodulator output contains the soft values of the sequence **x**' and the parity bits p'_1 and p'_2. These are used to initialize the decoder, as shown in Figure 2.7(b). The interleaved sequence is sent to decoder 2, while the sequence derived from **x**' is sent to decoder 1 and presented to decoder 2 through an interleaver. This re-sequences bits from streams **x**' and p_1 so that bits generated from the same bit in **x** are presented simultaneously to decoder 2, whether from **x**, p'_1 or p'_2.

The decoder may have some knowledge of the probability of the transmitted signal, for example, it may know that some messages are more likely than others. This *a priori* information assists the decoder, which adds information gained from the decoding process forming the *a posteriori* output. The decoder uses all this information to make its best estimate of the received sequence. The output is then deinterleaved and presented back to decoder 1, which makes its best estimate. Further iterations through decoders 1 and 2, with associated interleaving and deinterleaving, refine the estimate until a final version of the block, **x**", is presented at the output. This process is shown in Figure 2.7(c).

The two main types of decoder are maximum *a posteriori* probability (MAP) and the soft-output Viterbi algorithm (SOVA). MAP looks for the most likely symbol received, SOVA looks for the most likely sequence. Both MAP and SOVA perform similarly at high SNR. At low

SNR, MAP has a distinct advantage, gained at the cost of added complexity. MAP was first selected by [20] as the optimal decoder for turbo codes. MAP looks for the most probable value for each received bit by calculating the conditional probability of the transition from the previous bit, given the probability of the received bit. The focus on transitions, or state changes within the trellis, makes LLR a very suitable probability measure for use in MAP. SOVA is very similar to the standard Viterbi algorithm used in hard demodulators.

Most of the assessments of turbo code performance have resulted from simulation. In the ideal environment of a simulation, it is possible to produce highly impressive results. To apply turbo codes to real systems requires acceptance of real world constraints such as latency and computing power. Reference [21] has explored the performance of codes with parameters set to values that are more practical. The performance of turbo codes was influenced by four main factors, which are the number of iterations, K, interleaver design and puncturing. While there is considerable material reporting on the optimum performance of turbo codes, surprisingly, little material reporting on the performance of turbo codes in practical scenarios exists. Clearly, exploring the lower limits of turbo code performance can provide an insight into their practical limitations. Real decoders need to provide the best BER from the worst channel in the shortest time. A realistic implementation would have low bandwidth, and thus use punctured codes, short block sizes, few iterations and the lowest SNR capable of supporting the required service. With this in mind, some additional simulations were undertaken as part of this assignment to explore code performance in realistic implementations.

To show the performance of turbo codes, several simulations of Matlab^{TM} routines on turbo codes with a punctured turbo code at rate $\varphi = 1/2$ was used. The data block length was $1,024$ bits and a MAP decoder was used in the simulation. The results shown at Figure 2.8(a) are the BER against SNR curves for different number of iterations using 4-QAM modulation.

It can be seen that BERs of the order of 10^{-5} are achievable with SNR ≤ 3 decibels (dB) with modest numbers of iterations. However, with more iterations come more processing power and delay in processing. Therefore, it is apparent that there is a trade-off to be made between the number of iterations, processing power, and SNR when seeking a given BER. Another simulation was run to compare the performance of the MAP and SOVA decoders, particularly at low values of SNR. The results, shown at Figure 2.8(b) for iterations of 8, confirm that MAP is about 0.5 dB better than SOVA at low values of SNR. In addition to the costly operations of the detecting and decoding themselves, synchronization [22], channel estimation [23], and MIMO

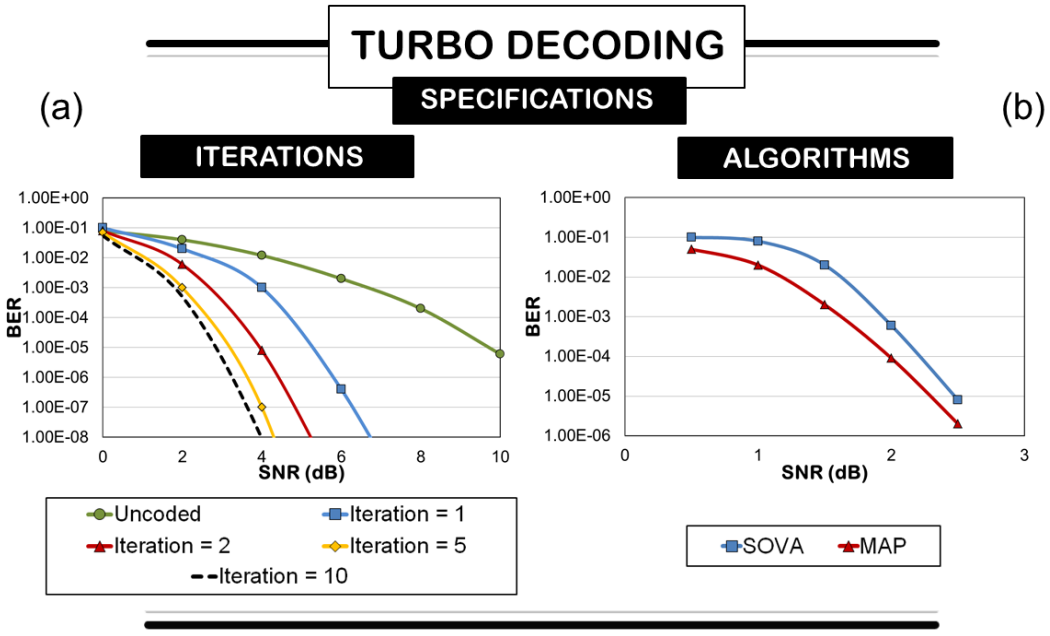

Figure 2.8: *BER for different specifications of turbo decoding according to (a) the number of decoding iterations and (b) different decoding algorithm*

preprocessing [24] also significantly account for the increased complexity of a MIMO receiver. Having discussed the detection and decoding algorithms and their corresponding approximations, the main goal of these receivers is almost always to optimize the BER performance while the required complexity is kept as low as possible. The worst-case complexity, however, remains exponential [25].

2.3 Hardware Architecture

In order to reduce the complexity and thus the power consumption of the detecting and decoding operations, efficient hardware implementations are needed. There are several power minimization techniques that may be incorporated concurrently with the detecting and decoding operations to reduce the overall complexity of the MIMO receivers. The descriptions of ones under investigation are included in the section.

2.3.1 Power Minimization Techniques

While there are many power minimization techniques at the processor core level that can be implemented on a programmable hardware, some work better than others depending on the hardware architecture and/or the applications. The power consumption of digital CMOS circuits is considered in terms of three components and will be described in detail in Chapter 4. Generally, they composed of the **dynamic power** component, which is related to the charging and discharging of the load capacitance at the gate output. The **short-circuit power** concerns component during the transition of the output line (of a CMOS gate) from one voltage level to the other. There is a period of time when the transistors are on, thus creating a path from supply voltage to ground. The **static power** component is mainly due to leakage that is present even when the circuit is not switching. This is composed of two components; the gate to source leakage, which is leakage directly through the gate insulator, mostly by tunnelling, and source-drain leakage attributed to both tunnelling and sub-threshold conduction.

This section aims to detail several power minimization techniques that can be applied and are potentially beneficial to iterative-MIMO receiver hardware design. They all have one thing in common whereby, the scaling of power enables the device to dynamically and proportionally change the energy consumption as its workload varies. This adaptivity in the hardware and software is the key solution to a more efficient design in the proposed algorithm and hardware design goals. It warrants an in-depth examination of the power equation in order to assess any given sustainably of chip architecture for power-sensitive applications today. This is realized by examining hardware power characteristics and their effects before diving into optimization tools and possible design solutions, which include, among others, clock and power gating, voltage and frequency scaling, partitioning of voltage, parallelization and pipelining.

2.3.1.1 Clock Gating

A straightforward technique to reduce dynamic power consumption is to reduce gate toggling either by reducing the number of gates in a device or minimizing the number of times each gate toggles i.e. the clock frequency. This technique achieves a power reduction by reducing the switching capacitance at the cost of computational speed.

The clock gating technique has been developed to avoid unnecessary power consumptions, like the power wasted by timing components during the time when the system is idle. Specifically

for flip-flops, clock gating means disabling the clock signal when the input data does not alter the stored data. It can be applied from the system level where the entire functional unit can be selectively set into *sleep mode*, or from the sequential/combinational circuit level where some parts of the circuit are in *sleep mode* while the rest of the block is operating. However, clock gating does not come for free. Extra logic and interconnects are required to generate the clock enabling signals, and the resulting area and power overhead must be considered [26]. Typically, the clock accounts for 20% to 40% of the total power consumption [27]. Figure 2.9(a) shows that without clock gating, the power consumption remains high. When clock gating is used to bypass the unused components of the system, as shown in Figure 2.9(b), a combinational logic where ENABLE controls when the clock signal is passed to the further stages.

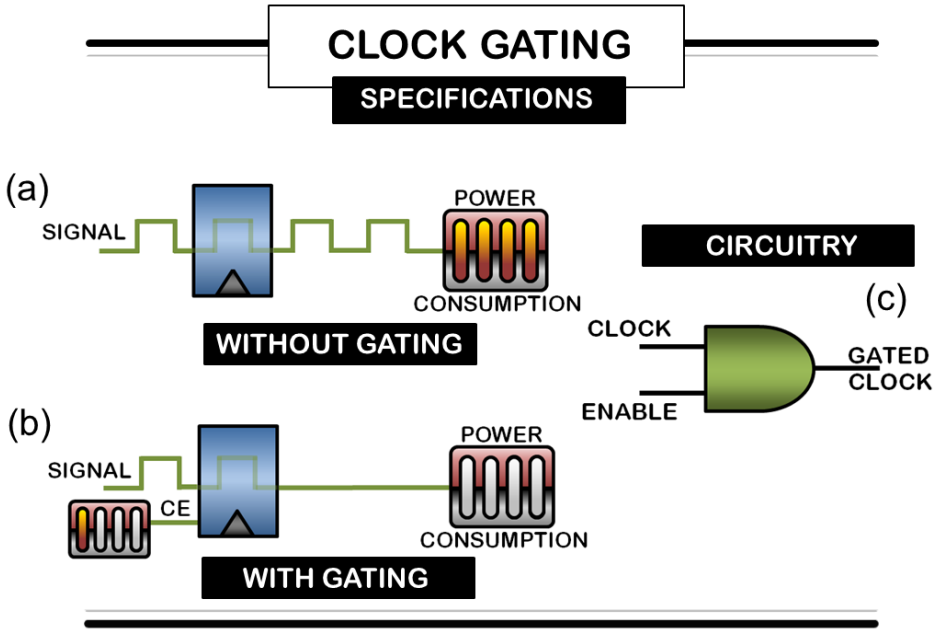

Figure 2.9: *The inner workings of clock gating where (a) is without clock gating, (b) is with clock gating and (c) is the clock gating circuitry*

Clock gating algorithms can be grouped into three categories [28], which are at **system-level**, **sequential** and **combinational**. **System-level** clock gating stops the clock for an entire block and effectively disables its entire functionality. On the other hand, **combinational** and **sequential** clock-gating selectively suspend clocking while the block continues to produce output. In [29], a logic synthesis approach for domino/skewed logic styles based on Shannon expansion is proposed that dynamically identifies idle parts of logic and applies clock gating to them to reduce power in the active mode of operation, which results improvements of 15% to 64% in

total power with minimal overhead in terms of delay and area compared to conventionally synthesized domino/skewed logic. The circuitry for implementation is simple, as shown in Figure 2.9(c), where an AND gate with clock and ENABLE signals are required to bypass the unused components. This circuitry adds little to no complexity when utilized to any system.

2.3.1.2 Power Gating

The basic strategy of power gating is to provide two power modes, which are a low power mode and an active mode. The goal is to switch between these modes at the appropriate time and in the appropriate manner to maximize power savings while minimizing the impact to performance. Power gating is one of the most effective techniques to reduce both sub-threshold leakage and gate leakage as it cuts off the path to the supply [30]. Figure 2.10(a) shows a simple schematic of a logic block that has been power gated by a header switch or a footer switch. While the logic block is not active, assertion of the SLEEP signal results in turning off either of the switches, thus disconnecting the logic block from supply, and reducing the leakage by orders of magnitude [31]. This technique is widely applied for implementing various *sleep modes* in control processing units (CPU). The examples of power gating architectures can be found in [32] [33]. Comparing both Figure 2.10(a) and Figure 2.10(b), where they show with and without power gating respectively, the amount of power leakage consumption is substantially reduced when the former is utilized, with the exception to a WAKE signal required earlier.

2.3.1.3 Dynamic Voltage and Frequency Scaling (DVFS)

Dynamic voltage and frequency scaling (DVFS) is an effective technique to attain low power consumption while meeting the performance requirements. Energy dissipation is reduced by dynamically scaling the supply voltage of the CPU, so that it operates at a minimum speed required by the specific task executed [34]. The technique principally involves scheduling in order to determine when each request of the task is to be executed by the processor and allows to slow down the processor, so that it consumes less power and takes greater time to execute. The tasks can be assigned priorities statically; when the priorities are fixed, or dynamical; when priorities are changed from one request to another. More information can be found in Chapter 4. However, to understand generally the power saving benefits of *DVFS*, consider a simple model for the dynamic circuit power consumption as shown in Figure 2.11(a). For easier understanding, only the dynamic power is discussed. Consider the completion of a task consisting

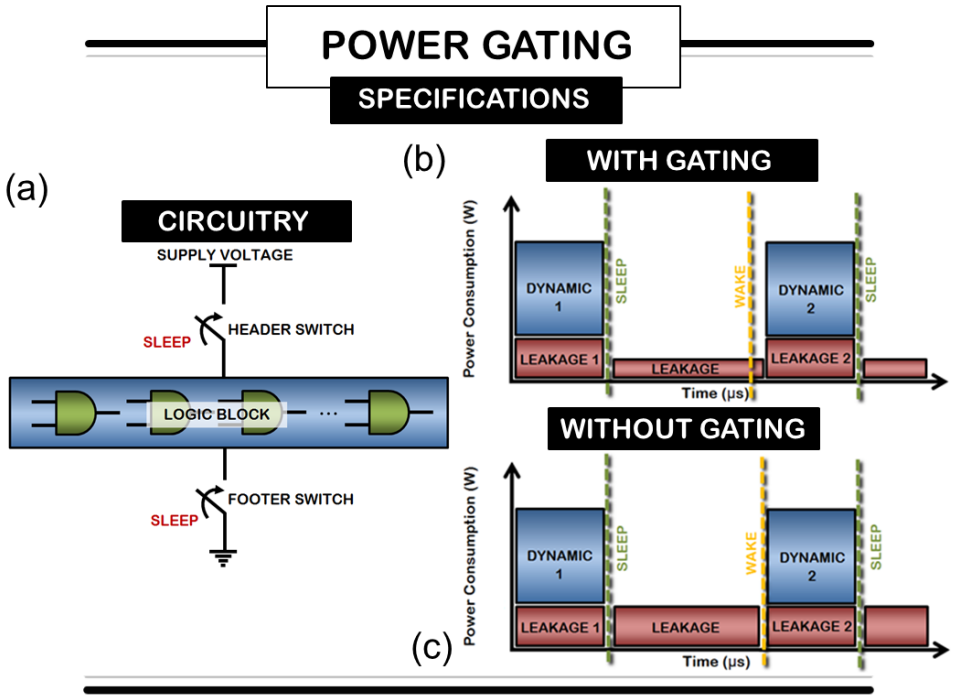

Figure 2.10: *The inner workings of power gating where (a) is the power gating circuitry, (b) is with clock gating and (c) is without clock gating*

of δ operations, which must be finished within a fixed time of τ seconds (s), and suppose that the voltage V, in volts (V) is chosen so that the processing finishes just-in-time, which is within the budgeted time frame allowed. The speed in which the circuit can be operated, and therefore the power, P in watts (W) required per operation, is a highly non-linear function of V, and it depends on the specific technology used and on the regime in which the circuit is operated [35]. Commonly, as a first-order approximation, the power required P for δ operations is modelled as a quadratic function of the circuit operating speed [36]. Therefore, for a fixed τ, P is:

$$P \propto \frac{\delta^3}{\tau} \tag{2.32}$$

An important insight when designing algorithms for circuitry that supports *DVFS* is that when there is a hard deadline at which the result must be available, and the quality of the computation result can be traded for a reduction in computational operations, then it is always better to run the circuit slower and finish just-in-time, as shown in Figure 2.11(c), than to run it fast and finish early so that it spends time idling, as shown in Figure 2.11(b). This is proved by Equations (2.33), (2.34) and (2.35).

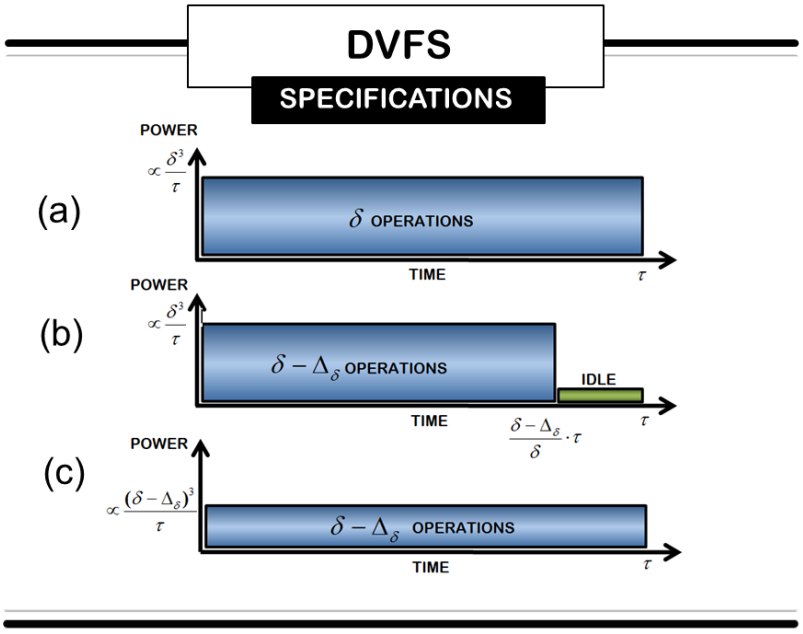

Figure 2.11: *The inner workings of DVFS where (a) without DVFS, (b) finishing early and (c) finishing just-in-time for dynamic power consumption*

Now, consider again the task above comprising δ operations. Suppose that the quality of the result may be compromised so that only $\delta - \Delta_\delta$ operations are needed as shown in Figure 2.11(c). If the circuit is run slowly in order to finish just-in-time after τ s, then the P required is given in Equation (2.33).

$$P_{run\ slowly} \propto \frac{(\delta - \Delta_\delta)^3}{\tau} \tag{2.33}$$

In contrast, if the circuit is run at nominal speed, it will finish at time $\frac{(\delta - \Delta_\delta)}{\delta} \cdot \tau$, thus requiring a power of:

$$P_{finish\ early} \propto \frac{(\delta - \Delta_\delta)\delta^2}{\tau} \tag{2.34}$$

By comparing Equation (2.33) and (2.34), it can be clearly seen that:

$$\frac{P_{finish\ early}}{P_{run\ slowly}} \propto \frac{\delta^2}{(\Delta_\delta)^2} \approx 1 + 2\frac{\Delta\delta}{\delta} > 1 \tag{2.35}$$

so running at full speed and finishing early would theoretically cost more energy than running slowly and finishing on time.

The calculations and modelling were based solely on one power component however, which is the dynamic power. Recent publication [37] has shown that the power consumption arises from powering up and keeping the chip active can no longer be ignored. The study also states that smaller chipsets require higher V in order to process the same data in comparison to their larger counterpart. Since the static power is highly dependent on the extrinsic properties of the chip as well as the operating temperature, the power can only be approximated, which is to be approximately a cubic function of the operating voltage. Therefore, new studies should no longer neglect the static power consumption when considering the power usage during hardware implementations. Chapter 4 confirms the fact that newer chipsets do have higher static consumption and therefore the power consumption considered in this study is a combination of both static and dynamic.

2.3.1.4 Multicores, Parallelization and Pipelining

The switch to parallel processing is natural with the rapid increase in computational demands of some applications, single-core architectures are just not capable of handling the amount of computation needed. Moreover, parallelization is related to energy consumption. Hardware architectures that perform many operations slowly in parallel are more power-efficient than architectures that perform a single operation very fast. Figure 2.12 shows that when a single core is used in comparison to multiple ones. Figure 2.12(a) depicts that even though over-clocking a single core does achieve higher performance, the power consumption is high in comparison to when a dual core is used, since the clocking can be lowered for lower power consumption.

To understand why this is so, consider again the above task comprising δ operations, and suppose that the computation is broken down into η parallel and equally large parts. Then each parallel circuit needs to perform δ/η operations within the time τ, and hence it can be fed with a lower V than supplied to the original circuit. The total amount of power consumed by the η parallel parts is:

$$P_{parallel} \propto \eta \cdot (\frac{\delta}{\eta})^3 = \frac{\delta^3}{\eta^2} \tag{2.36}$$

which is η^2 times less than Equation (2.32). Of course, this is an overoptimistic conclusion since if η is large, the static power consumption from leakage and the power from overhead in the circuit will be significant [35][36][38]. Moreover, the cores may then no longer operate in the super-threshold regime and the delay equations will drastically change [38].

Figure 2.12(b) on the other hand shows a simple power model for the power consumption taking 1 V to run one single core, in respect to the number of parallel cores utilized. It can be seen that the more cores are being used, the power is shared among them as given in Equation (2.36).

2.3.1.5 Multiple Voltage Islands

Voltage island is a popular method for implementing multiple supply voltages on a chip. It is an attractive method for reducing leakage power. Moreover, in comparison to *DVFS*, it is a static approach to reducing the dynamic power. Different blocks can be run at different voltages, saving power. Today's designs usually have multiple clocks running at different rates because

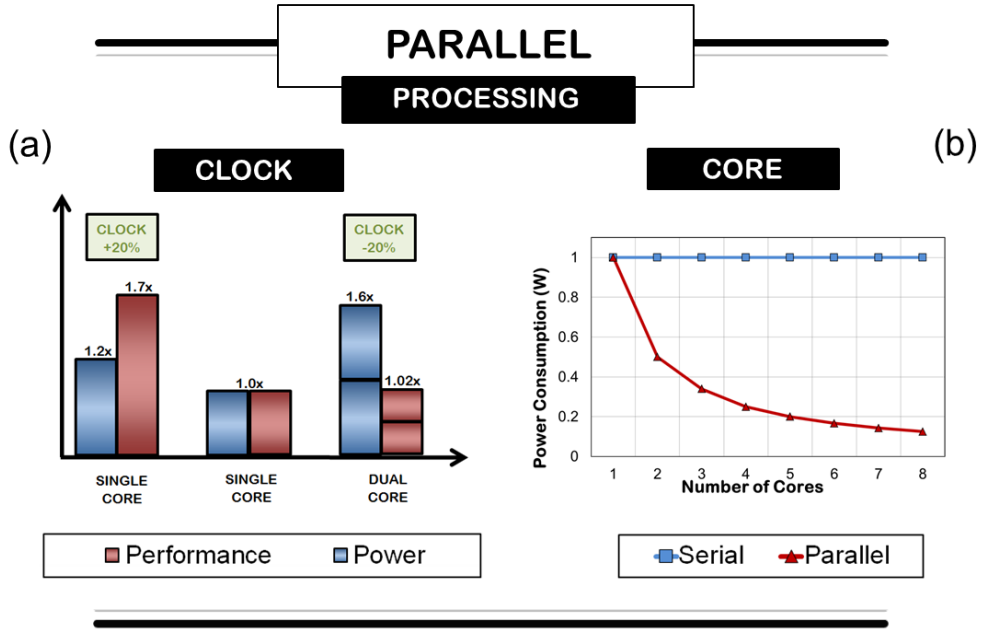

Figure 2.12: *The inner workings of parallel processing where (a) shows the effect of clocking and (b) the number of cores affecting the performance and power consumption on a hardware*

of the required performance of all functional blocks are not the same [39]. And the concept of voltage island was proposed in [40] to leverage voltage optimization of individual functional blocks of a system-on-chip (SoC) design. For example, the most performance-critical block like a processor core requires the highest voltage level while other functions such as memories or control logic, which co-exist on the SoC just require a low level of voltage. Voltage island formation can reduce the power consumption of a chip when there is a mixture of cores, which need to run at different levels of performance. A voltage island is a group of contiguous on-chip cores, which are powered by the same voltage level as shown in Figure 2.13. Without voltage islands as depicted in Figure 2.13(a), the chip voltage level has to be set at 1.0 V throughout. However, with voltage islands as shown in Figure 2.13(b), the total power consumption can be reduced by operating non-performance-critical cores at different voltages while the overall system performance is still maintained.

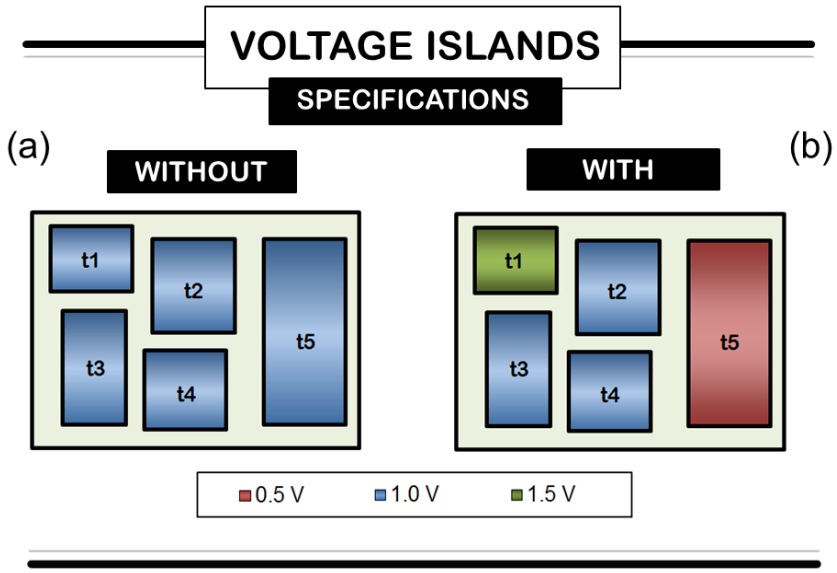

Figure 2.13: *The inner workings of voltage island where (a) without and (b) with voltage islands*

2.4 Chapter Summary

The aim of this chapter was to provide a comprehensive understanding in both areas of studies, which are the **wireless communication** and the **computer architecture**. Several algorithms and power minimization techniques have been described and their inner workings have been explored. Different detection algorithms are classified into two categories namely the high performance and the low complexity detectors. After careful deliberation on the **wireless communication** algorithms, FSD and V-BLAST working with ZF are chosen to construct the proposed efficient adaptive algorithm. This is due to FSD having the lowest complexity when compared to other high performance algorithms and it achieves comparable BER performance. Moreover, the parallel mechanism of the FSD may aid in power savings as well. The V-BLAST coupled with ZF is selected as the low complexity detector because of its similar mechanism to FSD. Therefore, this may lead to another power minimization technique, which is to share resources. This will help in reducing the chip size and thus the power consumption. The decoding process is chosen based on the latest LTE system, where a soft decision iterative turbo decoding using the MAP decoders is produced to give the best BER in comparison to other decoders. In **computer architecture**, to save power, in addition to parallelizing the algorithms and sharing of resources, the technique of *DVFS* and power gating will also be implemented on the hard-

ware design to investigate the effectiveness of the power minimization techniques to provide flexibility in the design.

Chapter 3
Adaptive Switching Algorithm

3.1 Chapter Contribution

The chapter presents an innovative design for the Adaptive Switching Algorithm in an iterative-MIMO detector suitable for both slow and fast fading environments for the purpose of saving power and optimizing energy consumption of the overall iterative-MIMO receivers. The algorithm works by switching between thresholds pre-calculated between the transmitters and receivers during each transmission in real-time. This novel idea is the first of its kind to produce an 'intelligent' system based on switching from a high to a low complexity detector, exploiting full information of the current channel condition of a MIMO system. The adaptivity has shown that potential savings can be gained in comparison to non-adaptive iterative-MIMO detectors. This positive outcome was also translated during preliminary implementation on an field-programmable logic array (FPGA), thus showing a promising design for future iterative-MIMO detectors.

3.2 Related Work

Current communication systems such as the LTE and the Institute of Electrical and Electronics Engineering (IEEE) 802.11 WiFi require immense resources to meet the demanding user data throughput needs. The ability to increase the throughput without requiring more computational power has always been a topic of interest amongst the wireless communication research community. Minimizing the power of the receiver, which is often limited, such as those that can be found on handheld mobile devices, is still under intensive study. Moreover, power and energy consumption of current base stations and proliferations of femtocells and/or wireless access points also need to exercise being 'green' since the sources are often shared among millions of devices. This amounts to substantial power usage, especially when there is an increasing trend [41] for the number of these devices to be active at one time, therefore, there are significant potential power and energy savings to be gained in these small mains powered devices as well.

This is where MIMO comes into play. MIMO promises higher throughput without additional transmit power [6]. It has been proven to be a promising technique in aiding this recent explosive growth of data volume by using multiple antennas in both the receive and transmit sides. It significantly improves the capacity and spectral efficiency of current wireless communication systems. Though this technique increases the data rate without affecting the power of the transmitter, the processing power of the receiver is often excessive. This chapter describes the attempts to minimize the power usage within the receiver, by designing a more efficient design for realistic implementation.

Fundamentally, an iterative-MIMO receiver is divided into two parts comprising the MIMO detector and the iterative decoder, working together to achieve the best performance. This iterative-MIMO scheme, which combines a spatial multiplexing MIMO detector and an outer forward error correction soft decoder with an interleaver in-between [42] [43], dubbed bit-interleaved coded modulation (BICM) [19], has very high computational complexity as the receiver detects and decodes symbols by searching through possible transmit symbols. Moreover, this is done iteratively in soft iterative-MIMO systems by the decoder.

There are many adaptive algorithms for these types of MIMO systems proposed in literature, many of which focus on the throughput [44] [45] and the overall performance [46] [47]. Only recently, a booming number of publications focus on power usage within the systems [48] [49] [50] [51]. However, the results are neither specific to hardware design implementation, nor do they concentrate on the latest wireless communication systems. Most adaptive systems study adaptivity in the form of changing between different MIMO techniques of beamforming, multiplexing and diversity [52] [53]. Though this helps in getting the best capacity in the MIMO, it does not convey the complexity of the system, and the power performance of the receiver. Some receiver-based studies such as [54] [55] [56] aim at linear detection using ZF for adaptivity in power allocation. These publications do not consider the latest state-of-the-art iterative-MIMO system such as the IEEE 802.11 WiFi or the LTE system. Reference [57] considers the LTE system and focuses only on the throughput while disregarding the burden the system has on power usage, which is an important parameter in current communication devices. Shifting specifically to the detectors and modes of power saving, most publications on adaptive MIMO detectors focus on saving power using the SNR [48], channel matrix condition number [58] or reducing the number of turbo decoding iterations [59] for the receiver as the method of switching parameters. Although they work to a certain extent, there is still

room for optimization where power usage is concerned. The SNR [48] does not determine the channel correlation relationship between the antennas in a MIMO system. Even if the channel is deemed good, due to high SNR values; strongly correlated antennas would still not make for a good transmission condition. This is because the correlated system provides insufficient diversity for reliable MIMO detections. Condition numbers [58] of the channel matrix on the other hand, would only take into account the input and output matrix of the transmitter and the receiver. This is not sufficient as a switching metric since it disregards the noise level. One publication, [60], presented a study of the MIMO adaptivity using the mutual information (MI) of the system. However, it only tracked the performance of the system while neglecting the effects it had on power consumption. The work described in this chapter has chosen to use the data readily available within the channel estimation block provided between the transmitters and the receivers. It considers the diversity of a MIMO system, which are the MI between the transmitters and the receivers, as well as the noise level of the current channel. This MI gives a maximum amount of information regarding a channel with minimal complexity in comparison to using either SNR [48] or the condition number [58] alone. In the upcoming Chapter 5, it shows that the MI does provide a more comprehensive knowledge about the channel. This is evident when the proposed algorithm is simulated on highly correlated channel conditions. It is discovered that unlike the channel matrix or the SNR, the MI is robust and is not affected by the change in antenna correlations. Thus, this further confirms that the usage of MI as a threshold design in the Adaptive Switching Algorithm would be beneficial to any systems regardless the channel conditions or antenna setup. The work of this chapter focuses primarily on the detector using MI as the threshold control in order to provide adaptivity, in the hope of achieving energy savings earlier at the processing stages i.e. by avoiding both detection and decoding processing.

The proposed Adaptive Switching Algorithm prevents the receiver from performing extensive computation under very low or very high SNR conditions, which ultimately yields significant savings in power and energy. The algorithm utilizes multiple thresholds to intelligently switch MIMO detection schemes according to the current environment. The Adaptive Switching Algorithm is unique in a sense that it is the first of its kind to utilize a high complexity "tree search" algorithm with a combination of low complexity "nulling and cancelling" algorithm adapting to the current channel condition in real-time. By exploiting the maximum information of the MIMO channel using the MI of each transmission condition, the diversity, spatial multiplexing and the noise level can be used to help decode the data using the right algorithm whilst maintaining the overall BER performance. Ultimately, using different detectors would only slightly

alter the thresholds that need to be implemented, confirming that MI is adaptive to any system for determining the threshold for switching. In other words, the idea behind the design is unique and can be implemented on any future communication system as well. This 'intelligence' is the key to efficient energy utilization in the receiver. The results of this work will be presented in terms of overall power and energy savings from both software and hardware design standpoints.

3.3 System Model Description

The system under consideration consists of four transmitters and receivers. The two parts of the receiver utilizes a BICM setup, which is a combination of a spatial multiplexing MIMO detector and an outer forward error correction soft iterative decoder with an interleaver in-between, both working together to achieve the best performance.

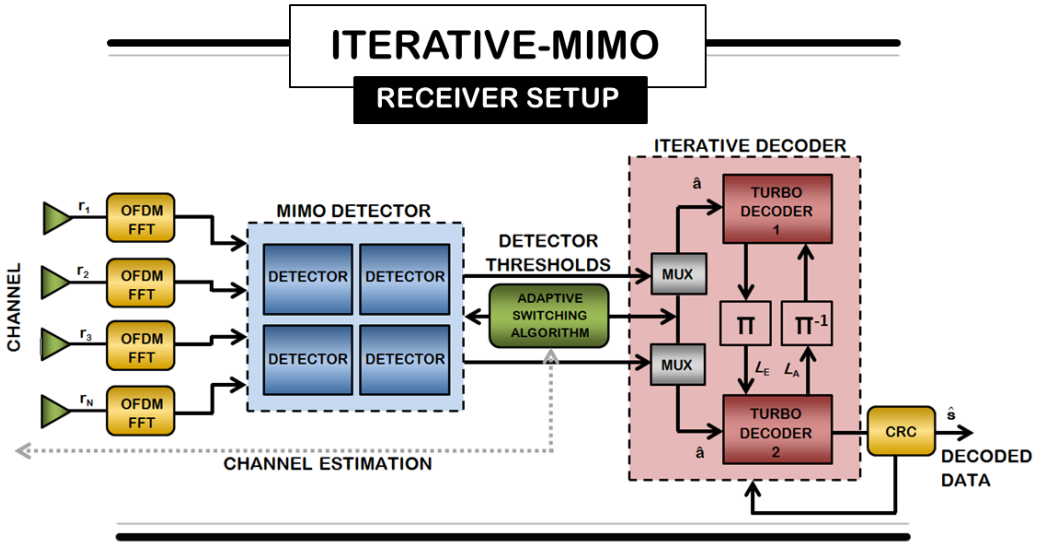

Figure 3.1: *Iterative-MIMO receiver system*

The system is simulated using a 4×4 MIMO system with QAM modulation symbols, $\mathcal{O}$, of point size $W = 4$, transmitting $1,024$ bits per packet of $100,000$ channel realizations utilizing an iterative-MIMO decoder of code rate, $\varphi = 1/2$, in a fast AWGN fading environment. The received data, $\mathbf{r}_N$, is processed through the detector before being passed to the decoder as shown in Figure 3.1 . The MIMO detector, where the focus of this chapter lies, then selects the appropriate detection algorithm depending on the MI calculated between the transmitter and the receiver in real-time. This threshold control provides adaptivity in the receiver, which is the key

to saving power in the computationally-expensive process. This is realized by selecting specific detection methods and consequently avoiding the decoding process in certain conditions. The detection methods chosen are explained in detail in the next sections. It should be noted that once the symbols are detected, they are passed to the iterative decoder, before a decision can be made. More details on the iterative decoding can be found in Chapter 5.

In the detector, there are many types of detection algorithms available. They can be generalized into "nulling and cancelling" methods, such as the ZF [61] and the MMSE [62] techniques as well as the "tree search" algorithms, for instance, the ML, SD [63], and the FSD [16] routines. For simple detectors, ZF and MMSE provide low complexity, however, they give poor performance in terms of BER. Linear detection methods, combined with "nulling and cancelling", seem to give a better BER whilst maintaining low complexity. In the system design, the combination of the simple V-BLAST and ZF is chosen and implemented due to it giving a balance of an acceptable BER performance and complexity in the high SNR region. This is particularly useful in good channel conditions, where the lack of noise in the channel means the symbols can be easily detected by the detection algorithm, using minimal computational resources.

On the other hand, for close to high ML performance, "tree search" algorithms such as FSD, layered orthogonal lattice detector (LORD), smart candidate adding algorithm (SOCA) and K-Best result in high complexity in order to meet the performance criteria. This drains quite significant power in order to decode data packets, especially when used in good channel conditions. However, these are useful during transmissions on noisy channels. In such poor channel conditions, FSD has been chosen as a detection method. Moreover, for easier hardware implementation, FSD is used as it is independent of the Φ, meaning, the complexity is fixed and minimal in comparison to other "tree search" algorithms. The computational power required to implement "tree search" MIMO detection every time a symbol is transmitted is unnecessary in some channel conditions. These two algorithms work in tandem according to the threshold design based on the MI of the current channel conditions. As each detection algorithm has a different performance and complexity, choosing between them depends on the unique requirements of the system. FSD and V-BLAST/ZF techniques are incorporated into an adaptive approach that has the ability to selectively operate according to the received signal conditions in real-time. These two detection algorithms are chosen due to their fixed data throughput, potential for hardware parallel implementation and relatively low complexity for their own particular detection group. Moreover, FSD can be seen as multiple V-BLAST/ZF algorithms working

together at the same time. This provides room for optimization for chip area utilization when the same parts of the chip can be reused for both algorithm implementations.

3.3.1 V-BLAST/ZF

ZF is a simple and effective technique for retrieving multiple transmitted data streams at the receiver. It has a relatively simple structure and good performance at high SNR. ZF provides sub-optimal performance offering significant complexity reduction with tolerable performance degradation. This method works by neglecting the constraint $\mathbf{s} \in \mathcal{O}^M$ in ML detection and uses different criteria to find the nulling vectors, the most common being the ZF or MMSE approach [64]. Generally, the symbol $\hat{\mathbf{s}}$ is given by a transformation of the received vectors $\mathbf{r}$ in the form of:

$$\hat{\mathbf{s}} = Q(\mathbf{G}\mathbf{r}) \tag{3.1}$$

where $\mathbf{G}$ is the Moore-Penrose pseudoinverse matrix that depends on channel $\mathbf{H}$ and Q is a quantizer that maps the argument into the closest point in $\mathcal{O}^M$. Even though this method has low complexity, it does have a major drawback of having a rather poor performance in terms of BER when implemented on an iterative-MIMO system, especially during bad channel conditions.

V-BLAST on the other hand, is a method proposed by [65] and it may achieve very high spectral efficiency promised for MIMO systems [3] [4] [66]. It gives slightly better BER performance in comparison to linear detection. However, due to the error propagation, it is still sub-optimal in performance. This is often overlooked due to its practicality during implementation. V-BLAST is a recursive procedure that works by minimizing the influence of noise by re-ordering the channel matrix according to the signal strength received. The algorithm simply makes a first detection of the most powerful signal, consequently subtracting that signal from the overall detected symbols. It then continues the same process by proceeding to the detection of the second most powerful signal and so forth.

Assuming the ordered set for a series of channel realization k to be:

$$\S \equiv \{k_1, k_2, \ldots, k_M\} \tag{3.2}$$

the detection algorithm operates on $\mathbf{r}_i$, given in Equation (3.3), while computing the decision statistics $y_{k_1}, y_{k_2}, \ldots y_{k_M}$, which are then quantized to form estimates of the received symbols $\hat{\mathbf{s}}_{k_1}, \hat{\mathbf{s}}_{k_2}, \ldots \hat{\mathbf{s}}_{k_M}$. The detection order is determined by the information about the channel conditions readily available within the estimation block. After computing Equation (3.1), the detection process uses linear combinatorial nulling and symbol cancellation to successively compute the received vectors.

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \hat{\mathbf{s}}_{k_i}(\mathbf{H})_{k_i} \tag{3.3}$$

In the original V-BLAST method [65], parallel data streams are simultaneously transmitted through multiple antennas in the same frequency band, and decoded at the receiver with ZF-SIC detector, which helps attain high spectral efficiency with reasonable computational decoding complexity. Therefore, it can be said that when combined with the ZF method, the V-BLAST/ZF method shows some improvement in BER while still maintaining low complexity. Due to these advantages, V-BLAST/ZF has gained lots of attention [17] [67] [68] [69]. The complete V-BLAST/ZF detection algorithm is summarized in Table 3.1, where $\mathbf{G}$ denotes the Moore-Penrose pseudoinverse of the current channel $\mathbf{H}$, and therefore, $(\mathbf{G}_i)_j$ is the j^{th} row of $\mathbf{G}_i$, $Q(\cdot)$ is a quantizer to the nearest constellation point, $(\mathbf{H})_{\bar{k}_i}$ is the k_i^{th} column of $\mathbf{H}$, $\mathbf{H}_{\bar{k}_i}$ denotes the matrix obtained by zeroing the columns $k_1, k_2, \ldots, k_i$ of $\mathbf{H}$, and $\mathbf{H}^+_{\bar{k}_i}$ denotes the pseudoinverse of $\mathbf{H}_{\bar{k}_i}$. This type of detection scheme is best deployed in high SNR environments.

3.3.2 FSD

FSD is an algorithm proposed by [70], which was derived from the original SD detection algorithm. SD reduces the complexity of the ML detection problem [71] [72] [73] by introducing a constraint within the search called the sphere radius, Φ.

$$\hat{\mathbf{s}}_{SD} = \underset{\mathbf{s}\in\mathcal{O}^M}{\arg\min} \parallel \mathbf{r} - \mathbf{Hs} \parallel^2 \leq \Phi^2 \tag{3.4}$$

The search can be visualized as a tree, traversing down each node until it encounters one with ED that is larger than Φ, where it will eliminate that branch from the search as shown in Figure 3.2(a). The minimum symbol is acquired once it has traversed down through every path of

Pseudo-Code

Channel realization:

$\mathbf{G_1} = \mathbf{G}$

$i = 1$

Recursion:

$k_i = \arg\min_{j \notin \{k_1, \cdots, k_{i_1}\}} \parallel (\mathbf{G}_i)_j \parallel^2$

$y_{k_i} = (\mathbf{G}_i)_{k_i} \mathbf{r}_i$

$\hat{\mathbf{s}}_{k_i} = Q(y_{k_i})$

$\mathbf{r}_{i+1} = \mathbf{r}_i - \hat{\mathbf{s}}_{k_i}(\mathbf{H}_{k_i})$

$\mathbf{G}_{i+1} = \mathbf{G}_{\bar{k_i}}$

$i = i + 1$

Table 3.1: *V-BLAST/ZF algorithm*

every level, i, reaching the end i.e. the leaf node(s). The SD has major drawbacks when it comes to hardware implementation due to its variable complexity and sequential nature. The complexity of the SD depends on the noise level and the channel conditions, which determine the size of Φ. Moreover, the linearity of the search prevents parallelism for newer hardware design implementation.

Parallelization has been proven to minimize power and energy consumption in circuit designs due to the workload being shared across multiple computational resources, so that the circuit can produce the same amount of throughput at a lower frequency of operation [74] [75] [76]. Therefore, [16] proposed a modified version, the FSD, in order to overcome both shortcomings. FSD is a combination of brute-force enumeration and a low complexity, approximate detector. Much like the SD, FSD traverses down the tree, as shown in Figure 3.2(b), whilst calculating the ED. Instead of having Φ, FSD determines in advance the number of lattice points $\hat{\mathbf{s}}$ around received signal $\mathbf{r}$ it would pass through, evaluating $\mathbf{r}$ independent of the noise level, giving it a fixed throughput. The algorithm makes use of the fact that the diagonal entries of $\mathbf{R}$ from the $\mathbf{QR}$-decomposition of the channel matrix satisfy [77]:

$$E[\mathbf{r}_{11}^2] < E[\mathbf{r}_{22}^2] < \cdots < E[\mathbf{r}_{NN}^2] \tag{3.5}$$

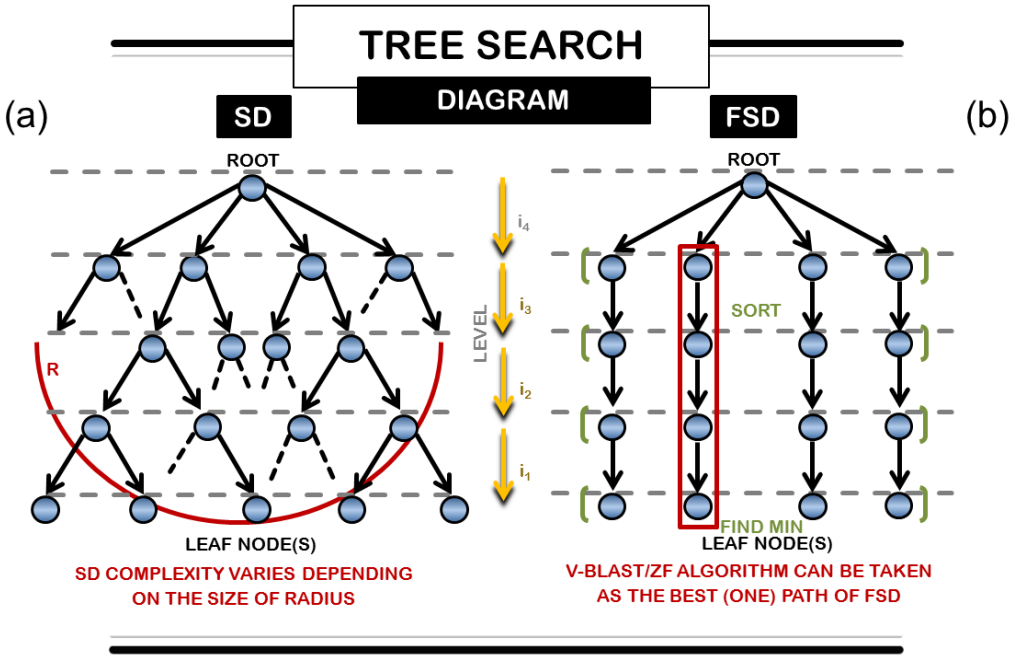

Figure 3.2: *Tree structure of (a) SD and (b) FSD and V-BLAST/ZF algorithms*

Thus, the number of candidates at antenna level i denoted by n_i should follow:

$$E[n_N] \geq E[n_{N-1}] \geq \cdots \geq E[n_1] \tag{3.6}$$

The main idea of FSD is to assign a fixed but distinct number of candidates, n, to be searched per antenna level. The FSD is considered a promising algorithm for soft iterative-MIMO detection. Since its introduction, the reduction of complexity in FSD has received significant attention [70] [78] [79] [80] [81]. After the matrix decomposition and removal of constant terms, Equation (3.4) can be written as:

$$\|\mathbf{U}(\mathbf{s} - \hat{\mathbf{s}})\|^2 \leq \Phi^2 \tag{3.7}$$

where $\mathbf{U}$ is an $M \times M$ upper triangular matrix with entries r_{ij}, obtained through the **QR** of $\mathbf{H}$ to give $\mathbf{G}$, and $\hat{\mathbf{s}}$ is the unconstrained ML estimate of $\mathbf{s}$ [19]. The solution for Equation (3.7) can be recursively calculated starting from $i = M$ until level $i = 1$ for each channel realization of k, which is checked on each iteration of i and the constellation point s_i are expanded as

Pseudo-Code

Channel realization:
$\mathbf{G_1} = \mathbf{QR}$
$i = M$
$\hat{\mathbf{y}} = \mathbf{Q}^H \mathbf{r}$

Expand all nodes on the first level

Recursion:
while $i \neq 1$
$d_i = D_i + d_{i+1}$
$d_{i+1} = \sum_{j=i+1}^{M} r_{jj}^2 |s_{k_j} - \hat{y}_{k_j}|^2$
$D_{ii} = r_{ii}^2 |s_{k_i} - \hat{y}_{k_i}|^2$
$i = i - 1$
end

Choose minimum path for ED
$\hat{\mathbf{s}} = \hat{s}_{k_{N-1}}, ..., \hat{s}_{k_1}, \hat{s}_{k_0}$

Table 3.2: *FSD algorithm*

partial ML candidates. When a point is found when $i = 1$, the solution is updated with the new minimum ED and the algorithm continues the search. The breakdown of the algorithm is given in Table 3.2. The recursion added the partial accumulated Euclidean distance (AED), d_i, to the ED, D_i, accumulating on each level until the search reached the bottom of the tree, which is the leaf node(s). Once the search reached the leaf node(s) or when the level $i = 1$, the minimum ED is chosen as the solution for $\hat{\mathbf{s}}$.

The V-BLAST/ZF algorithm works by predicting the best path of the FSD without authenticating hoping that it would yield the correct output. This is illustrated in Figure 3.2(b). Therefore, the former algorithm is inferior in performance in comparison to the latter detection algorithm. The chosen algorithms of FSD and V-BLAST/ZF are the cornerstones for the proposed detection algorithm for this chapter. They work together as one detector switching from one to the other based on the current channel condition and the noise level, which are the information between the transmitter and the receiver, MI. Moreover, they are chosen due to their similar mechanism in a way that they may be able to share hardware resources when searching through the possible transmit symbols. V-BLAST/ZF traverses one path of the FSD detection tree, choosing the one with the best SNR condition, optimistically assuming the path would yield

the correct results. The sharing of hardware may lead to further power and energy savings during implementation. This is the basic process for the proposed detection algorithm; the Adaptive Switching Algorithm.

3.4 Adaptive Switching Algorithm

Current MIMO detectors usually lack adaptivity whereby all receivers behave exactly the same way regardless of the received signal characteristics as well as the current channel conditions. This 'one size fits all' architecture does not work well in some situations, since different users experience distinct channel conditions and/or current channel conditions. For example, a stationary user who is physically near to a transmitter would often have a better data throughput than one who is further away. Doppler rates determined by motion in the environment also play a part in determining the current condition of the channel. To decode symbols in bad channel conditions would prove to be pointless since the data would not be likely to be decoded successfully anyway. Therefore, having 'intelligence' in the detector that could modify its behaviour according to current channel conditions would be ideal. This adaptivity in the proposed algorithm, dubbed the Adaptive Switching Algorithm, is controlled by the MI calculation between the transmitters and receivers. These MI values calculated in the channel block then determine, which detection methods to be deployed in the iterative-MIMO receiver, whether V-BLAST/ZF, during in high SNR regions, or FSD when the receiver needs extra support to decode the data due to bad channel readings. It is well-known that the MI of a MIMO channel is given by Equation (3.8) and the information required, $\mathbf{H}$, is already available within the channel estimation block. Different values of initial received soft information may lead to significantly different behaviour during the iterative decoding process. The study performed by [82], which compares the performance of iterative decoders using different received soft LLR information metrics, discovered that by computing the MI, the number of iterations in turbo decoding can be found using the highest complexity ML MIMO detection method. Reference [82] also proves that the best approximation of the received symbols obtained are lossless and that the exact LLR values are sufficient statistics of $\mathbf{r}$ about $\mathbf{s}$. Therefore, using this information and the principle of exploiting MI calculation in Equation (3.8), the work applies this approach for the first time to a MIMO detector to further save power and energy consumption in the overall receiver. With any given channel model in Equation (2.1), and a Gaussian constellation with $E[|\mathbf{s}_i|^2] = M^{-1}$, the MI for the ML method is

$$\bar{I}(\mathbf{H}_k) \triangleq \log_2 \det\left(\mathbf{I} + \frac{\mathbf{H}_k \mathbf{H}_k^H}{N_0}\right) \tag{3.8}$$

The values of MI are spread on a range for a given value of SNR. Figure 3.3 illustrates the accumulated MI performance of the detector as a function of probability of receiver failures and successes according to the system model description. The results obtained are specific to the system model setup, however, they can be translated to any modulation scheme, number of antennas with variations of channel modelling, only with the exception of minor alteration of the threshold values. The principle behind this design is therefore valid for any current and future communication systems as well, more information of which is included in Chapter 5.

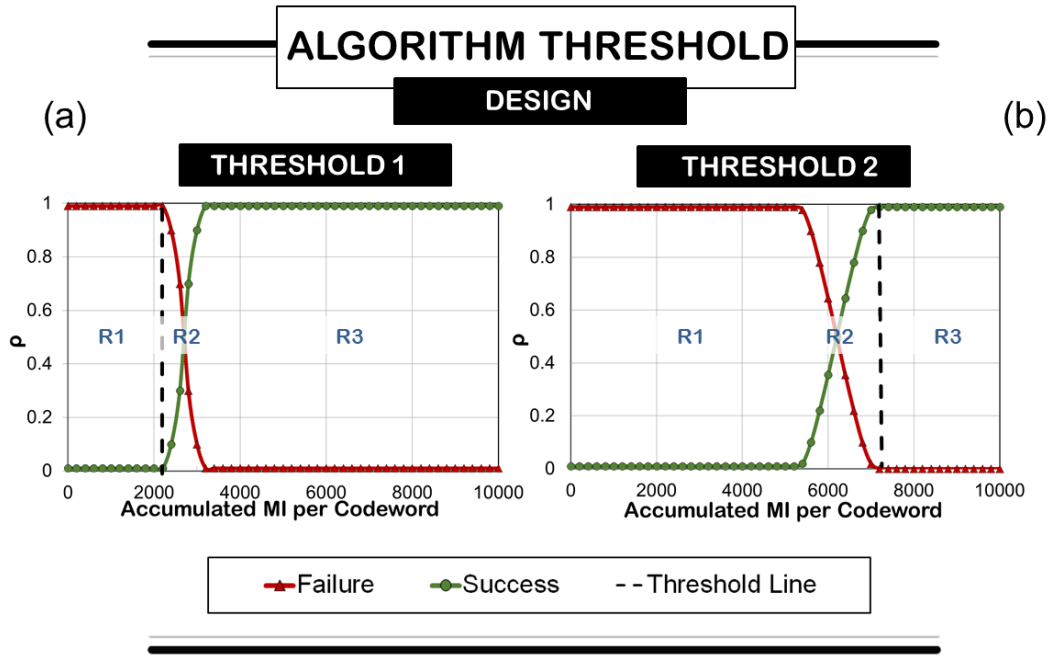

Figure 3.3: *Probability of receiver successes and failures for a* 4×4 *MIMO where (a) for the FSD method and (b) for the V-BLAST/ZF method*

Threshold 1, T_1 can be obtained in Figure 3.3(a), which shows the FSD performance. Region R1, which is below a certain MI threshold of approximately $2,200$, is where the receiver is certain to fail, with the error probability distribution of 1, when trying to decode a symbol message. With 100% decoding rate of failure, the best course of action for the receiver is to request a retransmission from the automatic repeat request (ARQ) block from the transmitter rather than to attempt decoding where it is unlikely to succeed, only wasting significant com-

Pseudo-Code

Channel realization:$\{\mathbf{H}_1, \mathbf{H}_2 \cdots, \mathbf{H}_k\}$

for $\mathbf{r}_i \leq \mathbf{r}_k$

$\bar{I}(\mathbf{H}_k) \triangleq \log_2 \det(\mathbf{I} + \frac{\mathbf{H}_k \mathbf{H}_k^H}{N_0})$

if $\bar{I}_i \leq T_1$
$\mathbf{r}_i$ error, request ARQ

elseif $T_1 \leq \bar{I}_i \leq T_2$
$\mathbf{r}_i$ with low MI: FSD

else $\bar{I}_i \geq T_2$
$\mathbf{r}_i$ with high MI: V-BLAST/ZF

endif

endfor

Table 3.3: *Adaptive Switching Algorithm*

putational energy, whilst yielding no correct output. Current wireless communication systems would attempt to decode nonetheless and to only stop until the number of set maximum iterations are completed. This is the limitation of current system designs. On the other hand, the V-BLAST/ZF performance is shown in Figure 3.3(b). In region R3, the value for threshold 2, T_2 of about $7,100$ can be seen. The receiver will decode the symbol message with very high probability above this MI value, therefore, a simpler detection method will suffice in detecting the symbol, which is the V-BLAST/ZF method. In addition, the area in-between (where the two curves intersect between the two figures i.e. $T_1 \leq \bar{I}_i \leq T_2$), region R2, the two thresholds shows that the receiver would sometimes fail to decode. Thus, a more powerful detection method is needed to assist the receiver in decoding the message. This is executed by deploying the FSD algorithm in the MIMO detector. By obtaining these thresholds, the design of the Adaptive Switching Algorithm can be described in Table 3.3.

3.5 Results and Analysis

The effectiveness of the Adaptive Switching Algorithm can be measured using the performance and complexity trade-off metrics. This section describes these efficiencies from both hardware and software perspectives.

3.5.1 Software Performance

The performance can be quantified by calculating the number of errors in a total frame, which is the BER analysis. The system design has been set to tolerate a BER of 10^{-3} or less in high SNR regions. The detector is designed in such a way that it may be able handle one error per 1000 packets transmitted, thus giving the BER threshold line of 10^{-3}. This BER threshold line is considered sufficient to maintain a satisfactory performance for the system under consideration. It should be noted that, when different coding schemes is added, this threshold may be adjusted lower to fit the requirement of any system. In the system model used, the BER is depicted in Figure 3.4. The Adaptive Switching Algorithm gives similar performance to the FSD and performs much better than the V-BLAST/ZF algorithm in low SNR regions. In very high SNRs of about 10 dB and above, the less complex algorithm of V-BLAST/ZF is adopted and the BER performance is below the set error tolerance line, which works under the design specification for the performance of the overall system design. The FSD does give a much better performance than the tolerance line, however, this level of performance is unnecessary and only adds extra complexity for the hardware. When the SNR is below 0 dB, the receiver abandons the detection process, subsequently avoiding the complexity of the iterative decoding process as well, gaining substantial power and energy savings by requesting an ARQ from the transmitter, saving power in the total iterative-MIMO receiver. Furthermore, the area above the set error tolerance line and before the area where retransmissions occur, which takes place circa 0 dB to 6 dB, the Adaptive Switching Algorithm provides much higher chances of successful processing in comparison to the V-BLAST/ZF method. The performance of the Adaptive Switching Algorithm is therefore better than the generic V-BLAST/ZF detector.

By obtaining the thresholds, the total usage of each MIMO detection algorithm throughout the span of the SNR can be obtained and is shown in Figure 3.5, where it depicts transmissions of $1,000$ packets of $1,024$ bits per frame over $100,000$ channel realizations. It clearly shows that below an SNR value of 0 dB i.e. T_1, no processing is taking place. In addition, in high SNR

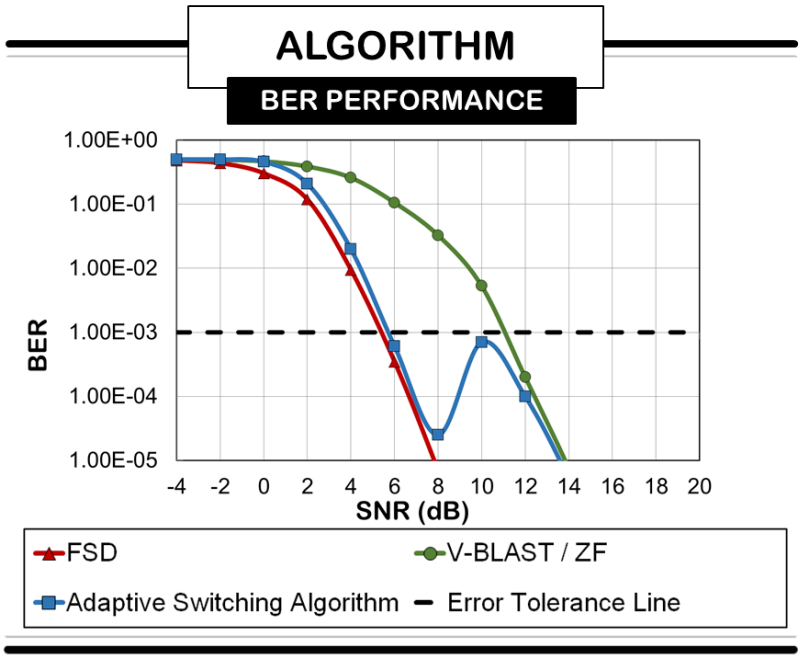

Figure 3.4: *BER performance of different detectors on a complex* 4×4 *MIMO system*

regions, V-BLAST/ZF is utilized. This figure concurs with Figure 3.4, where the performance coincides with the algorithm switching rate of success, particularly evident at SNR of below 2 dB for when ARQ is active and no decoding is taking place, and SNR values between 8 dB and 12 dB, when the switching between the high performance FSD to the low complexity V-BLAST/ZF. In addition, at an SNR of above 14 dB, only V-BLAST/ZF is utilized the entire time. From this, another part of the parameter, i.e. the complexity measurement of the software can be determined.

The complexity measurement gives an important overview of the hardware before the design implementation and provides initial indications of power and energy savings in hardware. A preliminary complexity analysis of the Adaptive Switching Algorithm is determined by the multiplier counts in the code. Assuming that the complexity of channel ordering is the same for both detection schemes, the multiplier counts for a transmission of one symbol for 4×4 M-QAM deploying FSD is M-times more than V-BLAST/ZF. Figure 3.6 plots the percentage complexity results against the SNR of the channels, where 100% equals the complexity of FSD, while the V-BLAST/ZF requires only 25%. Taking the FSD as a baseline for the complexity calculations, the complexity of the Adaptive Switching Algorithm can be calculated by

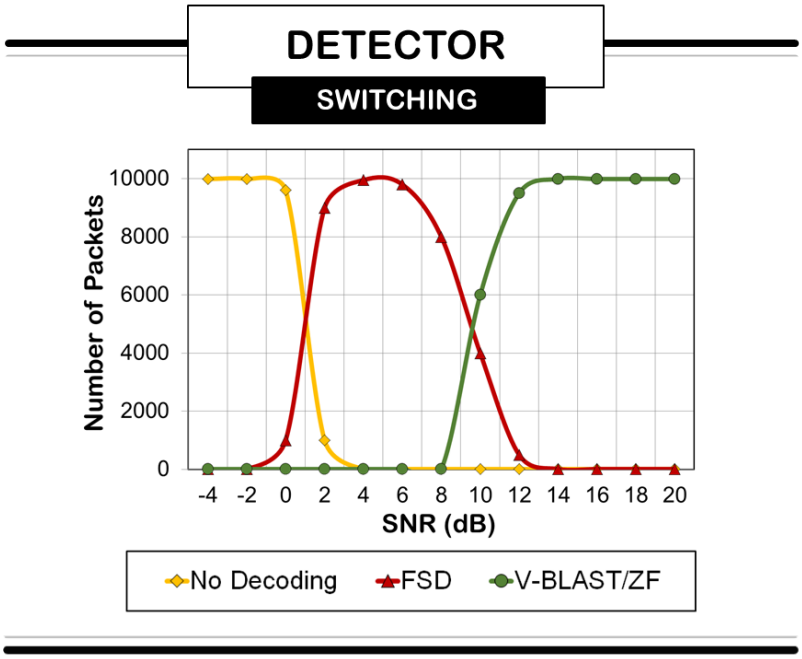

Figure 3.5: *Detection algorithm switching selection in iterative-MIMO receiver*

averaging over MI values shown at certain SNR and it is much lower than the FSD, which requires 62% of the multipliers required. In other words, a 38% complexity reduction can be achieved. Most power and energy savings can be gained during the "No Decoding" phase since no processing is required in this region. Furthermore, power and energy are saved during the utilization of V-BLAST/ZF algorithm i.e. where MI $> 7,100$, only 25% multiplier usage.

3.5.2 Hardware Performance

In order to comprehend the reason behind the complexity savings gained in Figure 3.6, consider four extreme scenarios of three transmission frames of $1,024$ data bits per frame size being transmitted using different detection algorithms. From this, it can be seen that if only ARQ is used such that depicted in scenario 3, the complexity would be equals to zero. The maximum complexity would be dominated by scenario 2, and using the results obtained in Figure 3.6, the complexity of the V-BLAST/ZF is approximately a quarter than that the FSD. If the FSD is set to be 100% and the V-BLAST/ZF is at 25%, scenario 4 would give a complexity of around 42%. From Figure 3.6, it is shown that the complexity of an FSD is M-times larger than that of the V-BLAST/ZF.

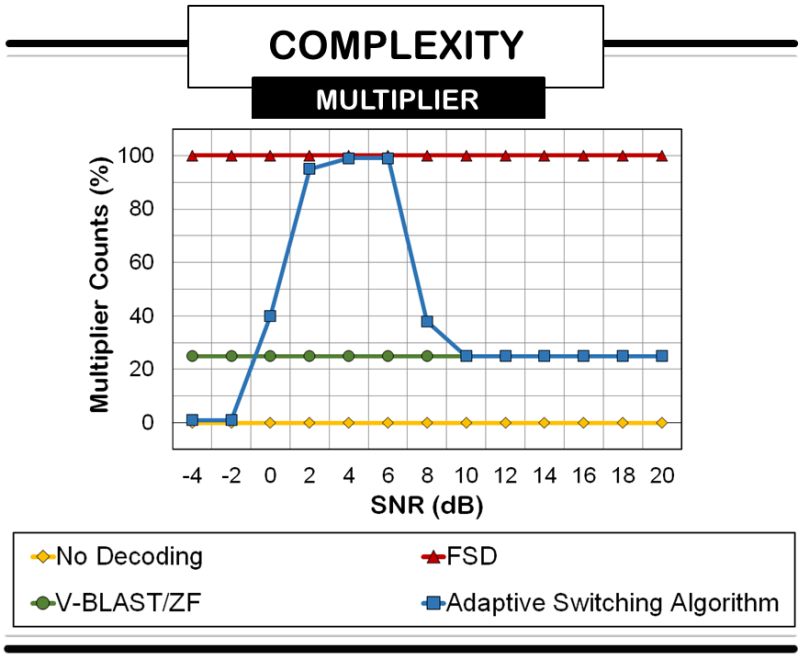

Figure 3.6: *Complexity measurements of multiplier counts between different MIMO detection schemes*

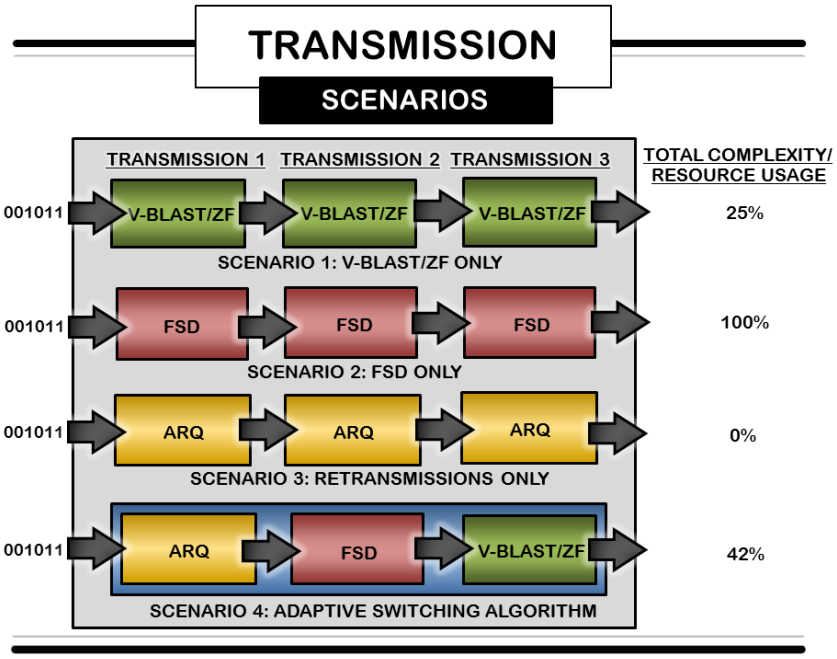

Figure 3.7: *Complexity count for simple mechanism of different detection algorithm*

Recall that the software results obtained previously was given as 62%. This can be concluded that both software and hardware standpoints show that approximately 50% savings can be gained when the Adaptive Switching Algorithm is utilized in comparison to the FSD baseline. To confirm this, the preliminary hardware performance is analysed using an exemplar FPGA design based on Xilinx® Virtex-5. The programmable hardware has a varying voltage range of 0.95 V to 1.05 V, and an operational frequency range of 60 megahertz (MHz) to 400 MHz [83]. In order to assess the efficacy of the Adaptive Switching Algorithm in saving power and energy consumption on hardware, both chosen iterative-MIMO detection algorithms, FSD and V-BLAST/ZF, are operated using the operating limits of the hardware capabilities spectrum. For the case of Xilinx® Virtex-5, it may operate at the lowest voltage of $V = 0.95$ V and the frequency of $f = 60$ MHz. For easier future reference, this work will dubbed this the "low power" mode. On the other end of the spectrum, the "high performance" mode can be fashioned using the high end spectrum limit of the design, which are to be at $V = 1.05$ V and $f = 400$ MHz. These modes are constructed in order to get an overview of the minimum and maximum capacity limitations of the hardware operation. The modes of operation information is determined using the Xilinx® integrated software environment (ISE) for the Xilinx® Virtex-5. The Xilinx® ISE comprises a combination of software/hardware setup performed in Matlab™ for modelling the transmitter and parts of the receiver, namely the channel reordering. The built-in Simulink® and Xilinx® System Generator cover the rest of the receiver parts, which are the components that make up the Adaptive Switching Algorithm detector. The power profile is estimated using a separate Xilinx® Power Estimator™ (XPE) tool.

Xilinx® Virtex-5: XC5VLX330TFF1738

Logic Resource Utilization	Available	Used V-BLAST/ZF	Utilization V-BLAST/ZF	Used FSD	Utilization FSD
Slice Registers	149,760	3,312	2%	13,683	9%
Flip Flops	37,440	892	2%	4,688	12%
4-Input LUTs	149,760	2,940	2%	12,161	8%
DSP48E	1,056	48	4%	132	12%
Memory (RAM)	516	12	2%	28	5%

Table 3.4: *Xilinx® Virtex-5 resource utilization for the V-BLAST/ZF and FSD detection algorithms*

The summary of the total number of the FPGA resources used are given in Table 3.4. The percentage of slices used can be seen as an indicator of the amount of control logic and intermediate buffers required in the Adaptive Switching Algorithm. It can be seen that the complexity of

the V-BLAST/ZF is approximately 25% less than FSD, and therefore, this result matches the software multiplier counts. This factor reflects hardware mapping and the resulting throughput. Though the work focuses on the power and energy savings, it is advisable to check that the other performance parameter, which is the average throughput, also behaves within the acceptable system requirement. In addition, by keeping the throughput in check, it helps to determine which modes of operations are better, either the "low power" or the "high performance" when considering the practicality of the Adaptive Switching Algorithm behaviour on hardware. The throughput, J, in Mbps is calculated according to:

$$J_{avg} = M \cdot \log_2 W \cdot f/\beta_{avg} \tag{3.9}$$

where β_{avg} is the average number of clock cycles required to detect a MIMO symbol.

For "low power" mode, where f = 60 MHz and the minimum number of cycles is β_{min} = 4, the maximum throughput is J_{min} = 240 Mbps while the "high performance" mode gives a throughput of J_{max} = $1,200$ Mbps. Increasing the clock frequency would result in a significant increase in the throughput, therefore, the ratio for $f = \beta_{avg}$ could be seen as an indicator of the level of optimization of the hardware design. The hardware setup parameters are included in Table 3.5.

Xilinx® Virtex-5: XC5VLX330TFF1738		
Operation Modes/ Parameters	Low Power	High Performance
Core Voltage	0.95 V	1.05 V
Clock Frequency	60 MHz	400 MHz
Max Throughput	240 Mbps	1,200 Mbps

Table 3.5: *Experiment parameters for different detection algorithms*

Figure 3.8 shows the total power usage given by the Xilinx® XPE™ tool for the Xilinx® Virtex-5. Major power components given by the software are four, the two dominating components being the dynamic and static power consumptions. The dynamic is mostly made of toggling of switching operations whereas the static is mainly caused by powering up the chip itself. More detailed information regarding the power components are given in Chapter 4. Similar to details reported in [84] [85] [86] [87], there are significant dynamic power savings in the circuit, portrayed in Figure 3.8, where "low power" mode uses 9%, of the overall power shown in Figure 3.8(a) in comparison to 29%, shown in Figure 3.8(b) when the circuit is run at full

"high performance" mode. However, these savings would be minimal in comparison due to the much larger static power, which dominates the overall chip power. The two other components being the transceiver and I/O power are negligible at this point in comparison to the dynamic and static components at approximately 0.1 W shown in Figure 3.8(c).

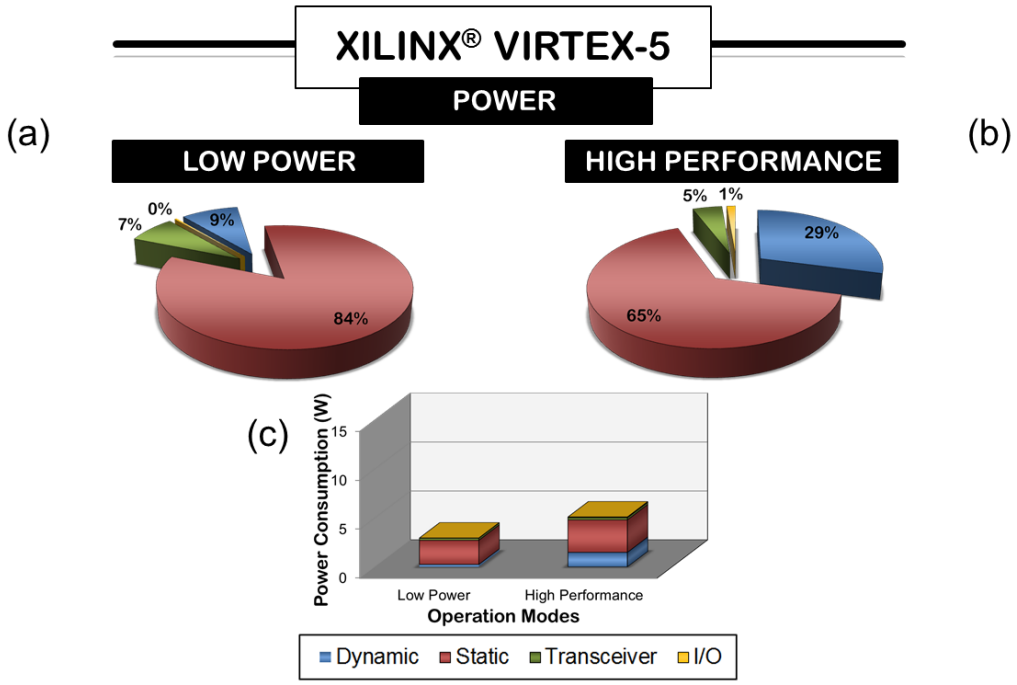

Figure 3.8: *Total power usage in Xilinx® Virtex-5 hardware design*

Figure 3.9 shows the "low power" results for (a) FSD and (c) V-BLAST/ZF as well as the "high performance" statistics, (b) and (d), for FSD and V-BLAST/ZF, respectively in terms of both the power and energy savings. It should be noted that some savings are gained when the Adaptive Switching Algorithm switches from the high complexity FSD to the simpler V-BLAST/ZF detection. The power saved during the swap is equivalent to 34% for "high performance" and 44% for "low power" mode. The energy savings when changing from "high performance" to "low power" and the energy savings for the swapping between the two detection algorithms can be calculated and are illustrated here. The total time computed that is obtained using the same system setup when operating at the lowest frequency of 60 MHz serves as a baseline, giving a completion time at approximately 20 μs. When operating at 400 MHz, the task completion time is approximately 8 times lower than when operating at the lower frequency. By finishing quickly, the hardware can be put into *sleep mode*, reducing the total energy, since the idle power

is negligible ≈ 0.08 mW [83]. More details on the hardware design can be found in Chapter 4.

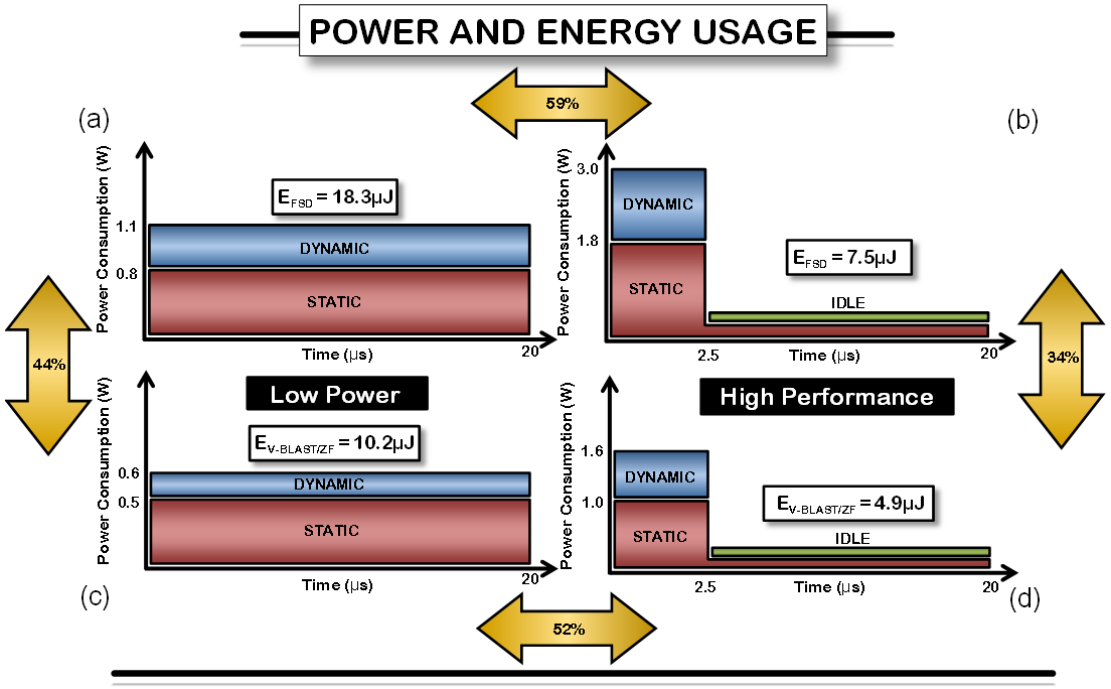

Figure 3.9: *MIMO detection FSD (a) and (b) in comparison with V-BLAST/ZF (c) and (d) for "low power" mode and "high performance" mode respectively*

By calculation, within the time budget, which is at the same total rate of completion, the energy required to complete one task is lower by 59% when the circuit operates quickly and switches into idle state in "high performance", taking 7.5 μ joules (J), than to run slowly and finish just-in-time, at lower frequency, "low power" mode, taking 18.3 μJ, when deploying FSD. Moreover, savings of 52% is gained for the V-BLAST/ZF algorithm, consuming 4.9 μJ and 10.2 μJ for "high performance" and "low power" modes respectively. These are the savings, which can be gained when putting the chip into *sleep mode* for more than 17 μs. The static power, resulting in 84% and 65% of the total power for "low power" and "high performance" mode respectively, shows that the static dominates the total consumption as shown in Figure 3.8. These findings coincide with the work reported in [88] however; stating that, as the manufacturing process gets smaller, the static component seems to dominate the overall chip power. Therefore, it can be concluded that running the circuit at a lower speed is not the answer to overall power savings in current and future programmable hardware technologies as the method of manufacturing of process nodes shrinks. Thus, the static component could no longer be neglected when designing a circuit, and it is now essential to take temperature as a

parameter in saving overall energy consumption, since the static component strongly depends on the heat generated by the circuit. Figure 3.8 and Figure 3.9 confirm the preliminary findings in Chapter 2, whereby the static power and energy should no longer be neglected when considering the power and/or energy consumption during hardware implementation. It can be seen that the static power is actually higher in comparison to the dynamic power, giving the more reason to include the component in the calculation to minimize the overall power and energy consumption.

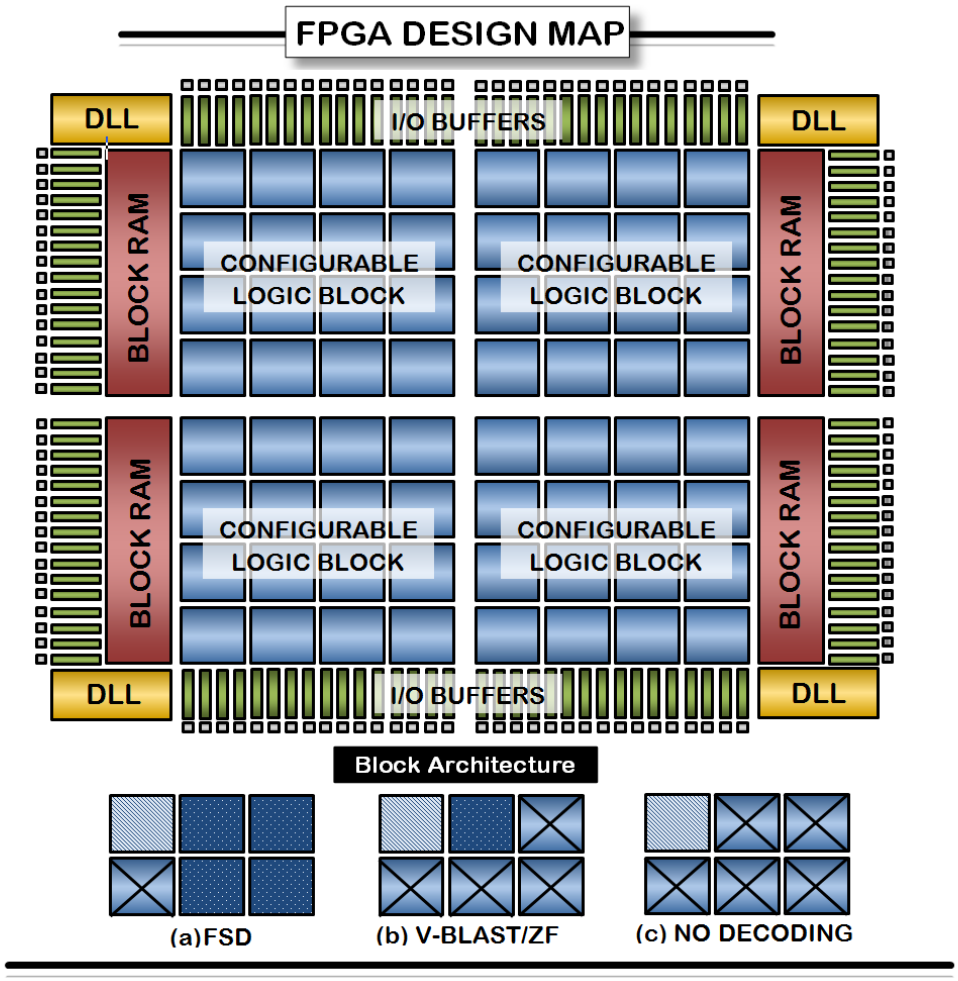

Figure 3.10: *Total resource allocation of Adaptive Switching Algorithm on a basic FPGA architecture*

In a nutshell, switching off parts of the FPGA chip would probably be the best method of power and energy savings. With this new information coming to light, the basic idea behind the im-

plementation of the Adaptive Switching Algorithm is illustrated in terms of a basic FPGA hardware given in Figure 3.10. It shows the overview of the algorithm flow within the chip. Only one detector is switched on at any given time according to the calculation from the threshold control block. The Adaptive Switching Algorithm is particularly useful for FPGA implementation since the hardware resources can be switched on and off as required. The configurable logic utilized for each detector is shown in (a) for FSD, (b) for V-BLAST/ZF and (c) when "No Decoding" is taking place. It can be seen that only certain parts of the overall chip hardware are turned on at any given time. Seeing that most power consumption is due to powering up the chip itself, which is the static power, the Adaptive Switching Algorithm takes advantage of this fact and therefore shuts down parts of the chip which are not in use. It is worth noting that since the workings of the FSD can be seen as V-BLAST/ZF detection simultaneously, one block for the latter detection algorithm can be re-used when designing FSD to compose the blocks. This hardware re-usability is a means of saving power if optimized. However, the work uses dedicated chip area space for each detection algorithm, where no hardware resources are shared amongst the common functionality between the two algorithms. Therefore, the power and energy savings outcome obtained in this chapter are not as promising in comparison to the potential gain that could be achieved when the optimization is realized. FSD resources shown in Figure 3.10(a) uses four configurable logic blocks for implementation while V-BLAST/ZF in Figure 3.10(b), would utilize one of the same blocks to perform the detection process. The threshold MI calculation would use one block and comprise negligible complexity of approximately 1% in the overall detector.

Shutting down parts of the chip, also known as *sleep modes*, are perhaps the key enablers in saving further energy in the design of hardware. More detailed analysis and results corroborate this in Chapter 4. By running the circuit at high frequency, the *sleep mode* can help prevent the circuit from running and powering up the entire logic gates all the time, consequently preventing the circuitry from overheating that leads to high static component consumption.

3.5.3 Rayleigh Fading Performance

For greater insight of the total power and energy savings that can be achieved in a realistic setting, Figure 3.11 considers the Adaptive Switching Algorithm in a Rayleigh fast fading channel. Rayleigh channel modelling may be used to replicate a real-life transmission environment, where the model varies over time, geographical position and radio frequencies. The preliminary

work uses this random process to mimic real-life wireless setup in order to confirm the robustness of the Adaptive Switching Algorithm in different environments. A more comprehensive study is evaluated in Chapter 5. The SNR range chosen is based on the operating SNR regions of the new wireless communication system LTE. In small cells, the transmit power is in the range of 23 dB to 46 dB, averaging at 26.5 dB [89]. The savings can be found by integrating the power, *P*, with respect to the probability density function, *F*, of the fading environment, ρ, as shown in Equation (3.10).

$$\int_{A}^{B} P(\rho)F(\rho)\,\mathrm{d}\rho \tag{3.10}$$

where *A* is the lower SNR value of -4 dB and *B* is the upper limit of the SNR, which is 40 dB in this case. Using a discrete approximation to this gives a representation of measure for the savings that can be as closely achieved as that in practice. The summary of the results for both AWGN and Rayleigh fading channel can be compared in Table 3.6, where the number of algorithm usage, the power and the energy for each detection method in both channel conditions are tabulated. It can be seen that "high performance" mode still uses less energy to decode the same data packet size in both channel setups, with slight power increase. It can be concluded that the proposed detector is best run at high frequency and be put into *sleep mode* as soon as possible to save power and energy.

	AWGN Fading		
Detection Algorithm	% of Complexity Usage	Low Power	High Performance
No Decoding	0%	0 W, 0.0 μJ	0 W, 0.0 μJ
FSD	100%	1.1 W, 18.3 μJ	2.7 W, 7.5 μJ
V-BLAST/ZF	25%	0.6 W, 10.2 μJ	1.6 W, 4.9 μJ
Adaptive Switching Algorithm	62%	0.8 W, 13.7 μJ	2.1 W, 6.2 μJ
	Rayleigh Fading		
Detection Algorithm	Complexity	Low Power	High Performance
No Decoding	0%	0 W, 0.0 μJ	0 W, 0.0 μJ
FSD	100%	1.3 W, 21.5 μJ	3.7 W, 10.8 μJ
V-BLAST/ZF	22%	0.7 W, 11.2 μJ	2.0 W, 5.9 μJ
Adaptive Switching Algorithm	74%	0.9 W, 16.0 μJ	2.7 W, 8.1 μJ

Table 3.6: *Comparison power and energy usage of different detection algorithms on different channel environment*

Taking the energy reading for "low power" mode for example, the Adaptive Switching Algorithm would use 13.7 μJ of energy to decode the $1,024$ bits data packet size in the AWGN

fading environment, and 16 μJ in a Rayleigh fading channel, with a slight increase of less than 15% of energy usage for the latter channel condition. Using the FSD and the Rayleigh fading distribution curves as baselines, the percentage of complexity, which determines the usage of the algorithm used during the span of SNR transmissions can be calculated. Moreover, since the behaviour of the Adaptive Switching Algorithm follows that of the Rayleigh fading channel for a 4×4 MIMO system, the proposed algorithm operates on 74% complexity usage, as shown in Figure 3.11 of the fading channel environment in comparison to only 62%, as shown in Figure 3.6, in AWGN fading channel. Power and energy savings can be achieved due to the *sleep mode* being implemented during appropriate times, for example, FSD is put on *sleep mode* at an SNR of 20 dB, with only V-BLAST/ZF being kept active. The results show that the Adaptive Switching Algorithm has the potential to save 26% of consumption in Rayleigh fading channel environment. Though this saving is lower than the ones obtained on the AWGN channel, it is significant nonetheless, which proves that the Adaptive Switching Algorithm has the potential to work under different channel setup and conditions.

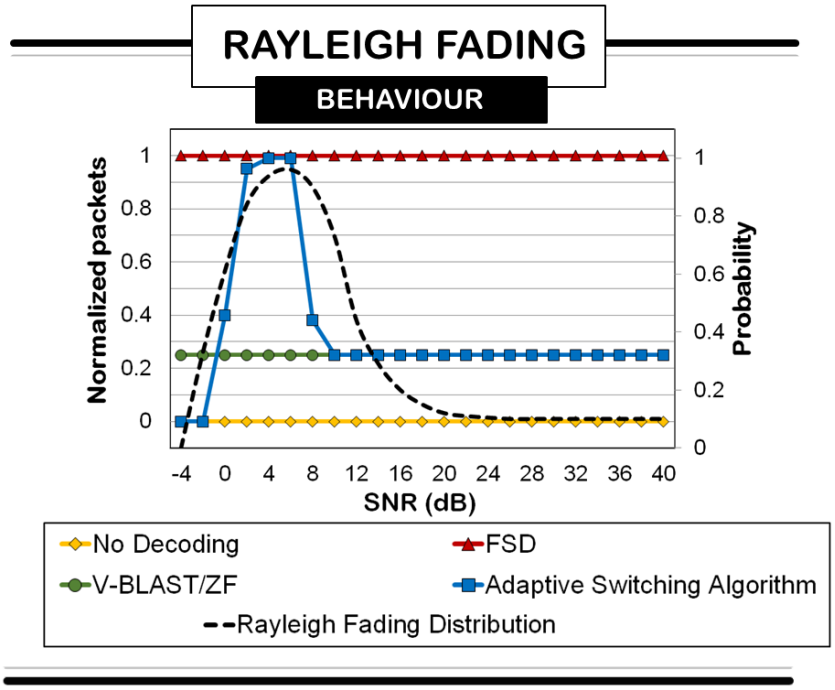

Figure 3.11: *Detection algorithm behaviours in a Rayleigh fading channel*

The energy saving results obtained can be optimized further by combining the common circuitry of the FSD and V-BLAST/ZF since they share some common functionality. By sharing

the circuitry resources between the two algorithms, additional energy savings can be gained. Detailed evaluation of the issues is the next major step of the project.

3.6 Chapter Summary

The Adaptive Switching Algorithm for an iterative-MIMO receiver is proposed in this chapter. It works by switching between low complexity "nulling and cancelling" detection algorithm of V-BLAST/ZF and the high close to ML performance of FSD. The switching occurs according to the MI calculated based on the current channel condition and noise level between the transmitter and receiver in real-time. The feasibility of the Adaptive Switching Algorithm has shown that up to 38% in AWGN fading channel based on the software standpoint across the SNR regions of -4 dB to 20 dB. Moreover, the switching of the FSD to V-BLAST/ZF in "high performance" and "low power" modes gives a saving range of 34% to 59% in resources consumption on both software and preliminary hardware design implementations respectively. Having 'intelligence' in the algorithm and the hardware design setup offers optimistic results in both performance and complexity for current and future iterative-MIMO systems. The adaptivity provided by the thresholds are controlled by the MI between the transmitters and receivers. They give significant information about the channel conditions as they offer comprehensive statistics regarding the MIMO setup. In addition, a preliminary study of having adaptivity in the hardware also shows that more power and energy can be saved if parts of the chip can be switched on and off accordingly. This is confirmed and can be improved further when incorporating *sleep modes* to reduce the static components in the hardware apparatus. The results of the above can be seen in detail in the Chapter 4. In addition, the proposed Adaptive Switching Algorithm is robust and works satisfactorily in a controlled Rayleigh fading channel setup that represents real-life deployment, where savings of 26% can be achieved. Results of detailed work and design implementation of the Adaptive Switching Algorithm in realistic environment settings can be found in Chapter 5.

Chapter 4
Design Trends of the Adaptive Switching Algorithm on the FPGA Hardware

4.1 Chapter Contribution

In this chapter, a comprehensive power performance analysis of the Adaptive Switching Algorithm for an iterative-MIMO system is carried out, with the primary goal of minimizing additional power and energy consumption within the overall receiver. This work builds upon the findings in the previous chapter by implementing the Adaptive Switching Algorithm onto the most recent FPGA hardware design map to achieve more of the power and energy savings on top of the proposed algorithm design. This savings incorporate both components of the power and energy, which is a combination of static and dynamic. Several power minimization techniques were tested during the implementation of the Adaptive Switching Algorithm to examine their potential benefits. In depth investigation has shown that power and energy usage can be further optimized when the proposed algorithm is deployed on the Xilinx® Virtex-5 and Virtex-7 due to the adaptivity of power minimization techniques implemented on the FPGA hardware design.

4.2 Related Work

The information theory for recent iterative-MIMO receiver systems has been thoroughly researched for various performance parameters such as the data throughput rate [90] [91], BER [92] [93] [94] and for power efficiency [95] [96] [97] [98]. By incorporating coding schemes into the structure of the layered space-time receiver, the systems have the ability to approach the theoretical capacities on a multiantenna channel [13]. The iterative-MIMO receiver iteratively performs channel decoding to recover the original data stream corresponding to each of the transmitted antennas from the received signal vectors and estimated channel information. One

of the biggest challenges in designing a testbed for iterative-MIMO systems capable of real-time wideband wireless communication processing is choosing the right hardware for implementation. Primarily, the major requirements for the receiver hardware need to be identified in the preliminary design stages. Firstly, it must possess enough processing power to implement a wide variety of complex algorithms, since the receiver usually needs to perform extensive computations when decoding especially with the usage of multiple transmit and receive antennas, which increase the complexity tremendously. Secondly, the hardware must have the ability to be re-programmable for rapid prototyping, and possess the flexibility for parallelization as well as switching parts of the cores on and off for power and energy saving techniques. Lastly, the hardware must have the ability for mapping and keeping track of the resource utilization measurements and power consumption calculations. This is the main focus of this chapter, which is to show the Adaptive Switching Algorithm behaviour in terms of computational efficiency and its suitability for real world use.

There are many hardware types available in the market today. Those most utilized for wireless communication devices are the DSP, the VLSI, the application specific integrated circuit (ASIC) and the FPGA. Due to the complexity of the two lattice decoding algorithms of the Adaptive Switching Algorithm, namely the FSD and V-BLAST/ZF, the high data dependency among the decoding procedures and the link between the detector and the decoder, the iterative-MIMO receivers are generally implemented on DSPs [99] [100] [101]. However, the speed of the DSP implementation is often limited, especially as the number of antennas increases because it does not support parallel computations [100]. To overcome this limitation, VLSI architectures of MIMO systems have been investigated recently. Several hardware implementations have been reported by prototyping either the V-BLAST/ZF algorithm, FSD algorithm, or their modified versions [7] [102] [103]. However, it is a challenging task to reduce the complexity of the VLSI implementation in order to achieve maximal performance in real-time [104]. This problem has been negated lately as the decoding rate was successfully increased by using ASIC implementation [14]. However, an ASIC implementation is generally defined for a fixed number of antennas and a certain signal constellation, and is optimized for low power, high frequency circuit design. The limitations of an ASIC implementation is that it may lack flexibility when the number of antennas or the signal constellation changes [105]. This brings us to FPGA devices, which are widely used in signal processing, communications, and network applications because of their reconfigurability and support of soft reconfigurable parallelization. The FPGA has at least three advantages over a DSP processor. The potential for parallelization

is perfect for FSD and for the implementation of power minimization techniques in general, since both require vector processing during implementation. Moreover, the processing capacity is scalable if the FPGA resource is available in comparison to VLSI and ASIC implementations. The disadvantage is that the development cycle of the FPGA design is usually longer than the DSP implementation, but once an efficient architecture is developed and the parallel implementation is explored, because of its intrinsic density advantage [106], the FPGA is able to significantly improve the processing speed. However, ASIC implementation still dominates the market. This is mainly due to the fact that ASIC designs are often faster than for FPGA, as the ASIC is designed for a specific application it can be optimized to a maximum. Moreover, an ASIC design often consumes less power than for an FPGA design, so it provides better power optimization. Due to the re-programmable nature of FPGAs, they are often used as ASIC prototypes. An ASIC hardware description language (HDL) code design is first loaded onto an FPGA and tested for accurate results. Once the design is error free then it is taken for further steps. However, an FPGA has advantages over an ASIC implementation, whereby an FPGA device is reconfigurable to accommodate system configuration changes even during run-time. In addition, it usually has a significantly reduced prototyping time compared to an ASIC (a few days vs. a few months).

The SoC concept has been adapted to FPGA lately by introducing one or more embedded processors into the FPGA design [107] [108], such as the PowerPC™ hard processor cores [109] and the MicroBlaze™ soft processors [110] on Xilinx® FPGAs as well as the Nios™ soft processors on Altera® FPGA devices [111]. The SoC architecture significantly improves the interoperability and reduces the design complexity of many complex computational algorithms. Consequently, the hardware/software co-design technique can be applied to partition the computational algorithm into customized hardware and embedded software. For instance, one or more embedded processors can be instantiated in an FPGA to execute processing tasks that are less time critical but highly sequential or considerably complicated for direct circuit implementation. Since the Adaptive Switching Algorithm comprises multiple detection algorithms and running these algorithms would potentially save power and energy, it makes it even more desirable to use this platform in the proposed work. Therefore, of all the options available, FPGA-based system architectures of the latest Xilinx® Virtex-5 [112] and Virtex-7 [113] for iterative-MIMO receiver system was chosen due to the FPGA providing the flexibility for varying the number of antennas and signal constellations and the flexibility in the algorithm design as well as the visibility of the resource utilization. Both chipsets include up to two embed-

ded IBM PowerPC™ cores targeted to the needs of SoC designers. Both V-BLAST/ZF and FSD decoding algorithms are implemented on an FPGA platform and are evaluated for BER performance and power consumption evaluation. Several power minimization techniques are implemented as well to further save power and energy consumption for the Adaptive Switching Algorithm. Even though in practice, FPGAs may not be the ideal platform for large scale wireless communication system deployment, they are nevertheless well suited for use in rapid prototyping and for research purposes. That being said, the design results on the FPGA can be easily transferable to other platforms or environments such as the DSP, VLSI or the commonly used ASIC.

4.3 System Model Description

In the previous chapter, the Adaptive Switching Algorithm [114] is demonstrated by two well-known detection algorithms, namely the FSD [16], and the V-BLAST/ZF [65] detection algorithms switching together efficiently, performing according to the BER performance of the system. The switching between algorithms is determined by thresholds pre-calculated from the MI between the transmitter and the receiver, according to the real-time channel conditions of each data transmission. The algorithm design has proven to achieve 38% reduction in computational complexity, and therefore this work investigates if more power and energy savings can be accomplished through hardware design as well.

In order to explore this, the experiment for this chapter uses a software/hardware setup performed in Matlab™ and its built-in Simulink® package as well as the Xilinx® System Generator for the FPGA. The transmission setup is kept as in the previous experiment, where it comprises $M = 4$ transmitters and $N = 4$ receivers, based on a BICM setup, which has a transmit frame size of $K_u = 1,024$ bits transmitting over a random independent AWGN fast fading propagation channel, $\mathbf{H}$, with independent elements, which is perfectly known at the receiver. The transmitted bits, K_u, are encoded using an iterative-turbo scheme at rate of $\varphi = 1/2$, which are then interleaved randomly to give, b coded bits, before mapping into a QAM constellation, $\mathcal{O}$, of size $W = 4$, forming a sequence of $K_s = K_e/\log_2 W$ symbols. The $K_s = 1,024$ symbols are divided equally using the spatial OFDM multiplexing between the transmitters for $100,000$ channel realizations. This part of the transmitter system is simulated purely using Matlab™.

The work focuses on the receiver, which is consequently divided into the software experimen-

tation and the hardware design implementation. For **software**, the Adaptive Switching Algorithm for the iterative-MIMO receiver is designed on Matlab™ and its built-in Simulink® modelling package. On the other hand, the **hardware** design implementation, the setup follows that, which is depicted in Figure 4.1. Similar to the software, the transmitter and parts of the receiver, which includes the **QR** decomposition of the channel matrix **H** and the channel ordering are simulated using Matlab™ . The circuitry for the proposed algorithm, which includes the detectors of FSD and V-BLAST/ZF algorithms, the Adaptive Switching Algorithm threshold control and the decoder are modelled on the Simulink® modelling and later forwarded to the Xilinx® System Generator, which are then mapped on to the latest Xilinx® Virtex-5 and Virtex-7 chip designs.

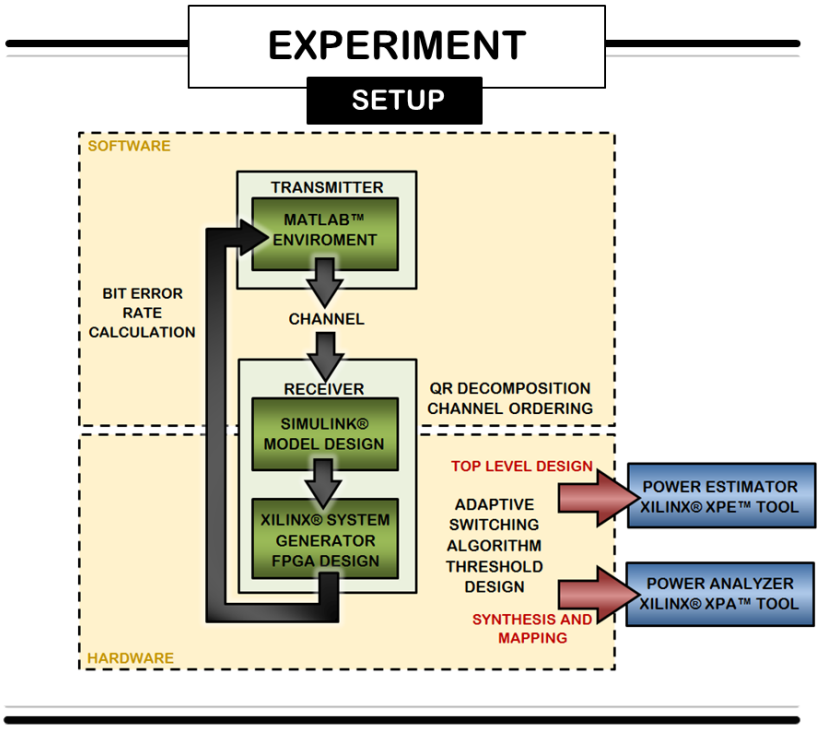

Figure 4.1: *Flowchart of the software/hardware experimental setup*

The power readings are initially estimated by the Xilinx® XPE™ tool based on the multiplier resource counter utilization during the software modelling portion. The power readings measured gives ballpark estimates for realistic hardware design implementation, which are later confirmed during the implementation using the Xilinx® System Generator using the Xilinx® Power Analyzer™ (XPA) tool after the model is synthesized and mapped onto the appropriate hardware of choice.

4.4 System Design Architecture

The operations of the Adaptive Switching Algorithm are realized using implementations on both software and hardware co-simulation. It should be noted that once the channel realizations, the **QR** decomposition and each corresponding channel ordering for specific detection algorithms are simulated on Matlab™ , the model of each detection method is demonstrated on Simulink® before being synthesized and mapped onto specific hardware using the Xilinx® System Generator. In order to understand how the Adaptive Switching Algorithm is implemented, consider the explanation in the next subsections, where each block of FPGA operations is described in detail.

4.4.1 V-BLAST/ZF

The first detection algorithms within the proposed algorithm, V-BLAST/ZF [65], is implemented on the FPGA chip as shown Figure 4.2. The FPGA part consists of three separate blocks, namely the "data estimation" block, where the ordered ZF channel sorts the signal according to the strongest signal with the highest SNR first as the received signals, $\mathbf{r}$, are augmented using the dot $(\cdot)$ operation with the channel matrix. The data is then quantized in the "data quantization" block, Q, to the nearest 4-QAM constellation to give $\hat{\mathbf{s}}$, which is then passed to the next block, "interference subtraction". This is where the quantized symbols are subtracted from the original data, $\mathbf{r}$, before repeating the whole process until $\mathbf{r}$ is fully nullified and all signals, $\hat{\mathbf{s}}$, are detected.

Similar to the previous chapter, the number of multiplier counts can be estimated for each block using the Xilinx® ISE software. For V-BLAST/ZF, the most complexity comes from the "data estimation" block since the process requires complex matrix multiplications, which takes almost 65% of the whole detection algorithm, followed by the "data quantization" of matching symbols on specific QAM constellation LUT at 26%. These results will provide an estimation for hardware design implementation.

4.4.2 FSD

The second more complex detection method, FSD, published in [16] can be viewed as running multiple V-BLAST/ZF detectors in parallel, each checking different transmit data combinations of possible modulation symbols. Figure 4.3 provides the breakdown of the algorithm.

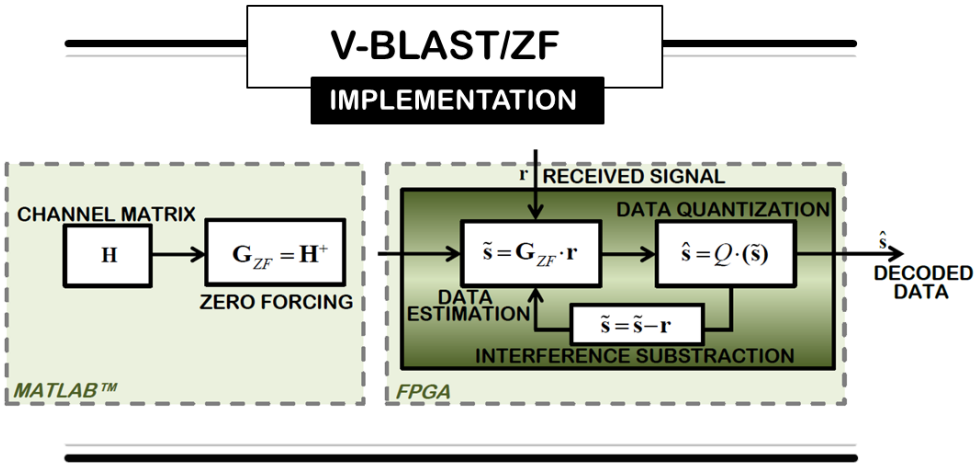

Figure 4.2: *Breakdown of V-BLAST/ZF FPGA implementation model*

The channel pseudoinverse, $\mathbf{G}$, is obtained by applying a $\mathbf{QR}$ decomposition to the channel matrix, which is implemented on Matlab$^{\text{TM}}$. There are two blocks of FPGA used for FSD implementation, namely the "metric calculation", which accumulate the ED, and the "path selection", which selects the minimum path to the lowest value for ED at the leaf node(s). Level i represents the i^{th} transmit antenna, therefore the partial ED, the AED, is calculated until the total ED is obtained for each path. The paths of selected ED at the leaf node(s) are then compared in order to find the minimum solution for received symbols, $\hat{\mathbf{s}}$. For the 4-QAM modulation scheme, after a full expansion on the first detected antennas, there are 4 paths to be selected, with 4 values of ED candidates for the minimum solution(s). The most complexity comes from the "metric calculation", where the dot $(\cdot)$ operation of channel matrix uses most of the resources, as well as the summation of the accumulated ED, taking almost 75% of the total FSD operation.

4.4.3 Adaptive Switching Algorithm

The main idea behind the Adaptive Switching Algorithm is shown in Figure 4.4. The "threshold control" block calculates the value of the accumulated MI and activates the appropriate detector, either the V-BLAST/ZF, when the channel condition is good i.e. when the MI is above T_2; or the FSD during bad channel conditions, i.e. when MI is above T_1 but below T_2. Once the threshold is determined, the appropriate FPGA blocks are switched on and off accordingly. If the threshold falls under T_1, an ARQ is required that consequently generates a new channel

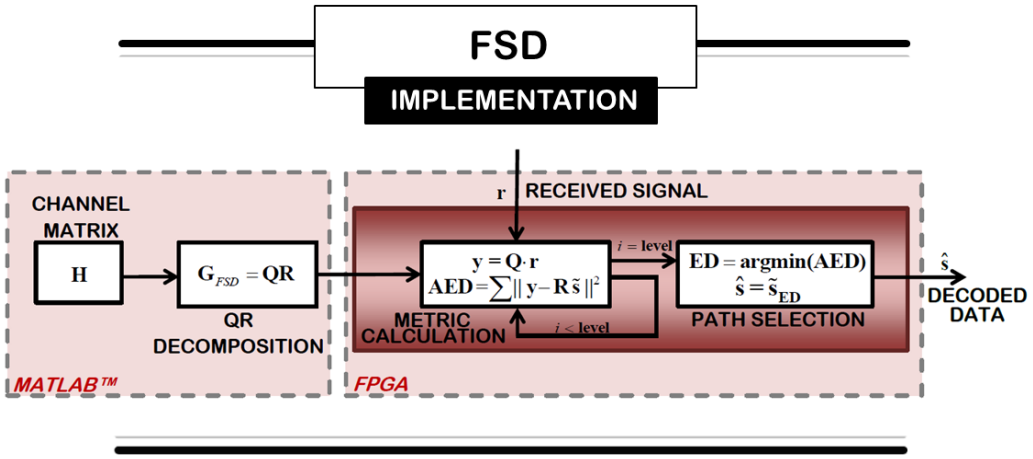

Figure 4.3: *Breakdown of FSD FPGA implementation model*

matrix, **H**, in the simulation process. Avoiding both detection algorithms in this way would also avoid the energy intensive iterative turbo decoding block. In this case, decoding is deemed superfluous in this transmission environment since symbol retrieval will experience close to 100% packet failure rate, which only wastes significant computational power. However, formally characterizing this decoding effect and ways of minimizing the corresponding power consumption are out of scope of this chapter and will be tackled and explained in Chapter 5.

4.5 Power and Energy Consumption

Power and energy consumption in recent communication devices, especially ones with battery powered sources are a major limiting factor in circuit designs. Fundamentally, most power and energy are consumed in dynamic, static, transceivers and I/O ports as specified by Equation (4.1); with dynamic and static dominating the process, as well as the transceiver and I/O powers being negligible at normally 1% of the total power usage [115]. For the purpose of the efficiency results, these two power components are omitted from the overall power consumption calculations. It should be noted that the energy is power used over specified timing constraints.

$$P_{total} = P_{dynamic} + P_{static} + P_{I/O} + P_{transceiver} \tag{4.1}$$

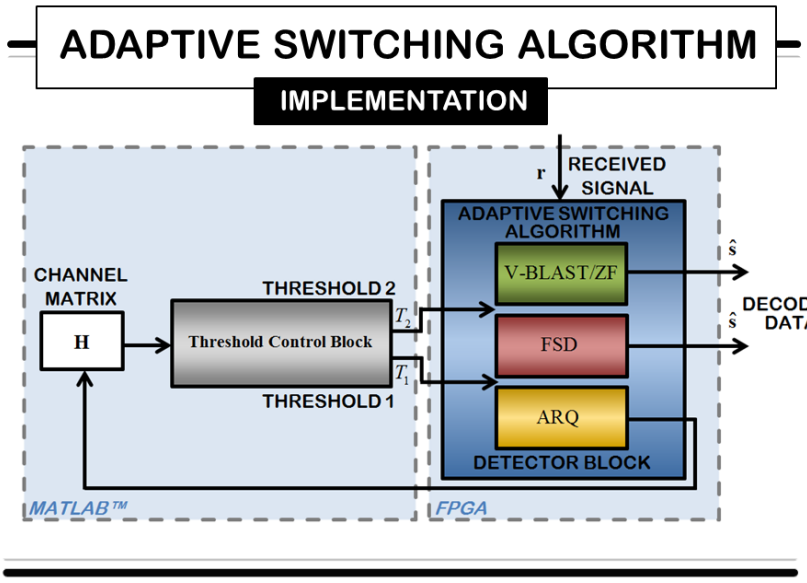

Figure 4.4: *Breakdown of Adaptive Switching Algorithm FPGA implementation model*

Most publications like [84], [85] and [86] have successfully reduced the dynamic power consumption, however, in newer chip technologies, the static power consumption is said to be high, [116], therefore, this work investigates ways to reduce both types, dynamic and static components in a circuit design, while ensuring the proposed algorithm performance behaves according to the design specifications set. This guarantees the Adaptive Switching Algorithm is properly optimized to meet power budget of the design. There are multiple ways to exploit power and energy savings in circuit designs and different type of power and energy have different approaches for executing these. For example, savings in dynamic component are achieved by scaling the voltage and frequency, while on the other hand, savings in static component depend on manipulating the parameters such as the manufacturing process, the temperature, and the core voltage used.

4.5.1 Dynamic Power and Energy

The dynamic power consumed within CMOS technology is due to toggling of transistors and is a function of clock frequency, which can be varied within some limit (before the circuit fails to function due to overheating), the value of the voltage, and the capacitance. Generally it can be said that with V, the power consumption is [87]:

$$P_{dynamic} \propto V^2 \tag{4.2}$$

The power consumption rises approximately with the V squared. Therefore, minimizing the V used is crucial where efficient implementation is concerned. Specifically and quantitatively, the dynamic component can be measured by the relation [37] given in Equation (4.3), where it depends on the number of toggling transistors, ξ, the circuit capacitance, C, the voltage swing, V, the toggling frequency, f and for the energy calculation, the time, τ, it takes to complete a set of operations as well.

$$P_{dynamic} = \xi C V^2 f \tag{4.3}$$

From this Equation (4.3), the power usage depends linearly on the clock frequency, f, therefore, both scaling in V and f were considered during efficient implementation designs.

4.5.2 Static Power and Energy

Static power is consumed due to transistor leakage and is highly dependent on the manufacturing process, the ambient temperature of the circuit, and the value of V. According to the study by [116], static components can dominate the overall power consumption within a circuit as the chip size shrinks. Therefore, these components can no longer be neglected when designing new algorithms into new chip technology. As the size of the recent hardware chipsets continue to scale down, the concerns for power and thus energy consumption should shift from the switching activity, which is the dynamic component, to the static, which is the component consumed when an idle element in a design has subthreshold current leakage, gate oxide current leakage, or reverse biased current leakage. Though it is hard to quantize the value for static consumption due to it being vastly different with every hardware chipset, a generalization of the relationship can be simplified as in Equation (4.4).

$$P_{static} \propto V^3 \tag{4.4}$$

All unused parts of the chip or idle logic in the hardware remains powered despite the lack of use, which contributes to high static power and energy consumption. While static power

was once considered secondary when looking at the total consumption, as the transistors logic shrink in size, the static power has increased exponentially while the dynamic power has stayed relatively stagnant due to lower operating V and decreased C associated with the switching nodes [117]. The smaller chipsets demand faster f and therefore require higher supply V to operate the same workload than their bigger predecessors. This trend can be seen in Figure 4.5. According to Moore's law, it is predicted that by the year 2020, the static power would have risen to almost 100 W for chipsets that are smaller than 14 nm. To overcome this rise of static power as well as other power components, several power minimization techniques have been devised and the descriptions can be seen in the next section.

4.5.3 Xilinx® Virtex-5 and Virtex-7

The Xilinx® Virtex-5 is considered due to its purpose suited for logic intensive and digital signal processing applications. This 65 nanometre (nm) design is fabricated in 1.0 V, triple-oxide process technology [117]. The power and efficiency of the FPGA chip correspond to the size of the manufacturing node, the previous chipset being 90 nm shown in Figure 4.5.

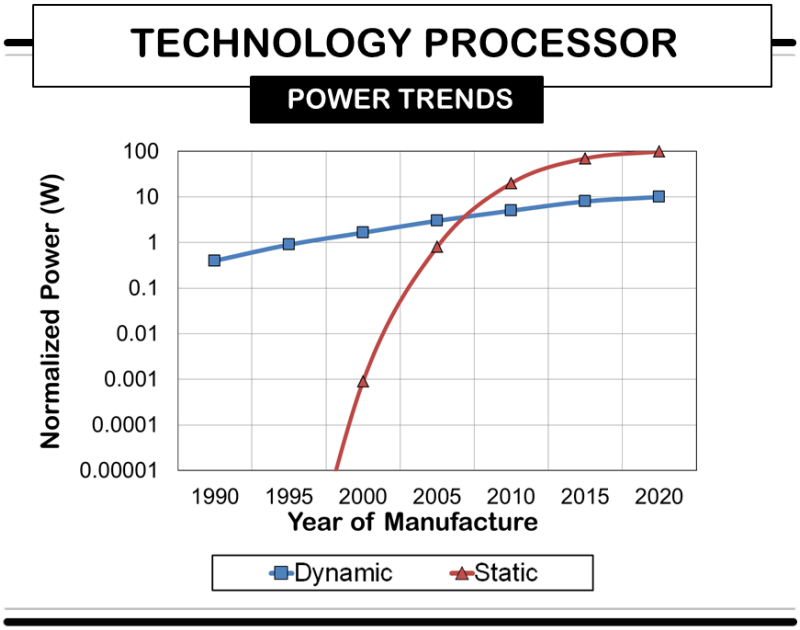

Figure 4.5: *Dynamic and static power consumption effects on process nodes [117]*

In contrast to the trend shown in Figure 4.5, Xilinx® Virtex-7, which is the company's most recent chipset based on an even smaller manufacturing node of 28 nm promises to deliver twice

the performance at 50% lower power, due to a newer lithography node in processing [113]. Moreover, reducing the feature size would reduce the energy required to switch transistors. Therefore, Xilinx® Virtex-7 is more energy efficient than Xilinx® Virtex-5. The work considers both chipsets when implementing the Adaptive Switching Algorithm to show the performance trends of the proposed algorithm and that it is suitable on all hardware; previous, current and on upcoming technology processes. In this experiment, the parameters of the operation modes under consideration are tabulated on Table 4.1 for both chipsets.

Power Component/	Xilinx® Virtex-5		Xilinx® Virtex-7	
Performance Mode	Low Power	High Performance	Low Power	High Performance
Voltage	0.95 V	1.05 V	0.97 V	1.03 V
Frequency	60 MHz	400 MHz	60 MHz	600 MHz
["Low Power" Mode : 0.97 V, 60 MHz; "High Performance" Mode : 1.03 V, 400 MHz]				

Table 4.1: *Operating parameters for the Xilinx® Virtex-5 and Virtex-7*

The Xilinx® Virtex-7 may operate at a much higher frequency of 600 MHz in comparison the its predecessor at 400 MHz, but at a lower voltage range of 0.97 V to 1.03 V as opposed to 0.95 V to 1.05 V, which suggests it may be suitable for faster processing at a low voltage utilization. For a fair comparison on both chipsets throughout the work, the Adaptive Switching Algorithm is implemented using the voltages and frequencies of 0.97 V and 60 MHz dubbed the "low power" and "high performance" mode having parameters of 1.03 V and 400 MHz respectively.

4.6 Power Minimization Techniques

Numerous power minimization techniques can be found throughout literature, however, the most common ones used in base stations and small cell devices are described as follows. These are the techniques applied during hardware design implementation of the proposed Adaptive Switching Algorithm.

4.6.1 DVFS

DVFS has shown significant power and energy savings when applied to circuit designs, evident in [87], [118] and [119]. Much like the Adaptive Switching Algorithm, *DVFS* has the ability to adjust its parameters to match the computational demand of the current workload. If the

workload requirement is high, *DVFS* will increase the V, to supply the circuit so that it can operate at a higher f in order to meet the desired data throughput within a particular time period. The opposite is also true; when the workload is minimal, the circuit could operate on a much lower f, which ultimately, according to Equation (4.3) and Equation (4.4) will decrease the overall power component as the task time lengthens. This adaptivity is appealing to the design of the Adaptive Switching Algorithm since now both software and hardware possess the same level of adaptivity and 'intelligence'. Combining both approaches yields significant overall power and energy savings. The basic principle detailed in [87] states that the power consumed by running the operation at a slower speed is less than to run it at full power and finishing early. Therefore, by budgeting the time for the workload to finish in time would save power and energy than to have the hardware run at maximum capacity and finishing early remaining switched on for the rest of the time. This study [87] considers only the dynamic power and discards other components of power consumption such as leakage, idle, overhead, static as well as the power needed to activate the chip. Figure 4.5 shows that the static component can no longer be ignored when considering the total power consumption of a circuit therefore, this work attempts to take all power components within the chip into consideration when applying the *DVFS* during the implementation of the Adaptive Switching Algorithm.

4.6.2 Sleep Mode

Sleep mode is when electronics operate on idle mode, with power so low, they are practically switched off for a certain period. When calculations do not possess the same task length and/or processing speed, they do not finish processing at the same time, meaning that for some proportion of the time, processor cores need not be on. Keeping the core activated would be wasteful, since the power to activate and keep the chip active is a significant contribution to its processing power, therefore, switching off the cores could be a means of saving power and energy. By running the application as fast as possible, longer *sleep modes* can be deployed. Instead of remaining active, the switched off cores will only consume 20 mW [120] of idle power for the remainder of the computational operation. The preliminary results found in the previous chapter state that the Adaptive Switching Algorithm is best run at high frequency of operation, and then put on *sleep mode*. Therefore, in addition to confirm this preliminary finding, this work attempts to discover, that this power minimization technique is best suited for the Adaptive Switching Algorithm detector when overall power consumption is considered.

4.6.3 Parallelization

Part of optimizing a system in current chip designs is to construct the algorithms in such a way that parallel operations are possible. *Parallelization* has been proven to save power and energy in recent hardware evident with the rise of multicore processors, multiple threads and pipelining approaches. The processors provide a trade-off between utilizing more chip space and increasing the throughput of a parallel algorithm. The cores split and share the computational load evenly amongst them. Therefore, each core performs only a fraction of the total computation depending on the number of cores available [87]. Furthermore, hardware architectures that can perform multiple tasks slowly in parallel should be more power efficient in comparison to running a single operation very fast on one processor core [36]. Due to the rise of multiple cores running simultaneously in chipsets, the work runs the Adaptive Switching Algorithm in parallel and the findings of its efficiency are positively discovered.

4.7 Results and Analysis

The total resource allocation of the Adaptive Switching Algorithm for the Xilinx® Virtex-5 and Virtex-7 is given in Table 4.2, and thus the number of the multiplier resources count is forwarded to the Xilinx® XPE™ tool to estimate the power measurements, which consequently give an overview of the effectiveness of each power minimization technique mentioned.

Utilization	Xilinx® Virtex-5 XC5VLX330TFF1738			Xilinx® Virtex-7 XC7VLX330TFFG1157		
Logic Resource	Available	Used	Utilization	Available	Used	Utilization
Slice Registers	149,760	16,995	11%	408,000	17,855	4%
Flip Flops	37,440	5,580	15%	51,000	5,692	11%
4-Input LUTs	149,760	15,101	10%	204,000	15,389	8%
DSP48E	1,056	180	17%	1,120	180	16%
Memory (RAM)	516	40	9%	1,500	38	3%

Table 4.2: *Resource utilization for Adaptive Switching Algorithm*

The available resources on the Xilinx® Virtex-7 are more than its predecessor, however, the number of registers, flip flops, look-up tables (LUT), DSP processors and the memory usage during the proposed algorithm implementation are generally similar on both chipsets, with a difference in resources of between 1% to 7%. The Adaptive Switching Algorithm iterative-MIMO detector is run in Matlab™ and its model counterpart on the Simulink® system. The

estimated power reading results for both Xilinx® Virtex-5 and Virtex-7 on the Xilinx® XPE™ tool are given in Table 4.3.

Power Component/	Xilinx® Virtex-5		Xilinx® Virtex-7	
Performance Mode	Low Power	High Performance	Low Power	High Performance
Dynamic	0.25 W	1.32 W	0.96 W	2.51 W
Static	2.16 W	2.87 W	0.23 W	0.73 W
Transceiver	0.02 W	0.03 W	0.00 W	0.00 W
I/O	0.01 W	0.06 W	0.07 W	0.08 W
Total	2.44 W	4.28 W	1.26 W	3.32 W

Table 4.3: *Power consumption of Adaptive Switching Algorithm on the Xilinx® Virtex-5 and Virtex-7*

The static power for the Xilinx® Virtex-5 is much higher than the Xilinx® Virtex-7. This finding contradicts the simple prediction given by the Figure 4.5 and agrees with the overview report for the Xilinx® Virtex-7 [120]. The new process nodes for the Xilinx® Virtex-7 do lower the overall static consumption by at least 89% and 74% for "low power" and "high performance" modes in comparison to its predecessor at 2.16 W and 2.87 W for "low power" and "high performance" respectively. This also coincides with [117] where the manufacturing node of the latter chipsets promises a much lower activation power. Due to this lower static power, the Xilinx® Virtex-7 operates in a slightly lower overall power when running the Adaptive Switching Algorithm; the power usage being lowered approximately 48% and 22% for the "low power" and "high performance" modes using 1.26 W and 3.32 W respectively. The dynamic power of the chipset however is slightly higher than the Xilinx® Virtex-5. This could be due to the slight increase in chip size as predicted in Figure 4.5, where the dynamic power increase steadily as the processing nodes decreases. The smaller chipsets need to process the same amount of data using limited chip area, therefore the hardware needs to perform more switching activities, which explains the rise in dynamic power. Moreover, the slight increase in resources needed for the Xilinx® Virtex-7 contributes to the higher dynamic power as well.

The amount of power used does not tell a lot about the Adaptive Switching Algorithm performance in terms of efficiency, therefore, a better parameter to consider would be in terms of the energy consumption. Simply reducing the power consumption in a processor may not decrease the energy demand if the task now takes longer to execute. Therefore, the energy information gives a better understanding of the efficiency of the system in transferring data packets of the same size within an allocated amount of time.

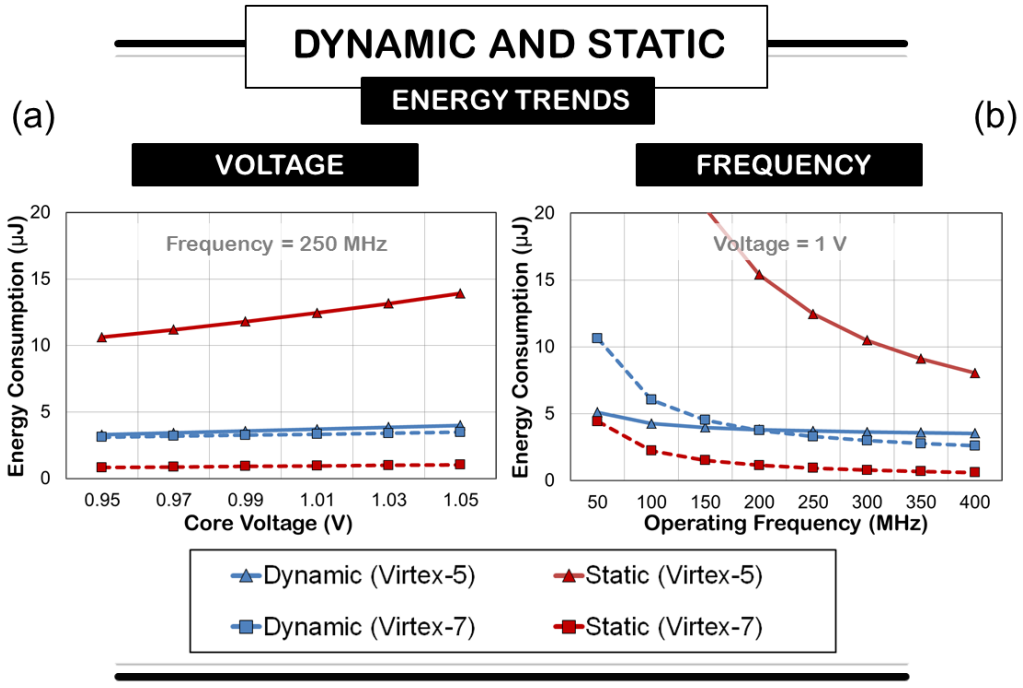

Figure 4.6: *Energy trends with (a) the voltage applied and (b) the variation of frequencies on the Xilinx® Virtex-5 and Virtex-7 respectively*

It should be noted that when considering the scaling for the voltages, the frequency is kept fixed at 250 MHz. On the other hand, when the frequency is scaled, the voltage is kept constant at 1 V. The energy trends are shown in Figure 4.6. By comparing the energy components in Figure 4.6(a), similar trends during scaling up the voltage can be observed, whereby, the voltage is directly proportional to the energy consumption. When comparing the frequency however, as shown in Figure 4.6(b), the energy consumption decreases with every frequency increment. First, the main difference to note here is that dynamic energy dominates in the Xilinx® Virtex-7 chipset, and therefore, the *DVFS* may be able to save power in the detector [87]. Secondly, the "high performance" and "low power" modes can be devised from taking the extreme ends of the scaling ranges. If running the proposed algorithm at the highest possible mode would save power, then *sleep mode* would be a good power minimization technique. Lastly, due to the small percentage of the area utilization, summarized in Table 4.2, ranging from 3% to 17% of total resource allocation, the proposed algorithm has the potential for *parallelization*, which is essentially having multiple copies of the detector within the chipset. The work looks at both the theoretical software simulations and the hardware design implementation standpoints to discover, which power minimization technique(s) mentioned is/are suitable for the Adap-

tive Switching Algorithm design implementation. The software simulations are based on both chipsets, while the hardware portion focuses on the Xilinx® Virtex-7.

4.7.1 DVFS

In addition, Figure 4.6 shows that due to the higher level of dynamic to static energy for the Xilinx® Virtex-5, where it is approximately six times larger, the overall energy of the circuit can be optimized using the *DVFS* as evaluated in [87]. However, when considering the total energy of the chip, including the static, the transceiver, the I/O and the leakage loss, this might no longer be the case.

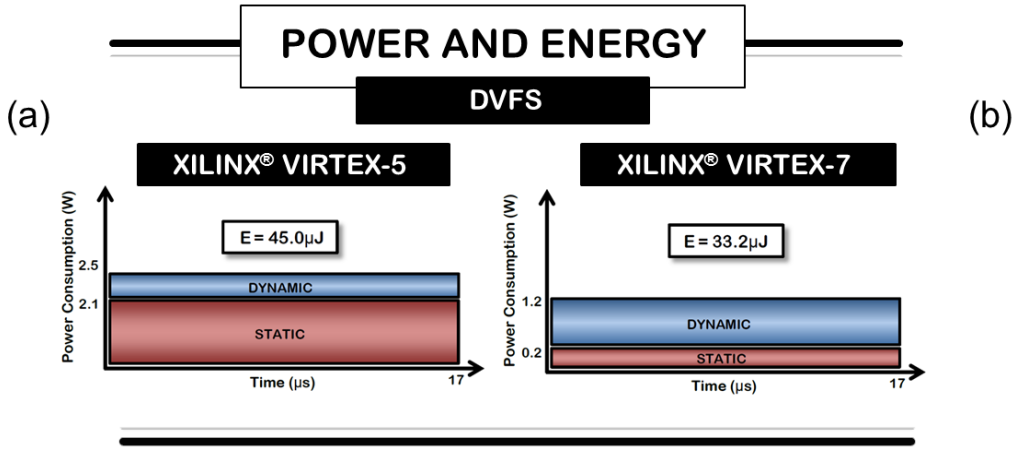

Figure 4.7: *Power and energy usage for (a) Xilinx® Virtex-5 and (b) Virtex-7 with DVFS applied*

Figure 4.6(b) confirms this as the static energy required to run the task for the Xilinx® Virtex-7 is much lower at higher speed with less than 0.6 μJ in comparison to 2.9 μJ, at 400 MHz and 100 MHz respectively, giving a difference of more than 79%. From this, even though the dynamic energy dominates, it can be said that running the algorithm as quickly as possible at the lowest possible voltage and switching it off would be better than running it at a slower speed.

The software results from Figure 4.7(a) and Figure 4.7(b) show the power and energy readings for both Xilinx® Virtex-5 and Virtex-7 respectively. By running the algorithm at the maximum allowed time of 17.1 μs for the same packet size, the difference of energy consumption between the two chipsets is approximately 12 μJ, which is almost 1.5 times less the energy usage for the Xilinx® Virtex-7.

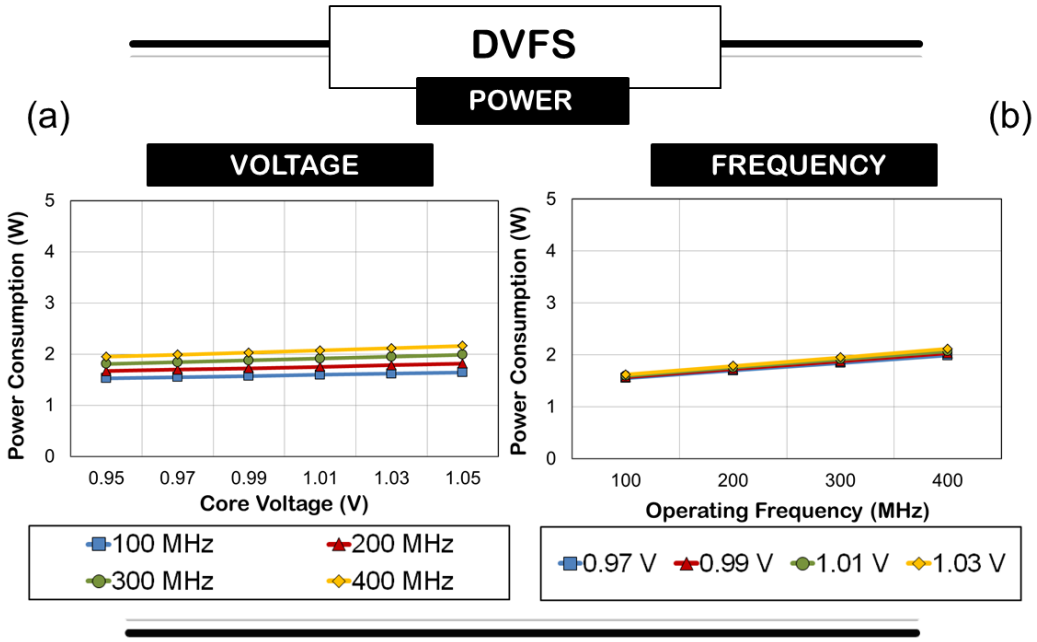

Figure 4.8: *Scaling effects where (a) is with voltage applied and (b) is with the variation of frequencies respectively for Xilinx® Virtex-7 platform*

For the hardware design based on the Xilinx® Virtex-7, the total power and energy consumption during the *DVFS* are given in Figure 4.8 and Figure 4.9. Similar to the previous experiments, the scaling of voltage is proportional to the power and energy consumption, which can be seen in Figure 4.8. Taking the 200 MHz as an example, at voltages of 0.97 V and 1.03 V, Figure 4.8(a) gives an increased power usage of 12%. Though minimal, it is still an undesired result. The scaling of frequency also shows minimal gain as shown in Figure 4.8(b). Taking 1.01 V as an example, at frequencies of 100 MHz and 400 MHz, the power gained is at 14%. The voltage scaling shown in Figure 4.9(a) illustrates that there is also minimal increment of energy, however, in frequency scaling, the reduction in energy is substantial. Looking at a voltage of 0.99 V, running the algorithm four times faster provides 69% energy savings.

Figure 4.9(b) shows that the total energy required to decode the same packet of data is less, due to the faster decoding process. It suggests that running the algorithm at full speed would be better than to finish just-in-time. This means that instead of having it running at "low power" and taking more than 20 μs to decode the data packet, the system would finish processing in less than 3 μs and be put into *sleep mode* for 78% of the time. This concludes that *DVFS* is not suitable as power minimization technique for the Adaptive Switching Algorithm on an

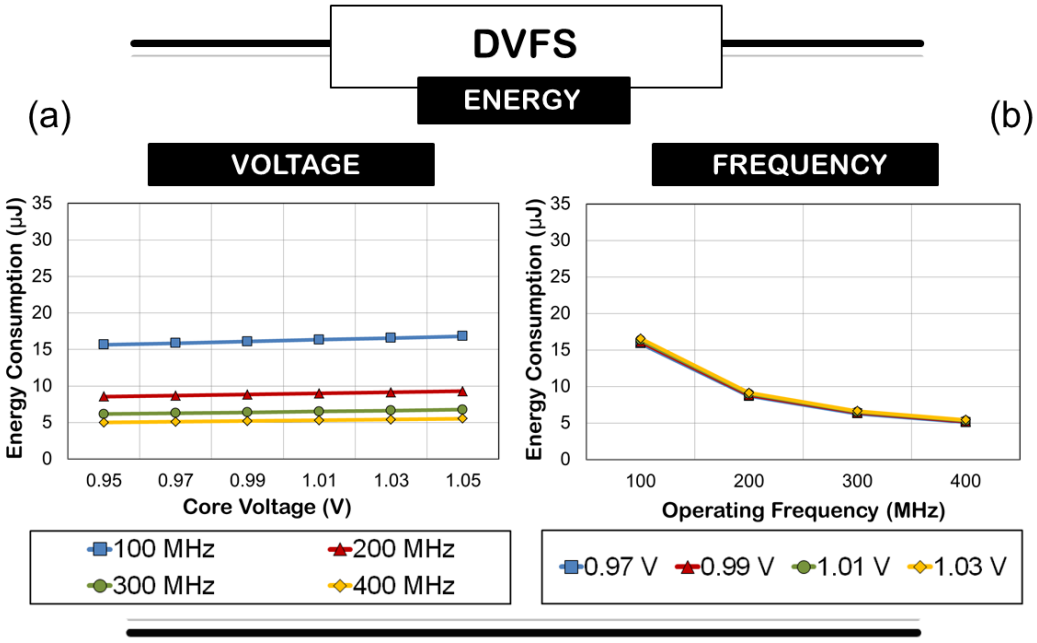

Figure 4.9: *Scaling effects where (a) is with voltage applied and (b) is with the variation of frequencies respectively for Xilinx® Virtex-7 platform*

architecture, since static power is a significant component of power consumption.

4.7.2 Sleep Mode

From the software standpoint, according to Figure 4.10, when *sleep mode* is utilized, i.e. running it at 400 MHz, the amount of energy required to process the same size data packet of $1,024$ bits is smaller than running it at a slower speed of 60 MHz. The power usage for both chipsets are similar with only about 5% difference between them. Running the algorithm as fast as possible finishing at 2.8 μs and shutting down 80% of the remaining time would give almost 70% and 64% energy savings for Xilinx® Virtex-5 and Virtex-7 respectively. The theoretical part of this work suggests that *sleep mode* is better suited for the Adaptive Switching Algorithm implementation.

The software results agree with the hardware design. In this situation, by taking the extreme cases of the *DVFS* into consideration, a "low power" and "high performance" modes can be articulated. Table 4.4 reviews the parameters of the Xilinx® Virtex-7 when running the Adaptive Switching Algorithm in two separate modes. The power usage analysed by the Xilinx® XPA™ tool is given as 1.5 W and 2.2 W for "low power" and "high performance" modes

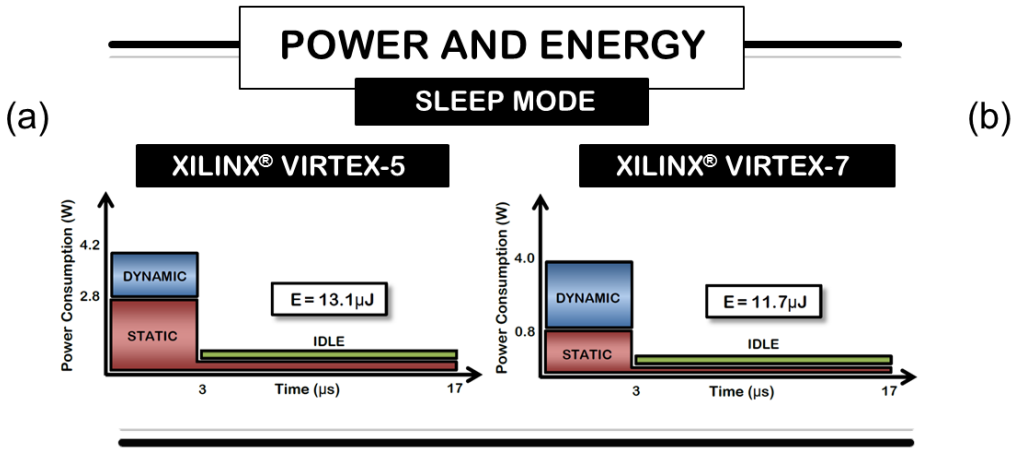

Figure 4.10: *Power and energy usage for (a) Xilinx® Virtex-5 and (b) Virtex-7 with sleep mode utilization*

respectively, contributing to 31% increase in power usage when "high performance" mode is selected. The total maximum energy savings is equivalent to 78%. Note that the maximum throughput is only achievable if the circuit is run 100% of the time and *sleep mode* is not active.

Xilinx® Virtex-7: XC7VLX330TFFG1157		
Operation Mode/ Parameters	"Low Power"	"High Performance"
Core Voltage	0.97 V	1.03 V
Operating Frequency	60 MHz	400 MHz
Max Throughput	240 Mbps	1200 Mbps
Power Consumption	-	31%
Total Energy Savings	-	78%

Table 4.4: *The "low power" and "high performance" parameters*

This section concludes that it takes less energy to transfer the same data packet in "high performance" mode. Therefore, by running the algorithm as fast as possible and then switching the cores off would save more energy, and thus, *sleep modes* are a good way to save power and energy in the Adaptive Switching Algorithm detector.

4.7.3 Parallelization

Starting with the software, the energy usage for parallel detector is compared in Figure 4.11 for both Xilinx® Virtex-5 and Virtex-7. The trend suggests that the more cores are used, the less energy is required to transmit the same amount of data. Savings of 75% for "low

power" mode can be gained when four cores are used instead of just one core whilst running the proposed algorithm on both hardware chipsets, at 45.1 μJ and 11.3 μJ as well as 33.3 μJ and 8.3 μJ respectively. Similarly, savings of more than 74% for "high performance" modes can be achieved, where a total energy consumption of 13.1 μJ and 3.3 μJ as well as 11.7 μJ and 2.9 μJ on Xilinx® Virtex-5 and Virtex-7 respectively. This coincides with the theory, which states as more cores are used, computations are divided evenly amongst the parallel cores [87]. *Parallelization* is an important way to achieve power savings for the algorithm as well.

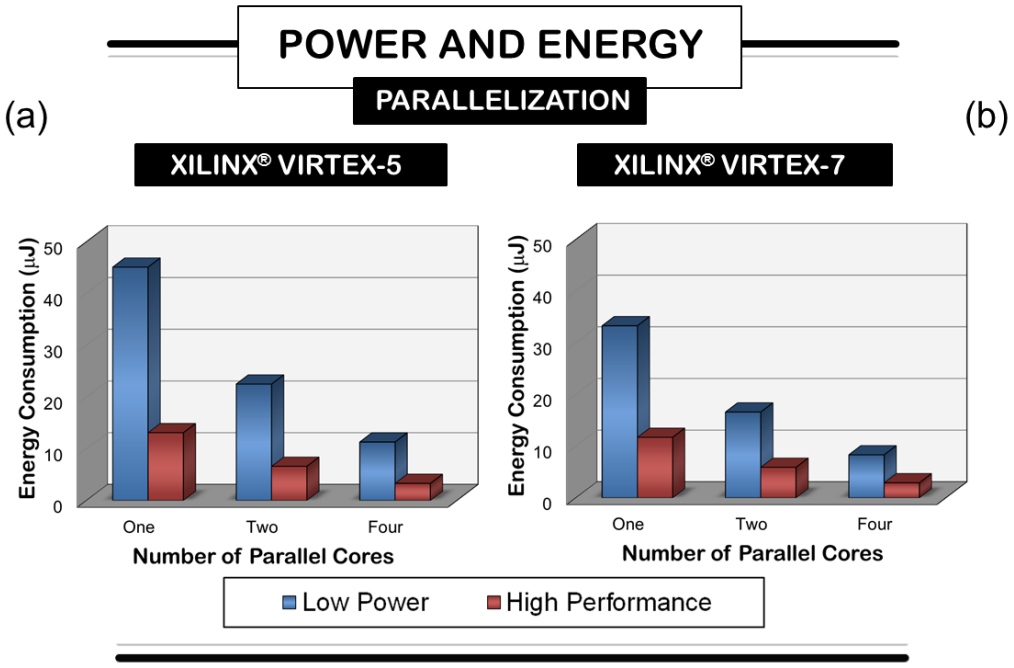

Figure 4.11: *Energy usage for (a) Xilinx® Virtex-5 and (b) Virtex-7 with parallel operations*

During the hardware design, it can be seen that the hardware utilization for the Adaptive Switching Algorithm is minimal. It uses a small percentage of the Xilinx® Virtex-7 as evident in Table 4.2. These promising results for parallel implementation are shown in Figure 4.12 and Figure 4.13.

Multiple copies of the Adaptive Switching Algorithm are utilized with one core being one copy of the algorithm being used. As predicted, the more cores used, the more power the chip needs as evident in Figure 4.12(a). This is due to the power needed to activate more area of utilization on the chip. However, the increase in power consumption is small in comparison to the energy savings gained, with only about 30% increment with every doubling in the number of cores used. Although the voltage scaling has little effect, the parallel setup does save significant

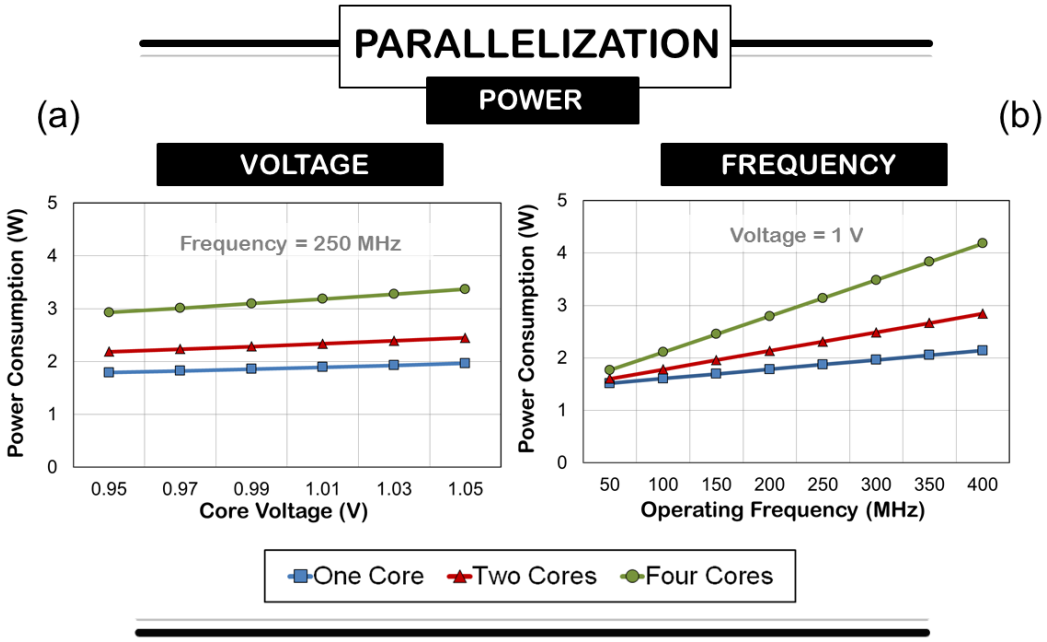

Figure 4.12: *Effects of scaling on power with parallel implementation where (a) with the voltage applied and (b) with the variation of frequencies respectively for Xilinx® Virtex-7 platform*

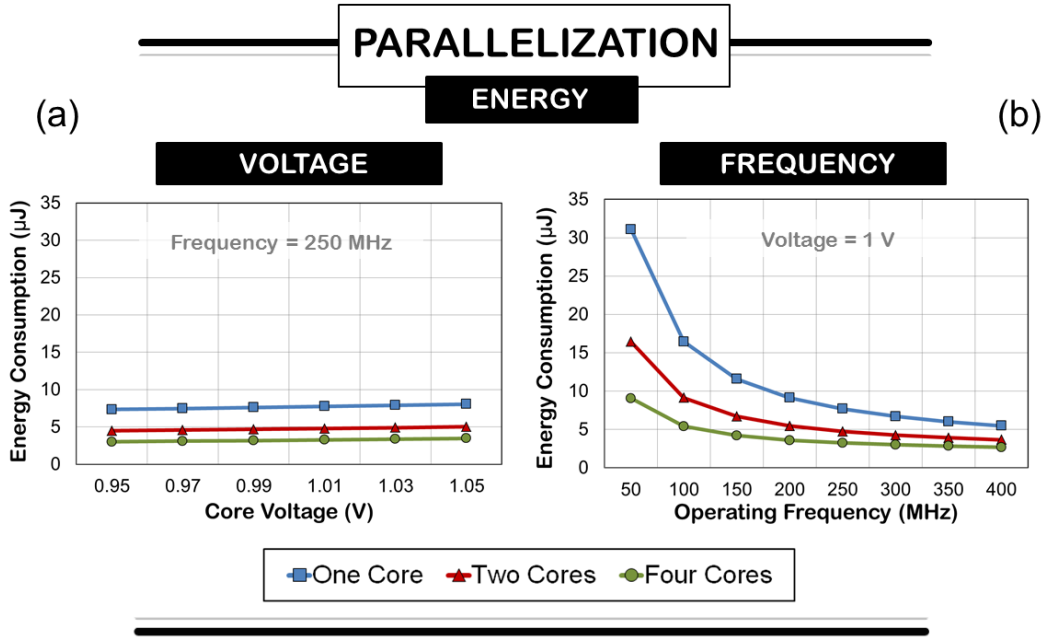

Figure 4.13: *Effects of scaling on energy with parallel implementation where (a) with the voltage applied and (b) with the variation of frequencies respectively for Xilinx® Virtex-7 platform*

overall energy savings seen in Figure 4.13(a).

Xilinx® Virtex-7: XC7VLX330TFFG1157						
Number of Cores/	One		Two		Four	
Parameters	Low	High	Low	High	Low	High
Power Consumption	1.5 W	2.2 W	1.6 W	2.8 W	1.7 W	4.3 W
Energy Consumption	25.6 μJ	5.6 μJ	18.8 μJ	3.6 μJ	7.3 μJ	2.8 μJ
Power Usage	-	31 %	6 %	46 %	12 %	65 %
Energy Savings	-	78 %	27 %	86 %	71 %	89 %

Table 4.5: *The "low power" and "high performance" parallel implementations*

The same can be said in frequency scaling, evident in Figure 4.12(b) and Figure 4.13(b), for power and energy respectively, where, taking frequency of 200 MHz as an example, running four cores instead of one give 52% energy savings with 29% increase in power. The energy saved whilst running on parallel cores in comparison to running a single thread is substantial, ranging from 3% to 83% across all frequencies, having particularly large differences at lower clock frequencies. These results show that *parallelization* is a good way to minimize the energy consumption.

4.7.4 Combination of Power Minimization Techniques

A combination of the techniques is performed to see if higher energy savings can be made. Table 4.5 summarizes the parameters of the power consumption and energy savings when the algorithm is run in parallel on "low power" and "high performance" modes, calculated against the "low power", single core baseline. The "low power" mode in fact uses more energy to process the same data packet in comparison to the "high performance" mode. Moreover, *parallelization* offers significant energy savings regardless of which mode is on, with a minimal increase in power to activate the extra cores. For example, by using four cores, in "low power" mode, the single core design uses 71% more energy than its multicore counterpart. This gain can be achieved with only 12% increase in power.

Figure 4.14 shows the energy used and time needed to decode the data packet received. These can be calculated from the power usage listed in Table 4.5. *Parallelization* causes the chip to use less energy on four cores, giving a total energy savings of 71% and 50% for considering separately the "low power" and "high performance" modes respectively. With these results, it can be concluded that the more cores deployed, the more efficient the Adaptive Switching

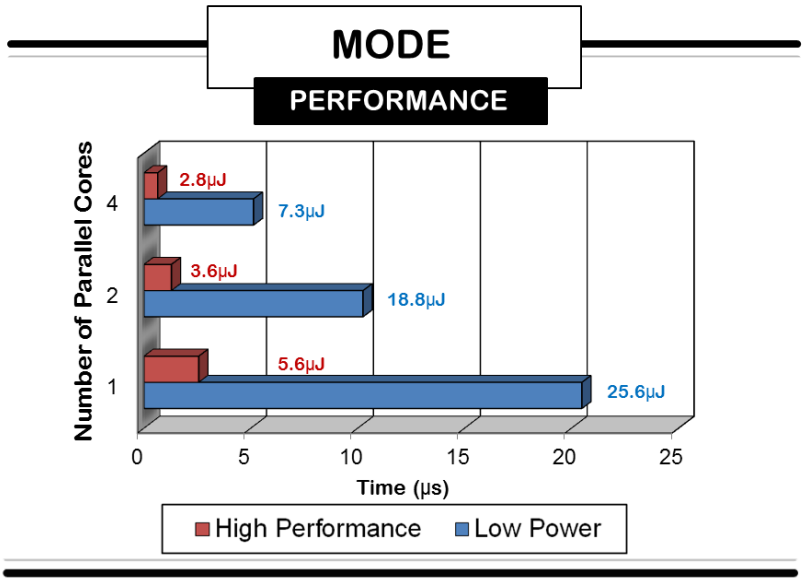

Figure 4.14: *Comparison of modes on parallel implementation*

Algorithm is. Instead of having one core running the algorithm for the entire 20.48 μs, using four cores running at for a quarter of the duration, and shutting them off for 75% of the time would minimize the energy consumption. Furthermore, the more cores being utilized, the more energy can be saved. When combining *DVFS* and *parallelization* techniques, i.e. comparing one core "low power" mode and "high performance" multicore mode, with values of 25.6 μJ and 2.8 μJ respectively, a total of more than 85% energy could be saved. This shows that combining the two power minimization techniques achieves significant overall energy savings.

4.8 Chapter Summary

The implementation of the Adaptive Switching Algorithm on both software and hardware are implemented to show the suitability of the algorithm for real world usage. During extensive study of several power minimization techniques of *DVFS*, *sleep mode* and *parallelization*, the best power minimization techniques for the Adaptive Switching Algorithm were established. It can be seen that for both "low power" and "high performance" modes at 25.6 μJ and 2.8 μJ respectively, a total of up to 89% of energy could be saved when four cores are running on "high performance" mode. The savings of power and energy can be seen from both standpoints, where they agree with the previous research that running the detector at a slower speed

would improve energy consumption. The results obtained for the Xilinx® Virtex-5 and Virtex-7 recommend the Adaptive Switching Algorithm to be run as fast as possible and then putting the chip into *sleep mode*. Additionally, the benefits of voltage scaling give inconclusive results due to the other power components dominating the chip, and due to its limited voltage scaling range, the chip gives negligible difference in energy consumption. However, larger savings may be possible on other ASIC or FPGA designs where a larger range of voltage values may be explored. On the other hand, the frequency scaling suggests that the algorithm works best when running at the highest frequency so that it can be put into *sleep mode* sooner, conserving energy. In addition, the more cores that are used, the faster the task completion and the faster it can be put into idle mode, thus achieving 75% energy savings. In the next chapter, the work continues to test the robustness of the proposed Adaptive Switching Algorithm by implementing it in realistic situations, and thus attempting to determine the total energy savings that can be gained in the overall iterative-MIMO receiver, which consists of both the MIMO detector and the iterative decoder.

Chapter 5
Practical Performance of the Adaptive Switching Algorithm in Spatially Correlated Channels

5.1 Chapter Contribution

In the previous chapters, the Adaptive Switching Algorithm has been proven to save significant complexity, power and energy consumption in both algorithmic design as well as during hardware design implementation in experimentally controlled AWGN fading channel conditions. In order to verify its effectiveness in realistic situations, the work in this chapter attempts to execute the proposed algorithm under spatially correlated channel conditions. The MI values used to design the thresholds were unaffected with the change in channel correlation proving that the MI is robust and provides a solid basis for the proposed algorithm design. It is found that the performance of the Adaptive Switching Algorithm detector in these channel conditions shows significant energy savings with slight BER degradation as the correlations between the transmitters and receivers increases. The chapter continues by forwarding the same MI calculations to be used as threshold information for the decoder. This provides the necessary information as a stopping criteria for the decoder that helps limit the number of iteration(s) required during each transmission. By combining both detector and decoder, the energy savings for the full Adaptive Switching Algorithm receiver shows significant savings gained in comparison to state-of-the-art, with lower hardware utilization complexity to boot.

5.2 Related Work

To meet the explosive growth in data rate currently caused by mobile devices such as smart phones and portable handheld multimedia devices, as well as data terminals such as wireless hotspots, femtocells and base stations, the technology of utilizing multiple antennas on both sides of the transmitter and receivers is imperative. Theoretical analysis has shown promising capacity growth by employing the MIMO scheme [4] [121], which helps in increasing

the spatial diversity and capacity of the system. However, the presence of spatial correlation between the multiple antennas reduces the capacity improvement [122]. Studies have evaluated the behaviour of detectors in such spatially correlated channel environments, for both low complexity linear MIMO detectors [123] [124] and high performance "tree search" detectors [125]. Generally, it is found that the BER degrades as the channel becomes more correlated. Studies are lacking however, on adaptive iterative-MIMO detection as well as for a full receiver setup that includes iterative decoding in such channel conditions. Moreover, to the best of the authors' knowledge, the energy analysis of adaptive algorithm implementations is not often considered in the literature. There are many adaptive detection algorithms proposed [44] [45] [46] [47], however, in addition to them using different switching criteria that do not fully exploit the available information regarding the MIMO channel setup [48] [58] [59] to provide the adaptivity, none of these papers considers the performance of such algorithms in spatially correlated channels or the energy saving potential for realistic implementations. Most publications focus on increasing throughput [44] [74] or the overall performance [46] [47] or provide generic energy saving results that are not specified to the latest state-of-the-art communication systems [48] [49] [50] [98]. A recently proposed Adaptive Switching Algorithm detector can achieve energy savings of about 38% in the algorithmic design [114], as shown in Chapter 3, and approximately 80% during hardware design implementation [126] found in Chapter 4, in experimentally controlled AWGN fading channel conditions. This chapter attempts to extend those findings by investigating the efficiency of the proposed algorithm usage in the detector in a realistic environment. In practice, the channels between different antennas are correlated and therefore the full multiantenna gains may not always be obtainable. Therefore, the work investigates the utilization of the Adaptive Switching Algorithm on simulated spatially correlated channels, whereby the information between the antennas, which is represented by the MI, may no longer be optimal.

In addition to the energy savings analysis of the detector in such channel conditions, this work explores the total iterative-MIMO receiver design, which includes the iterative turbo decoding that guarantees higher data rate support, and better performance in comparison to non-iterative systems [19]. The outstanding performance of the turbo decoder comes with a high price of computational complexity. To combat this, a number of early termination techniques or stopping criteria rules provided for the decoder iterations have been proposed in order to minimize the complexity of the decoder by reducing the number of iterations whilst maintaining the performance of the entire system. These criteria can be categorized into two groups, namely

soft-bit decisions and hard-bit decisions. Soft-bit decisions, which are considered in this chapter, such as cross-entropy (CE) [127] *a-priori* LLR measurement [128], and mean-estimation (ME) [129] updated threshold [130] are important methods. The most well-known CE stopping rule [129] works by using relative information between the two constituent decoders' soft output as the criteria. Decoding stops, or is considered converged, when the relative information is close to zero. Using the same concept as [129], different simplified versions are proposed in [130], where the LLRs are used instead to compute the relative soft information values. These concepts assist in lowering the complexity of the decoding process by minimizing the number of decoding iterations. Therefore, this trade-off of complexity and energy savings gained in both detector and iterative-decoding in spatially correlated channels are made and justified for realistic design implementations for the Adaptive Switching Algorithm receivers. In summary, this chapter investigates the applicability of a novel Adaptive Switching Algorithm detector under realistic channel conditions. By using the same MI values, the thresholds for the decoder can be constructed. With both detector and decoder thresholds obtained in the receiver, realistic performance for the proposed design is verified. These thresholds work according to the same calculated mutual information between the transmitters and receivers in real-time. The detector threshold determine whether the receiver would decode using a high performance detector, the low complexity detector or simply abandon further processing and reduce energy consumption by requesting a re-transmission. The decoder threshold works as a stopping criterion, where it determines the number of decoding iterations necessary for a transmission. This work provides the performance analysis for the proposed algorithm in realistic conditions by providing a detailed energy analysis of the algorithm for spatially correlated channel conditions. Analytical, simulation and implementation results show the practical behaviour of the proposed iterative-MIMO receiver in detecting and decoding.

5.3 Spatially Correlated MIMO Channels

In order to verify the effectiveness of the Adaptive Switching Algorithm in realistic conditions, spatially correlated MIMO channels are chosen as a reasonable model for providing simulated environments mimicking heavily built-up urban transmission settings on radio signals [131] [132]. Based on the flat fading standard MIMO model [6], with M transmitters and N receivers, where $M \leq N$, the channel setup considered in this portion of work utilizes the Kronecker model, where the correlation between the transmitters and receivers are assumed to

be independent and separable. This model is reasonable when there are multiple main signals scattering that occurs close to the transmitting and receiving antenna arrays. The results of this model has been validated by both outdoor and indoor measurements [133] [134]. In this case, with Rayleigh fading, the channel matrix can be factorized as in Equation (5.1).

$$\mathbf{H} = \mathbf{R}_{Rx}^{1/2}\mathbf{H}_w(\mathbf{R}_{Tx}^{1/2})^{\mathrm{T}} \tag{5.1}$$

The antenna correlation observed at the receiver is assumed the same on all transmitters, and similarly, the correlation for the transmitter is also the same on all receivers. The elements of $\mathbf{H}_w$ are i.i.d. as circular symmetric complex Gaussian with zero mean, μ, and unit variance, σ^2 with $\mathbf{vec}(\mathbf{H}) \sim \mathcal{CN}(\mathbf{0}, \mathbf{1})$ representing the MIMO uncorrelated channel. The $M \times M$ matrix $\mathbf{R}_{Tx}$ describes the fading correlation for the transmitter array while the $N \times N$ matrix $\mathbf{R}_{Rx}$ describes the received spatial correlation. The statistical behaviour of the channel matrix can be expressed as in Equation (5.2), where $\mathbf{vec}(\cdot)$ denotes the vec operator and $\otimes$ denotes the Kronecker product [133].

$$\mathbf{vec}(\mathbf{H}) \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_{Tx} \otimes \mathbf{R}_{Rx}) \tag{5.2}$$

The spatial correlation depends directly on the eigenvalue distribution of the correlation matrices, $\mathbf{R}_{Tx}$ and $\mathbf{R}_{Rx}$. Each eigenvector represents a spatial direction of the channel and the corresponding eigenvalue describes the average channel and signal gain in a specified direction. High spatial correlation indicated by a large eigenvalue spread in $\mathbf{R}_{Tx}$ and/or $\mathbf{R}_{Rx}$, mean(s) that some spatial directions are statistically stronger than others. Low spatial correlation on the other hand, is represented by a small eigenvalue spread in $\mathbf{R}_{Tx}$ and/or $\mathbf{R}_{Rx}$, meaning that almost the same signal power can be expected from all spatial directions. The higher the spatial correlation, the more impact it has on the performance of a given MIMO system [135]. The capacity of the channel is always degraded by the receiver side of spatial correlation as it decreases the number of (strong) spatial directions that the signal is received.

The correlation model considered in this paper can be calculated mathematically with respect to capacity, using generic definitions for the transmitter,

$$\mathbf{R}_{Tx} = \begin{pmatrix} 1 & \omega_{\mathrm{Tx}} & \dots & \omega_{Tx}^{N-1} \\ \omega_{Tx} & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \omega_{Tx}^{2} \\ \omega_{Tx}^{N-1} & \dots & \omega_{Tx}^{2} & 1 \end{pmatrix} \tag{5.3}$$

and receiver correlations.

$$\mathbf{R}_{Rx} = \begin{pmatrix} 1 & \omega_{Rx} & \dots & \omega_{Rx}^{N-1} \\ \omega_{Rx} & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \omega_{Rx}^{2} \\ \omega_{Rx}^{N-1} & \dots & \omega_{Rx}^{2} & 1 \end{pmatrix} \tag{5.4}$$

where ω_{Tx} and ω_{Rx} represents real-valued correlation coefficients. The correlation indexes considered are further simplified to give $\omega_{Tx} = \omega_{Rx} = \Omega$, yielding a single factor parameter. This means that the system considers the same correlation is present at both transmitter and receiver sides. The given model can range from the uncorrelated case i.e. $\Omega = 0$ to the fully correlated scenario of $\Omega = 1$.

Two points should be understood concerning the use of this model. First, while the channel model does represent close to realistic channel conditions, the results described above give pessimistic performance predictions for highly correlated fading scenarios where the model assumptions are no longer valid [136]. Secondly, though the correlation values between the transmitters and receivers are unlikely to be equal, this assumption is made to give an overall idea of the applicability of the Adaptive Switching Algorithm to spatially correlated channels.

5.4 System Model Description

The block diagram for the MIMO receiver under consideration is shown in Figure 5.1. Generally, a typical iterative-MIMO receiver comprises two blocks, a MIMO detector, and an iterative turbo decoder, where $\mathbf{r}$ is a series of received symbols from the transmitter, and $\hat{\mathbf{s}}$ is the estimated bit vectors for the transmitted data when the receiver processing is complete, similar to the experimental setup in the previous chapters. The experimental parameters are summarized in Table 5.1.

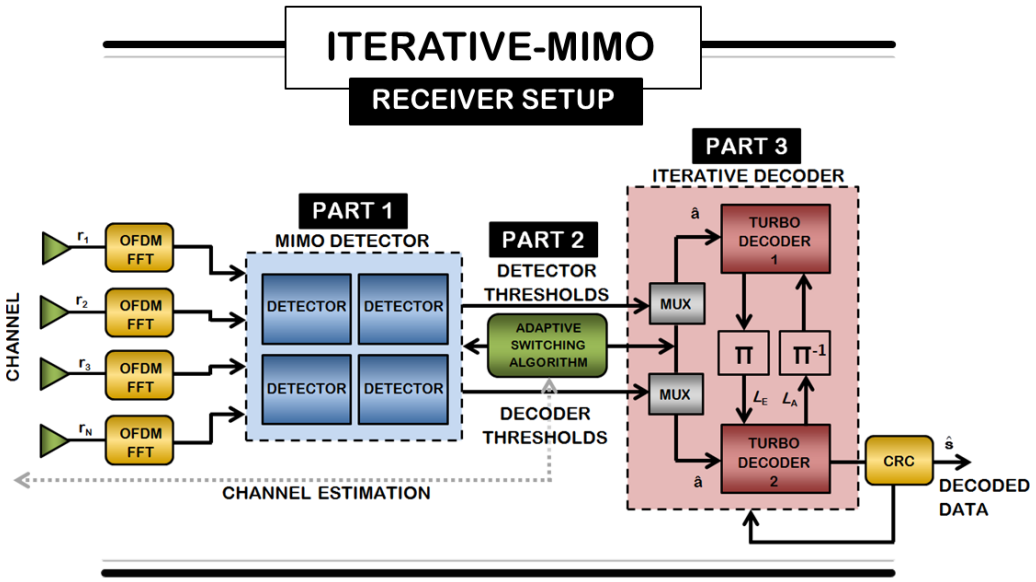

Figure 5.1: *Iterative-MIMO receiver system under consideration*

The detector first selects the appropriate detection algorithm, depending on the MI calculated between the transmitters and receivers in real-time, before being passed onto the iterative-turbo decoder with maximum number of iterations in spatially correlated fast fading channels. The experiments and results are divided into three parts, where the first focuses on the detector performance in spatially correlated channel conditions, i.e. **Part 1** in Figure 5.1. This then integrates itself onto the next part, which is **Part 2**, which will determine the suitability of the proposed algorithm as a link between the detector and the iterative decoder within the receiver system. Finally, once the link is successfully established, where the number of the required decoding iterations is determined, the addition of iterative-turbo decoder will complete the receiver design and thus the final analysis on energy and performance parameters is investigated and presented in **Part 3**. Lastly, the proposed receiver design is compared with the state-of-the-art LTE system and its deployment in realistic channel conditions is justified.

5.4.1 Iterative Turbo Decoding

As shown in Figure 5.1, after the detection process, the symbols are passed to the iterative decoder. Iterative decoding [137] is the key feature in turbo decoding. It is used right after the MIMO detector, where soft information extrinsic LLR (L_E) values are exchanged iteratively between the outer decoders with interleaving/deinterleaving operations in between until

Simulation	
MIMO System	4×4
Modulation	4-QAM
Code Rate	$1/2$
Turbo Iteration(s)	8
Packet Size	1,024 bits
Channel Realizations	100,000
SNR	-5 dB to 20 dB
Correlation Index (Ω)	0 - 1
Implementation	
Hardware	Xilinx®Virtex-7
Core Voltage	1 V
Clock Frequency	250 MHz

Table 5.1: *Experimental parameters*

a certain number of iterations have been executed to achieve the desired performance [20].

Generally, soft detection is used and it generates APP values in the form of LLR information, $L_E(b_k|\mathbf{r})$, about the interleaved bits, $\mathbf{b}$, for $1 \leq k \leq K_e$, while taking into account the channel observations $\mathbf{r}$ and the *a priori* LLR information, $L_A(b_k)$, coming from the outer decoder. For the FSD detector, assuming that the bits b_k are statistically independent due to the interleaving operation and making use of the Max-log approximation, $L_E(b_k|\mathbf{r})$ can be approximated by:

$$\begin{aligned} L_E(b_k|\mathbf{r}) \approx & \frac{1}{2} \max_{\mathbf{b} \in \mathcal{L} \cap \mathbb{B}_{k,+1}} \left(\frac{- \parallel \mathbf{r} - \mathbf{Hs} \parallel^2}{\sigma^2/2} + \mathbf{b}_{[k]}^{\mathrm{T}} \mathbf{L}_{A_{[}k]} \right) \\ & - \frac{1}{2} \max_{\mathbf{b} \in \mathcal{L} \cap \mathbb{B}_{k,-1}} \left(\frac{- \parallel \mathbf{r} - \mathbf{Hs} \parallel^2}{\sigma^2/2} + \mathbf{b}_{[k]}^{\mathrm{T}} \mathbf{L}_{A_{[}k]} \right) \end{aligned} \tag{5.5}$$

for $1 \leq k \leq K_e$, where, without loss of generality, $K_e = M \cdot \log_2 W$ has been used to simplify the index notation. In Equation (5.5), $\mathbf{b} = (b_1, b_2, b_3, \ldots, b_{K_e})^{\mathrm{T}}$, $\mathbf{b}_{[k]}$ denotes the subvector of $\mathbf{b}$ omitting b_k, $\mathbf{L}_A = [L_A(b_1), L_A(b_2), \ldots, L_A(b_{K_e})]^{\mathrm{T}}$, $\mathbf{L}_{A[k]}$ denotes the subvector of $\mathbf{L}_A$ omitting $L_A(b_k)$, $\mathbb{B}_{k,+1}$ and $\mathbb{B}_{k,-1}$ represent the sets of 2^{K_e-1} bit vectors $\mathbf{b}$ having $b_k = +1$ (logical '1') and $b_k = -1$ (logical '0') respectively, $\mathcal{L} \cap \mathbb{B}_{k,+1}$ and $\mathcal{L} \cap \mathbb{B}_{k,-1}$ denote the subgroups of vectors of $\mathcal{L}$ that have $b_k = +1$ and $b_k = -1$ respectively. The list of candidates $\mathcal{L} \subset \mathcal{O}^M$ is detector specific and subject to the overall performance and complexity of the iterative-MIMO receiver, since $\parallel \mathbf{r} - \mathbf{Hs} \parallel^2$ needs to be computed for all $\mathbf{s} \in \mathcal{L}$. It should

be noted that for V-BLAST/ZF detection, the LLR information can be simplified further by performing symbol by symbol likelihood calculations. In this model, $M \times 1$ coded bits are processed at one time and the LLR is defined as in Equation (5.6).

$$L(i,b) \approx \frac{1}{\sigma^2}\left(\min_{\mathbf{s}\in\mathcal{Z}_{i,b}^{(-1)}} \parallel \mathbf{r} - \mathbf{Hs} \parallel^2 - \min_{\mathbf{s}\in\mathcal{Z}_{i,b}^{(+1)}} \parallel \mathbf{r} - \mathbf{Hs} \parallel^2 \right) \tag{5.6}$$

under the assumption of equally distributed transmit symbols $\mathbf{s}$. The sets $\mathcal{Z}_{i,b}^{(+1)}$ and $\mathcal{Z}_{i,b}^{(-1)}$ are subsets of $\mathcal{O}$, where the b^{th} bit of the i^{th} stream is equal to +1 and 1, respectively.

Due to the iterative nature of decoding, the BER improves significantly at the output of the decoder as the iteration progresses. This improvement depends on the SNR, where it is dependent on the MIMO channel characteristics, and the MI between the transmitter and the receiver as well. Since the design for the detector considers the MI to provide the adaptivity, this work forwards the same MI to the iterative decoder, in order to gain the positive energy savings by stopping the system from dissipating useless energy in the decoding process by limiting the number of decoding iterations. When the next iteration of the decoder no longer provides significant or no improvement to the BER, early termination rules or stopping criteria are to be implemented. The criteria should find a balance and play a crucial part in terminating the decoding process without impacting the overall performance of the system. In some system setups, such as the state-of-the-art LTE systems [138], these iterative decoders are paired with the cyclic redundancy checks (CRC) and/or the valid code word checks (VCW) to ensure the system overall performance. A practical turbo decoder implementation typically sets a limit on the maximum number of iterations used [138]. Turbo decoding performance based on simple CRC assisted early stopping has been evaluated through simulations in [139] [140]. It is generally found that the average number of decoding iterations can be reduced substantially from the maximum while maintaining the same BER performance.

A CRC is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. Blocks of data entering these systems get a short check value attached, based on the remainder of a polynomial division of their contents; on retrieval, the calculation is repeated, and corrective action can be taken against presumed data corruption if the check values do not match. CRC uses redundancy where it expands the message without adding information and the algorithm is based on cyclic codes. CRCs are popular because they are simple to implement in binary hardware, easy to analyse mathematically, and particularly

good at detecting common errors caused by noise in transmission channels. In LTE systems, the CRC is implemented at every iteration. While this helps maintain the sustainability of the performance, it adds complexity to the system. Therefore, the work shows that the CRC can be omitted at every iteration, by replacing it with the threshold of the Adaptive Switching Algorithm, only to perform the checking at the end of the final iteration.

Typically, most stopping criteria work by setting a number of required decoding iterations according to certain rules, which can be generalized in Figure 5.2. The trend is that the number of the decoding iterations decreases as the channel condition improves, or at high SNR levels, whilst maintaining the desired BER performance. In theory, the number of decoding iterations may approach infinity as shown in Figure 5.2(a), however, due to delay limits in the receiver, all systems have set a maximum number of iterations as can be seen in Figure 5.2(b). At low SNRs, this number of iterations will not yield correct decoding. This failure point or error boundary is usually predicted by the usage of an extrinsic information transfer chart (EXIT) charts [141] [142]. However, EXIT charts are difficult to implement and uses a lot of hardware resources due to having a large LUT. In addition, EXIT charts are very specific to the design of the interleavers, which prevents the analysis of the asymptotically attainable performance. Furthermore, the task becomes time consuming, since the length of the interleavers is usually set as high as possible in order to reduce the correlation among the interleaved *a priori* and extrinsic LLRs [143]. These disadvantages can be negated by knowing in advance the number of minimum decoding iterations for the system by calculating the corresponding MI and using it as a basis of the threshold design. The basic principle of the proposed decoder that incorporates the Adaptive Switching Algorithm works by using the forwarded MI values from the detector. This MI values will determine the number of iteration(s) required depending on the current channel conditions of the transmissions. Moreover, the Adaptive Switching Algorithm decoder proposes that during transmissions where the channel conditions will yield close to 100% decoding failure, it would cease the process and requests for an automatic repeat request (ARQ) instead, with zero iterations used in the turbo decoding, resulting in significant energy savings. This design choice is shown in Fig. 5.2(c). The results for the MI threshold are obtained by numerical analysis and are presented in the next section.

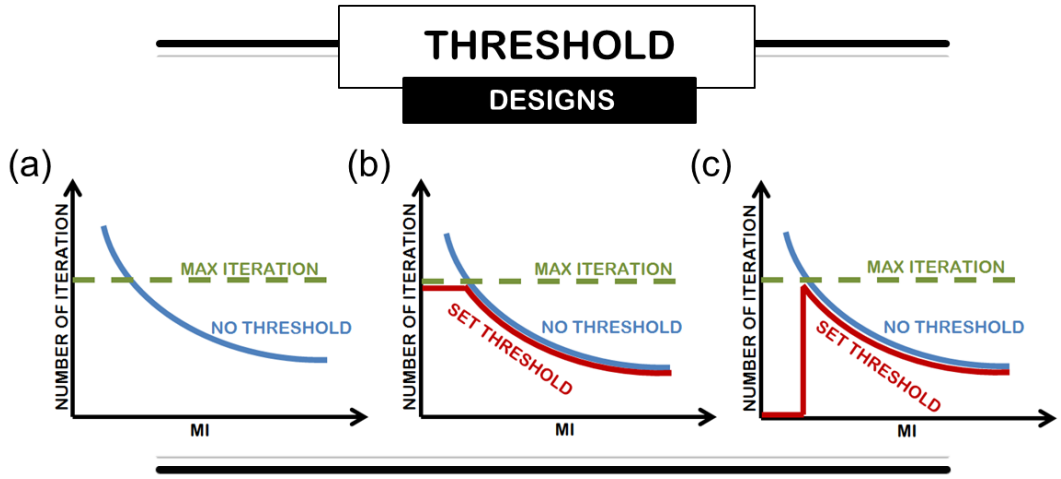

Figure 5.2: *General trends for thresholds used in different stopping criteria where (a) when no thresholds are used, (b) when a maximum threshold is used and (c) when both minimum and maximum thresholds are used*

5.5 Results and Analysis

The results are presented in sections according to the setup detailed in Figure 5.1, where each part is numerically labelled, and the energy performance analysis are based on the Xilinx® Virtex-7 chipset running at a core voltage of $V = 1$ V and an operating frequency of $f = 250$ MHz.

5.5.1 Part 1: The Detector in Spatially Correlated Channels

As shown in Figure 5.1, the first part of the work, labelled **Part 1**, involves running separate detection algorithms that make up the Adaptive Switching Algorithm on different correlated channel factors. In order to investigate the impact the Adaptive Switching Algorithm has on the channel correlations indexes, the channel correlations of $\mathbf{H}$ in Equation (5.1) are set to be $\mathbf{R}_{Tx} = \mathbf{R}_{Rx} = \Omega$. The total resource allocation provided by the Xilinx® ISE for both detection algorithms are given in Table 5.2. The V-BLAST/ZF uses less resources, about a quarter of that required the more complex FSD.

The number of multiplier counts can be estimated by breaking down the resource counter for each block using the Xilinx® ISE software. For V-BLAST/ZF, the most complexity comes from the estimating the symbols since the process requires complex matrix multiplications, which takes almost 65% of the whole detection algorithm, followed by the matching of symbols

Xilinx® Virtex-7: XC7VLX330TFFG1157		
Logic Resource	Utilization	
Utilization	V-BLAST/ZF	FSD
Slice Registers	3,228	14,628
Flip Flops	948	4,744
4-Input LUTs	3,080	12,309
DSP48E	48	132
Memory (RAM)	12	26

Table 5.2: *Xilinx® Virtex-7 resource utilization for the V-BLAST/ZF and the FSD detection algorithms*

to specific QAM constellation using an LUT at 26%. For FSD on the other hand, the highest complexity comes from calculating the distance metric where the dot $(\cdot)$ operation of channel matrix uses most of the resources, as well as the summation of the accumulated ED, taking almost 75% of the total FSD operation. These results will provide an estimation for hardware implementation.

When the FSD and V-BLAST/ZF detection algorithms are implemented on different factors of Ω, the BER degrades significantly for both detection algorithms as depicted in Figure 5.3(a) and Figure 5.3(b) for FSD and V-BLAST/ZF respectively. As the channel correlation increases, more profound differences are observed at higher SNR regions. This gets problematic at higher correlated channels when the V-BLAST/ZF is deployed, with BER of higher than 10^{-1} for $\Omega = 0.7$ for SNR ≤ 20 dB as depicted in Figure 5.3(b). In order to achieve the BER tolerance design for the entire system of 10^{-3}, SNR approximately ≥ 45 dB for V-BLAST/ZF is required when the $\Omega = 0.7$ in comparison to SNR of approximately 27 dB for uncorrelated channels as depicted in Figure 5.3(b). Similarly, a higher SNR is also needed or the FSD as shown in Figure 5.3(a), where the BER for $\Omega = 0.7$, is also higher, at 10^{-2} for SNR of 20 dB and lower, and it requires an SNR of more than 26 dB to obey the system performance requirements. However, the BER performance would improve significantly when the turbo decoder is included in the design, which may help in dealing with maintaining the overall performance of the system on spatially correlated channels.

With the performance verified, the MI values are calculated to provide the design of the thresholds for the Adaptive Switching Algorithm detector on different correlated channels. It is found that even though fading correlation does considerably affect the BER performance of each detection algorithm, the correlation index does not show any considerable changes to the MI

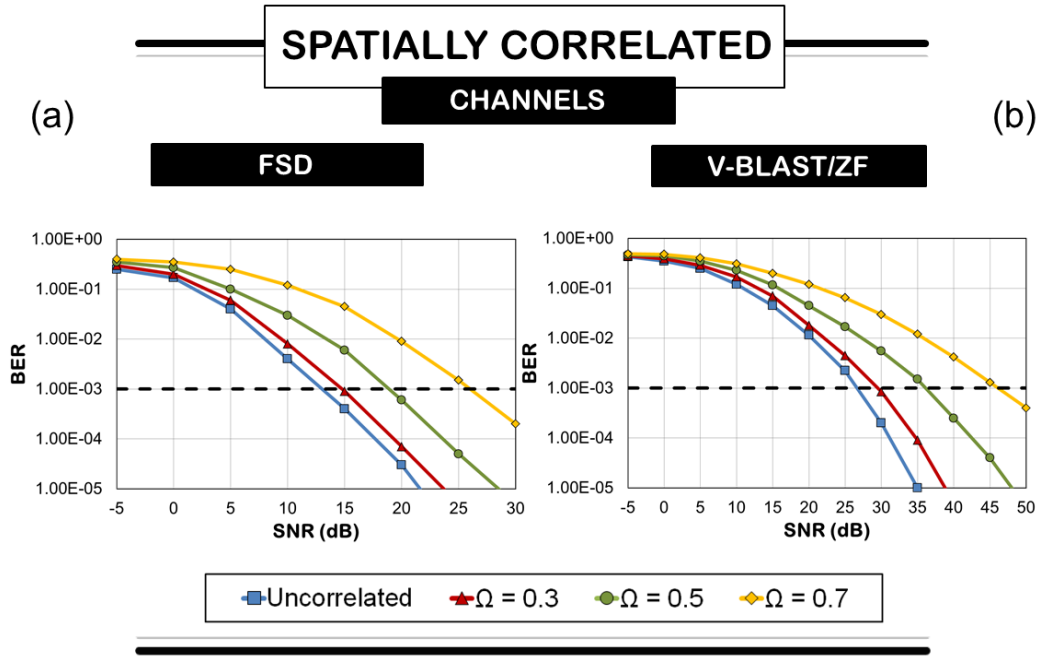

Figure 5.3: *Comparison of detector performance on spatially correlated channels*

values obtained. Monte Carlo simulations are run 10 times, where each run comprises $100,000$ channel realizations for each correlation index, Ω, at the SNR span of -5 dB to 20 dB. This can be observed in Figure 5.4.

The impact on the obtained MI thresholds shows only minor changes as the correlation of the channel increases. The two thresholds for the Adaptive Switching Algorithm detector lie in the range of $2,100$ to $2,300$ for, T_1, and $7,100$ to $7,800$ for threshold 2, T_2, for FSD and V-BLAST/ZF respectively. It gives a linear trend therefore, it can be concluded that the threshold values for the Adaptive Switching Algorithm detector remain the same even when applied spatially correlated channels and it can further be said that the detector design is only specific to the modulation and coding schemes in use. With these results, the design for the proposed algorithm is set as $2,200$ and $7,100$ for T_1 and T_2 respectively. T_1 corresponds to the BER = 0.5 and T_2 for a BER of 10^{-3}.

The other performance parameter, which is the energy consumption, can be calculated by taking the power readings provided by Xilinx® ISE and using the time it takes to transfer a packet bit size of $1,024$ at a core voltage of $V = 1$ V and an operating frequency of $f = 250$ MHz on the Xilinx® Virtex-7 chipset. For the span of the SNR levels of -5 dB to 20 dB, the average energy

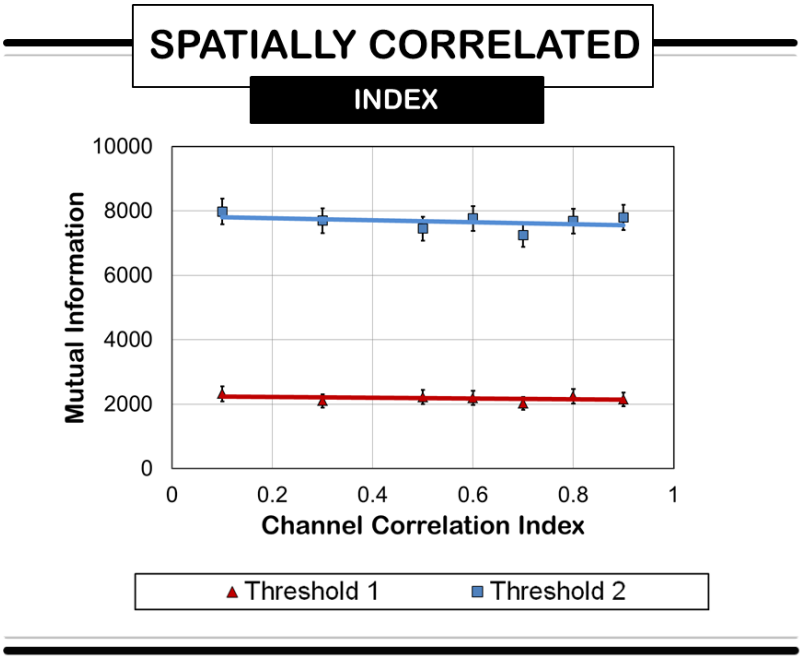

Figure 5.4: *Comparison of detector energy consumption on spatially correlated channels*

consumption of the two detection algorithms within the Adaptive Switching Algorithm against the correlated channel index range of 0 to 1 are computed for the FSD and the V-BLAST/ZF as 3.6 μJ and 0.9 μJ respectively. This shows that with the increase in correlation, the energy consumption of the detector is hardly affected. This could be due to both algorithms working independently of the noise level and have a fixed distinct search on any channel conditions. For the detector, it can be concluded that comparable energy savings can be gained in spatially correlated channels as well. When combining both algorithms to make the Adaptive Switching Algorithm, Figure 5.5 shows the energy consumption on spatially correlated channels. In the detector, the energy savings when utilizing the Adaptive Switching Algorithm on different correlated channel indexes can be calculated numerically for SNR range of 0 dB to 50 dB for a run of $100,000$ channel realizations on the chosen hardware. This is essentially the area under the graph of Figure 5.5 if the FSD is taken as the 100% baseline at 3.6 μJ. The results are tabulated in Table 5.3. It can be observed that though there are still savings gained, the energy savings decreases with higher channel correlation.

With the FSD consuming approximately four times the V-BLAST/ZF algorithm, both show no changes in the energy usage. The V-BLAST/ZF uses less energy than the FSD due to it being

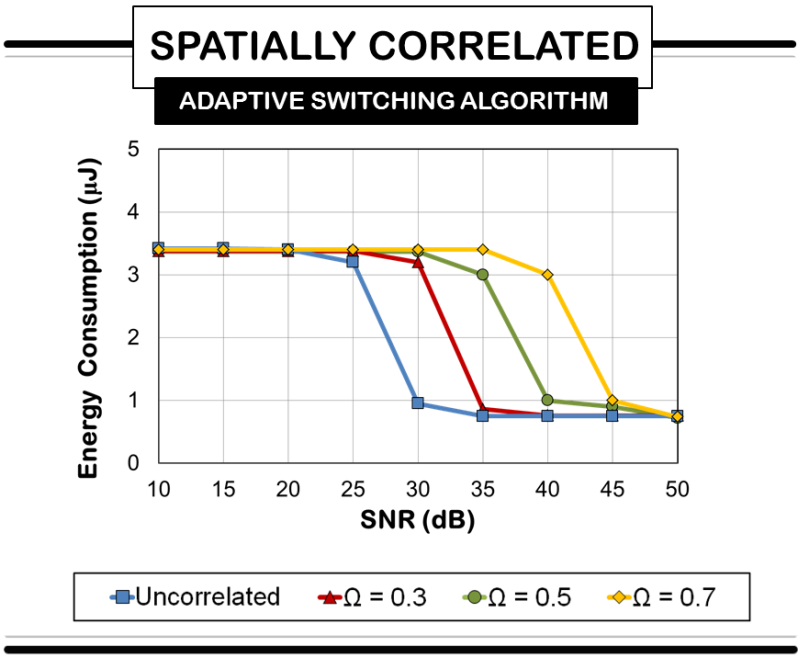

Figure 5.5: *Energy consumption of the Adaptive Switching Algorithm in spatially correlated channels*

less complex as a detector. It shows that similarly, though the correlation does not affect the overall power or energy consumption over a range of SNR observed, due to the adaptivity in the algorithm switching of FSD to the lesser complexity of V-BLAST/ZF at high SNRs, energy is saved.

Correlation Index (Ω)	Energy Savings (%)
Uncorrelated	40%
0.3	33%
0.5	27%
0.7	19%

Table 5.3: *Energy savings of Adaptive Switching Algorithm detector on spatially correlated channels*

Figure 5.5 also shows the reason for the reduced energy saving, which is that, the threshold T_2 between the two algorithms corresponds to a much higher SNR for higher channel correlation values. From the figure, it can be observed that the switching occurs at an SNR $\approx$ 25 dB for uncorrelated channels, and SNR $\approx$ 46 dB for Ω = 0.7. It can be concluded that the energy usage varies for the Adaptive Switching Algorithm with varying channel correlation factors,

with lower savings gained as the correlation increases.

5.5.2 Part 2: Joint Switching of the Detector and the Decoder

Since the effectiveness of the proposed algorithm detector can save energy regardless of channel correlation index, this part of the work investigates the next part of the receiver, labelled **Part 2** in Figure 5.1, which is the applicability of the Adaptive Switching Algorithm as a link between the detector and iterative decoder. **Part 2** is where the two thresholds for both the detector and decoder reside. When each part of the receiver, which are the detector(s) and the iterative decoder, are implemented on the Xilinx® Virtex-7, the multiplier counts and thus the complexity are determined. It can be found that about 76% of the total complexity of the receiver is from the iterative-MIMO turbo decoder, with 23% related to the MIMO detector with 1% reserved for the threshold control. Therefore, minimizing the complexity within the decoder would achieve greater energy savings than the ones obtained in **Part 1**, i.e. in the detector(s).

Shifting the focus to the decoder, the turbo decoders are divided into several blocks. If the total resource allocation for the entire decoder is set to be at 100%, the blocks with their corresponding complexity are detailed in Table 5.4. It can be noted that the highest complexity comes from the MAP decoders, therefore, limiting the number of iterations each received packet needs to go through would be the key to minimizing energy consumption within the turbo decoding. The Adaptive Switching Algorithm passes the MI calculated in the detector to the decoder, and thus the number of iteration iterative turbo decoder can be determined.

Block	Multiplier Counts (%)
Interleaver	1%
Demultiplexer	13%
logMAP Decoder(s)	80%
Trellis Tree	2%
Extrinsic LLR	4%

Table 5.4: *Complexity breakdown for turbo decoding*

Figure 5.6 gives the maximum, minimum and average number of iterations required when the experiment on the same Monte Carlo setup as in **Part 1**, where packets of $1,024$ bits over $100,000$ channel realizations are transmitted. The trend resembles the stopping criteria trends in Figure 5.2, whereas the MI increases, the number of decoding iterations decreases. Due to the design of the proposed algorithm, no decoding takes place when the MI is below T_1,

which is MI of $2,100$ and below, saving from unnecessary computations when the failure rate is extremely high. An ARQ or re-transmission is enabled in this region.

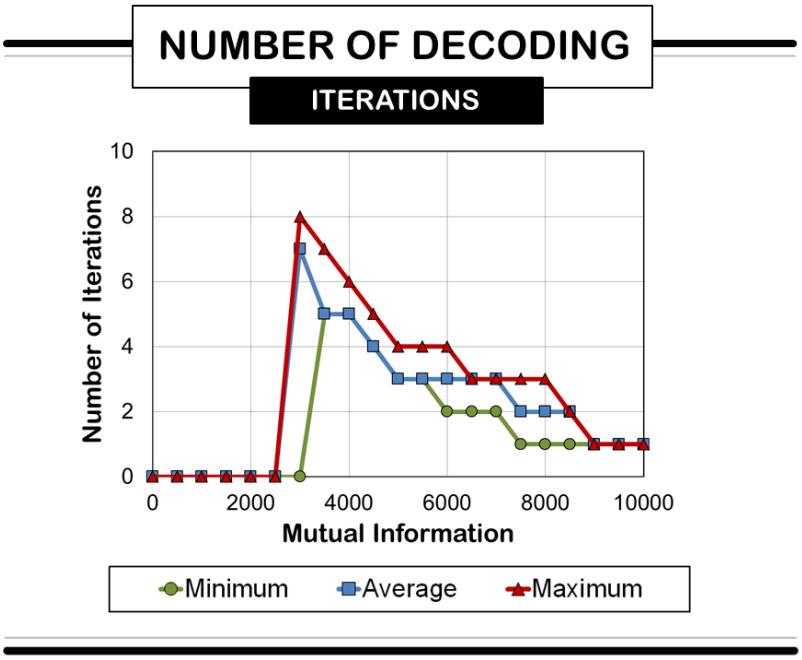

Figure 5.6: *Comparison of detector energy consumption on spatially correlated channels*

The trends provide a general idea for the range in iterations required in the turbo decoder over the considered number of transmissions. The average and maximum lines provide guidelines to the required number of iterations but are not directly used in the threshold design for the decoder. The minimum number of iterations is taken from Figure 5.6 as a foundation for the "Adaptive Switching Algorithm" threshold design in the decoder. Different stopping criteria for the decoder, one with the state-of-the-art used in LTE systems, the "CRC-24" method [138], and another without any stopping methods, with maximum of eight iterations throughout, labelled the "No Stopping Criteria" for the detector and decoder link are compared, as shown in Figure 5.7(a). The results are obtained using the Xilinx® System Generator software. For a fair comparison of the stopping criteria, the detector part is fixed to FSD with different stopping criteria usage on the decoder. It can be seen that the number of iterations required on Adaptive Switching Algorithm is the same as the CRC-24 method. The Adaptive Switching Algorithm has a fail-safe error checking method at the end of the final iteration, therefore, if a packet is not correctly decoded by the end of the final iteration, the decoder would increase the number of iterations up to a maximum of eight, after the CRC-24 check is implemented, giving it more

reliability in performance. In addition to the Adaptive Switching Algorithm using different iteration counts, Figure 5.7(b) shows that the proposed Adaptive Switching Algorithm also uses only about 18% multipliers needed as a stopping criterion when compared to the state-of-the-art CRC-24 method, when taking the latter as a baseline for percentage complexity calculations. This is due to the CRC having intricate calculations involving division of the data polynomials to get the remainder. For CRC-24, the degree of the polynomial is 24. It can be said that due to a smaller number of multiplier counts and comparable number of iterations needed, the Adaptive Switching Algorithm provides a better implementation when compared to the CRC-24 method.

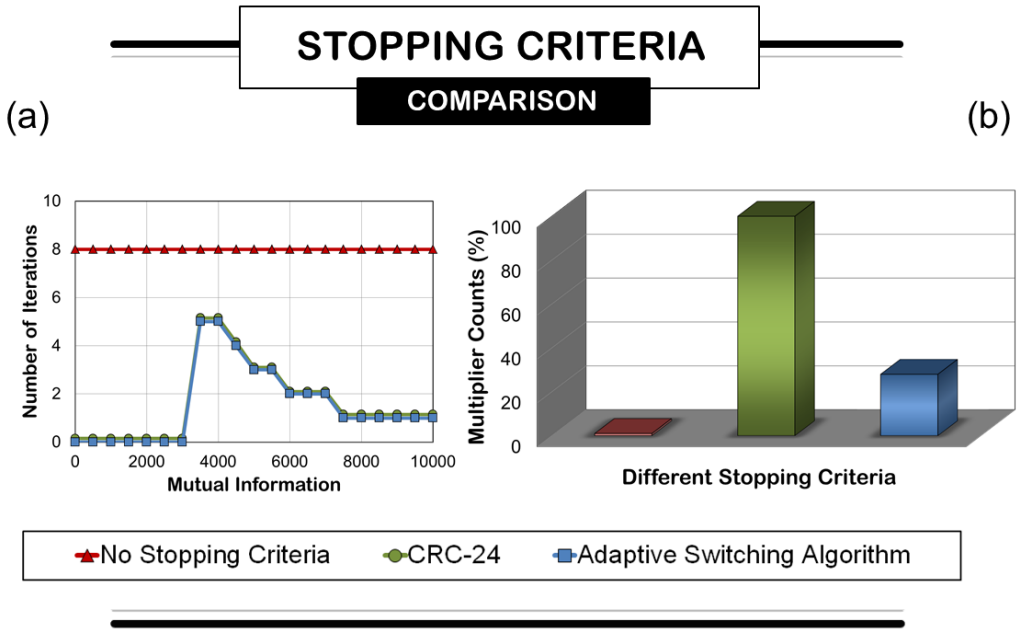

Figure 5.7: *Comparison of stopping criteria in turbo decoder*

When calculating the energy consumption using the same setup as **Part 1**, it can be observed that the "No Stopping Criteria" uses a lot more energy and is consistent throughout the span of considered SNR of -5 dB to 20 dB. Due to the minimization of the turbo decoding iterations, the energy consumption for both CRC-24 and the proposed decoder algorithm utilize a much lower energy consumption particularly at high SNR regions. Taking the "No Stopping Criteria" as the baseline for energy savings calculations, the overall percentages of energy savings are summarized in Table 5.5. The Adaptive Switching Algorithm decoder saves 7% more energy in comparison to the state-of-the-art CRC-24. Though this savings is not particularly large, this part of energy savings only considers the decoder part and more savings can be gained when a

full Adaptive Switching Algorithm is utilized in the iterative-MIMO receiver.

XILINX® VIRTEX-7: XC7VLX330TFFG1157	
Receiver Setup	Average Total Energy Savings
No Stopping Criteria	-
CRC-24	32%
Adaptive Switching Algorithm	39%

Table 5.5: *Average energy savings of the decoder on Xilinx® Virtex-7*

With both detector and decoder blocks verified, the receiver for the Adaptive Switching Algorithm can be constructed. The two thresholds LUT designs for the detector and the decoder that sit in **Part 2** are summarized in Table 5.6.

MIMO Detector			Turbo Decoder		
Label	MI	Type of Detector	Label	MI	No. of Iterations
ARQ	$\leq$ 2,200	No Detection	ARQ	$\leq$ 2,200	0
T_1	2,200 $< \bar{I}_i \leq$ 7,100	FSD	T_a	2,200 $< \bar{I}_i \leq$ 4,000	5
-	-	-	T_b	4,000 $< \bar{I}_i \leq$ 4,500	4
-	-	-	T_c	4,500 $< \bar{I}_i \leq$ 6,000	3
T_2	$>$ 7,100	V-BLAST/ZF	T_d	6,000 $< \bar{I}_i \leq$ 7,500	2
-	-	-	T_e	$>$ 7,500	1

Table 5.6: *Adaptive Switching Algorithm threshold designs for detector and decoder blocks of receiver*

In order to understand how the full Adaptive Switching Algorithm behaves, consider these four scenarios illustrated in Figure 5.8 on how a transmission can take place. "Scenario 1" is when the MI = $2,500$. Referring to the threshold designs in Table 5.6, this packet will go through the FSD detector and 5 iterations on the turbo decoder before the packet is successfully decoded. "Scenario 2" represents an MI = $4,700$, and thus, the packets will go through 3 iterations in the decoder after being detected by the FSD. If the accumulated MI = $8,000$ as in "Scenario 3", the packets will be detected by the V-BLAST/ZF and only iterate once in the decoder. Lastly, "Scenario 4" denotes MI = $1,800$. Since the MI is less than the necessary MI for any detecting and decoding to take place, an ARQ is activated so that the transmitter will re-transmit the same data packets in hope for a better channel condition.

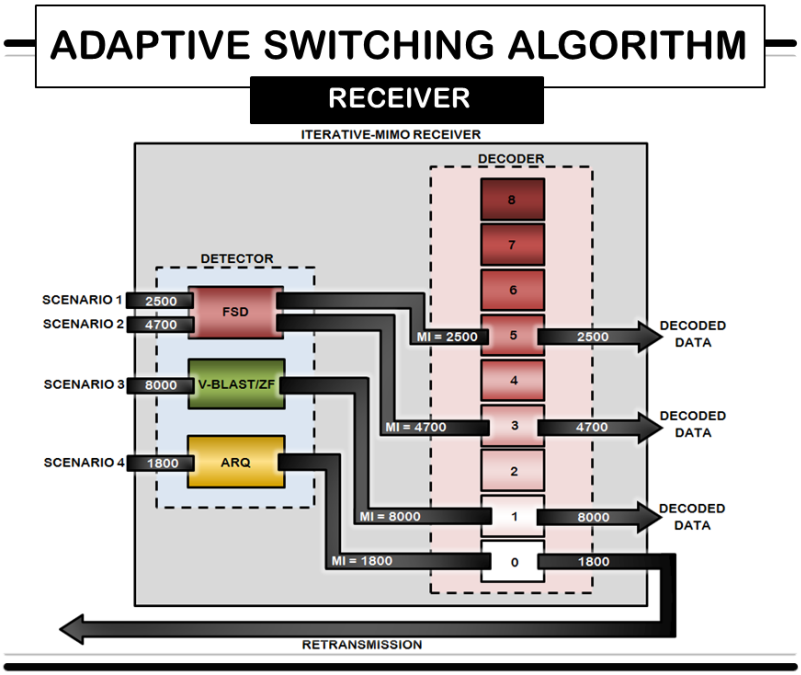

Figure 5.8: *Different transmission scenarios for Adaptive Switching Algorithm receiver*

5.5.3 Part 3: The Receiver Power Savings in Realistic Conditions

With the new design for the thresholds in the decoder, this section studies the different decoder setup to understand its performance of the newly Adaptive Switching Algorithm decoder by fashioning different designs for the receiver system. The work in **Part 3** therefore compares the full Adaptive Switching Algorithm with other systems as given in Table 5.7.

Name of System	Detector	Decoder
Full High Specification	FSD	No Stopping Criteria
State-of-the-Art	FSD	CRC
Half ASA	FSD	ASA
Full ASA	ASA	ASA

[ASA - Adaptive Switching Algorithm]

Table 5.7: *Receiver systems design parameters*

These four systems are compared to verify the effectiveness of different system designs. The "Full High Specification" consists of the high performance FSD for the detection and always performs the maximum eight iterations for the turbo decoding. In the second system, the FSD is used alongside the latest stopping criteria method used in the LTE systems, which is the

CRC-24. The proposed Adaptive Switching Algorithm design is investigated where the decoder coupled with the FSD as the detector to show the mechanism of the Adaptive Switching Algorithm as a stopping criterion in the system. This makes up the third system. Lastly, the full Adaptive Switching Algorithm system design, which operates the Adaptive Switching Algorithm on both parts of the system are measured for power and energy performance to confirm its validity in the iterative-MIMO receiver systems.

By incorporating the turbo decoder, the BER performance of receiver using the V-BLAST/ZF is explained in Figure 5.9(a). Similar to Figure 5.3, spatially correlated channels affect negatively on the BER performance. However, due to the decoder, the V-BLAST/ZF is now able to achieve a better BER performance. The required SNR for detector switching from FSD to V-BLAST/ZF is also illustrated here. This Figure 5.9(a) shows that the correlated channel requires a higher SNR ≈ 20 dB is needed for $\Omega = 0.7$ for the detection to occur in comparison to SNR ≈ 9 dB when the channel is uncorrelated. With these values, the BER for the Adaptive Switching Algorithm can be seen in Figure 5.9(b). From the figure, it can be observed the switch for transmissions during the uncorrelated MIMO channels occur at around $8 - 9$ dB, around 11 dB for $\Omega = 0.3$, 14 dB for $\Omega = 0.5$ and 20 dB for $\Omega = 0.7$. It can be seen that the BER performance is still under 0.5 and 10^{-3} for T_1 and T_2 respectively. Separate considerations of the Adaptive Switching Algorithm in the detector and decoder have proven that the adaptivity in the proposed algorithm has the ability to save energy whilst maintaining satisfactory BER performance. It can be concluded that the Adaptive Switching Algorithm works well for the full iterative-MIMO receiver design, since it is able to conform to the error tolerance requirement of the system of 10^{-3}.

Using the same energy calculation method, taking the "Full High Specification" as a baseline, the total energy usage can be calculated as areas under the graphs. In order to see how the extreme cases of correlation affect the energy savings, correlations of 0 and 0.9 are considered. Since most current systems normally operate between the range of 0 dB to 40 dB during real-life deployment [144], the results for the simulation under these SNR regions are given in Figure 5.10(a) for uncorrelated channels, i.e. $\Omega = 0$, and in Figure 5.10(b) for correlated channels of Ω close to 1, i.e. $\Omega = 0.9$. It can be seen that higher SNRs are required to reduce energy consumption for highly correlated channels. The energy savings are summarized in Table 5.8.

Energy savings of $74 - 78\%$ across SNR of 0 dB to 40 dB can be achieved when the "Full Adaptive Switching Algorithm" system is utilized for uncorrelated and correlated channel re-

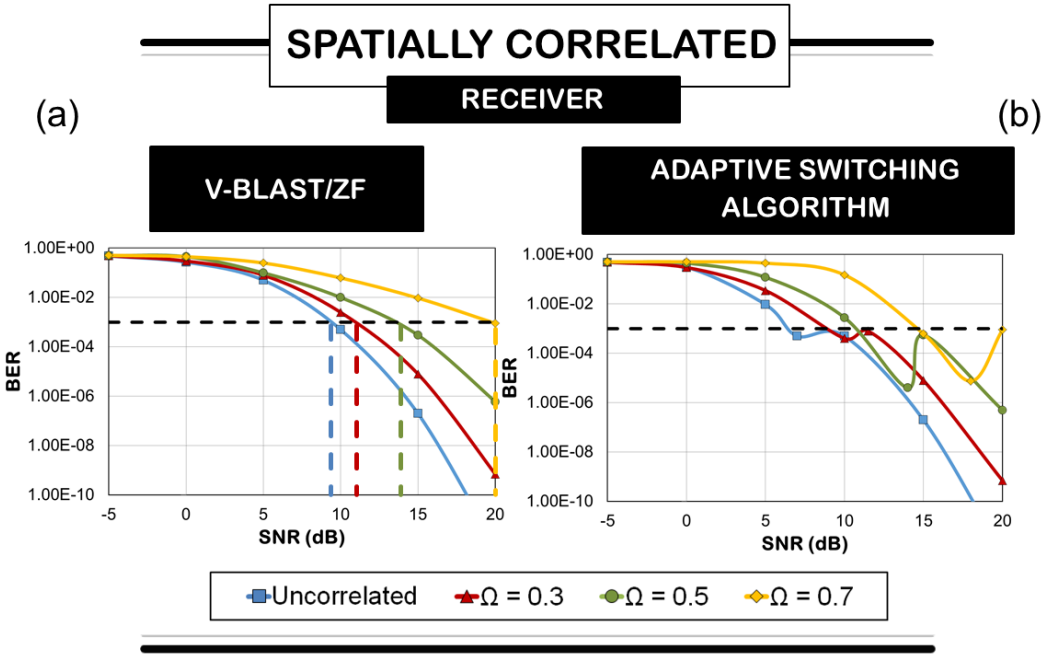

Figure 5.9: *Performance of turbo decoder in spatially correlated channels*

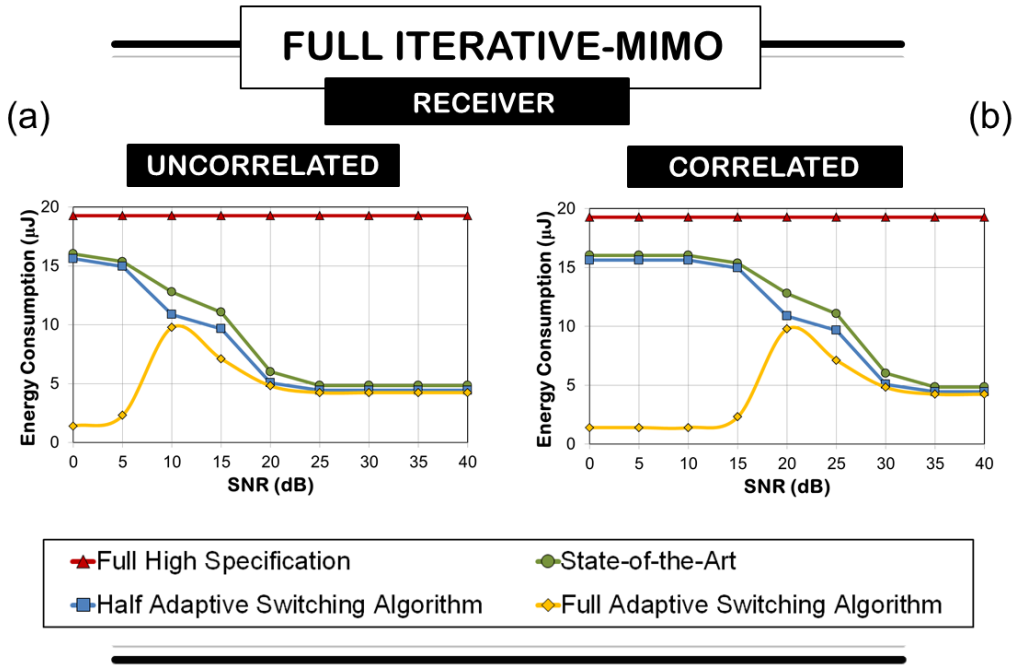

Figure 5.10: *Full receiver design with Adaptive Switching Algorithm*

XILINX® VIRTEX-7: XC7VLX330TFFG1157		
Receiver Setup Name	Energy Savings	
	Uncorrelated	Correlated
Full High Specification	-	-
State of the Art	54%	40%
Half Adaptive Switching Algorithm	59%	44%
Full Adaptive Switching Algorithm	78%	74%

Table 5.8: *Average energy savings of the iterative-MIMO receiver on Xilinx® Virtex-7*

spectively. Both the uncorrelated and correlated channels follow roughly the same energy trend with the exception of needing a higher SNR for the latter type of channel conditions. This gives a benefit of around $24 - 34\%$ savings gained in comparison to the state-of-the-art CRC-24 method. The savings lessen as the correlation increases, however, 74% energy savings can be gained when the channel is highly correlated, it can be concluded that the Adaptive Switching Algorithm works in an energy efficient manner regardless of the channel conditions.

5.6 Chapter Summary

The Adaptive Switching Algorithm was utilized in both detector and decoder to create a full adaptive iterative-MIMO receiver. The same threshold calculations involving the MI between the transmitters and receivers provide sufficient information in real-time regarding any channel conditions, whether uncorrelated or spatially correlated. The work has proven that the average energy savings in the detector that can be achieved throughout the span of considered SNR conditions of -5 dB to 20 dB, are within the range of 19% to 40% when implemented on Xilinx® Virtex-7 chipset. The design for the Adaptive Switching Algorithm was expanded to be a link between the detector and decoder, which helps reduce the energy consumption up to 39% by limiting the number of turbo decoding iterations in spatially correlated conditions, in comparison to the baseline system. When a full Adaptive Switching Algorithm is implemented on the receiver, 74% of the total energy consumption could be saved regardless of channel conditions. Thus, the proposed algorithm confirms that its adaptivity attribute in iterative-MIMO receivers is highly beneficial and the idea could be adopted in real-world future wireless communication devices.

Chapter 6
Conclusions

This chapter aims to provide a summary for the thesis, beginning with a brief description of the motivation behind the work. The specific, technical aims and objectives are explained in order to highlight the contributions of this work to the **wireless communication** and **computer architecture** communities' body of knowledge and the potential impact to society. This chapter also includes a discussion on the limitations of the current research and future research directions that can be considered in both wireless communication and computer architecture. The main objective of this research was to invent an adaptive algorithm suitable for an iterative-MIMO that could potentially gain power and energy savings in the algorithmic design as well as during implementation. The behaviour of the proposed algorithm, dubbed the Adaptive Switching Algorithm, as well as its mechanism were theoretically analysed in depth initially. Mapping onto selected platforms was then performed in order to identify the points of power optimization. Adaptivity seems to be the key to minimizing the power and energy consumption within the receiver design and this was successfully demonstrated during software and hardware design implementations. The novel algorithm was then put to the test in realistic channel conditions and the verified design was found to be suitable for the current and future wireless communication systems.

6.1 Summary

A significant breakthrough came about in the late 20^{th} century when the adaptive use of MIMO antenna systems was proposed in order to cater for the explosive data demand in wireless communication. This had triggered a great deal of research into algorithms and architectures that benefit from the increase in capacity. However, the increasing number of devices totalling almost 14 billion worldwide has resulted in an exigency in energy consumption. Solutions are therefore called for, and more efficient software and hardware designs are hence imperative. Adaptivity in algorithms and hardware implementations has been one of the approaches to accommodate this expanding predicament. The introduction of the Adaptive Switching Algorithm

provides a good solution in both software and hardware where it gives the system a level of 'intelligence' to adapt to any situation in real-time. The proposed algorithm has shown promising energy saving results and flexibility in both algorithmic and hardware design architectures. Moreover, the system performs well under the requirement of the overall iterative-MIMO receiver design. For clearer understanding, the work has been divided into two parts; the **wireless communication** part and the **computer architecture** part. The **wireless communication** part deals with the algorithmic design for the iterative-MIMO system. By combining two detection algorithms, the receiver system is able to behave according to certain channel conditions. The threshold design for the switching was controlled by information gathered between the transmission channel for the transmitter and receiver during real-time. The interchanging between low complexity detector used in high SNRs and high performance detector in bad channel conditions has been highly beneficial, where the adaptivity has given a good trade-off between the BER performance and the power and energy consumption. The power and energy consumption were analysed further where different power saving methods were investigated on hardware to enhance the power savings gained in the algorithm design. The **computer architecture** part ventured towards implementing several power minimization techniques onto an FPGA hardware. The amalgamation of the software and hardware consequently delivers huge power and energy savings when the Adaptive Switching Algorithm was mapped onto the Xilinx® Virtex-7. Once the algorithm and the hardware design implementation were verified, the full receiver system was analysed under realistic situations. The deployment of the system on simulated spatially correlated channels was investigated and positive outcomes were gained during the experiments.

6.2 Major Research Findings

The main contributions of this work have been elaborated in three separate technical chapters. The first contribution corresponds to the innovative design of the Adaptive Switching Algorithm, which works adaptively, switching from high performance FSD algorithm to the low complexity V-BLAST/ZF based on the current channel conditions. This led to the second contribution, where the design was implemented on the latest FPGA and several power minimization techniques were applied when the algorithm was designed on hardware. The third contribution is when the algorithm was put to the test under realistic channel conditions to see the gains it can achieved and to ascertain if it can exceed the state-of-the-art in terms of the

design and the efficiency on hardware performance. Specifically, the major research outcomes can be described as follows:

- The novel invention of the Adaptive Switching Algorithm for an iterative-MIMO receiver works by switching between low complexity detection algorithm and the high, close to ML, performance detector. The switching occurs according to the MI calculated based on the current channel condition and noise level between the transmitter and receiver in real-time. The applicability of the Adaptive Switching Algorithm has shown that more than half in resources consumption can be saved on both software and preliminary hardware implementations, respectively. Having 'intelligence' in the algorithm design and the hardware setup offers optimistic outcomes in both performance and complexity for the current and future iterative-MIMO systems. The adaptivity provided by the thresholds is controlled by the MI between the transmitters and receivers. They give significant information about the channel conditions as they offer comprehensive statistics regarding the MIMO conditions.

- The Adaptive Switching Algorithm for both software and hardware are implemented to show the suitability of the algorithm for realistic implementation. During extensive studies of several power minimization techniques of *DVFS*, *sleep mode* and *parallelization*, the best method of power minimization, which is a combination of *sleep mode* and *parallelization*, on the Adaptive Switching Algorithm was established. By utilizing combinations of the power minimization techniques, it can be seen that the system is able to save energy up to a total of 89%.

- The design for the Adaptive Switching Algorithm was utilized in both the detector and decoder to create a full adaptive iterative-MIMO receiver. The same threshold calculations involving the MI between the transmitters and receivers provide sufficient information in real-time regarding any channel conditions, uncorrelated or spatially correlated. The work has proven that the average energy savings in the detector can be achieved throughout the span of considered SNR conditions and they are in the range of between 19% to 40% when implemented on Xilinx® Virtex-7 chipset. The design for the Adaptive Switching Algorithm was expanded to be a link between the detector and decoder, which helps reduce the energy consumption up to 39% by limiting the number of turbo decoding iterations in spatially correlated conditions, in comparison to the baseline systems. Even though spatially correlated channel introduced BER degradation at high cor-

related channels, the threshold design for the decoder still meets the specified error performance. When a full Adaptive Switching Algorithm is implemented on the receiver, it shows that similar savings of up to 74% can be gained according to the prediction given in [1]. Thus, the proposed algorithm corroborates the fact that its adaptivity attribute in iterative-MIMO receivers is highly beneficial and should be adopted in future wireless communication devices.

6.3 Limitations and Further Research

There are limitations from both software and hardware standpoints. With regard to the **theoretical software** aspect, several assumptions have been made in the system and channel modelling, for example, ideal channel estimation, perfect timing, flat fading environments or that the results are based on numbered channel realizations. A possible extension of this work could take these assumptions into consideration and replace them with more realistic models of different parts of the system to analyse the performance of the proposed Adaptive Switching Algorithm. From the **hardware implementation** point of view, hardware mapping on different parts of the proposed algorithm could also be investigated, for example, hardware resources can be shared for the two detection algorithms of FSD and V-BLAST/ZF to further optimize the design for the proposed algorithm and/or a deeper analysis on fixed-point performance using a common quantization approach in order to compare them. In addition, the channel ordering for both detection algorithms have been done purely on simulation with fixed point arithmetic. In practice, it would be highly beneficial to incorporate the possible architectures for real-time implementation of the pseudoinverse calculations of the channel matrix to fully analyse the system's applicability. Moreover, in practice, the switching of different modes may create some latency in the hardware, which creates delays at the output. The lifetime of the hardware might also be affected by the rapid circuit switching in between the two modes of "high performance" to "low power" if occur too frequently. On a more **general** aspect, several possible routes could be considered for future work, for example, different modulation and code rates could be used to show the robustness of the idea behind the Adaptive Switching Algorithm. Since adaptivity is the key to power and energy savings, the design for the threshold can be further investigated to pinpoint the strengths and weaknesses of the proposed algorithm by incorporating different system parameters. Finally, a good direction for future research would be to implement the algorithm on dedicated hardware to see how it would perform under realistic conditions for both

indoor and outdoor conditions.

Appendix A
Publications

The list comprises papers that have been published or submitted in journal(s) and conference proceeding(s). The ones marked by * indicate that the paper(s) have been submitted and have not yet been published. All papers are included in the appendix section and are used as references throughout the thesis.

Journal Paper(s):

- N. Tadza, D. I. Laurenson, J. S. Thompson, "Adaptive Switching Detection Algorithm for Iterative-MIMO Systems to Enable Power Savings", *Journal of Radio Science*, vol. 49, no. 11, pp. 1065-1079, November 2014.
- N. Tadza, J. S. Thompson, D. I. Laurenson, "Practical Performance of the Adaptive Switching Algorithm for Iterative-MIMO Receivers in Spatially Correlated Channels", *submitted to IEEE Transactions on Consumer Electronics in May 2015*. *

Conference Paper(s):

- N. Tadza, D. I. Laurenson, J. S. Thompson, "Adaptive Switching Algorithm in Turbo-MIMO Systems", *URSI Festival of Radio Science*, vol. 1, no. 5, pp. 27-28, September 2013.
- N. Tadza, J. S. Thompson, D. I. Laurenson, "Power Performance Analysis of the Iterative-MIMO Adaptive Switching Algorithm Detector on the FPGA Hardware", *IEEE 81st Vehicular Technology Conference (VTC) - Spring*, vol. 81, no. 21, pp. 1-5, May 2015.

AGU PUBLICATIONS

Radio Science

RESEARCH ARTICLE
10.1002/2013RS005323

Special Section:
Green Radio Communications

Key Points:
- Adaptivity achieves optimal error rate performance with minimal power
- Receiver adapts to conditions according to the mutual information thresholds
- DVFS technique reduces the circuit power demands of the adaptive algorithm

Correspondence to:
N. Tadza,
n.tadza@ed.ac.uk

Citation:
Tadza, N., D. Laurenson, and J. S. Thompson (2014), Adaptive switching detection algorithm for iterative-MIMO systems to enable power savings, *Radio Sci.*, *49*, 1065–1079, doi:10.1002/2013RS005323.

Received 23 OCT 2013
Accepted 25 AUG 2014
Accepted article online 27 AUG 2014
Published online 13 NOV 2014

Adaptive switching detection algorithm for iterative-MIMO systems to enable power savings

N. Tadza[1], D. Laurenson[1], and J. S. Thompson[1]

[1]Institute for Digital Communications, School of Engineering, University of Edinburgh, Edinburgh, UK

Abstract This paper attempts to tackle one of the challenges faced in soft input soft output Multiple Input Multiple Output (MIMO) detection systems, which is to achieve optimal error rate performance with minimal power consumption. This is realized by proposing a new algorithm design that comprises multiple thresholds within the detector that, in real time, specify the receiver behavior according to the current channel in both slow and fast fading conditions, giving it adaptivity. This adaptivity enables energy savings within the system since the receiver chooses whether to accept or to reject the transmission, according to the success rate of detecting thresholds. The thresholds are calculated using the mutual information of the instantaneous channel conditions between the transmitting and receiving antennas of iterative-MIMO systems. In addition, the power saving technique, Dynamic Voltage and Frequency Scaling, helps to reduce the circuit power demands of the adaptive algorithm. This adaptivity has the potential to save up to 30% of the total energy when it is implemented on Xilinx® Virtex-5 simulation hardware. Results indicate the benefits of having this "intelligence" in the adaptive algorithm due to the promising performance-complexity tradeoff parameters in both software and hardware codesign simulation.

1. Introduction

The ability to increase throughput without requiring more computational power has always been a topic of interest amongst the wireless communication research community. Multiple Input Multiple Output (MIMO) promises high throughput without additional transmit power [*Goldsmith et al.*, 2007], however, minimizing the receiver's power, which is often limited, is still under intensive study. Current base stations, proliferations of femtocells and/or wireless access points also need to exercise being "green." The energy source is often shared amongst millions of devices. There are substantial potential of power savings to be gained in these small mains powered devices. In this paper, a field programmable gate array (FPGA) is used as a platform to show the inner workings of the adaptive algorithm. It is chosen due to its robustness, its reprogrammable capabilities and its potential for further energy savings by parallelization. The results obtained can be translated onto any hardware platform such as an application-specific integrated circuit (ASIC), which is more common in mobile devices. Fundamentally, a soft-MIMO receiver is divided into two parts, the MIMO detector and the soft decoder working together to achieve the best performance. The received data are processed through the detector before being passed into the decoder. Most publications focus on saving power using the signal-to-noise ratio (SNR) [*Wu*, 2011], channel matrix condition number [*Matthaiou et al.*, 2008], or reducing the number of turbo decoding iterations [*Zhang et al.*, 2009a, 2009b] for the receiver. Condition numbers of the channel matrix would only take into account the input and output matrix of the transmitter and the receiver. This is not sufficient as a switching metric since it disregards the noise level. SNR, on the other hand, does not compute the relationship between the antennas in a MIMO system. If the channel is deemed good, due to high SNR values, strongly correlated antennas would not make for a good transmission condition. This is because the correlated system provides insufficient diversity in the MIMO system. Therefore, mutual information (MI) is implemented due to its consideration of the diversity of a MIMO system, i.e., the transmitters and the receivers as well as the noise level. This gives a maximum amount of information regarding a channel with minimal complexity in comparison to using either condition number or SNR alone. This paper therefore shifts the attention to the detector using MI as the threshold control; in hope to gain energy savings earlier on the processing stages, i.e., by avoiding both detection and decoding processing. This iterative-MIMO scheme, which combines a spatial multiplexing MIMO detector and an outer forward error correction soft decoder with an interleaver in-between [*Ariyavisitakul*, 2000;

Sellathurai and Haykin, 2002], dubbed Bit-Interleaved Coded Modulation (BICM) [*Hochwald and Brink*, 2003], has a very high computational complexity as the receiver detects and decodes symbols by searching through possible transmit symbols. Moreover, this is done iteratively in soft-MIMO detection systems by the decoder.

This paper focuses on saving energy consumption in the MIMO detector, where it predicts symbols transmitted by each antenna by examining the channel noise and constellation modulation scheme. It should be noted that, though out of scope of the paper, after the process of detecting, the symbols are passed to the outer decoder before a hard decision can be made.

There are many types of different detection algorithms available, which can be generalized into "Nulling and Cancelling" methods, such as the Zero Forcing (ZF) [*Winters et al.*, 1994] and the Minimum Mean Square Error Estimation (MMSE) [*Li et al.*, 2006] techniques as well as the "tree search" algorithms, for instance, the Maximum Likelihood (ML), Sphere Decoding (SD) [*Fincke and Pohst*, 1985], and the Fixed Sphere Decoding (FSD) [*Barbero and Thompson*, 2008a] routines. For simple detectors, ZF and MMSE provide low complexity; however, they give poor performance in terms of bit error rate (BER). Linear detection methods, combined with nulling and cancelling, seem to give a better BER while maintaining the low complexity. This is why the combination of Vertical Bell Laboratories Layered Space Time (V-BLAST) and ZF is chosen. On the other hand, for close to ML performance, tree search algorithms such as FSD, layered orthogonal lattice detector, smart-ordered candidate adding algorithm, and K-Best result in high complexity in order to meet the performance criteria. This drains a lot of power in order to decode data packets, which is particularly wasteful in good channel conditions. In poor channel conditions, FSD has been chosen as a detection method as it is independent of the search radius, meaning, the complexity is fixed and minimal in comparison to other tree search algorithms. The novelty of this paper lies in the fact that the algorithm switches between high- and low-complexity detectors to give a bigger gain in energy savings. Ultimately, using different detectors would only slightly alter the thresholds that need to be implemented, confirming that MI is adaptive to any system for determining the threshold for switching.

The computational power required to implement tree search MIMO detection every time a symbol is transmitted is unnecessary in some channel conditions. As each detection algorithm has a different performance and complexity, choosing between them depends on the system's unique requirements. To construct an adaptive implementation that could fit on available hardware in the market, this study combines two detection algorithms. The Fixed Sphere Decoding (FSD) and the Vertical Bell Laboratories Layered Space Time/Zero Forcing (V-BLAST/ZF) techniques are incorporated into an adaptive approach that has the ability to selectively operate according to the received signal conditions. These two detection algorithms are chosen due to their fixed data throughput, potential for hardware parallel implementation and low complexity.

The proposed adaptive algorithm therefore prevents the receiver from performing extensive computation under very low or very high SNR conditions, which ultimately yields significant savings in energy. The algorithm utilizes multiple thresholds to intelligently switch MIMO detection schemes according to the current environment. This "intelligence" is the key to efficient energy utilization in the receiver. The results of this work will be presented in terms of overall energy savings from both software and hardware standpoints.

1.1. Contributions

The main contributions of this paper are summarized as follows:

1. An adaptive switching algorithm that adapts to real-time channel conditions by selecting to minimize the power consumption of iterative-MIMO detection systems is proposed. This is realized in the form of a threshold control unit, which selects the minimum complexity detector capable of meeting the desired BER performance.
2. The adaptive algorithm shows promising BER performance on a par with the current available detection schemes with lower computational complexity.
3. An evaluation of the new design in a Xilinx®Virtex-5 FPGA shows convincing dynamic and static power savings compared to baseline detectors.

AGU Radio Science 10.1002/2013RS005323

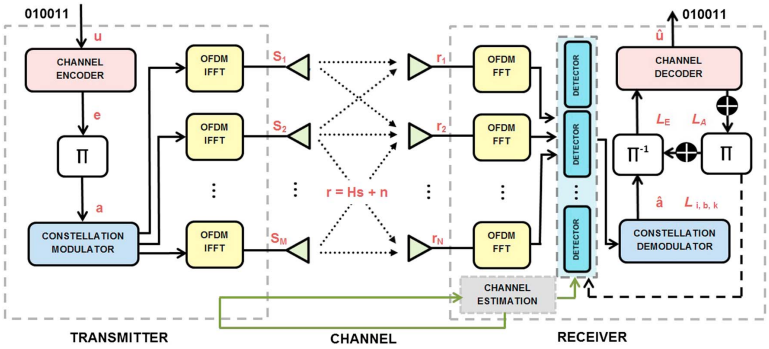

Figure 1. Iterative-MIMO (BICM) System.

2. Background

2.1. System Model

Consider an iterative-MIMO system comprising M transmit antennas and N receive antennas based on BCIM, transmitting frames of K_u bits as shown in Figure 1. At the transmitter, the K_u bits are encoded using an iterative encoding method such as convolutional or turbo coding [*Hagenauer et al.*, 1996] of rate R_c, where $K_u = K_e \cdot R_c$. The K_e-coded bits are then interleaved giving K_a bits, which are mapped into independent Quadrature Amplitude Modulation (QAM) constellations, $\mathcal{O}$, of P points, forming a sequence of $K_s = K_e / \log_2 P$ symbols. The symbols that are separated into M substreams blocks of $M \cdot K_{ch}$ symbols are transmitted in each channel realization, K_{ch}. These are transmitted over Rayleigh fading channels. In other words, a frame of K_e-coded bits requires a transmission of $K_s/(M \cdot K_{ch})$ blocks of data. Consequently, the received symbols, indexed by a sample time, k, can be written as

$$\mathbf{r}[k] = \mathbf{H}[k]\mathbf{s}[k] + \mathbf{n}[k] \tag{1}$$

where the channel matrix $\mathbf{H} \in \mathbb{C}^{M \times N}$ is assumed to be perfectly known at the receiver with independent elements $h_{i,j} \sim \mathcal{CN}(0, 1)$, for $1 \leq i \leq M$ and $1 \leq j \leq N$ representing a block Rayleigh fading propagation environment, $\mathbf{s} = (s_1, s_2, \ldots, s_M)^T$ is the transpose vector of the M-dimensional transmit symbol vector with $E[|\ \mathbf{s}_i\ |^2] = M^{-1}$, $\mathbf{n}$ is the $\mathbb{C}^{M \times 1}$ additive independent and identically distributed circular symmetric complex Gaussian noise vector of $h_{i,j} \sim \mathcal{CN}(0, \sigma^2)$ with $\sigma^2 = N_0$, and $\mathbf{r} = (r_1, r_2, \ldots, r_N)^T$ is the transpose N vector of received symbols. The set of all transmitted symbols forms an M-dimensional complex constellation $\mathcal{O}^M$ of P^M vectors, which specifies the dimensionality of the system.

2.2. MIMO Detection

The channel $\mathbf{H}$ is assumed to be known at the receiver through a preceding training period. This generates and saves data in the channel estimation block regarding the modulation schemes and the channel condition statistics. MIMO algorithms solve (1) by separating parallel data streams transmitted by antennas. They can generally be categorized into four types as described below.

2.2.1. ML

ML detection finds the minimum constellation point in (1) within the received symbols. It is given by

$$\hat{\mathbf{s}}_{ML} = \underset{\mathbf{s} \in \mathcal{O}^M}{\arg\min} \parallel \mathbf{r} - \mathbf{H}\mathbf{s} \parallel^2 \tag{2}$$

The ML detector is optimal and fully exploits all available diversity. Even though ML produces the best BER performance, due to its use of exhaustive search, it can have immense complexity for direct implementation. The complexity grows exponentially with the transmission rate R_c, since the detector needs to go through 2^{R_c} hypotheses for each received vector. For example, for the case of a 4×4 iterative-MIMO system employing 16-QAM, the detector would need to search a total of $K_s = 16^4 = 65{,}536$ candidates in order to find the correct transmitted vector. Several efficient suboptimal detection techniques have therefore

AGU Radio Science 10.1002/2013RS005323

Table 1. V-BLAST/ZF Algorithm[a]

Pseudo-Code
Channel realization:
$\mathbf{G}_1 = \mathbf{H}^+$
$i = 1$
Recursion:
$k_i = \arg\min_{j \notin \{k_1, \cdots, k_{i_1}\}} \parallel (\mathbf{G}_i)_j \parallel^2$
$y_{k_i} = (\mathbf{G}_i)_{k_i} \mathbf{r}_i$
$\hat{\mathbf{s}}_{k_i} = Q(y_{k_i})$
$\mathbf{r}_{i+1} = \mathbf{r}_i - \hat{\mathbf{s}}_{k_i}(\mathbf{H}_{k_i})$
$\mathbf{G}_{i+1} = \mathbf{H}^{\dagger}_{\bar{k}_i}$
$i = i + 1$

[a]Algorithm consists of channel ordering given by *Line 3*; *Line 4* performs nulling and computes the decision statistic; *Line 5* quantizes the computed decision statistic to yield the decision; *Line 6* performs cancellation by decision feedback, and *Line 7* computes the new pseudoinverse for the next iteration.

been proposed or adapted from the field of multiuser detection. Even though these techniques are much less computationally demanding than the ML detector, they are often unable to exploit a large part of the available diversity, and thus, their performance tends to be significantly poorer than that of ML detection. However, this tradeoff can be made for efficient hardware designs.

2.2.2. ZF: Linear Detection

This method neglects the constraint $\mathbf{s} \in \mathcal{O}^M$ in (2) and uses different criteria to find the nulling vectors, the most common being the ZF or MMSE approach [*Golub and Van Loan*, 1983]. Generally, the symbol $\hat{\mathbf{s}}$ is given by a transformation of the received vectors $\mathbf{r}$ in the form of

$$\hat{\mathbf{s}} = Q(\mathbf{H}^+ \mathbf{r}) \tag{3}$$

where $\mathbf{H}^+$ is the Moore-Penrose pseudoinverse matrix that depends on channel $\mathbf{H}$ and Q is a quantizer that maps the argument into the closest point in $\mathcal{O}^M$. Even though this method has low complexity, it does have a major drawback of having a rather poor performance in terms of BER.

2.2.3. V-BLAST: Ordered Successive Interference Cancellation

V-BLAST [*Golden et al.*, 1999] method gives slightly better BER performance in comparison to linear detection. However, due to the error propagation, it is still suboptimal in performance. This is often overlooked due to its practicality during implementation. V-BLAST is a recursive procedure that works by minimizing the influence of noise by reordering the channel matrix according to the signal strength received. The algorithm simply makes a first detection of the most powerful signal, consequently subtracting that signal from the overall detected symbols. It then continues the same process by proceeding to the detection of the second most powerful signal, and so forth. Assuming the ordered set to be

$$\S \equiv \{k_1, k_2, \ldots, k_M\} \tag{4}$$

the detection algorithm operates on $\mathbf{r}_i$, given in (5), while computing the decision statistics $y_{k_1}, y_{k_2}, \ldots y_{k_M}$, which are then quantized to form estimates of the received symbols $\hat{\mathbf{s}}_{k_1}, \hat{\mathbf{s}}_{k_2}, \ldots \hat{\mathbf{s}}_{k_M}$. The detection order is determined by the information about the channel conditions readily available within the estimation block. After computing (3), the detection process uses linear combinatorial nulling and symbol cancellation to successively compute the received vectors:

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \hat{s}_{k_i}(\mathbf{H})_{k_i} \tag{5}$$

When combined with the ZF method, it shows some improvement in BER while still maintaining low complexity. The complete V-BLAST/ZF detection algorithm is summarized in Table 1, where $\mathbf{G}$ denotes the Moore-Penrose pseudoinverse of the current channel $\mathbf{H}$, and therefore, $(\mathbf{G}_i)_j$ is the j^{th} row of $\mathbf{G}_i$, $Q(\cdot)$ is a quantizer to the nearest constellation point, $(\mathbf{H})_{\bar{k}_i}$ is the k_i^{th} column of $\mathbf{H}$, $\mathbf{H}_{\bar{k}_i}$ denotes the matrix obtained by zeroing the columns $k_1, k_2, \ldots, k_i$ of $\mathbf{H}$, and $\mathbf{H}^+_{\bar{k}_i}$ denotes the pseudoinverse of $\mathbf{H}_{\bar{k}_i}$. This type of detection scheme is best deployed in high-SNR environments.

2.2.4. SD and FSD

SD reduces the complexity of the ML detection problem [*Viterbo and Boutros*, 1999; *Pohst*, 1981; *Agrell et al.*, 2002] by introducing a constraint within the search called the sphere radius, R:

$$\hat{\mathbf{s}}_{\mathrm{SD}} = \arg\min_{\mathbf{s} \in \mathcal{O}^M} \parallel \mathbf{r} - \mathbf{H}\mathbf{s} \parallel^2 \leq R \tag{6}$$

The search can be visualized as a tree, traversing down each node until it encounters one with Euclidean Distance (ED) that is larger than R, where it will eliminate that branch from the search. The minimum symbol is acquired once it has traversed down through every path reaching the end, i.e., the leaf node(s). The SD has major drawbacks when it comes to hardware implementation due to having variable complexity

AGU Radio Science 10.1002/2013RS005323

and its sequential nature. The complexity of the SD depends on the noise level and the channel conditions. Moreover, the linearity of the search prevents parallelism for newer hardware design implementation. Parallelization has been proven to minimize energy consumption in circuit designs due to a workload being shared across multiple computational resources, so that the circuit can produce the same amount of throughput at a lower frequency of operation [*Chen et al.*, 2010; *Esmaeilzadeh et al.*, 2011; *Kumar et al.*, 2003]. Therefore, *Barbero and Thompson* [2008a] proposed a modified version, the FSD, in order to overcome both shortcomings. FSD is a combination of brute-force enumeration and a low complexity, approximate detector. Much like the SD, FSD traverses down the tree while calculating the ED; however, instead of having a radius constraint R, FSD determines in advance the number of lattice points $\hat{\mathbf{s}}$ around received signal $\mathbf{r}$ it would pass through, evaluating $\mathbf{r}$ independent of the noise level, giving it a fixed throughput. The algorithm makes use of the fact that [*Barbero and Thompson*, 2008b] the diagonal entries of $\mathbf{R}$ from the $\mathbf{QR}$ decomposition of the channel matrix satisfy

$$E[\mathbf{r}_{11}^2] < E[\mathbf{r}_{22}^2] < \cdots < E[\mathbf{r}_{NN}^2] \tag{7}$$

Thus, the number of candidates at antenna level k denoted by n_k should follow

$$E[n_N] \geq E[n_{N-1}] \geq \cdots \geq E[n_1] \tag{8}$$

The main idea of FSD is to assign a fixed but distinct number of candidates to be searched per antenna level. The FSD is considered a promising algorithm for soft-MIMO detection. Since its introduction, the reduction of complexity in FSD has received significant attention [*Barbero et al.*, 2008; *Lei et al.*, 2010; *Liu et al.*, 2011; *Li et al.*, 2012; *Wu and Thompson*, 2011].

2.3. Iterative Decoding

An iterative decoder [*Hagenauer et al.*, 1996] is used right after the MIMO symbols have been detected, where soft information extrinsic log-likelihood ratio (LLR) values are exchanged iteratively between the outer decoders with interleaving/deinterleaving operations in between until the desired performance is achieved [*Berrou et al.*, 1993]. The idea behind soft detection is to generate a posteriori probability values in the form of LLR information, $L_E(b_k|\mathbf{r})$, about the interleaved bits, $\mathbf{b}$, for $1 \leq k \leq K_e$, while taking into account the channel observations $\mathbf{r}$ and the a priori LLR information, $L_A(b_k)$, coming from the outer decoder. For the system under consideration, assuming that the bits b_k are statistically independent due to the interleaving operation and making use of the Max-log approximation, $L_E(b_k|\mathbf{r})$ can be approximated by

$$L_E(b_k|\mathbf{r}) \approx \frac{1}{2}\max_{\mathbf{b}\in\mathcal{L}\cap\mathbb{B}_{k,+1}}\left(\frac{-\parallel \mathbf{r}-\mathbf{Hs}\parallel^2}{\sigma^2/2}+\mathbf{b}_{[k]}^{\mathrm{T}}\mathbf{L}_{A[k]}\right)-\frac{1}{2}\max_{\mathbf{b}\in\mathcal{L}\cap\mathbb{B}_{k,-1}}\left(\frac{-\parallel \mathbf{r}-\mathbf{Hs}\parallel^2}{\sigma^2/2}+\mathbf{b}_{[k]}^{\mathrm{T}}\mathbf{L}_{A[k]}\right) \tag{9}$$

for $1 \leq k \leq K_e$, where, without loss of generality, $K_e = M \cdot \log_2 P$ has been assumed to simplify the index notation. In (9), $\mathbf{b} = (b_1, b_2, b_3, \ldots, b_{K_e})^{\mathrm{T}}$, $\mathbf{b}_{[k]}$ denotes the subvector of $\mathbf{b}$ omitting b_k, $\mathbf{L}_A = [L_A(b_1), L_A(b_2), \ldots, L_A(b_{K_e})]^{\mathrm{T}}$, $\mathbf{L}_{A[k]}$ denotes the subvector of $\mathbf{L}_A$ omitting $L_A(b_k)$, $\mathbb{B}_{k,+1}$ and $\mathbb{B}_{k,-1}$ represent the sets of 2^{K_e-1} bit vectors $\mathbf{b}$ having $b_k = +1$ (logical 1) and $b_k = -1$ (logical 0) respectively, $\mathcal{L}\cap\mathbb{B}_{k,+1}$, and $\mathcal{L}\cap\mathbb{B}_{k,-1}$ denote the subgroups of vectors of $\mathcal{L}$ that have $b_k = +1$ and $b_k = -1$, respectively. The list of candidates $\mathcal{L} \subset \mathcal{O}^M$ is detector specific and subject to the overall performance and complexity of the iterative-MIMO receiver, since $\parallel \mathbf{r}-\mathbf{Hs}\parallel^2$ needs to be computed for all $\mathbf{s} \in \mathcal{L}$. Although iterative decoding does contribute to the overall complexity of a MIMO receiver, numerous studies have been done in reducing the total complexity of iterative decoding [*Li et al.*, 2013; *Mathana et al.*, 2013; *Wu*, 2011; *Zhang et al.*, 2009b]; therefore, this paper focuses on minimizing energy consumption in the MIMO detector. It should be noted that some of the complexity of iterative decoding will be avoided due to the proposed adaptive algorithm design; however, this is out of scope of this paper.

2.4. Power Savings

Energy consumption in mobile devices with battery-powered sources is a major limiting factor in circuit designs. Fundamentally, energy is consumed in both dynamic and static aspects as specified by (10). Most publications like *Mirsad et al.* [2011], *Andrei et al.* [2009], and *Salehi et al.* [2011] have successfully reduce the dynamic power consumption; however, in newer chip technologies, the static power consumption is said to be high [*Telikepalli*, 2006]; therefore, this work investigates ways to reduce both types, dynamic and static energy consumptions, in a circuit design, while ensuring that the algorithm performance is sufficient. This will ensure that the adaptive algorithm is properly optimized to meet power budget of the design. In

order to evaluate the overall power savings gained by the adaptive algorithm, both software and hardware savings should be analyzed:

$$E_{\text{total}} = E_{\text{dynamic}} + E_{\text{static}} + E_{I/O} + E_{\text{transceiver}} \tag{10}$$

There are multiple ways to exploit energy savings in circuit designs, and different energies have different approaches to execute these. For example, savings in E_{dynamic} are achieved by deploying the Dynamic Voltage and Frequency Scaling (DVFS) technique [*Rabaey*, 2009] while on the other hand, savings in E_{static} depend on the manufacturing process, the temperature, and the voltage, V.

2.4.1. Dynamic Energy

Dynamic energy, E_{dynamic}, spent within complementary metal-oxide-semiconductor (CMOS) technology is due to toggling of transistors and is a function of clock frequency, f, which can be varied within some limit (before the circuit fails to function due to overheating), the value of V, and the capacitance. The E_{dynamic} is given by the relation [*Abusaidi et al.*, 2008] below:

$$E_{\text{dynamic}} = \frac{nCV^2 f}{t} \tag{11}$$

where n is the number of toggling transistors, C is the circuit capacitance, V is the voltage swing, f is the toggling frequency, and t is the time it takes to complete an operation. DVFS has shown significant energy savings when applied to circuit designs, evident in *Larson and Gustafsson* [2011], *ARM Industry* [2009], and *Kim et al.* [2008]. Much like the adaptive algorithm, DVFS has the ability to adjust its parameters to match the computational demand of the current workload. If the workload requirement is high, DVFS will increase the V, to supply the circuit so that it can operate at a higher f in order to meet the desired data throughput within a particular time period. The opposite is also true; when the workload is minimal, the circuit could operate on a much lower f, which ultimately, according to (11) will decrease the overall E_{dynamic} as the task time lengthens. This adaptivity is appealing to the design of the adaptive algorithm since now both hardware and software possess the same level of adaptivity and intelligence. Both approaches will in turn yield significant overall energy savings.

2.4.2. Static Energy

Static energy, E_{static}, is consumed due to transistor leakage and is highly dependent on the manufacturing process, the ambient temperature of the circuit, and the value of V. According to the study by *Telikepalli* [2006], E_{static} seems to dominate the overall power consumption within a circuit as the chip size shrinks. Therefore, E_{static} can no longer be neglected when designing new algorithms into new chip technology.

3. Adaptive Algorithm Methodology

Current MIMO detectors usually lack adaptivity whereby all receivers behave exactly the same way regardless the received signal characteristics. This "one size fits all" architecture does not work well in some situations, since different users experience distinct channel conditions. For example, a stationary user who is physically near to a transmitter would often have a better data throughput than one who is further away. Doppler rates determined by motion in the environment also play a part in determining the current condition of the channel. To decode symbols in bad channel conditions would prove to be pointless since the data would not be likely to be decoded successfully anyway. Therefore, having intelligence in the detector that could modify its behavior according to current channel conditions would be ideal. This adaptivity in the algorithm is controlled by the MI calculation between the transmitters and receivers. It is well known that MI of a MIMO channel is given by (12) and the information required, **H** is already available within the channel estimation block. Different values of initial received soft information may lead to significantly different behavior during the iterative decoding process. The study done by *Zhang et al.* [2009a], which compares the performance of iterative decoders using different received soft LLR information metrics, discovered that by computing the MI, the number of iterations in turbo decoding can be found using the highest complexity ML MIMO detection method. *Zhang et al.* [2009a] also proves that the best approximation of the received symbols obtained are lossless and that the exact LLR values are sufficient enough statistic of **r** about **s**. Therefore, using this information and the principle of exploiting MI calculation in (12), the paper applies this approach for the first time to a MIMO detector to further save energy consumption in the overall receiver.

AGU Radio Science 10.1002/2013RS005323

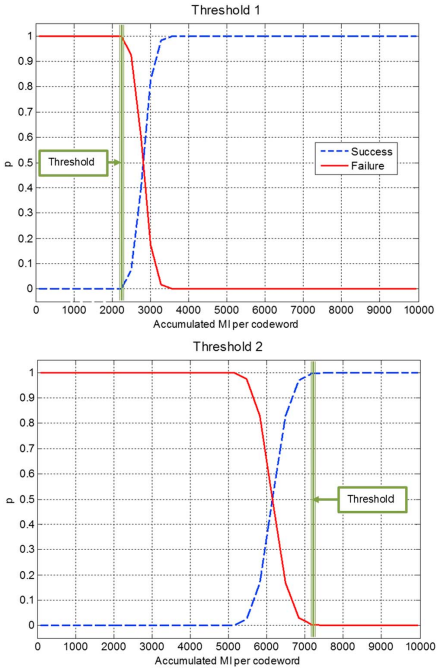

Figure 2. Probability of receiver successes and failures on 4 × 4 MIMO where (a) threshold 1 is for FSD method and (b) threshold 2 for V-BLAST/ZF method.

With any given channel model in (1), and a Gaussian constellation with $E[|\ \mathbf{s_i}\ |^2] = M^{-1}$, the MI for the ML method is

$$\bar{I}(\mathbf{H}_k) \triangleq \log_2 \det\left(I + \frac{\mathbf{H}_k\mathbf{H}_k^T}{N_0}\right) \quad (12)$$

The values of MI spread at specific SNR conditions. Figure 2 illustrates the accumulated MI performance of the detector as a function of probability of receiver fails and successes. The system is simulated using a 4 × 4 MIMO system with 16-QAM modulation symbols transmitting 1024 bits per packet of 10,000 channel realizations utilizing an iterative-MIMO decoder of $\frac{1}{2}$ code rate in a fast-fading environment. Threshold 1 can be obtained in Figure 2a, which shows the FSD performance. Below a certain MI threshold of approximately 2200, the receiver is certain to fail when trying to decode a symbol message. Therefore, the best cause of action for the receiver is to request a retransmission, i.e., Automatic Repeat Request, from the transmitter rather than to attempt decoding where it is unlikely to succeed, wasting significant computational energy, which is the limitation of today's system designs. On the other hand, the V-BLAST/ZF performance is shown in Figure 2b, where a value of about 7100 for threshold 2 can be seen. The receiver will decode the symbol message with very high probability above this MI value; therefore, a simpler detection method will suffice in detecting the symbol, i.e., the V-BLAST/ZF method. In addition, the area in-between the two thresholds shows that the receiver would sometimes fail to decode. Thus, a more powerful detection method is needed to assist the receiver in decoding the message.

Table 2. Adaptive Algorithm

Pseudo-Code
Channel realization: $\{\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H_k}\}$
for $\mathbf{r}_l \leq \mathbf{r}_k$
$\bar{I}(\mathbf{H}_k) \triangleq \log_2 \det\left(I + \frac{\mathbf{H}_k\mathbf{H}_k^T}{N_0}\right)$
if $\bar{I}_l \leq$ Threshold 1
$\mathbf{r}_l$ error, request retransmission
else if Threshold 1 $\leq \bar{I}_l \leq$ Threshold 2
$\mathbf{r}_l$ with low MI : FSD
else $\bar{I}_l \geq$ Threshold 2
$\mathbf{r}_l$ with high MI : V-BLAST/ZF
end if
end for

This is done by deploying the FSD algorithm in the MIMO detector. By obtaining these thresholds, the design of the adaptive algorithm can be described in Table 2. It should be noted that the thresholds obtained are catered specifically for 16-QAM modulation scheme on a 4 × 4 MIMO system; however, the idea behind adaptive algorithm can be adjusted to fit any communication systems. The same analysis can be applied to all other modulation and coding schemes, with the exception of having different threshold values when calculated using (12).

AGU Radio Science 10.1002/2013RS005323

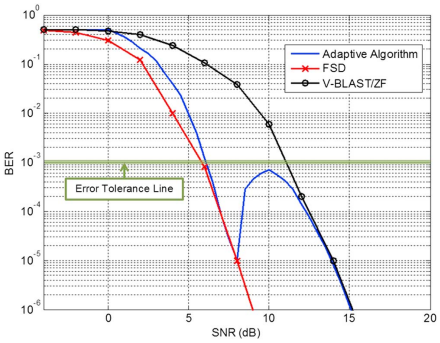

Figure 3. Performance measurement of BER on complex 4 × 4 MIMO system.

4. Results and Analysis

The effectiveness of the adaptive algorithm can be measured using the performance and complexity tradeoff metrics. This section describes these efficiencies from both hardware and software perspectives.

4.1. SOFTWARE: Performance

The performance can be quantified by calculating the number of errors in a total frame, i.e., the BER analysis. The system design has been set to tolerate a BER of 10^{-3} or less in high-SNR regions. In the system model used, the BER is depicted in Figure 3. The adaptive algorithm gives similar performance to the FSD and performs much better than the V-BLAST/ZF algorithm in low-SNR regions. In very high SNRs, i.e., 10 dB and above, the less complex algorithm of V-BLAST/ZF is adopted and the BER performance is below the set error tolerance line. The FSD does give a much better performance than the tolerance line; however, this level of performance is unnecessary and only adds extra complexity for the hardware. When the SNR is below 0 dB, the receiver abandons the detection process (subsequently avoiding the complexity of the iterative decoding process as well, gaining substantial power savings) and requests a retransmission from the transmitter, whereas the area above the set threshold, approximately 0 dB to 6 dB, the adaptive algorithm provides much higher chances of successful processing in comparison to the V-BLAST/ZF method.

4.2. SOFTWARE: Complexity

By obtaining the thresholds, the total number of usage of each MIMO detection algorithm throughout the span of the SNR is shown in Figure 4, depicting transmissions of 10,000 packets of 1024 bits per frame. It clearly shows that below an SNR value of 0 dB, i.e., threshold 1, no processing is taking place. In addition, in high-SNR regions, V-BLAST/ZF is utilized. This figure concurs with Figure 3, where the performance coincides with the algorithm switching rate of successfulness. From this, another part of the parameter, the complexity measurement of the software can be determined.

Complexity measurement gives an important overview of the hardware before implementation and provides an initial indication of power savings in the design. A preliminary complexity analysis of the adaptive algorithm is determined by the multiplier counts in the code. Assuming that the complexity of channel ordering is the same for both detection schemes, the multiplier counts between the FSD and V-BLAST/ZF detection schemes for a transmission of one symbol for 4 × 4 *M*-QAM deploying FSD is *M*-times more complex than the V-BLAST/ZF. Figure 5 plots the percentage complexity results against the SNR of the channels, where 100% equals the complexity of FSD, while the V-BLAST/ZF requires only 25%. The complexity of the adaptive algorithm can be calculated by averaging over MI values shown at certain SNR in the figure, and it is much lower than the FSD, i.e., 62% of the multipliers required. Most energy

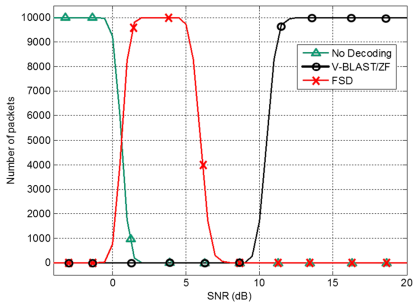

Figure 4. Algorithm switching selection in receiver.

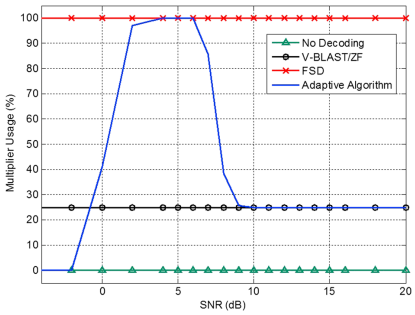

Figure 5. Complexity measurements of multiplier counts between different MIMO detection schemes.

savings can be gained during the "No Decoding" phase since no processing is required in this region. Furthermore, energy are saved during the utilization of V-BLAST/ZF algorithm, i.e., where MI > 7100, which gives a total of only 25% multiplier usage.

4.3. HARDWARE: Performance and Complexity

Xilinx®Virtex-5 has a varying voltage range of 0.95 V to 1.05 V and an operational frequency range of 60 MHz to 400 MHz [*Klein*, 2009]. In order to assess the efficacy of the DVFS technique in saving energy consumption in wireless communication, both MIMO detection algorithms, FSD and V-BLAST/ZF, are operated at low-power mode (0.95 V, 60 MHz) and high-performance mode (1.05 V, 400 MHz) to get the minimum and maximum thresholds of operation. This information is determined using the Xilinx®Design Suite software for the Xilinx®Virtex-5. The Xilinx®Design Suite software comprises a codesign software/hardware setup performed in MATLAB™ and Xilinx®System Generator, which is a part of the Xilinx®ISE. In addition, the power profile is analyzed using the Xilinx®Power Estimator tool. The summary of the total number of the FPGA resources used are given in Table 3. The percentage of slices used can be seen as an indicator of the amount of control logic and intermediate buffers required in the adaptive algorithm. This factor affects hardware mapping and the resulting throughput. The average throughput of the system is a parameter of importance when considering the performance of the algorithm. The throughput in megabits per second (Mbps) is calculated according to

$$Q_{\text{avg}} = M \cdot \log_2 P \cdot f / C_{\text{avg}} \tag{13}$$

where C_{avg} is the average number of clock cycles required to detect a MIMO symbol.

For low-power mode, where f = 60 MHz and the minimum number of cycles is C_{min} = 4, the maximum throughput is Q_{min} = 240 Mbps while the high-performance mode gives a throughput of Q_{max} = 1200 Mbps. Increasing the clock frequency would result in a significant increase in the throughput; therefore, the $f = C_{\text{avg}}$ could be seen as an indicator of the level of optimization of the hardware design.The hardware setup parameters are included in Table 4.

Similar to details reported in *Mirsad et al.* [2011], *Andrei et al.* [2009], *Salehi et al.* [2011], and *Larson and Gustafsson* [2011] there are significant dynamic power savings in the circuit, portrayed in Figure 6, where low-power mode uses 9% of the overall power in comparison to 29% when the circuit is run at full power, i.e., the high-performance mode. However, these savings would be minimal in comparison due to the much larger static power, which dominates the overall chip power. Figure 7 shows the low-power results for FSD (a) and V-BLAST/ZF (c) as well as the high-performance statistics, (b) and (d), for FSD and V-BLAST/ZF, respectively. It is shown that some savings are gained when the adaptive algorithm switches from the high-complexity FSD to the simpler V-BLAST/ZF detection. The power saved during the swap is equivalent

Table 3. Virtex-5 Resource Utilization of Adaptive Algorithm

Logic Resource Utilization	Used	Available	Utilization
Slice Registers	13,683	149,760	9%
Flip Flops	4,688	37,440	12%
4-Input LUTs[a]	12,161	149,760	8%
DSP48E	132	1,056	12%
Memory (RAM[b])	28	516	5%

[a]Look-up tables.
[b]Random access memory.

Table 4. Experiment Parameters of Adaptive Algorithm

Virtex 5: XC5VLX330TFF1738		
MIMO Setup 4 × 4	Modulation Scheme 16-QAM	Bit Frame Size 1024 Bits
Operation Mode Parameters	Low Power	High Performers
Core Voltage	0.95 V	1.05 V
Clock Frequency	60 MHz	400 MHz
Max Throughput	240 Mbps	1200 Mbps

to 20% for high performance and 8% for low-power mode. The energy savings when changing from high performance to low power are also illustrated here. The total time computed is obtained by transmitting one packet of 1024 bit frame using a 16-QAM modulation symbol over the 4 × 4 MIMO channel when operating at the lowest frequency of 60 MHz. When operating at 400 MHz, the task completion time takes approximately 7 times less than when operating at lower frequency. By finishing quickly, the hardware can be put into sleep mode, reducing the total energy, since the idle power is negligible ≈ 0.08 mW. By calculation, at the same total rate of completion, the energy required to complete one task is lower by 42% when the circuit operates quickly and switches into idle state (high performance) than to run slowly and finishes just in time, at lower frequency (low power) when deploying FSD, and 52% for the V-BLAST/ZF algorithm. These are the savings which can be gained when putting the chip into sleep mode for more than 15 μs. Even though in theory, verified in (11), the longer the task runs, the lower the dynamic energy consumption, this is not the case here because when evaluating the total energy consumption of the circuit, the E_{static} required in powering up the Xilinx®Virtex-5 hardware is too large, occupying most of the power demand of the chip, resulting in 84% and 65% of the total power for low-power and high-performance mode, respectively, as shown in Figure 6. These findings coincide with the work reported in *Hasan and Bird* [2011], stating that as manufacturing process get smaller, the E_{static} seems to dominate the overall chip power. Therefore, it can be concluded that running the circuit at a lower speed is not the answer to overall power savings in this technology. E_{static} could no longer be neglected when designing a circuit, and it is now more essential to take temperature as a parameter in saving overall energy consumption, since E_{static} strongly depends on the heat generated by the circuit.

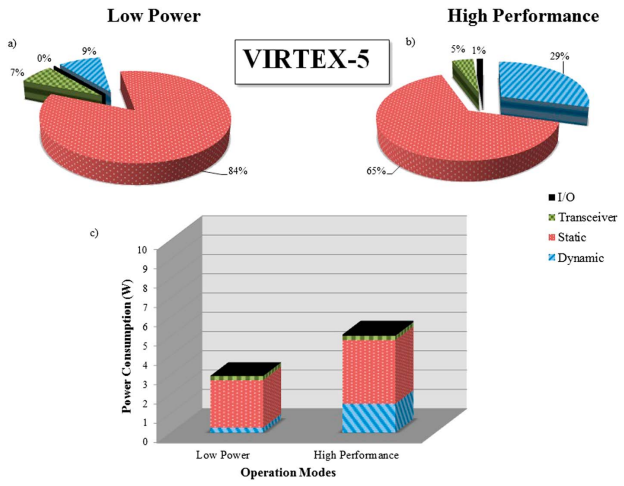

Figure 6. Total power usage in Xilinx®Virtex-5 hardware apparatus.

AGU Radio Science 10.1002/2013RS005323

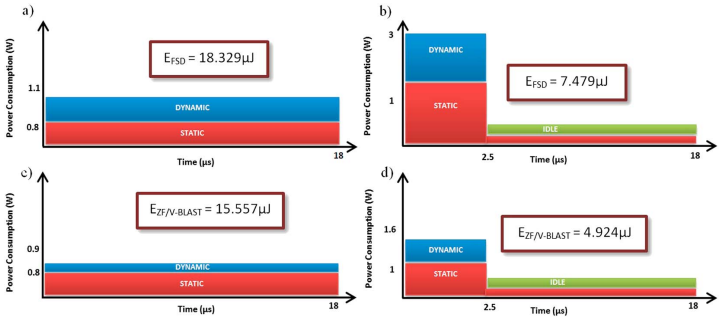

Figure 7. MIMO detection (a) FSD and (b) in comparison with (c and d) V-BLAST/ZF for low-power mode and high-performance mode, respectively.

Figure 8 shows the overview of the algorithm flow within the chip. Only one detector is switched on at any given time according to the calculation from the threshold control block. This is particularly useful for FPGA implementation since the hardware resources are switched on and off as required. The implementation of the adaptive algorithm is illustrated in terms of the FPGA hardware given in Figure 9. The configurable logic utilized for each detector is shown in (a) for FSD, (b) for V-BLAST/ZF, and (c) when "No Decoding" is taken place. It can be seen that only certain parts of the overall chip hardware are turned on at any given time. Seeing that most power consumption is due to powering the up the chip itself, i.e., static power, the adaptive algorithm takes advantage of this fact and therefore shuts down parts of the chip which are not in use. To show how the adaptive algorithm behaves, consider four extreme scenarios of three frames of 1024 data bits per frame size being transmitted over different environments, where T_1 is when the MI is at a high value, T_2 is for when MI is acceptable, and T_3 is for MI being low and not suitable for further decoding. From Figure 5, it is shown that the complexity of an FSD is 4 times larger than that of the V-BLAST/ZF. Therefore, if the complexity of the V-BLAST/ZF is set to 1, the FSD will have the equivalent complexity of 4. The overall chip area usage is given in Figure 10. Using the same complexity ratio, consider a transmission of 100,000 frames of 1024 bits per frame on random fast-fading channel realizations over various ranges of SNR values from −4 dB to 20 dB. The adaptive algorithm saves approximately 30% of the overall resource in comparison to the FSD detector while maintaining the BER performance at a satisfactory region.

Shutting down parts of the chips, i.e., sleep modes, is the key enabler in saving further energy in the design on Virtex-5 hardware. By running the circuit at high frequency, the sleep modes can help prevent the circuit

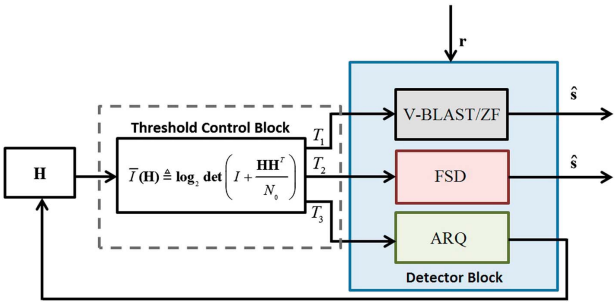

Figure 8. Simple adaptive algorithm implementation model.

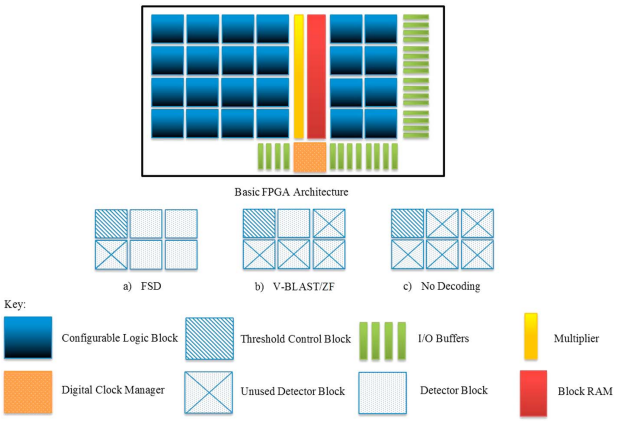

Figure 9. Total resource allocation of adaptive algorithm on a basic FPGA architecture.

from running and powering up the entire logic gates all the time, consequently preventing the circuitry from overheating that leads to high-E_{static} consumption.

For greater insight of the total energy savings that can be achieved in a realistic setting, Figure 11 considers the adaptive algorithm in a Rayleigh fading channel. The SNR chosen are based on the operating SNR regions of the Long-Term Evolution. In small cells, the transmit power is to be in the range of 23 dB to 46 dB, averaging at 26.5 dB [*Nakamura*, 2013]. The savings can be found by integrating the power, *P*, with respect

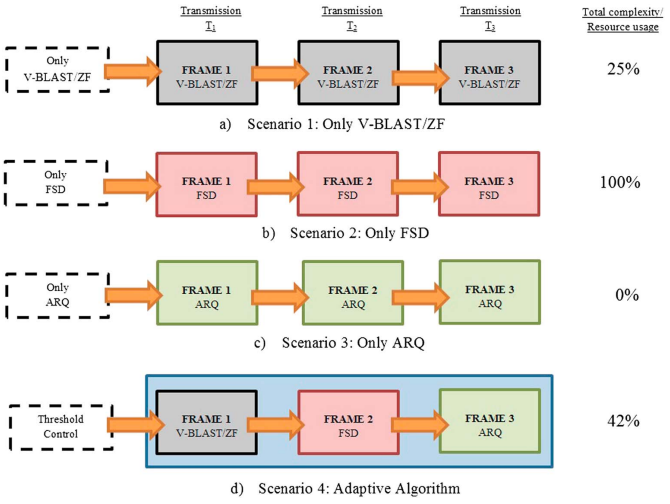

Figure 10. Basic overview of the inner workings of the adaptive algorithm.

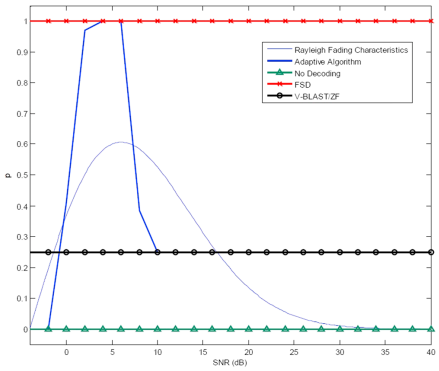

Figure 11. Behaviors of different detection algorithms in a Rayleigh fading channel.

to the probability density function, f, of the fading environment, ρ, as shown in (14):

$$\int_a^b P(\rho) f(\rho)\, \mathrm{d}\rho \tag{14}$$

where a is the lower SNR value of −4 dB and b is the upper limit of the SNR, which is 40 dB in this case. Using a discrete approximation to this gives a measure of the savings that can be achieved in practice. For example, taking the FSD as a benchmark would use 8 J (in high performance) of energy to decode the 1024 bits data packet size. Utilizing the adaptive algorithm would use 70% less resources since the FSD does not take into account the transmit power nor the SNR values, which results in unnecessary power wastage. In addition, the behavior of the adaptive algorithm follows that of the Rayleigh fading channel for a 4 × 4 MIMO system, operating on 74% of the fading channel environment, gaining energy savings due to sleep implemented in the appropriate regions; i.e., FSD is on sleep mode at SNR of 20 dB, and only V-BLAST is active.

The energy saving results obtained can be optimized further by combining the common circuitry of the FSD and V-BLAST since they share some common functionality. By sharing the circuitry resources between the two algorithms can gain additional energy savings. Detailed evaluation of the issues is the next major step of the project.

5. Future Direction

Research is still ongoing in the field of both hardware and software designs. This section describes some of the planned future work.

5.1. SOFTWARE: Algorithm Switching Selection

The SNR values at which the adaptive algorithm switches between the different thresholds is illustrated in Figure 4. The selection of adaptive algorithm can be optimized. At a particular SNR, the MI varies, and must be calculated by the receiver. The effect is that the detector switches between approaches in regions corresponding to the MI thresholds. The transitions across the MI thresholds result in switching from one to the other rapidly. This switching could have an impact on the power consumption. One possible improvement is to enforce use of FSD during these situations when V-BLAST/ZF fails to decode a packet, or when there would be rapid switching between FSD and "'No Decoding." However, even though this would increase the likelihood of decoding, it would be at a cost of higher-energy consumption.

5.2. HARDWARE: New Xilinx® Virtex 7

Newer technology chips such as the Xilinx®Virtex-7, based on a different manufacturing process, have an improved solution to the high-E_{static} consumption of previous circuit technologies [*Hussein et al.*, 2013]. It may therefore be that DVFS can be applied to minimize power consumption in this type of hardware, due to E_{static} no longer dominating the total chip power.

6. Conclusion

Having intelligence in the algorithm design and the hardware offers both adequate performance and reduced complexity in future iterative-MIMO systems. The adaptive algorithm within the MIMO receiver demonstrates significant energy savings in both software and hardware implementation. It has the potential to save up to 30% energy in the software design and in the Xilinx®Virtex-5 hardware. This can be improved further when incorporating sleep modes to reduce the E_{static} in the hardware apparatus.

AGU Radio Science 10.1002/2013RS005323

Acknowledgment
This work is funded by the University of Tun Hussein Onn Malaysia as a part of the main author's PhD program.

References

Abusaidi, B. P., M. Klein, and B. Philofsky, (2008), Virtex-5 FPGA system power design considerations, *Tech. Rep. WP 285 (v1.0)*, pp. 1–23, Xilinx Inc., San Jose, Calif.
Agrell, E., T. Eriksson, A. Vardy, and K. Zeger (2002), Closest point search in lattices, *IEEE Trans. Inf. Theory*, *48*(8), 2201–2214.
Andrei, A., P. Eles, Z. Peng, S. Link, M. Schmitz, and B. M. Al-Hashimi (2009), Energy optimization of multiprocessor systems on chip by voltage selection, in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, pp. 262–275, IEEE Educational Activities Department, Piscataway, N. J.
Ariyavisitakul, S. L. (2000), Turbo space-time processing to improve wireless channel capacity, *IEEE Trans. Commun.*, *48*(8), 1347–1359.
ARM Industry (2009), The ARM Cortex-A9 processors, White Paper, pp. 1–11, ARM Holdings, U. K.
Barbero, L. G., and J. S. Thompson (2008a), Extending a fixed-complexity sphere decoder to obtain likelihood information for Turbo-MIMO systems, *IEEE Trans. Vehicular Technol.*, *57*(5), 2804–2810.
Barbero, L. G., and J. S. Thompson (2008b), Fixing the complexity of the sphere decoder for MIMO detection, *IEEE Trans. Wireless Commun.*, *7*(6), 2131–2142.
Barbero, L. G., T. Ratnarajah, and C. Cowan (2008), A low-complexity soft-MIMO detector based on the fixed-complexity sphere decoder, in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2669–2672, IEEE, Las Vegas, Nev.
Berrou, C., A. Glavieux, and P. Thitimajshima (1993), Near Shannon limit error - Correcting coding and decoding: Turbo codes, in *IEEE International Conference on Communications*, vol. 2, pp. 1064–1070, IEEE, Geneva, Switzerland.
Chen, Y. K., C. Chakrabarti, and B. Bougard (2010), Signal processing on platforms with multiple cores: Part 2 – Applications and design, *IEEE Signal Process. Mag.*, *2*(1), 20–21.
Esmaeilzadeh, H., E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger (2011), Dark silicon and the end of multicore scaling, in *Proceeding of the 38th Annual International Symposium on Computer Architecture*, pp. 365–376, ACM, New York.
Fincke, B. U., and M. Pohst (1985), Improved methods for calculating vectors of short length in a lattice, including a complexity analysis, *Math. Comput.*, *44*(170), 463–471.
Golden, G. D., C. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky (1999), Detection algorithm and initial laboratory results using V-BLAST space time communication architecture, *IEEE Electron. Lett.*, *35*(1), 14–16.
Goldsmith, A., E. Biglieri, R. Calderbank, A. Constantinides, A. Paulraj, and H. V. Poor (2007), *MIMO Wireless Communications*, pp. 1–559, Cambridge Univ. Press, Cambridge, U. K.
Golub, G. H., and C. F. Van Loan (1983), *Matrix Computations*, 3rd ed., 476 pp., The Johns Hopkins Univ. Press, Baltimore, Md.
Hagenauer, J, E. Offer, and L. Papke (1996), Iterative decoding of binary block and convolutional codes, *IEEE Trans. Inf. Theory*, *42*(2), 429–445.
Hasan, M. Z., and M. Bird (2011), Energy reductions for embedded processors in reconfigurable hardware, in *IEEE International Conference on Electro/Information Techonology*, pp. 1–8, IEEE, Mankato, Minn.
Hochwald, B. M., and S. T. Brink (2003), Achieving near-capacity on a multiple-antenna channel, *IEEE Trans. Commun.*, *51*(3), 389–399.
Hussein, B. J., M. Klein, and M. Hart (2013), *Lowering Power at 28 nm With Xilinx® 7 Series Devices*, *389*, 1–25.
Kim, W., M. S. Gupta, G. Wei, and D. Brooks (2008), System level analysis of fast, per-core DVFS using on-chip switching regulators, in *IEEE 14th International Symposium on High Performance Computer Architecture*, pp. 123–134, IEEE, Salt Lake City, Utah.
Klein, M. (2009), Power consumption at 40 and 45 nm, White Paper, vol. 298, pp. 1–21, Xilinx Inc., San José, Calif.
Kumar, R,, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen (2003), Single-ISA heterogeneous multi-core architectures: The potential for processor power reduction, in *Proceedings of the 36th International Symposium on Microarchitecture*, pp. 81, IEEE Computer Society, Washington, D. C.
Larson, E. G., and O. Gustafsson (2011), The impact of dynamic voltage and frequency scaling on multicore dsp algorithm design the impact of dynamic voltage and frequency scaling on multicore DSP algorithm design, *IEEE Signal Process. Mag.*, *28*, 127–144.
Lei, S., Q. Tu, D. Yang, and J. Chen (2010), Probabilistic tree pruning for fixed-complexity sphere decoder in MIMO systems, in *International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, IEEE, Suzhou, China.
Li, G., X. Zhang, S. Lei, C. Xiong, and D. Yang (2012), An early termination-based improved algorithm for fixed-complexity sphere decoder, in *IEEE Wireless Communications and Networking Conference: PHY and Fundamentals*, vol. 1, pp. 629–634, IEEE, Shanghai, China.
Li, L., R. G. Maunder, B. Al-Hashimi, and L. Hanzo (2013), A low-complexity turbo decoder architecture for energy-efficient wireless sensor networks, *IEEE Trans. Very Large Scale Integr.*, *21*(1), 14–22.
Li, P,, D. Paul, R. Narasimhan, and J. Cioffi (2006), On the distribution of SINR for the MMSE MIMO receiver and performance analysis, *IEEE Trans. Inf. Theory*, *52*(1), 271–286.
Liu, L., J. Lofgren, and P. Nilsson (2011), Low complexity soft-output signal detector for spatial-multiplexing MIMO system, in *IEEE International Wireless Communications and Mobile Computing Conference*, pp. 988–993, IEEE, Toronto, Canada.
Matthaiou, M., D. I. Laurenson, and C. X. Wang (2008), Reduced complexity detection for ricean mimo channels based on channel number thresholding, in *22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1718–1722, IEEE, Crete, Greece.
Mathana, J. M., P. Rangarajan, and J. Raja Paul Perinbam (2013), Low complexity reconfigurable turbo decoder for wireless communication systems, *Arabian J. Sci. Eng.*, *38*(10), 2649–2662.
Mirsad, C., D. Persson, and E. G. Larson (2011), Allocation of computational resources for soft MIMO detection, *IEEE J. Sel. Top. Signal Process.*, *5*(8), 1451–1461.
Nakamura, T. (2013), Trends in small cell enhancements in LTE advanced, *IEEE Commun. Mag.*, *51*(2), 98–105.
Pohst, M. (1981), On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications, *Newslett. ACM SIGSAM Bull.*, *15*(1), 37–44.
Rabaey, J. (2009), Low power design essentials, pp. 289–316.
Salehi, M. E., M. Samadi, M. Najibi, A. Afzali-Kusha, M. Pedram, and S. M. Fakhraie (2011), Dynamic voltage and frequency scheduling for embedded processors considering power/performance tradeoffs, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, *19*(10), 1931–1935.
Sellathurai, M., and S. Haykin (2002), TURBO-BLAST for wireless communications: Theory and experiments, *IEEE Trans. Signal Process.*, *50*(10), 2538–2546.
Telikepalli, A. (2006), Power vs. performance: The 90 nm inflection point reducing power in FPGAs–The triple challenge, White Paper, vol. 223, pp. 1–18, Xilinx Inc., San José, Calif.
Viterbo, E., and J. Boutros (1999), A universal lattice code decoder for fading channels, *IEEE Trans. Inf. Theory*, *45*(5), 1639–1642.
Winters, J. H., S. Member, J. Salz, and R. D. Gitlin (1994), The impact of antenna diversity on the capacity of wireless communication systems, *IEEE Trans. Commun.*, *42*(2), 1740–1751.

Wu, P. H.-Y. (2011), On the complexity of turbo decoding algorithms, in *Proceedings of IEEE Vehicular Technology Conference*, vol. 2, pp. 1439–1443, IEEE, Rhodes, Greece.
Wu, X., and J. S. Thompson (2011), A fixed-complexity soft-MIMO detector via parallel candidate adding scheme and its FPGA implementation, *IEEE Commun. Lett.*, *15*(2), 241–243.
Zhang, J., M. A. Armand, P. Y. Kam, and A. T. Mi (2009a), A mutual information approach for comparing LLR metrics for iterative decoders, *IEEE Commun. Soc.*, *1*(4), 1–4.
Zhang, J., M. A. Armand, P. Y. Kam, and A. T. Mi (2009b), Low hardware complexity parallel turbo decoder architecture, *Int. Symp. Proc. Circuits Syst.*, *1*(4), 1–4.

IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, VOL. ??, NO. ?, MAY 2015

Practical Performance of an Adaptive Switching Algorithm for Iterative-MIMO Receivers in Spatially Correlated Channels

Nina Tadza, *Student,* John S Thompson, *Professor,* and David I Laurenson, *Senior Lecturer*

***Abstract*—This paper investigates the applicability of a novel adaptive algorithm, dubbed the Adaptive Switching Algorithm, for iterative-multiple-input multiple-output (MIMO) detection in realistic channel conditions. The thresholds in the receiver, that control the adaptivity, provide various settings for the detector and decoder operation. These thresholds work according to the same calculated mutual information between the transmitters and receivers in real-time. The detector threshold determines whether the receiver would decode using a high performance detector, a low complexity detector or simply abandon further processing and reduce energy consumption by requesting a re-transmission. The threshold also works as a decoder stopping criterion, where it determines the number of decoding iterations necessary for a transmission. This paper provides the performance analysis for the proposed algorithm in realistic conditions by providing a detailed energy analysis of the algorithm for spatially correlated channel conditions. Analytical, simulation and implementation results show that the practical behavior of the proposed iterative-MIMO receiver in detection and decoding saves significant energy with a tolerable bit error rate performance degradation.**

***Index Terms*—turbo decoding, stopping criteria, energy savings, iterative-MIMO, mutual information, adaptive switching algorithm**

I. Introduction

To meet the explosive growth in data rates currently caused by mobile devices such as smart phones and portable handheld multimedia devices, as well as data terminals such as wireless hotspots, femtocells and base stations, the technology of utilizing multiple antennas on both sides of the transmitter and receiver is imperative. Theoretical analysis has shown promising capacity growth by employing the multiple-input multiple output (MIMO) scheme [1] [2], which helps in increasing the spatial diversity and capacity of the system. However, the presence of spatial correlation between the multiple antennas reduces the capacity improvement [3]. Studies have evaluated the behavior of detectors in such spatially correlated channel environments, for both low complexity linear MIMO detectors [4] [5] and high performance tree search detectors [6]. Generally, it is found that the bit-error-rate (BER) degrades as the channel gets more correlated. Studies are lacking however, for adaptive iterative-MIMO detection as well as for a full receiver setup that includes iterative decoding in such channel conditions. Moreover, to the best of the authors' knowledge, the energy analysis of adaptive algorithm implementations is also sparse in the literature. There are many adaptive detection algorithms proposed [7] [8] [9] [10], however, in addition to them using different switching criteria that does not fully exploit the available information regarding the MIMO channel setup [11] [12] [13] to provide the adaptivity, none of these papers considers the performance of such algorithms in spatially correlated channels or the energy savings potential for realistic hardware implementations. Most publications focus on increasing throughput [7] [8] or the overall performance [9] [10] or provide generic energy saving results that are not specified to the latest state-of-the-art communication systems [11] [14] [15] [16]. A recently proposed Adaptive Switching Algorithm detector can achieve energy savings of about 38% in the algorithmic design [17], and approximately 80% during hardware design implementation [18] in experimentally controlled additive white Gaussian noise (AWGN) channel conditions. This paper attempts to extend the findings of [18] by investigating the efficiency of the proposed algorithm usage in the detector in a realistic environment. In practice, the channels between different antennas are correlated and therefore the full multiantenna gains may not always be obtainable. Therefore, the work investigates the utilization of the Adaptive Switching Algorithm on simulated spatially correlated channels, whereby the information between the antennas, which is the mutual information (MI) is not optimal.

In addition to the energy saving analysis of the detector in such channel conditions, this work explores the total iterative-MIMO receiver design, which includes the iterative turbo decoding that guarantees higher data rate support, and better performance in comparison to non-iterative systems [19]. The outstanding performance of the turbo decoder comes with a high price of computational complexity. To combat this, a number of early termination techniques or stopping criteria rules provided for the decoder iterations have been proposed in order to minimize the complexity of the decoder by reducing the number of iterations whilst maintaining the performance of the entire system. These criteria can be categorized into two groups, which are soft-bit decisions and hard-bit decisions. Soft-bit decisions, which are considered in this paper, such as Cross-Entropy (CE) [20] *A-Priori* Log Likelihood Ratio (LLR) Measurement [21], and Mean-Estimation (ME) [22] Updated Threshold [23] are important methods. The most well-known CE stopping rule [22] works by using relative information between the two constituent decoders' soft output

N. Tadza is with the Institute for Digital Communications, School of Engineering, The University of Edinburgh, EH9 3JL, Edinburgh, UK email : (n.tadza@ed.ac.uk)

J. S. Thompson and D. I. Laurenson are with The University of Edinburgh.

Manuscript received May 1, 2015; revised May 1, 2015.

as the criteria. Decoding stops, or converges, when the relative information is close to zero. Using the same concept as [22], different simplified versions are proposed in [23], where the LLR are used instead to compute the relative soft information values. These concepts assist in lowering the complexity of the decoding process by minimizing the number of decoding iterations. Therefore, this trade-off of complexity and energy savings gained in both detector and iterative-decoding in spatially correlated channels are made and justified for realistic design implementations for the Adaptive Switching Algorithm receivers.

The main contributions of this paper are summarized as follows:

- The proposed adaptive algorithm is found to control both the detector; to choose the appropriate detection method, and iterative decoder; as a stopping criteria tool to help determine and thus minimize the number of decoding iterations needed per transmission.
- Energy analysis and hardware design implementation for the Adaptive Switching Algorithm saves energy whilst maintaining the performance of the receiver in spatially correlated channels with only a slight increase in hardware utilization complexity, and higher signal-to-noise ratio (SNR).

The rest of this paper is organized as follows. Section II describes the MIMO channel model under consideration by explaining a brief background information on spatially correlated channels; Section III gives a detailed description of the novel Adaptive Switching Algorithm and how each algorithm involved in both detector and decoder is being implemented on the hardware of choice; Section IV summarizes the analysis and results and lastly, the paper is concluded in Section V.

II. Spatially Correlated MIMO Channel Model

In order to verify the effectiveness of the Adaptive Switching Algorithm in realistic conditions, spatially correlated MIMO channels are chosen as a reasonable model for providing simulated environments mimicking heavily built-up urban transmission settings for radio signals [25] [26]. Based on a flat fading standard MIMO model [24], with M transmitters and N receivers where $M \leq N$, the channel setup considered in this paper utilizes the Kronecker model, where the correlation between the transmitters and receivers are assumed to be independent and separable. This model is reasonable when there is a lot of signal scattering that occurs close to the transmitting and receiving antenna arrays. The results of this model has been validated by both outdoor and indoor measurements [27] [29]. In this case, with Rayleigh fading, the channel matrix can be factorized as in equation (1).

$$\mathbf{H} = \mathbf{R}_{Rx}^{1/2}\mathbf{H}_w(\mathbf{R}_{Tx}^{1/2})^T \tag{1}$$

The antenna correlation observed at the receiver is assumed to be the same for all transmitters, and similarly, the correlation for the transmitters is also the same on all receivers. The elements of $\mathbf{H}_w$ are independent and identically distributed (i.i.d) as circular symmetric complex Gaussian with zero mean, μ, and unit variance, σ with $\mathbf{vec}(\mathbf{H}) \sim \mathcal{CN}(\mathbf{0}, \mathbf{1})$ representing the MIMO uncorrelated channel. The $N \times N$ matrix $\mathbf{R}_{Tx}$ describes the fading correlation for the transmitter array while the $M \times M$ matrix $\mathbf{R}_{Rx}$ described the received spatial correlation. The statistical behavior of the channel matrix can also be expressed as in equation (2), where the $\mathbf{vec}(\cdot)$ denotes the vec operator and $\otimes$ the Kronecker product [27].

$$\mathbf{vec}(\mathbf{H}) \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_{Tx} \otimes \mathbf{R}_{Rx}) \tag{2}$$

The spatial correlation depends directly on the eigenvalue distribution of the correlation matrices, $\mathbf{R}_{Tx}$ and $\mathbf{R}_{Rx}$. Each eigenvector represents a spatial direction of the channel and the corresponding eigenvalue describes the average channel and signal gain in a specified direction. High spatial correlation indicated by a large eigenvalue spread in $\mathbf{R}_{Tx}$ or $\mathbf{R}_{Rx}$ means that some spatial directions are statistically stronger than others. Low spatial correlation on the other hand, is represented by a small eigenvalue spread in $\mathbf{R}_{Tx}$ or $\mathbf{R}_{Rx}$, meaning that almost the same signal power can be expected from all spatial directions. The higher the spatial correlation, the more impact it has on the performance of a given MIMO system [28]. The capacity of the channel is always degraded by the receiver side of spatial correlation as it decreases the number of (strong) spatial directions that the signal is received.

The correlation model considered in this paper can be calculated mathematically with respect to capacity, using generic definitions for the transmitter,

$$\mathbf{R}_{Tx} = \begin{pmatrix} 1 & \mathrm{C_{Tx}} & \dots & \mathrm{C}_{Tx}^{N-1} \\ \mathrm{C}_{Tx} & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \mathrm{C}_{Tx}^{2} \\ \mathrm{C}_{Tx}^{N-1} & \dots & \mathrm{C}_{Tx}^{2} & 1 \end{pmatrix} \tag{3}$$

and receiver correlations.

$$\mathbf{R}_{Rx} = \begin{pmatrix} 1 & \mathrm{C}_{Rx} & \dots & \mathrm{C}_{Rx}^{N-1} \\ \mathrm{C}_{Rx} & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \mathrm{C}_{Rx}^{2} \\ \mathrm{C}_{Rx}^{N-1} & \dots & \mathrm{C}_{Rx}^{2} & 1 \end{pmatrix} \tag{4}$$

where C_{Tx} and C_{Rx} represents real-valued correlation coefficients. The correlation indexes considered are further simplified to give $\mathbf{R}_{Tx} = \mathbf{R}_{Rx} = \mathbf{C}$, yielding a single factor parameter. This means that the system considers the same correlation is present at both transmitter and receiver sides. The given model can range from the uncorrelated case i.e. $\mathbf{C} = 0$ to the fully correlated scenario of $\mathbf{C} = 1$.

Two points should be understood concerning the use of this model in the paper. First, while the channel model does represent close to realistic channel conditions, the results give pessimistic performance predictions for highly correlated fading scenarios where the model assumptions described above are no longer valid [30]. Secondly, though the correlation values between the transmitters and receivers are unlikely to be equal, this assumption is made to give an overall idea of the applicability of the Adaptive Switching Algorithm to spatial correlated channels.

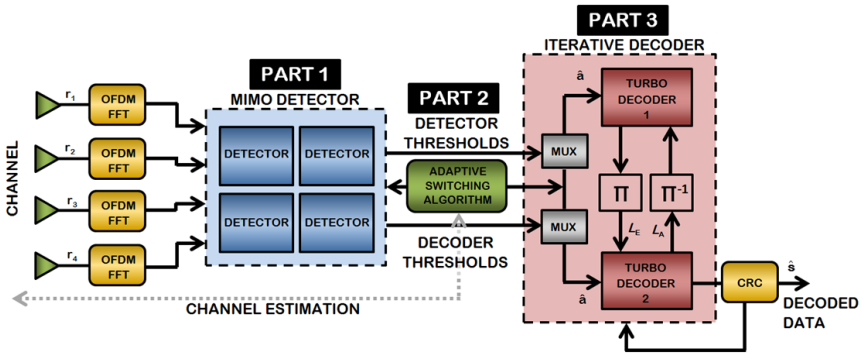

Fig. 1. Iterative-MIMO Receiver System Under Consideration.

III. Adaptive Switching Algorithm Description

The block diagram for the MIMO receiver under consideration is shown in Fig. 1. Generally, a typical iterative-MIMO receiver comprises two parts, the MIMO detector, and the iterative turbo decoder, where $\mathbf{r}$ is a series of received symbols from the transmitter, and $\hat{\mathbf{s}}$ is the estimated bit vectors for the transmitted data, when receiver processing is complete. The Adaptive Switching Algorithm detector first selects the appropriate detection algorithm depending on the MI calculated between the transmitters and receivers in real-time. The detection results are passed onto the next part of the receiver, which is the iterative-turbo decoder with a specified number of decoding iterations.

From the authors' work in [17], it can be seen that the Adaptive Switching Algorithm comprises selecting between by two well-known detection algorithms, namely the Fixed Sphere Decoder (FSD) [31] and the Vertical Bell Laboratory Layered Space Time with Zero Forcing (V-BLAST/ZF) [32] detection algorithms according to the BER performance of the system. Switching between the two algorithms is determined by thresholds pre-calculated from the MI between the transmitter and the receiver, according to the real-time channel conditions of each data transmission. The algorithm design has been shown to achieve 38% reduction in computational complexity in the detector [17], thus this work investigates if more power and energy savings can be accomplished for realistic channel conditions. It also extends the study considering the power-hungry iterative-decoder block in the receiver system.

In order to explore this, the experiments for the proposed work uses a software/hardware setup performed in Matlab™ and its built-in Simulink® package as well as Xilinx® System Generator to compile into a field programmable gate arrays (FPGA). The transmission setup comprises $M = 4$ transmitters and $N = 4$ receivers, based on a bit-interleaved coded modulation (BICM) setup, which has a transmit frame size of $K_u = 1,024$ bits for transmission over a random independent fast fading propagation channel, $\mathbf{H}$, and it is constant over a frame and changes independently from frame to frame following the Kronecker model, which is perfectly known at the receiver. The transmitted bits, K_u, are encoded using an iterative-turbo scheme at rate of $\varphi = 1/2$, which are then interleaved randomly to give, K_a coded bits, before mapping into a quadrature amplitude modulation (QAM) constellation, $\mathcal{O}$, of size $W = 16$ points, forming a sequence of $K_s = K_e/\log_2 W$ symbols. This gives $K_s = 512$ symbols, which are divided equally between the transmitters for $100,000$ channel realizations. This part of the transmitter system is simulated purely using Matlab™.

The work focuses on the receiver, which is consequently divided into the theoretical software experimentation and the hardware design implementation. On the software side, the Adaptive Switching Algorithm for the iterative-MIMO receiver is designed in Matlab™ and its built-in Simulink® modeling package. On the other hand, the hardware design involves constructing the circuitry of the receiver using Xilinx® System Generator based on the latest Xilinx® Virtex-7 hardware. The system setup for both software and hardware co-simulation is shown in Fig. 2.

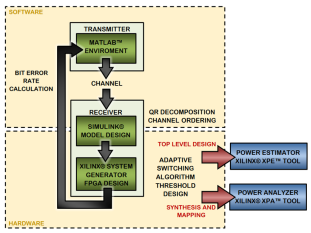

Fig. 2. Flowchart of the Software/Hardware Experimental Setup

The power readings are initially estimated by the Xilinx® Power Estimator (XPE)™ tool based on the multiplier re-

source counter utilization during the software modeling portion. The power readings measured gives ballpark estimates for hardware design, which are later confirmed during the implementation using the Xilinx® System Generator using the Xilinx® Power Analyzer (XPA)^{TM} tool after the model is synthesized and mapped onto the appropriate hardware.

It should be noted that, once the channel realizations and each corresponding channel ordering for specific detection algorithms that make up the Adaptive Switching Algorithm are simulated on Matlab^{TM}, the model of each detection and decoding method is demonstrated on Simulink® before being synthesized and mapped onto the Xilinx® Virtex-7 using the Xilinx® System Generator. The hardware is run at a core voltage of 1 V and at the operating frequency of 250 MHz. The energy can then be calculated from the power and the time it takes to transfer and decode a packet. In order to understand how the Adaptive Switching Algorithm receiver is implemented specifically, each block of the FPGA design is described in detail in the next subsections.

A. V-BLAST/ZF

The first detection algorithm within the proposed algorithm, V-BLAST/ZF [32], is implemented on the FPGA chip as shown Fig 3. The V-BLAST/ZF algorithm traverses the one path choosing the one with the best SNR condition during channel ordering as shown in Fig. 4(b), explained in detail later, optimistically assuming that the selected path yields the correct output. This type of non-linear detection works best during high SNR environments, where the noise is unlikely to distort the original transmits symbols. The FPGA part consists of three separate blocks, namely the "data estimation" block, where the ordered ZF channel sorts the signal according to the strongest signal with the highest SNR first as the received signals, $\mathbf{r}$ is augmented using the dot $(\cdot)$ operation with the channel matrix. The data is then quantized in the "data quantization" block, Q, to the nearest 16-QAM constellation to give $\hat{\mathbf{s}}$, which is then passed to the next block, "interference subtraction". This is where the quantized symbols are subtracted from the original data, $\mathbf{r}$ before repeating the whole process until $\mathbf{r}$ is fully nullified and all signals, $\hat{\mathbf{s}}$, are detected.

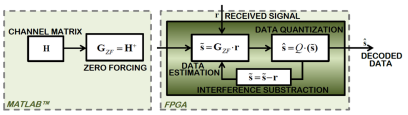

Fig. 3. Breakdown of V-BLAST/ZF FPGA Implementation Model

B. FSD

The second more complex detection method, FSD, published in [31], can be viewed as running multiple V-BLAST/ZF detectors in parallel, each checking different transmit data combinations of possible modulation symbols. FSD was derived from the original sphere decoding (SD) algorithm to reduce and fix the complexity of the algorithm due to the ever changing search radius and to eliminate the sequential nature of the SD search procedure. The diagram for the SD, the FSD and the V-BLAST/ZF can be seen in Fig. 4(a) and Fig. 4(b) respectively. Both the search for the algorithms can be visualized as a tree, traversing down each path until the end of the branches, where the possible solutions for the received symbols are accumulated. The main idea behind the FSD is to pre-determine a fixed but distinct number of candidates to be searched per antenna level.

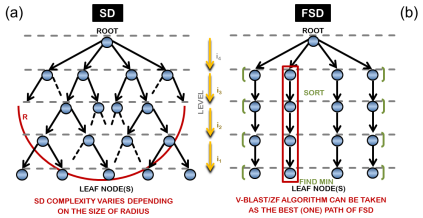

Fig. 4. Tree Search Structure for (a) SD, and (b) FSD and V-BLAST/ZF Algorithms

For the FPGA implementation, Fig. 5 provides the breakdown of the algorithm. The channel pseudoinverse, $\mathbf{G}$, is obtained by applying a $\mathbf{QR}$ decomposition to the channel matrix, which is implemented on Matlab^{TM}. There are two blocks of FPGA used for FSD implementation, namely the "metric calculation", which accumulates the Euclidean distance (ED), and the "path selection", which selects the minimum path to the lowest value for ED at the leaf node(s). Level i represents the i^{th} transmit antennas, therefore the partial accumulated ED, the AED, is calculated until the total ED is obtained for each path. The path of selected ED at the leaf node(s) then compared in order to find the minimum solution for received symbols, $\hat{\mathbf{s}}$. For the 16-QAM modulation scheme, after the full expansion on the first detected antenna, there are 16 paths to be selected, with 16 values of ED candidates for the minimum solution(s).

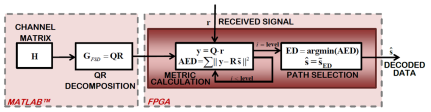

Fig. 5. Breakdown of FSD FPGA Implementation Model

V-BLAST/ZF and FSD are the two approaches that made up the Adaptive Switching Algorithm. They work together as one detector, switching from one another based on the noise level and the current channel conditions, i.e. based on the MI between the transmitters and receivers. Moreover, they are

chosen due to their similar algorithmic layout that means they are able to share hardware resources when being designed on hardware.

C. Adaptive Switching Algorithm

The Adaptive Switching Algorithm proposed in [17], works according to the specified design BER performance of the system. Efficient switching between the algorithms is performed by pre-determined thresholds calculated by the MI shown in equation (5).

$$\bar{I}(\mathbf{H}) \triangleq \log_2 \det\left(\mathbf{I} + \frac{\mathbf{H}\mathbf{H}^H}{N_0}\right) \tag{5}$$

where $\bar{I}$ is the accumulated MI within certain transmission frames, $\mathbf{I}$ is identity matrix, $\mathbf{H}$ is the real-time channel condition, $\mathbf{H}^H$ is the Hermitian transpose matrix of the channel with transpose vector of the M-dimensional transmit symbol vector with $E[|\mathbf{s}_i|^2] = M^{-1}$ and N_0 is the one sided spectral density expressed in decibels (dB) in the system. The main idea behind the Adaptive Switching Algorithm is explained in Fig. 6. The "threshold control" block calculates the value of the accumulated MI and activates the appropriate detector, either the V-BLAST/ZF, when the channel condition is good i.e. when the MI is above T_2; or the FSD during bad channel conditions, i.e. when MI is above T_1 but below T_2. Once the threshold is determined, the appropriate FPGA blocks are switched on and off accordingly. If the threshold falls under T_1, a re-transmission is required at a later time that consequently generates a new channel matrix, $\mathbf{H}$, in the simulation process. Avoiding using either detection algorithm in this way would also avoid processing the energy intensive iterative turbo decoding block. This process is deemed superfluous in this transmission environment since symbol retrieval will experience close to 100% failure rate, which only wastes significant computational power.

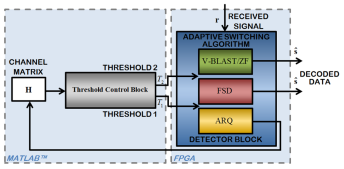

Fig. 6. Breakdown of Adaptive Switching Algorithm FPGA Implementation Model

After the symbols are detected, they are passed to the turbo decoder for error correction, and is run for a specific number of iterations. In state-of-the-art receiver, the data is processed through the Cyclic Redundancy Checksum (CRC) as an extra checking policy to check if the packet is decoded correctly during each iteration, which adds complexity in the system. This paper shows that this complexity can be reduced by re-using the same MI calculation of the Adaptive Switching Algorithm in the detector to design the threshold for early termination of the turbo decoder. This has three benefits: (1) canceling the need for having a fixed number of iterations of the turbo decoding, (2) avoiding the extra complexity of the CRC at each iteration and (3), avoiding using separate calculations for the threshold designs on the detector and decoder. These three points lead to the energy savings in the proposed receiver design.

D. Iterative Turbo Decoding

As shown in Fig. 1, after the detection process, the symbols are passed to the iterative decoder. Iterative decoding [33] is the key feature in turbo decoding. It is used right after the MIMO detector, where soft information extrinsic LLR (L_E) values are exchanged iteratively between the outer decoders with interleaving/de-interleaving operations in between until a certain number of iterations have been executed to achieve the desired performance [34].

Generally, soft detection is used and it generates *a posteriori* probability (APP) values in the form of LLR information, $L_E(b_k|\mathbf{r})$, about the interleaved bits, $\mathbf{b}$, for $1 \leq k \leq K_e$, while taking into account the channel observations $\mathbf{r}$ and the *a priori* LLR information, $L_A(b_k)$, coming from the outer decoder. For the FSD detector, assuming that the bits b_k are statistically independent due to the interleaving operation and making use of the Max-log approximation, $L_E(b_k|\mathbf{r})$ can be approximated by:

$$\begin{aligned} L_E(b_k|\mathbf{r}) \approx \frac{1}{2} \max_{\mathbf{b}\in\mathcal{L}\cap\mathbb{B}_{k,+1}} \left(\frac{-\parallel \mathbf{r}-\mathbf{H}\mathbf{s} \parallel^2}{\sigma^2/2} + \mathbf{b}^{\mathrm{T}}_{[k]}\mathbf{L}_{A[k]}\right) \\ -\frac{1}{2} \max_{\mathbf{b}\in\mathcal{L}\cap\mathbb{B}_{k,-1}} \left(\frac{-\parallel \mathbf{r}-\mathbf{H}\mathbf{s} \parallel^2}{\sigma^2/2} + \mathbf{b}^{\mathrm{T}}_{[k]}\mathbf{L}_{A[k]}\right) \end{aligned} \tag{6}$$

for $1 \leq k \leq K_e$, where, without loss of generality, $K_e = M \cdot \log_2 W$ has been assumed to simplify the index notation. In equation (6), $\mathbf{b} = (b_1, b_2, b_3, \ldots, b_{K_e})^{\mathrm{T}}$, $\mathbf{b}_{[k]}$ denotes the subvector of $\mathbf{b}$ omitting b_k, $\mathbf{L}_A = [L_A(b_1), L_A(b_2), \ldots, L_A(b_{K_e})]^{\mathrm{T}}$, $\mathbf{L}_{A[k]}$ denotes the subvector of $\mathbf{L}_A$ omitting $L_A(b_k)$, $\mathbb{B}_{k,+1}$ and $\mathbb{B}_{k,-1}$ represent the sets of 2^{K_e-1} bit vectors $\mathbf{b}$ having $b_k = +1$ (logical '1') and $b_k = -1$ (logical '0') respectively, $\mathcal{L}\cap\mathbb{B}_{k,+1}$ and $\mathcal{L}\cap\mathbb{B}_{k,-1}$ denote the subgroups of vectors of $\mathcal{L}$ that have $b_k = +1$ and $b_k = -1$ respectively. The list of candidates $\mathcal{L} \subset \mathcal{O}^M$ is detector specific and subject to the overall performance and complexity of the iterative-MIMO receiver, since $\parallel \mathbf{r}-\mathbf{H}\mathbf{s} \parallel^2$ needs to be computed for all $\mathbf{s} \in \mathcal{L}$.

It should be noted that, for V-BLAST/ZF detection, the LLR information can be simplified further by performing symbol by symbol likelihood calculations. In this model, $M \times 1$ coded bits are processed at one time and the LLR is defined as:

$$L(i,b) \approx \frac{1}{\sigma^2}\left(\min_{\mathbf{s}\in\mathcal{Z}^{(-1)}_{i,b}} \parallel \mathbf{r}-\mathbf{H}\mathbf{s} \parallel^2 - \min_{\mathbf{s}\in\mathcal{Z}^{(+1)}_{i,b}} \parallel \mathbf{r}-\mathbf{H}\mathbf{s} \parallel^2\right) \tag{7}$$

under the assumption of equally distributed transmit symbols $\mathbf{s}$. The sets $\mathcal{Z}^{(+1)}_{i,b}$ and $\mathcal{Z}^{(-1)}_{i,b}$ are subsets of $\mathcal{O}$, where the b^{th} bit of the i^{th} stream is equal to $+1$ and 1, respectively.

Due to the iterative nature of decoding, the BER improves significantly at the output of the decoder as the iterations progress. This improvement depends on the SNR, where is it dependent on the MIMO channel characteristics, and the MI between the transmitter and the receiver as well. Since the design for the detector considers the MI to provide the adaptivity, this work forwards the same MI value to the iterative decoder. By passing the same MI value to both the detector and the iterative decoder, we hope to gain positive energy savings by stopping the system from dissipating useless energy in the decoding process by limiting the number of decoding iterations. When the next iteration of the decoder no longer provides significant improvement to the BER, early termination rules or stopping criteria are to be implemented.

Typically, most stopping criteria work by setting a number of required decoding iterations according to certain rule, which can be generalized in Fig. 7. The trend is that the number of the decoding iterations decreases as the channel condition improves, or at high SNR levels, whilst maintaining the desired BER performance. In theory, the number of decoding iterations may approach infinity as shown in Fig. 7(a), however, due to the delay limits in the receiver, all systems have set a maximum number of iterations as can be seen in Fig. 7(b). At low SNRs, this number of iterations will not yield correct decoding. This failure point or error boundary is usually predicted by the usage of an extrinsic information transfer chart (EXIT) charts [35] [36]. However, EXIT charts are difficult to implement and uses a lot of hardware resources due to having a large look-up table (LUT). In addition, EXIT charts are very specific to the design of the interleavers, which prevents the analysis of the asymptotically attainable performance. Furthermore, the task become time consuming, since the length of the interleavers are usually set as high as possible in order to reduce the correlation among the interleaved *a priori* and extrinsic LLRs [37]. These disadvantages can be negated by knowing in advance the number of minimum decoding iterations for the system by calculating the corresponding MI and using it as a basis of the threshold design. The basic principle of the proposed decoder that incorporates the Adaptive Switching Algorithm works by using the forwarded MI values from the detector. This MI values will determine the number of iteration(s) required depending on the current channel conditions of the transmissions. Moreover, the Adaptive Switching Algorithm decoder proposes that during transmissions where the channel conditions will yield close to 100% decoding failure, it would cease the process and requests for a automatic repeat request (ARQ) instead, with zero iterations used in the turbo decoding, resulting in significant energy savings. This design choice is shown in Fig. 7(c). The results for the MI threshold are obtained by numerical analysis and are presented in the next section.

IV. Results and Analysis

The results are presented in subsections according to the setup detailed in Fig. 1, where each part is numerically labeled, and the energy performance analysis are based on the Xilinx® Virtex-7 chipset running at a core voltage of 1 V and an operating frequency of 250 MHz.

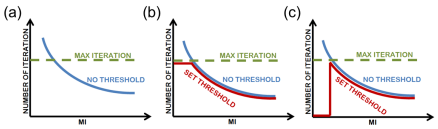

Fig. 7. General Trends for Thresholds used in Different Stopping Criteria; (a) when no thresholds are used, (b) when a Maximum Threshold is used, (c) when both Minimum and Maximum Thresholds are used.

A. Part 1 - The Behavior of the Detector in Spatially Correlated Channels

As shown in Fig. 1, the first part of the work, labeled **Part 1**, involves in running separate detection algorithms that make up the Adaptive Switching Algorithm with different correlated channel factor. In order to investigate the impact they have on the channel correlation indexes, the channel correlations of $\mathbf{H}$ in equation (1) are set to be $\mathbf{R}_{Tx} = \mathbf{R}_{Rx} = \mathbf{C}$. The total resource allocation provided by the Xilinx® Integrated Synthesis Environment (ISE) for both detection algorithms is given in Table I. The V-BLAST/ZF uses less resources, about a quarter of that required the more complex FSD.

TABLE I
Xilinx® Resource Utilization for the V-BLAST/ZF and the FSD Detection Algorithms

XILINX® VIRTEX-7 : XC7VLX330TFFG1157

Logic Resource	Utilization	
Utilization	V-BLAST/ZF	FSD
Slice Registers	3,312	13,683
Flip Flops	892	4,688
4-Input LUTs	2,940	12,161
DSP48E	48	132
Memory (RAM)	12	28

The number of multiplier counts can be estimated by breaking down the resource counter for each block using the Xilinx® ISE software. For V-BLAST/ZF, as shown in Fig. 3, the most complexity comes from the "data estimation" block since the process requires complex matrix multiplications, which takes almost 65% of the whole detection algorithm, followed by the "data quantization" of matching symbols on specific QAM constellation LUT at 26%. For FSD on the other hand, which is depicted in Fig. 5, the highest complexity comes from the "metric calculation", of the channel matrix against the transmitted symbols, uses most of the resources, as well as the summation of the accumulated ED, taking almost 75% of the total FSD operation. These results will provide an estimation for hardware design implementation.

When the two detection algorithms are implemented on different factors of $\mathbf{C}$, the BER degrades significantly for both detection algorithms as depicted in Fig. 8(a) and Fig. 8(b) for FSD and V-BLAST/ZF respectively. As the channel correlation increases, getting more profound differences at

higher SNR regions. This gets problematic at higher correlated channels when the V-BLAST/ZF is deployed, with BER of higher than 10^{-1} for **C** = 0.7 for SNR $\leq$ 20 dB as depicted in Fig. 8(b). In order to achieve the BER tolerance design for the entire system of 10^{-3}, SNR approximately $\geq$ 45 dB for V-BLAST/ZF is required when the **C** = 0.7 in comparison to SNR of approximately 27 dB for uncorrelated channels as depicted on Fig. 8(b). Similarly, a higher SNR is also needed or the FSD as shown in Fig. 8(a), where the BER for **C** = 0.7, is also higher, at 10^{-2} for SNR of 20 dB and lower, and it requires an SNR of more than 26 dB to obey the system performance requirements. However, the BER performance would improve significantly when the turbo decoder is included in the design, which may help in dealing with maintaining the overall performance of the system on spatially correlated channels.

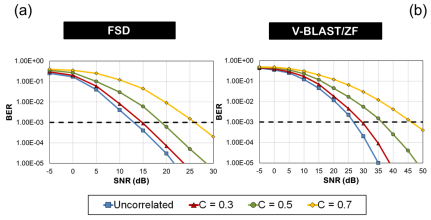

Fig. 8. Comparison of Detector BER Performance on Spatially Correlated Channels for (a) FSD and (b) V-BLAST/ZF

With the performance verified, the MI values are calculated to provide the design of the thresholds for the Adaptive Switching Algorithm detector on different correlated channels. It is found that even though fading correlation does considerably affect the BER performance of each detection algorithm, the correlation index does not show any considerable changes to the MI values obtained. Monte Carlo simulations are run 10 times, where each run comprises $100,000$ channel realizations for each correlation index, **C**, at the SNR span of -5 dB to 20 dB. This can be observed in Fig. 9. More information on how the thresholds are determined can be found in [17].

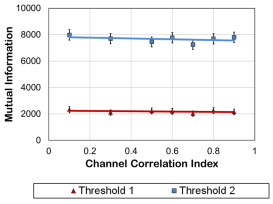

Fig. 9. Comparison of Detector Power Consumption on Spatially Correlated Channels

The impact on the obtained MI thresholds shows only minor changes as the correlation of the channel increases. The two thresholds for the Adaptive Switching Algorithm detector lie in the range of $2,100$ to $2,300$ for, T_1, and $7,100$ to $7,800$ for threshold 2, T_2, for FSD and V-BLAST/ZF respectively. It gives a linear trend therefore, it can be concluded that the threshold values for the Adaptive Switching Algorithm detector remain the same even when applied spatially correlated channels and it can be said that the detector design is only specific to the modulation and coding schemes in use. With these results, the design for the proposed algorithm is set as $2,200$ and $7,100$ for T_1 and T_2 respectively. T_1 corresponds to the BER = 0.5 and T_2 for a BER of 10^{-3}.

The other performance parameter, which is the energy consumption, can be calculated by taking the power readings provided by Xilinx® ISE and using the time it takes to transfer a packet bit size of $1,024$ at a core voltage of 1 V and an operating frequency of 250 MHz on the Xilinx®Virtex-7 chipset. For the span of the SNR levels of -5 dB to 20 dB, the average energy consumption of the two detection algorithms within the Adaptive Switching Algorithm against the correlated channel index range of 0 to 1 are computed for the FSD and the V-BLAST/ZF as 3.6 μJ and 0.9 μJ respectively. This shows that with the increase in correlation, the energy consumption of the detector is hardly affected as well. This could be due to the both algorithms work independently of noise level and have a fixed distinct search on any channel conditions. For the detector, it can be concluded that, comparable energy savings can be gained in spatially correlated channels as well. When combining both algorithms to make the Adaptive Switching Algorithm, Fig. 10 shows the energy consumption on a spatially correlated channels. In the detector, the energy savings when utilizing the Adaptive Switching Algorithm on different correlated channel indexes can be calculated numerically for SNR range of 0 dB to 50 dB for a run of $100,000$ channel realizations on the chosen hardware. This is essentially the area under the graph of Fig. 10 if the FSD is taken as the 100% baseline at 3.6 μJ. The results are tabulated in Table II. It can be observed that though there are still savings gained, the energy savings decreases with higher channel correlation.

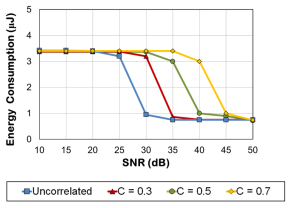

Fig. 10. Energy Consumption of the Adaptive Switching Algorithm Detector in Spatially Correlated Channels

Fig. 10 also shows the reason for the reduced energy saving, which is that, the threshold T_2 between the two algorithms corresponds to a much higher SNR for higher channel correlation values. From the figure, it can be observed that the switching occur an SNR $\approx$ 25 dB for uncorrelated channels,

TABLE II
ENERGY SAVINGS OF ADAPTIVE SWITCHING ALGORITHM DETECTOR ON SPATIALLY CORRELATED CHANNELS

Correlation Index (C)	Energy Savings (%)
Uncorrelated	40%
0.3	33%
0.5	27%
0.7	19%

and SNR $\approx$ 46 dB for $\mathbf{C}$ = 0.7. It can be concluded that, the energy usage varies for the Adaptive Switching Algorithm with varying channel correlation factors, with lower savings can be gained as the correlation increases.

B. Part 2 - Joint Switching of the Detector and the Decoder

Since the effectiveness of the proposed algorithm detector can save energy regardless of channel correlation index, this part of the work investigates the next part of the receiver, labeled **Part 2** in Fig. 1, which is the applicability of the Adaptive Switching Algorithm as a link between the detector and iterative decoder. **Part 2** is where the the two thresholds for both the detector and decoder reside. When each part of the receiver, which are the detector(s) and the iterative decoder, are implemented on the Xilinx® Virtex-7, the multiplier counts and thus the complexity are determined. It can be found that about 76% of the total complexity of the receiver is from the iterative-MIMO turbo decoder, with 23% related to the MIMO detector with 1% reserved for the threshold control. Therefore, minimizing the complexity within the decoder would achieve greater energy savings than the ones obtained in **Part 1**, i.e. in the detector(s).

Shifting the focus to the decoder, the turbo decoders are divided into several blocks. If the total resource allocation for the entire decoder is set to be at 100%, the blocks with their corresponding complexity are detailed in Table III. It can be noted that the highest complexity comes from the Maximum *A Posteriori* (MAP) decoders, therefore, limiting the number of iterations each received packet needs to go through would be the key to minimizing energy consumption within the turbo decoding. The Adaptive Switching Algorithm passes the MI calculated in the detector to the decoder, and thus the number of iteration iterative turbo decoder can be determined.

TABLE III
COMPLEXITY BREAKDOWN FOR TURBO DECODING

Block	Multiplier Counts (%)
Interleaver	1%
Demultiplexer	13%
logMAP Decoder(s)	80%
Trellis Tree	2%
Extrinsic LLR	4%

Fig. 11 gives the maximum, minimum and average number of iterations required when the experiment on the same Monte Carlo setup as in **Part 1**, where packets of 1,024 bits over 100,000 channel realizations are transmitted. The trend resembles the stopping criteria trends in Fig. 7, where as the MI increases, the number of decoding iterations decreases. Due to the design of the proposed algorithm, no decoding is taken place when the MI is below T_1, which is MI of 2,100 and below, saving the unnecessary computations when the failure rate is extremely high. An ARQ or re-transmission is enabled in this region.

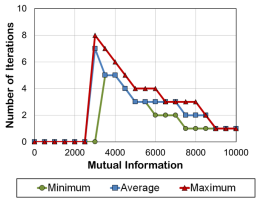

Fig. 11. Comparison of Detector Energy Consumption on Spatially Correlated Channels

The trends provide a general idea for the range in iterations required in the turbo decoder over the considered number of transmissions. The average and maximum lines provide guidelines to the required number of iterations but are not directly used in the threshold design for the decoder. The minimum number of iterations is taken from Fig. 11 as a foundation for the "Adaptive Switching Algorithm" threshold design in the decoder. Different stopping criteria for the decoder, one with the state-of-the-art used in Long Term Evolution (LTE) systems, the "CRC-24" method [38], and another without any stopping methods, with maximum of eight iterations throughout, labeled the "No Stopping Criteria" for the detector and decoder link are compared, as shown in Fig. 12(a). The results are obtained using the Xilinx® System Generator software. For a fair comparison of the stopping criteria, the detector part is fixed to FSD with different stopping criteria usage on the decoder. It can be seen that the number of iterations required on Adaptive Switching Algorithm is the same as the CRC-24 method. The Adaptive Switching Algorithm has a fail-safe error checking method at the end of the final iteration, therefore, if a packet is not correctly decoded by the end of the final iteration, the decoder would increase the number of iterations up to a maximum of eight, after the CRC-24 check is implemented, giving it more reliability in performance. In addition to the Adaptive Switching Algorithm using different iteration counts, Fig. 12(b) shows the fact that the proposed Adaptive Switching Algorithm also uses only about 18% multipliers needed as a stopping criteria when compared to the state-of-the-art CRC-24 method, when taking the latter as a baseline for percentage complexity calculations. This is due to the CRC having intricate calculations involving division of the data polynomials to get the remainder. For CRC-24, the degree of the polynomial is 24. This can be said due to a smaller number of multiplier counts and comparable number of iterations needed, the Adaptive Switching Algorithm provides a better implementation when compared to the CRC-24 method.

When calculating the energy consumption using the same

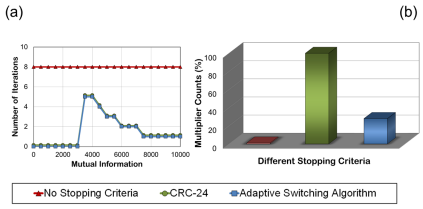

Fig. 12. Comparison of Stopping Criteria in Turbo Decoder

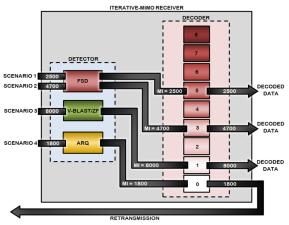

Fig. 13. Different Transmission Scenarios for Adaptive Switching Algorithm Receiver

setup as **Part 1**, it can observed that the "No Stopping Criteria" uses a lot more energy and is consistent throughout the span of considered SNR of -5 dB to 20 dB. Due to the minimization of the turbo decoding iterations, the energy consumption for both CRC-24 and the proposed decoder algorithm utilize a much lower energy consumption particularly at high SNR regions. Taking the "No Stopping Criteria" as the baseline for energy savings calculations, the overall percentage of energy savings are summarized in Table IV. The Adaptive Switching Algorithm decoder saves 7% more energy in comparison to the state-of-the-art CRC-24. Though this savings is not particularly large, this part of energy savings only considers the decoder part and more savings can be gained when a full Adaptive Switching Algorithm is utilized in the iterative-MIMO receiver.

TABLE IV
AVERAGE ENERGY SAVINGS OF THE DECODER ON XILINX ®VIRTEX-7

XILINX®VIRTEX-7: XC7VLX330TFFG1157

Receiver Setup	Average Total Energy Savings
No Stopping Criteria	-
CRC-24	32%
Adaptive Switching Algorithm	39%

With both detector and decoder blocks verified, the receiver for the Adaptive Switching Algorithm can be constructed. The two thresholds LUT designs for the detector and the decoder that sit in **Part 2** are summarized in Table V.

In order to understand how the full Adaptive Switching Algorithm behaves, consider these four scenarios illustrated in Fig. 13 on how a transmission can take place. "Scenario 1" is when the MI = $2,500$. Referring to the threshold designs in Table V, this packet will go through the FSD detector and 5 iterations on the turbo decoder before the packet successfully is decoded. "Scenario 2" represents an MI = $4,700$, and thus, the packets will go through 3 iterations in the decoder after being detected by the FSD. If the accumulated MI = $8,000$ as in "Scenario 3", the packets will be detected by the V-BLAST/ZF and only iterate once in the decoder. Lastly, "Scenario 4" denotes MI = $1,800$. Since the MI is less than the necessary MI for any detecting and decoding to take place, an ARQ is activated so that the transmitter will re-transmit the same data packets in hope for a better channel condition.

C. Part 3 - The Total Iterative-MIMO Receiver Energy Savings in Realistic Conditions

Having demonstrated the Adaptive Switching Algorithm threshold designs in both the detector and the decoder in **Part 2**, the work in **Part 3** compares the full Adaptive Switching Algorithm with other systems as given in Table VI.

TABLE VI
RECEIVER SYSTEMS DESIGN PARAMETERS

Name of System	Detector	Decoder
Full High Specification	FSD	No Stopping Criteria
State of the Art	FSD	CRC
Half ASA	FSD	ASA
Full ASA	ASA	ASA

[ASA - Adaptive Switching Algorithm]

These four systems are compared to verify the effectiveness of different system designs. The "Full High Specification" consists of the high performance FSD for the detection and always performs the maximum eight iterations for the turbo decoding. In the second system, the FSD is used alongside the latest stopping criteria method used in the LTE systems, which is the CRC-24. The proposed Adaptive Switching Algorithm design is investigated where the decoder coupled with the FSD as the detector to show the mechanism of the Adaptive Switching Algorithm as a stopping criterion in the system. This makes up the third system. Lastly, the full Adaptive Switching Algorithm system design, which operates the Adaptive Switching Algorithm on both parts of the system are measured for power and energy performance to confirm its validity in the iterative-MIMO receiver systems.

By incorporating the turbo decoder, the BER performance of receiver using the V-BLAST/ZF is explained in Fig. 14(a). Similar to Fig. 8, spatially correlated channels affect negatively on the BER performance. However, due to the decoder, the V-BLAST/ZF now able to achieve a better BER performance. The required SNR for detector switching from FSD to V-BLAST/ZF is also illustrated here. It shows an SNR ≈ 20 dB is needed for $\mathbf{C} = 0.7$ for the detection algorithm switching can occur in comparison to SNR ≈ 46 dB when no decoder is present. With these values, the BER for the Adaptive Switching Algorithm can be seen in Fig. 8(b). From the figure, it can be observed the switch for transmissions during the uncorrelated MIMO channels occur at around $8-9$ dB, around

TABLE V
ADAPTIVE SWITCHING ALGORITHM THRESHOLD DESIGNS FOR DETECTOR AND DECODER BLOCKS OF RECEIVER

MIMO Detector			Turbo Decoder		
Label	MI	Type of Detector	Label	MI	No. of Iterations
ARQ	$\leq$ 2,200	No Detection	ARQ	$\leq$ 2,200	0
T_1	2,200 $<$ MI $\leq$ 7,100	FSD	T_a	2,200 $<$ MI $\leq$ 4,000	5
-	-	-	T_b	4,000 $<$ MI $\leq$ 4,500	4
-	-	-	T_c	4,500 $<$ MI $\leq$ 6,000	3
T_2	$>$ 7,100	V-BLAST/ZF	T_d	6,000 $<$ MI $\leq$ 7,500	2
-	-	-	T_e	$>$ 7,500	1

11 dB for **C** = 0.3, 14 dB for **C** = 0.5 and 20 dB for **C** = 0.7. It can be seen that the BER performance is still under 0.5 and 10^{-3} for T_1 and T_2 respectively. Separate considerations of the Adaptive Switching Algorithm in the detector and decoder have proven that the adaptivity in the proposed algorithm has the ability to save energy whilst maintaining satisfactory BER performance. It can be concluded that the Adaptive Switching Algorithm works well for the full iterative-MIMO receiver design, since it is able to conform with the error tolerance requirement of the system of 10^{-3}.

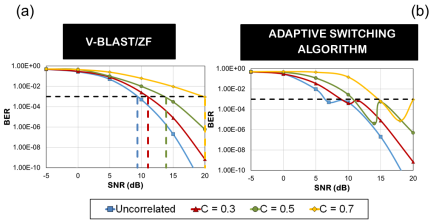

Fig. 14. Performance of Turbo Decoder in Spatially Correlated Channels for (a) V-BLAST/ZF and (b) Adaptive Switching Algorithm

Using the same energy calculation method, taking the "Full High Specification" as a baseline, the total energy usage can be calculated as areas under the graphs. In order to see how the extreme cases of correlation affect the energy savings, correlations of 0 and 0.9 are considered. Since most current systems normally operates between the range of 0 dB to 40 dB during real-life deployment [39], the results for the simulation under these SNR regions are given in Fig. 15(a) for uncorrelated channels, i.e. **C** = 0, and in Fig. 15(b) for correlated channels of **C** close to 1. It can be seen that higher SNRs are required to reduce energy consumption for highly correlated channels. The energy savings are summarized in Table VII.

The energy savings of $74-78\%$ across SNR of 0 dB to 40 dB can be achieved when the "Full Adaptive Switching Algorithm" system is utilized for uncorrelated and correlated channel respectively. Both the uncorrelated and correlated channel follow roughly the same energy trend with the exception of needing a higher SNR for the latter type of channel conditions. This gives a benefit of around $24-34\%$ savings gained in comparison to the state-of-the-art CRC-24 method. The savings lessen as as the correlation increases, however,

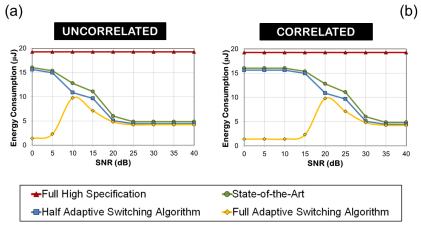

Fig. 15. Energy Savings Comparison between Different Systems under Consideration for (a) Uncorrelated and (b) Correlated Channel Conditions

TABLE VII
AVERAGE ENERGY SAVINGS OF THE ITERATIVE-MIMO RECEIVER ON XILINX ®VIRTEX-7

XILINX®VIRTEX-7: XC7VLX330TFFG1157

Receiver Setup Name	Energy Savings Uncorrelated	Correlated
Full High Specification	-	-
State of the Art	54%	40%
Half Adaptive Switching Algorithm	59%	44%
Full Adaptive Switching Algorithm	78%	74%

74% energy savings can be gained when the channel is highly correlated, it can be concluded that the Adaptive Switching Algorithm works in an energy efficient manner regardless of the channel conditions.

V. CONCLUSION

The Adaptive Switching Algorithm was utilized in both detector and decoder to create a full adaptive iterative-MIMO receiver. The same threshold calculations involving the MI between the transmitters and receivers provide sufficient information in real-time regarding any channel conditions, whether uncorrelated or spatially correlated. The work has proven that the average energy savings in the detector can be achieved throughout the span of considered SNR conditions of -5 dB to 20 dB, and they are to be at the range of 19% to 40% when implemented on Xilinx® Virtex-7 chipset. The design for the Adaptive Switching Algorithm was expanded to be a link between the detector and decoder, which helps reduce the energy consumption up to 39% by limiting the number of turbo decoding iterations in spatially correlated conditions, in comparison to the baseline system. When a full Adaptive Switching Algorithm is implemented on the receiver, it can

save up to 74% of the total energy consumption regardless of channel conditions. Thus, the proposed algorithm confirms the fact that its adaptivity attribute in iterative-MIMO receivers is highly beneficial and the idea could adopted in real-world future wireless communication devices.

REFERENCES

[1] E. Telatar, *"Capacity of Multi-Antenna Gaussian Channels"*, European Transactions on Telecommunication, vol. 10, no. 6, pp. 585-595, Dec. 1999.

[2] G. J. Foschini, *"Layered Space-Time Architecture for Wireless Communication in a Fading Environment when using Multi-Element Antennas"*, Journal of Bell Laboratory Technology, vol. 1, no. 2, pp. 41-59, Aug. 2002.

[3] D. S. Shiu, G. J. Foschini, M. J. Gans, J. M. Kahn, *"Fading Correlation and its Effect on the Capacity of Multielement Antenna Systems"*, IEEE Transactions on Communications, vol. 48, no. 3, pp. 502-513, Mar. 2000.

[4] D. Wübber, V. Kühn, K. D. Kammeyer, *"On the Robustness of Lattice-Reduction Aided Detectors in Correlated MIMO"*, Proceedings IEEE 60th Vehicular Technology Conference, vol. 5, no. 1, pp. 3639-3643, Sep. 2004.

[5] Q. Meng, Z. Pan, X. You, Y. H. Kim, *"On Performance of Lattice Reduction Aided Detection in the Presence of Receive Correlation"*, Proceedings IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communications, Transactions on Information Theory, vol. 1, no. 1, pp. 89-92, Jun. 2004.

[6] L. G. Barbero, J. S. Thompson, *"Performance of the Complex Sphere Decoder in Spatially Correlated Channel"*, Journal of Institution of Engineering and Technology, vol. 1, no. 1, pp. 122-130, Feb. 2007.

[7] M. R. McKay, I. B. Collings, A. Forenza, R. W. Heath, *"A Throghput-Based Adaptive MIMO-OFDM Approach for Spatially Correlated Channels"*, IEEE International Conference on Communications, vol. 1, no. 1, pp. 1374-1379, Jun. 2006.

[8] Y. L. Chen, C. Z. Zhan, T. J. Jheng, A. Y. Wu, *"Reconfigurable Adaptive Singular Value Decomposition Engine Design for High-Throughput MIMO-OFDM Systems"*, IEEE Transactions on Very Large Scale Integration Systems, vol. 21, pp. 747-760, Apr. 2013.

[9] P. J. Smith, M. Shafi, L. M. Garth, *"Performance Analysis for Adaptive MIMO SVD Transmission in a Cellular System"*, IEEE Transactions on Parallel Distributed Systems, vol. 1, pp. 49-54, Feb. 2006.

[10] J. Sarrazin, Y. Mahe, S. Avrillon, S. Toutain, *"On the Performance of Adaptive MIMO Systems using Radiation Pattern Reconfigurable Antennas"*, IEEE International Symposium Antennas and Propagation Society, vol. 1, pp. 1-4, Jul. 2008.

[11] X. Wu, J. S. Thompson, A. M. Wallace, *"An Improved Sphere Decoding Scheme for MIMO Systems Using An Adaptive Statistical Approach"*, 17th European Signal Processing Conference, vol. 1, pp. 2668-2672, Aug. 2009.

[12] M. Matthaiou, D. I. Laurenson, X. I. Wang, *"Reduced Complexity Detection for Ricean MIMO Channels Based on Channel Number Thresholding"*, IEEE International Proceedings of Wireless COmmunications and Mobile Computing Conference, pp. 1718-1722, Aug. 2008.

[13] Z. Wang, Y. Tan, Y. Wang, *"Low Hardware Complexity Parallel Turbo Decoder Architecture"*, IEEE International Symposium on Circuits and Systems, vol. 2, pp. 53-56, May. 2003.

[14] J. S. Chung, *"Fast Power Allocation Algorithm for Adaptive MIMO Systems"*, PhD Thesis, University of Canterbury, Mar. 2005.

[15] L. Xian, H. Liu, *"An Adaptive Power Allocation Scheme for Space-Time Block Coded MIMO Systems"*, IEEE Wireless Communications and Networking Conference, vol. 1, pp. 504-508, Mar. 2005.

[16] Z. Wang, Y. Tan, Y. Wang, *"Performance Analysis of Variable-Power Adaptive Modulation in Space-Time Block Coded MIMO Diversity Systems"*, Science China Information Science Press, vol. 53, pp. 2106-2115, Jun. 2010.

[17] N. Tadza, D. I. Laurenson, J. S. Thompson, *"Adaptive Switching Detection Algorithm for Iterative-MIMO Systems to Enable Power Savings"*, Journal of Radio Science, vol. 49, no. 11, pp. 1065-1079, Nov. 2014.

[18] N. Tadza, D. I. Laurenson, J. S. Thompson., *"Power Performance Analysis of the Iterative-MIMO Adaptive Switching Algorithm Detector on the FPGA Hardware"*, accepted to IEEE Vehicular Technology Conference (VTC), May 2015. A copy of the paper can be accessed at this website, http://tinyurl.com/on6wqle.

[19] B. M. Hochwald, S. T. Brink, *"Achieving Near-Capacity on a Multiple-Antenna Channel"*, IEEE Transactions on Communications, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[20] J. Hagenauer, E. Offer, L. Papke, *"Iterative Decoding of Binary Block and Convolutional Codes"*, IEEE Transaction of Information Theory, vol. 42, pp. 429-445, Mar. 1996.

[21] D. Bokolamulla, T. Aulin, *"A New Stopping Criterion for Iterative Decoding"*, IEEE Communication Society, vol. 1, pp. 538-541, Jan. 2004.

[22] F. Zhai, I. Fair, *"New Error Detection Techniques and Stopping Criteria for Turbo Decoding"*, Proceedings 2000 Canadian Conference on Electrical and Computer Engineering, vol. 1, pp. 58-62, Mar. 2000.

[23] N. Y. Yu, M. G. Kim, Y. S. Kim, S. U. Chung, *"Efficient Stopping Criterion for Iterative Decoding of Turbo Codes"*, IEEE Electronic Letters, vol. 39, pp. 73-75, Jan. 2003.

[24] A. Goldsmith, E. Biglieri, R. Calderbank, *"MIMO Wireless Communications"*, Stanford University, pp. 559, May. 2007.

[25] J. G. Proakis, *"Digital Communications (3rd ed.)."* Singapore: McGraw-Hill Book Co., pp. 767-768, Jan. 1983.

[26] B. Sklar, *"Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization"*, IEEE Communications Magazine, vol. 35, no. 7, pp. 90-100, Jul. 1997.

[27] J. Kermoal, L. Schumacher, K. I. Pedersen, P. Mogensen, F. Frederiksen, *"A Stochastic MIMO Radio Channel Model With Experimental Validation"*, IEEE Journal on Selected Areas of Communications, vol. 20, pp. 1211-1226, May. 2002.

[28] A. M. Tulino, A. Lozano, S. Verdu, *"Impact of Antenna Correlation on the Capacity of Multiantenna Channels"*, IEEE Transaction on Information Theory, vol. 51, no. 7, pp. 2491-2509, Jul. 2005.

[29] K. Yu, M. Bengtsson, B. Ottersten, D. McNamara, P. Karlsson, M. Beach, *"Modeling of Wide-Band MIMO Radio Channels Based on NLoS Indoor Measurements"*, IEEE Transactions on Vehicular Technology, vol. 53, pp. 655-665, Jun. 2004.

[30] H. Özcelik, M. Herdin, W. Weichselberger, J. Wallace, E. Bonek, *"Deficiencies of 'Kronecker' MIMO Radio Channel Model"*, IEEE Electronics Letters, vol. 36, no.16, pp. 1209-1210, Aug. 2003.

[31] L. G. Barbero, J. S. Thompson *"Fixing the Complexity of the Sphere Decoder for MIMO Detection"*, IEEE Transactions on Wireless Communications, vol. 7, no. 6, pp. 2131-2142, Jun. 2008

[32] G. D. Golden, C. J. Foschini, R. A. Valenzuela, P. W. Wolniansky, *"Detection Algorithm and Initial Laboratory results using V-BLAST Space Time Communication Architecture"*, IEEE Electronic Letters, vol. 35, no. 1, pp. 14-16, Mar. 1999.

[33] J. Hagenauer, E. Offer, L. Papke, *"Iterative Decoding of Binary Block and Convolutional Codes"*, IEEE Transactions on Information Theory, vol. 42, no. 2, pp. 429-445, Mar.1996.

[34] C. Berrou, A. Glavieux, P. Thitimajshima, *"Near Shannon Limit Error - Correcting Coding and Decoding: Turbo Codes"*, IEEE International Conference on Communications, vol. 2, pp. 1064-1070, May 1993.

[35] C. Hermocilla, C. Szczecinski, *"Exit Charts for Turbo Receivers in MIMO Systems"*, International Symposium on Signal Processing and its Applications, vol. 1, pp. 1209-212, Jul. 2003.

[36] W. Li, H. Dai, *"EXIT Chart Analysis of Turbo-BLAST Receivers in Rayleigh fading Channels"*, IEEE Vehicular Technology Conference, vol. 1, pp. 1396-1400, Sep. 2004.

[37] H. Chen, R. G. Mauder, L. Hanzo, *"An Exit-Chart Aided Design Procedure for Near-Capacity N-Component Parallel Concatenated Codes"*, IEEE Global Telecommunications Conference, vol. 1, pp. 1-5, Dec. 2010.

[38] Long Term Evolution, *"3GPP Technical Specifications Series 36 for E-UTRA (Release 8)"*, http://www.3gpp.org, Jan. 2015.

[39] J. Geier, *"How to: Define Minimum SNR values for Signal Coverage"*, http://www.wireless-nets.com/resources/tutorials/define_SNR_values.html, Dec. 2013.

Nina Tadza graduated with first class honors in Computer and Electronics Engineering from the University of Nottingham. She holds a teaching position at the Malaysian University of Tun Hussein Onn (UTHM) and is currently pursuing her PhD in the fields of wireless communications and computer architecture at the University of Edinburgh under the sponsorship of the Ministry of Education (MoE).

Power Performance Analysis of the Iterative-MIMO Adaptive Switching Algorithm Detector on the FPGA Hardware

Nina Tadza and John S. Thompson and David I. Laurenson
Institute for Digital Communications
School of Engineering
University of Edinburgh, UK
Email: {n.tadza, john.thompson, dave.laurenson}@ed.ac.uk

***Abstract*—In this paper, a comprehensive power performance analysis of a novel Adaptive Switching Algorithm for an iterative-MIMO system is investigated with the prime goal of minimizing energy consumption in the receiver. The algorithm works by switching between a high performance detection method, the Fixed Sphere Decoding, and a much lower complexity algorithm, the Vertical-Bell Laboratories Layered Space-Time Zero Forcing technique, controlled by a threshold according to the mutual information calculated during each transmission. Results show significant improvements over current non-adaptive receivers, where energy savings of more than 60% can be obtained using on the latest Xilinx®Virtex-7 FPGA hardware.**

I. Introduction

Decoding received signals from an iterative-Multiple Input Multiple Output (MIMO) wireless system is computationally-expensive. Receiver performance is not only based on the success rate of the receiver recovering the data sent by the transmitter, but also in achieving this with minimal energy consumption. A system that could operate with low energy consumption whenever feasible is advantageous. Such detection algorithms currently active include Zero Forcing (ZF) with Decision Feedback (ZF-DF) [1], Sphere Decoder (SD) [2], Semidefinite-Relaxation (SDR) [3] etc. Although most work well in the detection process, they lack adaptivity, whereby, most detectors behave independently of the received signal characteristics and current channel conditions, which may waste computational resources. Several detection methods have been proposed to date to overcome this problem, however, one that could fit perfectly with MIMO characteristics has not been very well investigated. Most publications focus on saving power by using the Signal-to-Noise Ratio (SNR) [4], channel matrix condition number [5] or reducing the number of decoding iterations. These criteria are not enough to optimize the information on the entire MIMO setup. To tackle this, this paper considers the Mutual Information (MI) between the MIMO transmitters and receivers so that the diversity of the channels is fully exploited. Combining the MI with the noise level give better information regarding a channel in comparison to using either condition number or SNR alone.

This paper builds on the work of [7] by implementing the Adaptive Switching Algorithm onto an Field Programmable Gate Arrays (FPGA) hardware, in hope to gain further power and energy savings during hardware implementation. This gain is on top of the energy savings due to the switching receiver described in [7]. The FPGA is chosen as an exemplar platform for rapid prototyping purposes. Generally, even though it is more efficient to use an Application-Specific Integrated Circuit (ASIC) implementation, they usually require very long design times. FPGA is expected to produce generally similar trends and trade-offs with a fraction of the design time as other hardware platforms due to its re-programmability. This means it can provide a suitable platform for evaluating the implementation of the Adaptive Switching method in an iterative-MIMO system. The key to power savings comes from the algorithm exploiting the adaptivity in the detector according to the current conditions. The main contributions of this paper are summarized as follows:

- Realistic power and energy savings trends of the Adaptive Switching Algorithm are computed for example hardware circuitry.
- Detailed power analysis and the potential benefits of *Sleep Modes* and *Parallelization* as power savings techniques show more promising results in contrast to the *Voltage and Frequency Scaling*.

The rest of the paper is organized as follows; the Adaptive Switching Algorithm and its hardware design is given in Section II; Section III outlines several power saving methods evaluated in this paper; Section IV discusses the MIMO system under consideration; whilst the key findings are summarized in Section V; lastly, Section VI concludes the paper.

II. Adaptive Switching Algorithm

The Adaptive Switching Algorithm [7] is demonstrated with two well-known detection algorithms, namely the Fixed Sphere Decoder (FSD) [6], and the Vertical-Bell Laboratories Layered Space-Time [1] with the Zero Forcing (V-BLAST/ZF) technique. Switching between algorithms is determined by thresholds pre-calculated from the MI between the transmitter and the receiver, according to the real-time channel conditions.

A. Breakdown of Algorithm Design

1) V-BLAST/ZF: V-BLAST/ZF [1] is best deployed in high SNR environments, when the chances of successful decoding are high. Figure 1 illustrate the block diagram of the algorithm during implementation.

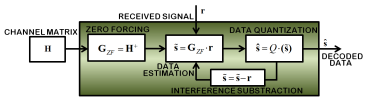

Fig. 1. Breakdown of V-BLAST/ZF Implementation Model

The algorithm minimizes the impact of noise by re-ordering the beamformer matrix, $\mathbf{G}$, which is the Moore-Penrose pseudoinverse of $\mathbf{H}$, with respect to the received signal strength. It processes the symbols, $\mathbf{r}$, according to this order i.e. handling the highest SNR antenna first. The signals are quantized to the nearest estimates, Q, using the quantizer function followed by linear combinatorial nulling and successive cancellation until all signals, $\hat{\mathbf{s}}$, are decoded.

2) FSD: The more complex detection method, the FSD, published in [6] can be viewed as running multiple V-BLAST/ZF detectors in parallel, each checking different transmit data combinations. Figure 2 provides the breakdown of the algorithm. The channel pseudoinverse, $\mathbf{G}$, is obtained by applying the $\mathbf{QR}$ decomposition to the channel matrix, $\mathbf{H}$.

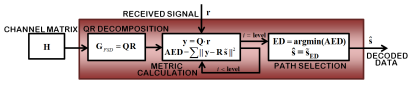

Fig. 2. Breakdown of FSD Implementation Model

The algorithm traverses down the tree, i, until the end of the tree i.e. the leaf is discovered, computing the Euclidean Distance (ED). FSD determines beforehand the number of nodes $\tilde{\mathbf{s}}$ around signal $\mathbf{r}$ that will be explored independent of the noise level, which means the search of an FSD is fixed for each candidate per antenna level. This yields an algorithm suitable for parallel implementation. The symbols $\hat{\mathbf{s}}$ associated with the minimum ED are the final solution.

3) Adaptive Switching Algorithm: The main idea behind the Adaptive Switching Algorithm is shown in Figure 3. The Threshold Control Block calculates the value of the accumulated MI, denoted by $\bar{I}$, obtained in the transmitter in relation to the receiver and activates the appropriate detector, either V-BLAST/ZF or FSD. Within the Threshold Control Block sits the MI calculation as shown in Equation (1).

$$\bar{I}(\mathbf{H}) \triangleq \log_2 \det\left(I + \frac{\mathbf{H}\mathbf{H}^T}{N_0}\right) \tag{1}$$

This calculation assumes the channel matrix $\mathbf{H}$ is perfectly known at the receiver with independent elements representing a block Rayleigh fading propagation environment, where T denotes the transpose operator and N_0 is the power of additive, independent and identically distributed (i.i.d.) circular symmetric complex Gaussian noise.

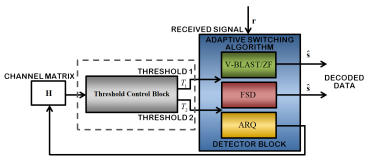

Fig. 3. Breakdown of Adaptive Switching Algorithm Implementation Model

The accumulated MI, $\bar{I}$, is dependent on the current channel conditions i.e. the noise level, N_0. The thresholds, T_1 and T_2, are pre-determined. If the MI computed is higher than the T_1 threshold, V-BLAST/ZF is chosen. FSD is selected when the transmitting environment is acceptable, which is when the MI value is in-between T_1 and T_2. When the channel is too poor for reliable recovery of the received signals, the detector block would send an Automatic Repeat reQuest (ARQ) for a re-transmission, avoiding forward error correction decoding when this is expected to fail, however, formally characterizing this decoding effect is out of scope of the present paper.

III. Power Saving Techniques

This paper investigates several power saving techniques when the Adaptive Switching Algorithm is implemented in FPGA hardware.

A. Voltage and Frequency Scaling

The power and energy consumption of a circuit depends on the number of computations performed over a fixed duration. By lowering the number of computations and varying the supply voltage to lower the internal clock frequency of the chip at run-time, the overall power consumption is lowered. The basic principle detailed in [8] states that the power consumed by running the operation at a slower speed is less than to run it at full power and finishing early. This study considers only the dynamic power however, discarding other components of power such as leakage, idle, overhead, static as well as the power needed to activate the chip. This present paper attempts to take all power components into consideration.

B. Sleep Mode

Sleep Mode is when the electronics operate in idle mode with a very low power consumption so they appear switched off for a certain period. When calculations do not possess the same task length and/or processing speed, they do not finish processing at the same time, meaning for some proportion of the time, processor cores need not be active at all times. Therefore, switching off the cores could be a means of saving power. By running the application as fast as possible, longer *Sleep Modes* can be deployed. This is a direct contradiction

to the findings of [8], where the reduction in dynamic power is inferior to the savings gained by scaling above. This paper attempts to discover, which power savings mode is best when other power components are also considered.

C. Parallelization

Part of optimizing a system in current chip designs is to construct the algorithms in such a way that parallel operations are possible. Using multiple processors provide a trade-off between utilizing more chip space and increasing the throughput of the algorithm. The cores split and share the computational load evenly amongst them. Therefore, each core performs only a fraction of the total computation depending on the number of cores available [8]. Furthermore, hardware architectures that can perform multiple tasks slowly in parallel should be more power efficient in comparison to computing a single operation at a higher clock speed [9]. Therefore, this paper will study how to combine the level of *Parallelization* with *Voltage and Frequency Scaling* technique.

IV. Experimental Setup

The experiment uses a software/hardware setup using Matlab™ and Xilinx® System Generator. The iterative-MIMO system under consideration comprises $M = 4$ transmitters and $N = 4$ receivers based on a Bit-Interleaved Coded Modulation (BICM) setup, which has a transmit frame size of K_u = 1,024 bits transmitting over a random independent Rayleigh fading propagation channel, $\mathbf{H}$, with independent fading elements, which is perfectly known at the receiver.

The transmitted bits, K_u, are encoded using an iterative-turbo scheme at rate of $R_c = 1/2$, which are then interleaved randomly to give, K_a coded bits, before mapping into a Quadrature Amplitude Modulation (QAM) constellation, $\mathcal{O}$, of $J = 16$, forming a sequence of $K_s = K_e/\log_2 J$ symbols. The 512 symbols are divided equally between the transmitters for 100,000 channel realizations. This part of the system is simulated using Matlab™. At the receiver, where the focus of this paper lies, is where the Adaptive Switching Algorithm detector is implemented using Xilinx®Virtex-7 chip. The receiver FPGA implementation is obtained using the Xilinx® System Generator.

V. Results and Findings

The total resource allocation of the Adaptive Switching Algorithm is given in Table I. The power usage can be calculated when the algorithm is implemented on the Xilinx® System Generator using the Xilinx® XPA™ tool. The power readings specified by the tool is generally dominated by the dynamic and static power terms, where dynamic is the power spent within a chip due to toggling of transistors, the value of voltage, the capacitance and is a function of the FPGA clock frequency. Static power is consumed due to transistor leakage and is highly dependent on the manufacturing process, the ambient temperature of the circuit, and the operating voltage. In order to determine the effectiveness of the algorithm, instead of power, a better parameter to consider is the energy, which is the power multiplied by the processing time. This information gives a better understanding of the system's efficiency in transferring the same size data packets within an allocated amount of time. Since this paper studies the energy efficiency of the system instead of maximizing the throughput, it is assumed that the system adopts low channel utilization policy, where packets are decoded at a maximum time of 20 μs.

TABLE I
Resource Allocation of the Adaptive Switching Algorithm

XILINX®VIRTEX-7 : XC7VLX330TFFG1157

Logic Resource Utilization	Used	Available	Utilization
Slice Registers	12,528	408,000	3%
Flip Flops	4,361	51,000	8%
4-Input LUTs	11,429	204,000	5%
DSP48E	132	1,120	11%
Memory (RAM)	41	1,500	2%

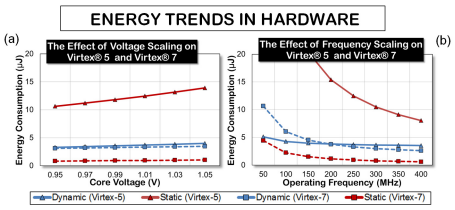

Fig. 4. Energy Trends with (a) the voltage applied and (b) the variation of clock frequencies on the Xilinx®Virtex-5 and Virtex-7 respectively

The energy trends are shown in Figure 4, where the dynamic and static energy consumption are compared on Xilinx®Virtex-5 [7] and Virtex-7. By comparing Figure 4(a), similar trends for scaling up the voltage in both chips can be observed, whereby, the energy is directly proportional to the voltage. When comparing the frequency however shown in 4(b), the energy consumption decreases with every frequency increment. From now on, the only chip under consideration is the Xilinx®Virtex-7. First, the main difference to note here is that dynamic energy dominates and therefore, the *Voltage and Frequency Scaling* may be able to save power in the detector [8]. Secondly, "high performance" and "low power" modes can be obtained by taking the extreme ends of the scaling ranges. If running the algorithm at the highest possible mode would save power, then the *Sleep Mode* would be an energy efficient method for the algorithm. Lastly, due to the small percentage of the area utilization, summarized in Table I, the algorithm has the potential for *Parallelization*, i.e. having multiple copies of the detector. This paper attempts to instigate the three techniques mentioned in Section III and determines if they might increase energy savings.

A. Voltage and Frequency Scaling

Figure 4 shows that, due to the higher level of dynamic to static energy, where it is approximately six times larger, the

overall energy of the circuit can be optimized. However, when considering the total energy of the chip, this might no longer be the case.

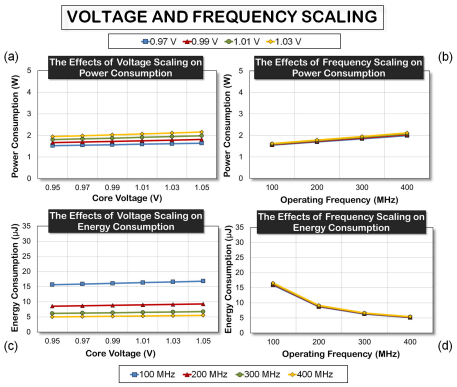

Fig. 5. Voltage and Frequency Scaling Effects where (a) and (b) are with voltage applied, (c) and (d) are with the variation of frequencies respectively

Figure 4(b) confirms this as the energy required to run the task at 400 MHz is less than 0.6 μJ in comparison to 2.9 μJ at 100 MHz, giving a difference of more than 65%. From this, it can be said that running the algorithm as quickly as possible at the lowest possible voltage and switching it off would be better than running it at a slower clock speed. The total power and energy consumption during the *Voltage and Frequency Scaling* are given in Figure 5. Similar to the previous experiment, the scaling of voltage is proportional to the power and energy consumption, which can be seen in Figure 5(a) and 5(c). Taking a clock speed of 200 MHz as an example, at voltages of 0.97 V and 1.03 V, the latter gives an increased power usage of 12%. Though minimal, it is still an undesired result. In contrast, Figure 5(b) illustrates that even with a minimal increment of power in frequency scaling, the reduction in energy shown in Figure 5(d) is substantial. Looking at a voltage of 0.99 V, running the algorithm four times faster provides 51% energy savings.

Moreover, Figure 5(d) shows the total energy required to decode the same packet of data is less, due to the faster decoding process. It suggests that running the algorithm at full speed would be better than to finish processing just in time. This means that instead of having it running at lower power and taking the maximum 20 μs to decode the data packet, the system would finish processing in less than 3 μs and be put into *Sleep Mode* for 78% of the time. This concludes that voltage scaling is not suitable as a power savings technique for the Adaptive Switching Algorithm on an architecture where static power is a significant component of power consumption.

B. Sleep Mode

Taking the extreme cases of the chip's lower and upper limit of voltage and frequency operations into consideration, "low power" and "high performance" modes can be evaluated. Table II reviews the parameters of the Xilinx®Virtex-7 when running the Adaptive Switching Algorithm in two separate modes. The power usage analyzed by the Xilinx® XPA™ tool are given as 1.5 W and 2.2 W for low power and high performance modes respectively, contributing to 19% increase in power usage when high performance mode is selected. The total maximum energy saving is equivalent to 69%.

TABLE II
LOW POWER AND HIGH PERFORMANCE PARAMETERS

Operation Mode/ Parameters	Low Power	High Performance
Core Voltage	0.97 V	1.03 V
Operating Frequency	60 MHz	400 MHz
Max Throughput	240 Mbps	1200 Mbps
Total Power Consumption	-	19%
Total Energy Savings	-	69%

This section confirms the previous conclusion where, it takes less energy to transfer the same data packet in "high performance" mode. Therefore, by running the algorithm as fast as possible and then switching the cores off would save more energy, and thus, *Sleep Modes* are a good way to save energy (and power) in the detector.

C. Parallelization

The Adaptive Switching Algorithm has quite low complexity and only uses a small percentage of the Xilinx®Virtex-7 as evident in Table I. This suggests promising results for parallel implementation, which are shown in Figure 6.

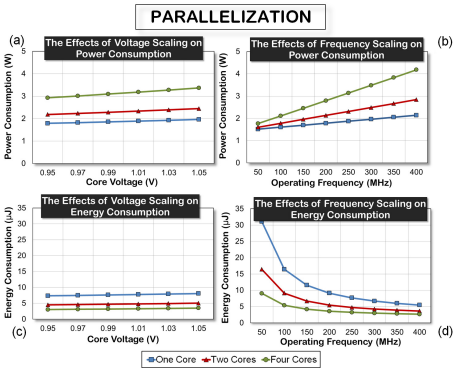

Fig. 6. Results for Parallel Implementation, (a) and (c) with the voltage applied, (b) and (d) with the variation of frequencies respectively

Multiple copies of the Adaptive Switching Algorithm are utilized with one core matching one copy of the algorithm

being used on the FPGA. As predicted, the more cores are used on the FPGA, the more power the chip needs as evident in Figure 6(a). This is due to the extra power needed to activate the multiple cores on the chip. However, the increase in power consumption is small, at maximum, 20%, with every quadruple number of cores used, which is evident at every voltage point. When it comes to energy however, although voltage scaling has little effect, the parallel setup does save significant overall energy savings seen in Figure 6(d).

TABLE III
"LOW POWER" AND "HIGH PERFORMANCE" PARALLEL IMPLEMENTATIONS

Number of Cores / Parameters	One		Two		Four	
	Low	High	Low	High	Low	High
Power Consumption	1.5 W	2.2 W	1.6 W	2.9 W	1.7 W	4.3 W
Energy Consumption	30.4 µJ	5.6 µJ	16.0 µJ	3.8 µJ	8.8 µJ	2.8 µJ
Power Usage	-	19%	3%	32%	6%	48%
Energy Savings	-	69%	31%	78%	55%	83%

Low Power: *0.97V, 60MHz*; **High Performance:** *1.03V, 400MHz*

The same can be said in frequency scaling, evident in Figure 6(b) and Figure 6(d), for power and energy respectively, where, taking frequency of 200 MHz as an example, running four cores instead of one gives 42% energy savings with only a 14% increase in power. The energy saved whilst running parallel cores in comparison to running a single thread is substantial, ranging from 3% to 68% across all frequencies, with particularly large differences at lower clock frequencies. These results show that, *Parallelization* is a good way to minimize the energy consumption.

A combination of the techniques is evaluated to see if more energy savings can be gained. Table III summarizes the parameters of the power consumption and energy savings when the algorithm is run in parallel on "low power" and "high performance" modes, calculated against the "low power", single core baseline. The "low power" mode in fact uses more energy to process the same data packet in comparison to the "high performance" mode. Moreover, *Parallelization* offers significant energy savings regardless of which mode is on, with a minimal increase in power to activate the extra cores. For example, by using four cores, in "low power" mode, the single core design uses 55% more energy than its multicore counterpart, with only a 6% increase in power.

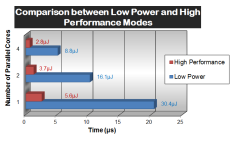

Fig. 7. Modes Comparison on Parallel Implementation

Figure 7 shows the energy used and time needed to decode the data packet received. These can be calculated from the power usage listed in Table III. *Parallelization* causes the chip to use less energy on four cores, giving a total energy savings of 55% and 33% for considering separately the "low power" and "high performance" modes respectively. With these results, it can be concluded that, the more cores deployed, the more energy efficient the Adaptive Switching Algorithm becomes. Instead of having one core running the algorithm for the entire 20 μs, using four cores running at for a quarter of the duration, and shutting them off for 75% of the time would minimize the energy consumption. Furthermore, the more cores being utilized, the more energy can be saved. When combining *Voltage and Frequency Scaling* and *Parallelization* techniques, i.e. comparing one core "low power" mode and "high performance" multicore mode, with energy values of 30.4 μJ and 2.8 μJ respectively, saves a total of more than 80%. This shows that combining the two saving techniques achieves significant combined energy savings.

VI. CONCLUSION

In contrast that running the detector at a slower speed would improve energy consumption [8], when considering the overall power usage, i.e. dynamic and static, the results obtained for the Xilinx®Virtex-7 recommend the Adaptive Switching Algorithm to be run as fast as possible and be put into *Sleep Mode*. Additionally, the benefits of voltage scaling are not significant as the limited voltage scaling range gives a negligible difference in energy consumption. On the other hand, the frequency scaling suggests that the algorithm works best when running at the highest frequency so that it can be put into *Sleep Mode* sooner, conserving energy. In addition, the more cores are used, the faster the task completion, the faster it can be put into idle mode, thus saves significant energy, where more than 60% can be saved.

ACKNOWLEDGMENT

NT is funded by the Universiti Tun Hussein Onn Malaysia.

REFERENCES

[1] G. D. Golden et al., *"Detection Algorithm and Initial Laboratory Results Using V-BLAST Space Time Communication Architecture"*, IEEE Elec. Letters, Vol. 35, No. 1, pp. 14, 1999.
[2] E. Viterbo et al., *"A Universal Lattice Code Decoder for Fading Channels,"*, IEEE Trans. Info Thy, Vol. 45, pp. 1639, 1999.
[3] P. Tan et al., *"The Aplication of Semidefinite Programming for Detection in CDMA,"*, IEEE JSAC, Vol. 19, pp. 1442, 2001.
[4] X. Wu et al., *"A Fixed-Complexity Soft-MIMO Detector via Parallel Candidate Adding Scheme and its FPGA Implementation"*, IEEE Comms. Letters, Vol. 15, No. 2, pp. 241, 2011.
[5] M. Matthaiou et al., *"Reduced Complexity Detection for Ricean MIMO Channels Based on Channel Number Thresholding"*, IEEE PIMRC, pp. 1718, 2008.
[6] L. G. Barbero et al., *"Fixing the Complexity of the Sphere Decoder for MIMO Detection"*, IEEE Trans. Wireless Comms., Vol. 7, No. 6, pp. 2131, 2008.
[7] N. Tadza et al., *"Adaptive Switching Detection Algorithm for Iterative-MIMO Systems to Enable Power Savings"*, accepted by the AGU Radio Science Journal in Aug. 2014. A copy of the journal can be accessed at this website, http://edin.ac/1pomwMd
[8] E. G. Larsson et al., *"The Impact of Dynamic Voltage and Frequency Scaling on Multicore DSP Algorithm Design"*, IEEE Signal Processing Mag., Vol. 28, No. 3, 2011.
[9] E. Seo et al., *"Energy Efficient Scheduling of Real-Time Tasks on Multicore Processors"*, IEEE Trans. Parallel Distributed Systems, Vol. 19, No. 11, pp. 1540, 2008.

References

[1] International Energy Agency, "More Data, Less Energy: Making Network Standby More Efficient in Billions of Connected Devices." `http://www.iea.org/publications/freepublications/publication/MoreData_LessEnergy.pdf`. Accessed January 14, 2015.

[2] F. Wanlass and C. Sah, "Nanowatt Logic using Field-Effect Metal-Oxide Semiconductor Triodes," Technical Report, Xilinx Inc., December 2001.

[3] G. J. Foschini and M. J. Gans, "On the Limits of Wireless Communications in a Fading Environment when using Multiple Antennas," in *IEEE Wireless Personal Communications*, vol. 6, pp. 311–335, March 1998.

[4] E. Telatar, "Capacity of Multi-Antenna Gaussian Channels," in *European Transactions on Telecommunications*, vol. 56, pp. 619–630, December 1999.

[5] A. J. Paulraj and T. Kailath, "Increasing Capacity in Wireless Broadcast Systems using Distributed Transmission/Directional Reception (DTDR)," patent, The Board Of Trustees Of The Leland Stanford Junior University, September 1994.

[6] A. Goldsmith, E. Biglieri, R. Calderbank, A. Constantinides, A. Paulraj, and H. V. Poor, *MIMO Wireless Communications*. Cambridge: Cambridge University Press, 2007.

[7] D. Garrett, L. Davis, S. ten Brink, B. Hochwald, and G. Knagge, "Silicon Complexity for Maximum Likelihood MIMO Detection using Spherical Decoder," in *IEEE Journal on Solid State Circuits*, vol. 39, pp. 1544–1552, September 2004.

[8] C. Studer, A. Burg, and H. Bölcskei, "Soft-Output Sphere Decoding: Algorithms and VLSI Implementations," in *IEEE Journal on Selected Area in Communications*, vol. 26, pp. 290–300, February 2008.

[9] D. Wübben, R. Bohnke, V. Kühn, and K. D. Kammeyer, "MMSE Extension of V-BLAST Based on Sorted QR Decomposition," in *IEEE Proceedings of Vehicular Technology Conference*, vol. 1, pp. 508–512, October 2003.

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

[11] A. Burg, N. Felber, and W. Fichtner, "A 50 Mbps 44 Maximum Likelihood Decoder for Multiple-Input Multiple-Output Systems with QPSK Modulation," in *IEEE International Conference on Electronics, Circuits and Systems*, vol. 1, pp. 332–335, December 2003.

[12] C. P. Schnorr and M. Euchner, "Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems," in *Math Programming*, vol. 1, pp. 181–191, July 1993.

[13] M. O. Damen, H. E. Gamal, and G. Caire, "On Maximum Likelihood Detection and the Search for Closest Lattice Point," in *IEEE Transactions on Information Theory*, vol. 49, pp. 2389–2402, October 2003.

[14] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI Implementation of MIMO Detection using Sphere Decoding Algorithms," in *IEEE Journal of Solid State Circuits*, vol. 40, pp. 1566–1577, July 2005.

[15] K. W. Wong, C. Y. Tsui, R. S. K. Cheng, and W. H. Mow, "A VLSI Architecture of a K-Best Lattice Decoding Algorithm for MIMO Channels," in *IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 273–276, May 2002.

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

[17] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channels," in *International Symposium on Signals, Systems and Electronics*, vol. 1, pp. 295–300, February 1998.

[18] C. Studer and H. Bölcskei, "Soft-Input Soft-Output Sphere Decoding," in *IEEE International Symposium on Information Theory*, vol. 1, pp. 2007–2011, July 2008.

[19] B. M. Hochwald and S. T. Brink, "Achieving Near Capacity on a Multiple-Antenna," in *IEEE International Conference on Communications*, vol. 51, pp. 389–399, March 2003.

[20] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error - Correcting Coding and Decoding: Turbo Codes," in *IEEE International Conference on Communications*, vol. 1, pp. 1064–1070, May 1993.

[21] C. Wang, "On the Performance of Turbo Codes," in *IEEE Proceedings on Military Communications Conference*, vol. 3, pp. 987–992, October 1998.

[22] C. D. Perels, *Frame-Based MIMO-OFDM Systems: Impairment Estimation and Compensation*. PhD thesis, ETH Zürich, Switzerland, October 2007.

[23] S. Häne, *VLSI Circuits for MIMO-OFDM Physical Layer*. PhD thesis, ETH Zürich, Switzerland, October 2008.

[24] P. J. Lüthi, *VLSI Circuits for MIMO Preprocessing*. PhD thesis, ETH Zürich, Switzerland, October 2010.

[25] J. Jaldén and B. Ottersten, "On the Complexity of Sphere Decoding in Digital Communications," in *IEEE Transactions on Signal Processing*, vol. 53, pp. 1474–1484, April 2004.

[26] J. Kermoal, L. Schumacher, K. I. Pederson, P. Mogensen, and F. Frederikson, "A Review of Clock Gating Techniques," in *MIT International Journal of Electronics and Communication Engineering*, vol. 1, pp. 106–114, August 2011.

[27] F. Emnett and M. Biegel, "Power Reduction Through RTL Clock Gating," in *SNUG (Synopsis User Group) Conference in San Jose*, 2000.

[28] M. Dale, "The Power of RTL Clock-gating." `http://www.http://chipdesignmag.com/display.php?articleId=915`. Accessed January 19, 2015.

[29] D. Banerjee, K. Roy, H. Mahmoodi, and S. Bhunia, "Low Power Synthesis of Dynamic Logic Circuits using Fine-Grained Clock Gating," in *Proceedings on Design, Automation and Test: European Design and Automation Association*, vol. 1, pp. 1–6, March 2006.

[30] S. V. Kosonocky, A. J. Bhavnagarwala, K. Chin, G. D. Gristede, A.-M. Haen, W. H. M. B. Ketchen, S. Kim, D. R. Knebel, K. W. Warren, and V. Zyuban, "Low-Power Circuits and Technology for Wireless Digital Systems," in *ISM Journal of Research and Development*, vol. 47, pp. 283–298, April 2003.

[31] R. Puri, L. Stok, and S. Bhattacharya, "Keeping Hot Chips Cool," in *Proceedings on Design Automation Anaheim, California, USA: ACM*, vol. 42, February 2005.

[32] A. Calimera, A. Pullini, A. V. Sathanur, L. Benini, A. Macii, E. Macii, and M. Poncino, "Design of a Family of Sleep Transistor Cells for a Clustered Power Gating Flow in 65 nm Technology," in *17th Great Lakes Symposium on VLSI Stresa-Lago Maggiore, Italy: ACM*, vol. 17, pp. 1–6, January 2007.

[33] H.-O. Kim, Y. Shin, H. Kim, and I. Eo, "Physical Design Methodology of Power Gating Circuits for Standard-Cell-Based Design," in *Proceedings of the 43rd Annual Conference on Design Automation San Francisco, CA, USA: ACM*, vol. 43, July 2006.

[34] J. Henkel and S. Parameswaran, *Designing Embedded Processors: A Low Power Perspective*. Springer Netherlands, 2007.

[35] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "The Limit of Dynamic Voltage Scaling and Insomniac Dynamic Voltage Scaling," in *IEEE Transactions on VLSI Systems*, vol. 13, pp. 1239–1252, November 2005.

[36] E. Seo, J. Jeong, S. Park, and J. Lee, "Energy Efficient Scheduling of Real-Time Task on Multicore Processors," in *IEEE Transactions on Parallel Distributed Systems*, vol. 19, pp. 1540–1552, November 2008.

[37] B. P. Abusaidi, M. Klein, and B. Philofsky, *Virtex-5 FPGA System Power Design Considerations*. Xilinx Inc., 2008.

[38] V. Gutnik and A. P. Chandrakasan, "Embedded Power Supply for Low-Power DSP," in *IEEE Transactions on VLSI Systems*, vol. 5, pp. 425–435, December 1997.

[39] H. P. Su, A. C. Wu, and Y. L. Lin, "A Timing-Driven Soft-Macro Placement and Resynthesis Method in Interaction with Chip Floorplanning," in *Proceedings of the 36th Conference on Design Automation*, vol. 36, pp. 262–267, June 1999.

[40] D. E. Lackey, P. S. Zuchowski, T. R. Bednar, D. Stout, S. Gould, and J. M. Cohn, "Managing Power and Performance for System-on-Chip Designs using Voltage Islands," in *IEEE/ACM International Conference on Computer Aided Design*, vol. 1, pp. 195–202, November 2002.

[41] H. Holma and A. Toskala, *LTE for UMTS OFDMA and SC-FDMA Based Radio Access*. John Wiley and Sons Ltd., 2009.

[42] S. L. Ariyavisitakul, "Turbo Space-Time Processing to Improve Wireless Channel Capacity," in *IEEE Transactions on Communications*, vol. 48, pp. 1347–1359, August 2000.

[43] M. Sellathurai and S. Haykin, "TURBO-BLAST for Wireless Communications: Theory and Experiments," in *IEEE Transactions on Signal Processing*, vol. 50, pp. 2538–2546, October 2002.

[44] M. R. McKay, I. B. Collings, A. Forenza, and R. W. Heath, "A Throughput-Based Adaptive MIMO-OFDM Approach for Spatially Correlated Channel," in *IEEE International Conference on Communications*, vol. 1, pp. 1374–1379, June 2006.

[45] Y. L. Chen, C. Z. Zhan, T. J. Jheng, and A. Y. Wu, "Reconfigurable Adaptive Singular Value Decomposition Engine Design for High-Throughput MIMO-OFDM Systems," in *IEEE Transactions on Very Large Scale Integration Systems*, vol. 21, pp. 747–760, April 2013.

[46] P. J. Smith, M. Shafi, and L. M. Garth, "Performance Analysis for Adaptive MIMO SVD Transmission in a Cellular System," in *IEEE Transactions on Parallel Distributed Systems*, vol. 1, pp. 49–54, February 2006.

[47] J. Sarrazin, Y. Mahe, S. Avrillon, and S. Toutain, "On the Performance of Adaptive MIMO Systems using Radiation Pattern Reconfigurable Antennas," in *IEEE International Symposium Antennas and Propogation Society*, vol. 1, pp. 1–4, July 2008.

[48] X. Wu, J. S. Thompson, and A. M. Wallace, "An Improved Sphere Decoding Scheme for MIMO Systems using an Adaptive Statistical Threshold," in *17th European Signal Processing Conference*, vol. 1, pp. 2668–2672, August 2009.

[49] J. S. Chung, *Fast Power Allocation Algorithm for Adaptive MIMO Systems*. PhD thesis, University of Canterbury, October 2009.

[50] L. Xian and H. Liu, "An Adaptive Power Allocation Scheme for Space-Time Block Coded MIMO Systems," in *IEEE Wireless Communications and Networking Conference*, vol. 1, pp. 504–508, March 2005.

[51] X. Yu and H. Shi, "Performance Analysis of Variable-Power Adaptive Modulation in Space-Time Block Coded MIMO Diversity Systems," in *SCIENCE CHINA Information Sciences Press*, vol. 53, pp. 2106–2115, June 2010.

[52] A. Forenza, M. R. McKay, A. Pandharipande, and R. W. Heath, "Adaptive MIMO Transmission for Exploting the Capacity of Spatially Correlated Channels," in *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 619–630, March 2007.

[53] C. Kim and J. Lee, "Dynamic Rate-Adaptive MIMO Mode Switching between Spatial Multiplexing and Diversity," in *Journal on Wireless Communications and Networking*, vol. 1, pp. 1–12, July 2012.

[54] J. L. Yu, C. L. Hung, and I. T. Lee, "A Two-Stage Partially Adaptive Linear Receiver for CDMA MIMO Systems with Alamouti's Space-Time Block Codes," in *Journal of Digital Signal Processing*, vol. 17, pp. 244–260, January 2007.

[55] E. K. S. Au, C. Wang, S. Sfar, R. D. Murch, W. H. Mow, V. K. N. Lau, R. S. Cheng, and K. B. Letaief, "Error Probability for MIMO Zero-Forcing Receiver with Adaptive Power Allocation in the Presence of Imperfect Channel State Information," in *IEEE Signal Processing Magazine*, vol. 28, pp. 127–144, May 2011.

[56] R. C. de Lamare and R. Sampaio-Neto, "Blind Adaptive MIMO Receivers for Space-Time Block-Coded DS-CDMA Systems in Multipath Channels using Constant Modulus Criterion," in *IEEE Transactions on Communications*, vol. 58, pp. 21–27, January 2010.

[57] D. Banerjee, S. Devarakond, S. Sen, and A. Chatterjee, "Real-Time Use-Aware Adaptive MIMO RF Receiver Systems for Energy Efficiency under BER Constraints," in *IEEE Conference in Design Automation*, vol. 1, pp. 1–7, June 2013.

[58] M. Matthaiou, D. I. Laurenson, and C. X. Wang, "Reduced Complexity Detection for Ricean MIMO Channels Based on Channel Number Thresholding," in *IEEE International Proceedings of Wireless Communications and Mobile Computing Conference*, pp. 1718–1722, August 2008.

[59] Z. Wang, Y. Tan, and Y. Wang, "Low Hardware Complexity Parallel Turbo Decoder Architecture," in *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. II–53–II–56, May 2003.

[60] I. Berenguer, X. Wang, and V. Krishnamurthy, "Adaptive MIMO Antenna Selection via Discrete Stochatic Optimization," in *IEEE Transactions on Signal Processing*, vol. 53, pp. 4315–4328, November 2005.

[61] J. H. Winters, S. Member, J. Salz, and R. D. Gitlin, "The Impact of Antenna Diversity on the Capacity of Wireless Communication Systems," in *IEEE Transactions on Communications*, vol. 42, pp. 1740–1751, April 1994.

[62] G. Li, X. Zhang, S. Lei, C. Xiong, and D. Yang, "An Early Termination-Based Improved Algorithm for Fixed-Complexity Sphere Decoder," in *IEEE Wireless Communications and Networking Conference: PHY and Fundamentals*, vol. 1, pp. 629–634, April 2012.

[63] B. U. Fincke and M. Pohst, "Improved Methods for Calculating Vectors of Short Length in a Lattice, including a Complexity Analysis," in *Mathematics of Computations*, pp. 463–471, June 1985.

[64] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore and London: The Johns Hopkins University Press, 1983.

[65] G. D. Golden, C. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Achieving Near Capacity on a Multiple-Antenna," in *IEEE International Conference on Communications*, vol. 51, pp. 389–399, March 2003.

[66] T. L. Marzetta and B. M. Hochwald, "Capacity of a Mobile Multiple-Antenna Communication Link in Rayleigh Flat Fading," in *IEEE Transactions on Information Theory*, vol. 45, pp. 139–157, January 1999.

[67] C. Shen, H. Zhuang, L. Dai, and S. Zhou, "Detection Algorithm Improving V-BLAST Performance over Error Propogation," in *IEEE Electronics Letters*, vol. 39, pp. 1007–1008, June 2003.

[68] G. J. Foschini, G. D. Golden, R. A. Valenzuela, and P. W. Wolniansky, "Simplified Processing for High Spectral Efficiency Wireless Communication Employing Multi-Element Arrays," in *IEEE Journal in Selected Areas of Communications*, vol. 17, pp. 1841–1852, November 1999.

[69] S. Loyka, "V-BLAST Outage Probability: Analytical Analysis," in *IEEE Proceedings in Vehicular Technology Conference*, vol. 4, pp. 24–28, September 2002.

[70] X. Wu and J. S. Thompson, "A Fixed-Complexity Soft-MIMO Detector via Parallel Candidate Adding Scheme and its FPGA Implementation," in *IEEE Communications Letters*, vol. 15, pp. 241–243, February 2011.

[71] E. Viterbo and J. Boutros, "A Universal Lattice Code Decoder for Fading Channels," in *IEEE Transactions on Information Theory*, vol. 45, pp. 1699–1642, August 1999.

[72] M. Pohst, "On the Computation of Lattice Vectors of Minimal Lengths, Successive Minima and Reduced bases with Applications," in *Newsletter ACM SIGSAM Bulletin*, vol. 15, pp. 37–44, February 1981.

[73] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest Point Search in Lattices," in *IEEE Transactions on Information Theory*, vol. 48, pp. 2201–2214, August 2002.

[74] Y. K. Chen, C. Chakrabarti, and B. Bougard, "Signal Processing on Platforms with Multiple Cores: Part 2 - Applications and Designs," in *IEEE Signal Processing Magazine*, vol. 2, pp. 20–21, March 2010.

[75] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark Silicon and the End of Multicore Scaling," in *Proceedings of the 38th Annual International Symposium on Computer Architecture*, pp. 365–370, June 2011.

[76] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen, "Single ISA Heterogeneous Multicore Architectures : The Potential for Processor Power Reduction," in *Proceedings of the 36th Annual International Symposium on Microarchitecture*, vol. 1, p. 81, December 2003.

[77] L. G. Barbero and J. S. Thompson, "Extending a Fixed-Complexity Sphere Decoder to Obtain Likelihood Information for Turbo-MIMO Systems," in *IEEE Transactions on Vehicular Technology*, vol. 57, pp. 2804–2810, September 2008.

[78] L. G. Barbero, T. Ratnarajah, and C. Cowan, "A Low Complexity Soft-MIMO Detector Based on the Fixed Complexity Sphere Decoder," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 2669–2672, April 2008.

[79] S. Lei, Q. Tu, D. Yang, and J. Chen, "Probabilistic Tree Pruning for Fixed-Complexity Sphere Decoder in MIMO Systems," in *International Conference on Wireless Communications and Signal Processing*, vol. 1, pp. 1–6, October 2010.

[80] L. Liu, J. Lofgren, and P. Nilsson, "Low Complexity Soft-Output Signal Detector for Spatial-Multiplexing MIMO System," in *IEEE International Wireless Communications and Mobile Computing Conference*, vol. 1, pp. 988–993, September 2011.

[81] G. Li, X. Zhang, S. Lei, C. Xiong, and D. Yang, "An Early Termination-Based Improved Algorithm for Fixed-Complexity Sphere Decoder," in *IEEE Wireless Communications and Networking Conference: PHY and Fundamentals*, vol. 1, pp. 629–634, April 2012.

[82] J. Zhang, M. A. Armand, P. Y. Kam, and T. Mi, "A Mutual Information Approach for Comparing LLR Metrics for Iterative Decoders," in *IEEE International Conference on Communications*, vol. 1, pp. 1–4, June 2009.

[83] M. Klein, "Power Consumption at 40 and 45 nm," in *Xilinx Inc. White Paper : WP298*, vol. 298, pp. 1–21, April 2009.

[84] M. Čirkić, D. Persson, and E. Larson, "Allocation of Computational Resources for Soft-MIMO Detection," in *IEEE Journal of Selected Topics on Signal Processing*, vol. 5, pp. 1451–1461, December 2011.

[85] A. Andrei, P. Eles, Z. Peng, S. Link, M. Schmitz, and B. M. Al-Hashimi, "Energy Optimization of Multiprocessor Systems on Chip by Voltage Selection," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, pp. 1451–1461, April 2009.

[86] M. E. Salehi, M. Samadi, M. Najibi, A. Afzali-Kusha, M. Pedram, and S. M. Fakhraie, "Dynamic Voltage and Frequency Scheduling for Embedded Processors Considering Power/Performance Trade-offs," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, pp. 1931–1935, August 2011.

[87] E. G. Larson and O. Gustafsson, "The Impact of Dynamic Voltage and Frequency Scaling on Multicore DSP Algorithm Design," in *IEEE Signal Processing Magazine*, vol. 28, pp. 127–144, May 2011.

[88] M. Z. Hasan and M. Bird, "Energy Reductions for Embedded Processors in Reconfigurable Hardware," in *IEEE Conference on Electro/Information Technology*, vol. 1, pp. 1–8, May 2011.

[89] T. Nakamura, "Trends in Small Cell Enhancements in LTE Advanced," in *IEEE Communication Magazine*, vol. 51, pp. 98–105, February 2013.

[90] Y. An, Y. Xiao, P. Wu, and X. Gao, "Research on Improving Throughput of Wireless MIMO Networks," in *International Conference on Information Computer Applications*, vol. 7030, pp. 258–265, February 2011.

[91] G. Yue, N. Prasad, M. Jiang, M. Khojastepour, and S. Rangarajan, "Improving Downlink Multiuser MIMO Througput in LTE-Advanced Cellular Systems," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, pp. 2009–2013, September 2011.

[92] S. Joshi, P. C. Bapna, and n. Kothari, "Bit Error Rate Performance of MIMO Channels for Various Modulation Schemes using Maximum Likelihood Detection Technique," in

Special Issue Edition of International Journal of Computer Applications, vol. 1, pp. 15–18, December 2011.

[93] V. W. Sonone and N. B. Chopade, "Techniques for Improving BER and SNR in MIMO Antenna for Optimum Performance," in *International Journal of Electrical and Electronics Engineering*, vol. 1, pp. 11–14, May 2014.

[94] A. Jemmali, J. Conan, and M. Torabi, "Bit Error Rate Analysis of MIMO Schemes in LTE Systems," in *International Conference on Wireless and Mobile Communications*, vol. 1, pp. 190–194, December 2013.

[95] F. Héliot, M. A. Imran, and R. Tafazolli, "On the Efficiency Gain of MIMO Communication under Various Power Consumption Models," in *Conference Proceedings of Future Networks and Mobile Summit*, vol. 1, pp. 1–9, June 2011.

[96] A. S. Gowda and A. Goldsmith, *MIMO Wireless Communications*. Cambridge: Cambridge University Press, 2011.

[97] A. He, S. Srikanteswara, K. K. Bae, T. R. Newman, J. H. Reed, W. H. Tranter, M. Sajadieh, and M. Verhelst, "Power Consumption Minimization for MIMO Systems - A Cognitive Radio Approach," in *IEEE Journal on Selected Areas in Communication*, vol. 29, pp. 469–479, February 2011.

[98] H. Yu, "Power Management of MIMO Network Interfaces on Mobile Systems," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, pp. 1175–1186, June 2012.

[99] N. Horner, A. Kwasinski, and A. Mondragon, "Improving the Performance of DSP Systems for MIMO Processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 1681–1684, May 2011.

[100] H. Bölcskei, "Digital Signal Processing Challenges in MIMO Wireless Communications," in *IEEE Workshop on Signal Processing Systems*, vol. 1, pp. 1–9, September 2001.

[101] T. N. Canh, "Implementation of a MIMO-OFDM System Based on the TI C64x+ DSP," in *IEEE International COnference on Ubiquitous Information Management and Communication*, vol. 1, pp. 1–6, January 2013.

[102] Z. Guo and P. Nilsson, "The VLSI Architecture of the Soft-Output Sphere Decoder for MIMO Systems," in *IEEE International Proceedings Midwest Symposium Circuit Systems*, vol. 1, pp. 1195–1198, May 2005.

[103] M. O. Damen, A. Chkeif, and J. C. Belfiore, "Lattice Code Decoder for Space-Time Codes," in *IEEE Communication Letters*, vol. 4, pp. 161–163, May 2000.

[104] M. M. Khan, D. R. Lester, A. R. L. A. Plana, X. Jin, E. Painkras, and S. B. Furber, "SpiNNaker: Mapping Neural Networks onto a Massive Parallel Chip Multiprocessor," in *IEEE International Joint Conference on Neural Networks and World Congress on Computational Intelligence*, vol. 1, pp. 2850–2857, June 2008.

[105] R. Jayaraman, "When Will FPGAs Kill ASICs?," Technical Report, Xilinx Inc., December 2001.

[106] A. DeHon, "The Density Advantage of Configurable Computing," in *IEEE Journal on Computer*, vol. 33, pp. 41–49, April 2002.

[107] X. Huang, C. Liang, and J. Ma, "System Architecture and Implementation of MIMO Sphere Decoders on FPGA," in *IEEE Transactions on Very Large Scale Integration Systems*, vol. 16, pp. 188–197, February 2008.

[108] J. S. Park and T. Ogunfunmi, "Efficient FPGA-Based Implementations of MIMO-OFDM Physcial Layer," in *International Conference of Circuits Systems Signal Process*, vol. 3, pp. 1–25, April 2012.

[109] Xilinx Inc., "Embedded Design with the PowerPC 440 Processor Core." `http://www.xilinx.com/training/embedded/embedded-design-with-powerpc-440-video.htm`. Accessed November 18, 2014.

[110] Xilinx Inc., "Embedded Design with the MicroBlaze Soft Processor Core." `http://www.xilinx.com/training/embedded/embedded-design-with-microblaze-video.htm`. Accessed November 18, 2014.

[111] Altera, "Nios II Processor: The World Most Versatile Embedded Processor." `http://www.altera.co.uk/devices/processor/nios2/ni2-index.html`. Accessed November 18, 2014.

[112] Xilinx Inc., "Virtex-5 FPGA User Guide." `http://www.xilinx.com/support/documentation/user_guides/ug190.pdf`. Published on March 16, 2012.

[113] Xilinx Inc., "7 Series FPGAs Overview." `http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf`. Published on October 8, 2014.

[114] N. Tadza, D. I. Laurenson, and J. S. Thompson, "Adaptive Switching Detection Algorithm for Iterative-MIMO Systems to Enable Power Savings," in *Journal of Radio Science*, vol. 49, pp. 1065–1079, November 2014.

[115] Xilinx Inc., "System Generator for DSP." `http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/sysgen_user.pdf`. Published on April 24, 2012.

[116] A. Telikepalli, *Power vs. Performance: The 90 nm Inflection Point in Reducing Power in FPGAs: The Triple Challenge*. Xilinx Inc., 2006.

[117] N. Instruments, *Advantages of the Virtex-5 FPGA*. National Instruments, 2010.

[118] The ARM Team, *The ARM Cortex-A9 Processors*. The ARM Industry, 2009.

[119] W. Kim, M. S. Gupta, G. Wei, and D. Brooks, "System Level Analysis of Fast, Per-Core DVFS using On-Chip Switching Regulators," in *IEEE 14th International Symposium*, vol. 16, pp. 123–134, February 2008.

[120] J. Hussein, M. Klein, and M. Hart, *Lowering Power at 28 nm with Xilinx 7 Series Devices*. Xilinx Inc., 2013.

[121] G. J. Foschini, "Layered Space-Time Architecture for Wireless Communication in a Fading Environment when using Multi-Element Antennas," in *Journal of Bell Laboratory Technology*, vol. 1, pp. 41–59, August 2002.

[122] D. S. Shiu, G. J. Foschini, M. J. Gans, and J. M. Khan, "Fading Correlation and its Effect on the Capacity of Multi-Element Antenna Systems," in *IEEE Transactions on Communications*, vol. 48, pp. 502–513, March 2000.

[123] D. Wübben, V. Kühn, and K. D. Kammeyer, "On the Robustness of Lattice Reduction Aided Detectors in Correlated MIMO," in *60th IEEE Proceedings of Vehicular Technology Conference*, vol. 5, pp. 3639–3643, September 2004.

[124] Q. Meng, Z. Pan, X. You, and Y. H. Kim, "On Performance of Lattice Reduction Aided Detection in the Presence of Receive Correlation," in *IEEE 6th Proceedings of Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communications, Transactions on Information Theory*, vol. 1, pp. 89–92, June 2004.

[125] L. G. Barbero and J. S. Thompson, "Performance of Complex Sphere Decoder in Spatially Correlated Channel," in *Journal of Institution of Engineering and Technology*, vol. 1, pp. 122–130, February 2007.

[126] N. Tadza, J. S. Thompson, and D. I. Laurenson, "Power Performance Analysis of the Iterative-MIMO Adaptive Switching Algorithm Detector on the FPGA Hardware," in *IEEE Conference of Vehicular Technology*, vol. 81, p. ??, May 2015.

[127] D. Bokolamulla and T. Aulin, "A New Stopping Criterion for Iterative Decoding," in *IEEE on Communication Society*, vol. 1, pp. 538–541, January 2004.

[128] F. Zhai and I. Fair, "A New Error Detection Techniques and Stopping Criteria for Turbo Decoding," in *Proceedings 2000th Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 58–62, March 2000.

[129] N. Y. Yu, M. G. Kim, Y. S. Kim, and S. U. Chung, "Efficient Stopping Criterion for Iterative Decoding of Turbo Codes," in *IEEE Electronics Letters*, vol. 39, pp. 73–75, January 2003.

[130] C. Gimmler, T. Lehnigk-Emden, and N. Wehn, "Low-Complexity Iteration Control for MIMO-BCIM Systems," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 21, pp. 241–246, September 2010.

[131] J. G. Proakis, *Digital Communications (3rd Edition)*. Singapore: McGraw Hill, 1983.

[132] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization," in *IEEE Communications Magazine*, vol. 35, pp. 90–100, July 1997.

[133] J. Kermoal, L. Schumacher, K. I. Pederson, P. Mogensen, and F. Frederikson, "A Stochastic MIMO Radio Channel Model with Experimental Validation," in *IEEE Journal on Selected Areas of Communications*, vol. 20, pp. 1211–1226, May 2002.

[134] K. Yu, M. Bengtsson, B. Ottersten, D. McNamara, P. Karlsson, and M. Beach, "Modelling of Wide-Band MIMO Radio Channels Based on NLoS Indoor Measurements," in *IEEE Transactions on Vehicular Technology*, vol. 53, pp. 655–665, June 2004.

[135] A. M. Tulino, A. Lozano, and S. Verdu, "Impact of Antenna Correlation on the Capacity of Multiantenna Channels," in *IEEE Transactions on Information Theory*, vol. 51, pp. 2491–2509, July 2005.

[136] H. Özcelik, M. Herdin, W. Weichselberger, J. Wallace, and E. Bonek, "Deficiencies of 'Kronecker' MIMO Radio Channel Model," in *IEEE Electronics Letters*, vol. 36, pp. 1209–1210, August 2003.

[137] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," in *IEEE Transactions on Information Theory*, vol. 42, pp. 429–445, March 1996.

[138] 3GPP Technical Team, "3GPP Technical Specifications Series 36 for E-UTRA (Release 8)." `http://www.3gpp.org`. Published on August 8, 1999.

[139] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative Decoding for MIMO Channels via Modified Sphere Decoder," in *IEEE Transactions on Wireless Communications*, vol. 3, pp. 2299–2311, June 2004.

[140] R. C. de Lamare and R. Sampaio-Neto, "Minimum Mean-Squared Error Iterative Successive Parallel Arbitrated Decision Feedback Detectors for DS-CDMA Systems," in *IEEE Transactions on Communications*, vol. 56, pp. 778–789, May 2012.

[141] C. Hermocilla and C. Szczecinski, "Exit Charts for Turbo Receivers in MIMO Systems," in *IEEE International Symposium on Signal Processing and its Applications*, vol. 1, pp. 209–212, July 2003.

[142] W. Li and H. Dai, "EXIT Chart Analysis of Turbo-BLAST Receivers in Rayleigh Fading Channels," in *IEEE Vehicular Technology Conference*, vol. 1, pp. 1396–1400, September 2004.

[143] H. Chen, R. G. Mauder, and L. Hanzo, "An Exit-Chart Aided Design Procedure for Near-Capacity N-Component Parallel Concatenated Codes," in *IEEE Global Telecommunications Conference*, vol. 1, pp. 1–5, December 2010.

[144] J. Geier, "How to: Define Minimum SNR Values for Signal Coverage." `http://www.wireless-nets.com/resources/tutorials/define_SNR_values.html`. Accessed April 20, 2015.