ABSTRACT

JENKAL, RAVI SOMNATH Testbed Design for Throughput Characterization of a Multiple Antenna Structure. (Under the direction of Prof. William Rhett Davis).

Multiple Input Multiple Output (MIMO) schemes have generated a lot of interest due to their ability to efficiently use the limited spectral resources available to users around the world. The MIMO concept has afforded work in many subtending fields of research, such as channel modeling, signaling schemes and receiver designs, coding and adaptability of MIMO to existing and future standards, which aim towards making this technique a more feasible and popular choice.

One of the interesting fields of study that has spawned from MIMO is the building of testbeds and prototypes of MIMO systems. The need for testbed creation comes from the fact that there is a need for validation of results generated by theoretical analysis. The use of testbeds and prototypes of a system can help characterize the gains of a proposed solution and hence can provide an initial sanity check for the solution at hand. It also provides an initial cost estimate which would have a major bearing on the marketability of a given idea. An end-to-end implementation would enable testing of different blocks of the design independently and the evaluation of their effect in conjunction with methods being used in other parts of the design, thus giving a much needed indication of system design complexities which go a long way in determining the design choice and hence the overall design architecture.

This work aims to implement such a system to enable the performance evaluation of a newly proposed antenna design as a means of providing an example solution to the prototyping concept. This document focuses on the core ideas that constitute a non-realtime implementation, consisting of the design of a simplified RF frontend and a software-defined backend. The applicability of this implementation in providing sufficient measurement information to characterize the solutions for MIMO is justified.

Testbed Design for Throughput Characterization of a Multiple Antenna Structure

by

Ravi Somnath Jenkal

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Master of Science

Department of Electrical and Computer Engineering

Raleigh

2004

Approved By:

Prof. Kevin Gard	Prof. Gianluca Lazzi

Prof. William Rhett Davis
Chair of Advisory Committee

To

To Mom, Dad and Vijey

Biography

Ravi Jenkal was born on 28^{th} July 1980 in Mangalore, India. He received his Bachelor of Engineering (B.E) Degree in Electronics and Communication Engineering from Sri Jayachamarajendra College of Engineering (S.J.C.E), Vishveshwaraiah Technological University in 2002. In Fall 2002, he began his graduate studies in the Electrical and Computer Engineering Department at North Carolina State University, Raleigh, NC. He is a student member of the Institute of Electrical and Electronics Engineers (IEEE). Since Spring 2003, he has been working in the Methodologies for User-friendly System-on-a-chip Experimentation (MUSE) group of Prof. Rhett Davis in the field of multiple-antenna system architecture issues. He will be continuing on for his doctoral degree from Fall 2004.

He has also taken keen interest in community participation and has been the president of the NC State Indian Graduate Student Association called MAITRI. Moreover, he has actively participated in volunteering for the Office of International Scholar and Student Services and other affiliated organizations.

Acknowledgements

Above all I would like to thank my parents for the invaluable gift of life and love that they have showered upon me all through. My brother, Vijey, has been the best role-model that I could have asked for. I donot need to look much beyond them for inspriation, determination and achievement. I am truly blessed.

I sincerely thank my advisor Prof. Rhett Davis for giving me an opportunity to work for him. He is an unceasing source of enthusiasm and a great person to work with. There are an innumerable things to learn from him and I hope to do just that as I go further in my learning experience. I will always be thankful for his patience with me.

I also thank Prof. Gard for his tireless patience in answering all my questions on the frontend design and also thank Prof. Lazzi for allowing me to use his antenna design for the project.

I greatly appreciate the help given to me by Ambarish Sule and Hao Hua, my research group mates, for showing me what dedicated work is all about and helping me with design and simulation issues along the way.

A special thanks to my closest friends (very-closeans for those who know us) for letting me work in peace and not having too much fun when I was working on my thesis. My roommates- Mithun, Mitul and Yathi with my friendly neighbors - Shalu, Tanu, Ramya, Anu and Ratna I am indebted to for reasons of nutrition, entertainment and an unnatural ability to put up with me. Thanks to Liz for giving me evening distractions with coffee breaks. They ensured my sanity in their own special way. They have made me feel at home 8000 miles away from it.

Contents

List of Figures

Chapter 1

INTRODUCTION

The wireless industry is on a continual quest for higher throughput communications. The most important constraint in this exercise lies in the availability of spectral resources. Thus there has always been an interest in methods that would lead to the goal of throughput improvement without necessitating an equivalent increase in the spectral requirements. This criterion is met by a class of algorithms called MIMO (Multiple Input Multiple Output).

MIMO can be defined simply. As defined by Gesbert, et all [14], in this concept of communications, both the transmitter and the receiver are equipped with more than one antenna (usually equal in number). In this case the signals on the transmitter and the receiver are "combined" in such a manner that the quality (Bit Error Rate) or the data rate of the communication for each user is improved.

The core idea of MIMO systems is Space-Time signal processing which adds the spatial dimension to the time dimension that is inherent in any digital communication data. This spatial dimension comes from the use of multiple antennas in different configurations. The most important consequence of the use of MIMO concepts is the conversion of multipath fading, traditionally a pitfall of wireless communication

due to its introduction of phase and amplitude distortions, into a requirement for spatial signature generation. The tremendous possibilities of MIMO has made it one of the most widely researched ideas in recent times with publications coming out on different aspects of multiple antenna communications. A special issue of IEEE Journal on Selected Areas in Communications on MIMO in April 2003 would shed greater light on this concept and its research directions. This is because of the recognition of MIMO as an emerging area with a possibility of contributing greatly to changing the face of wireless communications, provided the bottlenecks of hardware implementation in a cost effective manner are solved. Some of the areas of research that have a direct bearing on the feasibility of MIMO communications are channel modeling, information theory and coding, signal processing and antenna design.

Though there has been a lot of work going on the theoretical side of MIMO, the implementation side has not reached the same rate of growth [14]. There are a number of issues such as development of channel models, adaptation to standards and the complexity of hardware implementations for MIMO detection that, if overcome, can make this technology commercially viable [14, 4].

As a general case, it is to be noted that though there may exist methods in theory that would improve the quality in a field of interest, the final usefulness of the above methods are contingent upon their feasibility in real world scenarios and adaptability to standards. Thus it is very important that there be enough thought given to the marketable applications of concepts and the ability to test real-world performance of systems. The first step towards this is the generation of testbeds that would create an end-to-end wireless link and thus enable checks to be done along with design while dealing with the varied areas of multiple antenna communication and keep cost factors in mind too. It is precisely this that we attempt to do here. The aim of this work is to generate a reconfigurable testbed encompassing both the one time design RF frontend system for acquisition of data from the wireless channel and reconfigurable complex digital (combined hardware-software approach) backend that would perform digital data acquisition, Space-time coding, MIMO detection (information

estimation), channel encoding/decoding and error checking functionalities. All this would be done with a simplified RF frontend and a software-defined backend.

As has been stated earlier, one of the areas of interest is that of antenna design. The number of antennas used has a bearing at multiple levels. The power requirements of the system increases linearly with the number of antennas as for each antenna there is a need for an entire RF frontend to be replicated. Also, area resources are also an important criterion since the target market is one based on ease of portability. A good antenna design can also relax the constraints on the design of the rest of the system when considering the link budget. A proper antenna design can provide a greater degree of decorrelation in the MIMO channels which effectively gives a greater number of independent channels. [21] is an attempt at designing an antenna structure which uses the concepts of co-location and co-polarization to increase the capacity of a given channel. We attempt here to design and build a system that, to start off with, would characterize this new antenna structure and check its improvement over present antenna configurations used. By this process, the final aim is to understand the issues related to an implementation of such nature.

1.1 Thesis Organization

Chapter 2 describes the principle behind MIMO, its fundamental assumptions and the options available for the implementation of the different blocks of a general MIMO system. Chapter 3 then deals in greater depth with the design that we have aimed to build. There are also justifications for the design choices for the different macro-level entities of the design. We then move on, in chapter 4, to the implementation details of the ML-APP algorithm and a comparative study of this algorithm, in terms of its performance and hardware complexity, with the other options i.e. ZF, MMSE and nulling-and-cancelling approaches. Chapter 5 then deals with the turbo coding implementation in the system wherein the justification for its use and the encoding and decoding algorithms are explained. The next block to be dealt with is the RF

system block. This is more of an overview of the choice between the Direct-conversion architecture and the other options available. The Data Acquisition Module is then described where we deal with the FSMs within the system that are used to perform serial communication. We then consider the testing of the blocks already designed and draw conclusions from the results obtained.

1.2 Related work

As has been stated earlier, the development of implementations for the MIMO concept has not been able to match the theoretical work in this field. This is precisely what keeps MIMO from becoming the dominant technology in wireless data transfers. There has been work on the development of full-custom design solutions for MIMO detection and decoding [18, 17]. Lucent has also come out with a Maximum Likelihood detector chip [13] for the HSDPA extension of 3GPP. Most of the work in the design of full-custom solutions has been the result of a race towards providing solutions that would work with the next generation standards, for example, 3G UMTS. But the central idea to prototyping and testbed design is that there might be a simpler solution to a given problem than one that is presently available, which requires a platform for testing. Full-custom solutions are not economically viable solutions to this problem. Being a widely researched area, there is always going to be a constant stream of possible solutions which need to be chosen from and further worked on. Thus, prototyping and testbed creation remains an important area of research in MIMO. Initial experimental results [7, 15] show working of concept. The need for reconfigurability has ensured the use of DSPs and FPGAs. Some interesting papers on prototyping and hardware testbed creation show the general methodology used in creation of such platforms. [7, 16, 35, 9].

1.3 Contribution

This thesis demonstrates the idea of testbed design for MIMO. The major focus is on determining the design choices available. We will also then have a more in-depth look at certain examples, keeping in mind the scenario of characterizing the antenna of interest. A central motivation in this implementation is to enable an efficient characterization at low cost in an end-to-end and modular manner. Most of the implementations such as [35, 7] are high cost ventures.

Chapter 2

MIMO OVERVIEW

MIMO got its first major exposure through a press release by Lucent [1] wherein they announced that their scientists had been successful in achieving spectral efficiencies in excess of that predicted by Shannon. This was not actually a breakthrough that happened overnight. It was a natural progression that was borne out of the concept of diversity. Diversity in reception (or equivalently SIMO) uses multiple receive antenna elements to improve the reliability of reception in a fading environment. This is attributed to the fact that, in the presence of multiple receive antennas, there are multiple paths for receiving the same signal and thus, even if severe fading is present in one of the paths, it would not affect all the paths at the same time. This increases the receiver's ability to detect signals. But the requirement is that of increased hardware at the users end. This can be shifted to the base-station side giving MISO requiring preprocessing or coding of the information before transmission to enable the receiver to separate out the different streams of information transmitted. An increase in the number of transmitting antennas increases the bit rate for a given frequency of operation. The end result had to be MIMO by adding multiple antennas at the receiver as well [12].

A general MIMO system is shown in figure 2.1 There are no hardware details of the system in the figure shown which we will discuss later in chapter 3. The most

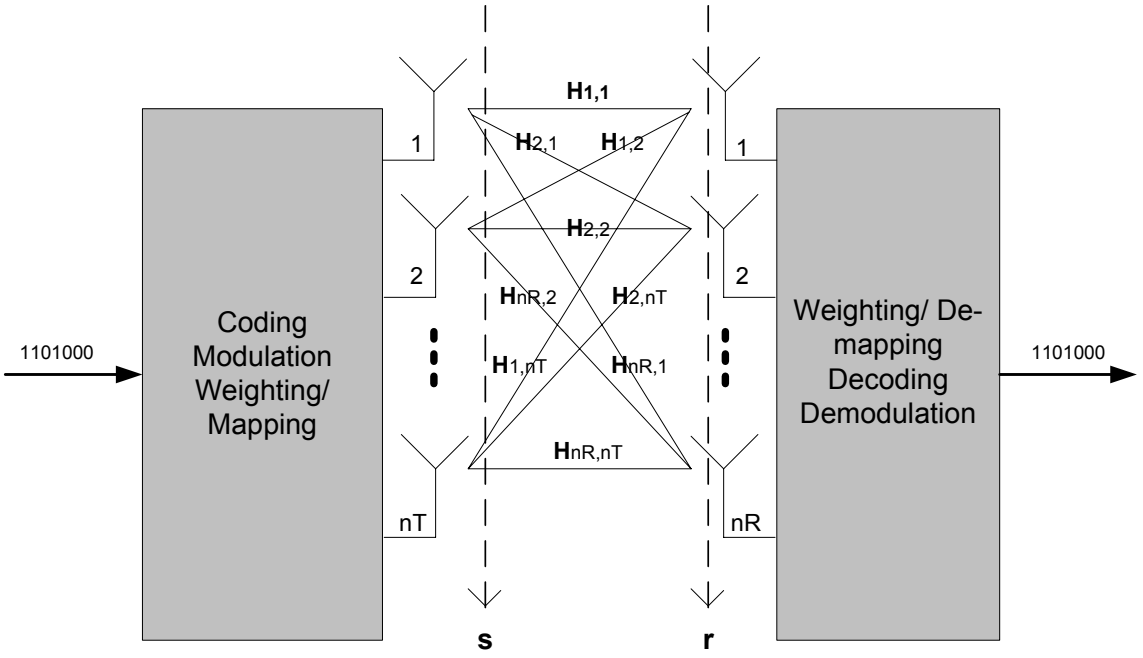

Figure 2.1: General MIMO system model

distinct and obvious difference from a standard communication system is the use of multiple antennas for transmission and reception. It is on the basis of this system architecture that MIMO as a concept is built.

To understand the reasons behind this technology, it is first necessary to talk about the transmission model. Due to the use of multiple antennas, the transmission is done as a vector of symbols. The mathematical analysis takes a vector form. This leads to multiple variants of gain like diversity gain, multiplexing gain, array again [12]. Let us first look at the transmission model.

Terminals at each end of the link have multiple antenna elements:nT at the transmitter and nR at the receiver. If the channel is subject to fading, the propagation between each pair of transmit/receive antennas (say from transmit antenna k to receive antenna i) will be a random (complex) variable, $\mathbf{H_{ik}}$.

In this implementation, we assume an indoor environment which has relatively small variation with time. Under such an assumption, the channel is taken to have slow and flat fading. By flat fading, we assume that the effect of the environment does not vary with the frequency, i.e., all the frequencies received would have undergone the same type and proportion of variation. This enables representation as a scalar complex value. Moreover, since the environment is assumed to be varying only a little bit at a time, i.e., people moving sometimes but in general the environment being the same, the channel can be assumed to be slow fading i.e. its characteristics do not change vary often. This assumption enables the same coefficients of the channel to be used over a period of time. The complex representation of the coefficient represents the presence of both amplitude and phase shifting of the signals transmitted. Also, since the distances tend to be small between the transmit and receive paths, the delay spread of the signals received is very small compared to the symbol interval. We normally model this channel as a matrix, $\mathbf{H}$, while the signals at the transmitter and receiver are treated as complex (column) vectors, $\mathbf{s}$ and $\mathbf{r}$, of size nT and nR respectively (in most cases $nT = nR$). Then we can write:

$$\mathbf{r} = \mathbf{Hs} + \mathbf{n};$$

where $\mathbf{n}$ is a size nR column vector of noise associated with each receive antenna. (It is to be noted that each antenna has all the required components like LNA, power amplifier and mixer, separate from the other antenna elements). Each entry in $\mathbf{H}$ corresponds to a transmit-receive pair.

As seen from the above equation, the channel is now a matrix instead of a vector and this brings with it a number of advantages. It has been shown that under certain conditions one can transmit $\min(nR, nT)$ independent data streams over the eigenmodes of the channel matrix [14]. Let us look at why this is possible using an analogy to solving a problem of linear equations.

Let us assume that the number of transmit and receive antennas are the same. As can be seen from the model previously shown there is a high bit rate stream

coming into the system at the transmitter side. This high bitrate stream is then split into sub-streams equal to the number of transmission antennas. These are then transmitted over the same channel where they mix with each other and with the interferers in the environment. It is to be noted that each stream is transmitted over the same frequency band. At the receiver, each antenna receives the signals transmitted by each transmit antenna. Also, the channel is estimated and hence all the coefficients in the channel matrix are known. At the same time the information bits are separated and estimated. Now, it can be seen that we have a system of linear equations where we have nT number of unknowns and nT equations. The separation of the unknowns is only possible if the coefficients are independent which translates to the channel having a different effect on the signal transmitted from each antenna and received by each antenna. This "effect" is brought about by multipath fading.

In wireless systems, radio waves bounce and scatter randomly off objects in the environment. This scattering is known as multipath, as it results in multiple copies (or "images") of the transmitted signal arriving at the receiver via different scattered paths. In conventional wireless systems, multipath represents a significant impediment to accurate transmission, because the images arrive at the receiver at slightly different times and can thus interfere destructively, cancelling each other out. However, MIMO techniques exploit multipath by using the scattering characteristics of the propagation environment to enhance, rather than degrade, transmission accuracy by treating the multiplicity of scattering paths as separate parallel sub-channels which result in different coefficients in the channel matrix.

Some things need to be noted from this discussion.

1. The same frequency band is used for communication and hence spectrum is used efficiently. Together these substreams occupy a frequency band that is nT times smaller than what would be required to transmit the original stream on a single signal.

2. Since the user's data is being sent in parallel over multiple antennas, the effective transmission rate is increased roughly in proportion to the number of transmitter

antennas used.

3. The channel is assumed to be constant for a certain length of time. If this were not the case, there would be a need to determine the matrix coefficients for every symbol sent by the transmitter. Thus for a certain block of information, the channel coefficients remain unchanged. This is called a "quasi-static" assumption for the channel.

4. Independence of channels also means that the receiver will have more than one independent copy of the transmitted signal. This phenomenon, known as *diversity*, is exploited to provide reliable communication.

There are two main techniques which exploit this channel: spatial multiplexing and space-time coding. Space-Time Codes are error-correcting codes, and as the name suggests, use both spatial diversity (that is due to spacing between antennas) and temporal diversity (due to redundancy in time domain). It has been shown that codes can be designed to provide diversity up to $nT * nR$ [10]. The $nT * nR$ diversity is possible for cases where there is an independent fading path between each transmit antenna and receive antenna pair.

Spatial multiplexing involves multiplexing the input data stream between the various transmit antennas. It exploits the independence of channels to achieve very high spectral efficiency. It has been shown in that under certain assumptions capacity increases linearly with the number of antennas used without increasing bandwidth or power[25] [26]. The original and best-known example of this is BLAST, which was the technique announced by Lucent. The acronym stands for *"Bell labs LAyered Space Time"* Architecture. These form the signal-processing effort in the MIMO algorithm. In this implementation, we have used spatial multiplexing in the V-BLAST form with no coding done on each substream since it is a simple to construct and such a rudimentary system should suffice when it comes to characterization of a the antenna.

2.0.1 Role of the Antenna

As has already been described, the concept of MIMO is based on the independence of the individual paths in the channel. The presence of correlations between the paths will reduce the diversity in the system and also bring down the capacity [10]. In the absence of sufficient antenna spacing, or when the antennas are designed poorly, the correlations between the received signals increase. Thus, proper antenna design is a must for the feasibility of MIMO communications.

2.1 MIMO System Implementation Details and Options

A more detailed system model is shown in figures 2.2 2.3

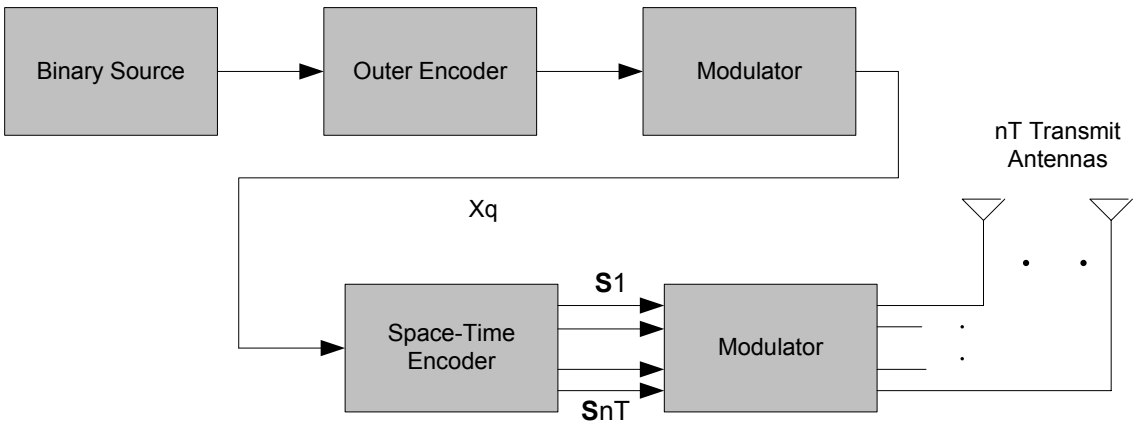

Figure 2.2: General MIMO transmitter model

Space-time transmission starts from the information source. The information bits that are provided need to be looked at in a vector manner rather than as singular bits of information. In the above case let us consider M vectors of length K, i.e., M*K bits, provided by the source. The encoder will generate V bits for every K bits

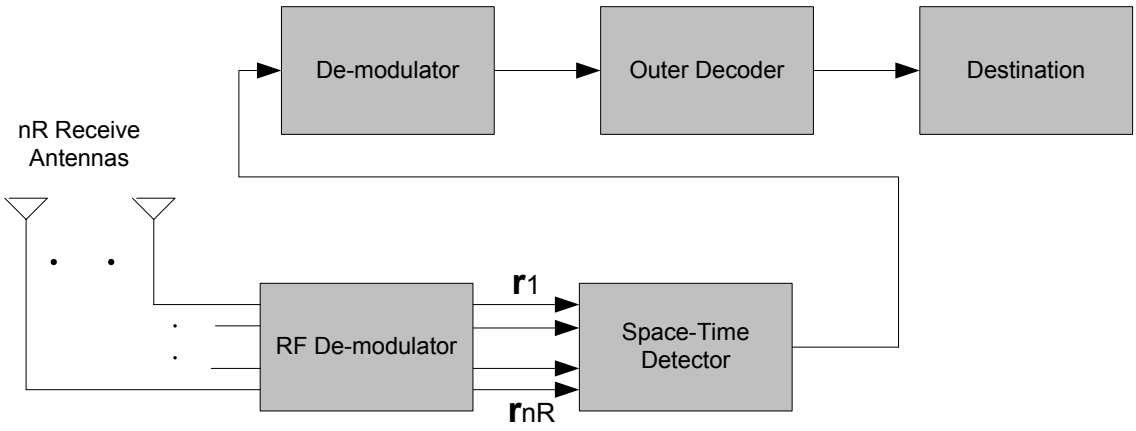

Figure 2.3: General MIMO receiver model

that it takes in i.e. we now have V*M bits from the first K*M transmitted. The modulator block just performs bit to symbol mapping. If Q is the modulation order, i.e., 2^Q points are present in the constellation being mapped, then there would be a total of V*M/Q symbols that would be sent to the space time encoder. It is to be remembered that the outer encoding can be done by a coded modulation scheme. In this case, we might think of the modulator as a sub-block of the outer encoder.

The space-time encoder is at the heart of the entire system and is what distinguishes implementations. This transforms the V*M/Q say, x_q , symbols into nT vectors of complex signals that need to be transmitted from nT antennas. Each is of length L which equals the number of symbol periods that it takes to complete the transmission of the x_q symbols.

The RF modulator performs the required amplification, mixing, upconversion and transmission of the symbols generated by the space-time encoder.

Let us look at each block in greater detail and assess the popular options that can be considered for implementation.

1. Outer Channel Coding/decoding:

In this block, the high bit-rate data steam that comes in will undergo channel encoding. This is done to protect the digital information from noise and interference and therefore reduce the number of bit errors. This results in the ability to increase distance of operation, reduce power requirements or operate under greater stress in terms of noise and interference than when channel encoding is not done. On the flip side, information (or useful data) is sent with a reduced data rate or an expansion in bandwidth. The main channel coding methods that are in use are block coding and convolutional coding.

Convolutional codes are generally more complicated than linear block codes, more difficult to implement, and have lower code rates (usually below 0.90). But they have powerful error correcting capabilities. They are popular in cases where bandwidth is essentially unlimited, but the BER is much higher and retransmissions are infeasible.

Block codes are very useful in situations where the BER of the channel is relatively low, bandwidth availability is limited in the transmission, and it is easy to retransmit data. They have very high code rates, usually above 0.95. They have low coding overhead, but they have limited error correction capabilities.

An important option which is an extension of the convolutional coding is the emergence of turbo coding which is nothing but the concatenation, either serially or parallelly, of two convolutional coders separated by a random interleaver. This scheme has been able to show results with values closest to Shannon's limit [11]. Turbo coding is one of the options for implementation in the UMTS standard for the third generation mobile standard. The combination of providing excellent SNR characteristics and the fact that there has been research that has shown the possibility of mapping the turbo decoder onto FPGAs [2] to achieve very high data rates makes this a very good choice for implementation and hence is being used in this thesis. The decoding strategy for turbo decoding is SISO (soft-in-soft-out) where the input into and the output from the decoding implementation is in the form of soft information i.e. not a 0 or a 1 for a bit but an intermediate value. This provides valuable information about the

channel. The most common form of this soft information is LLR (Log-Likelihood-Ratio) which encodes in it the probability of the bit being a 0 or a 1. This leads to two popular choices for the decoding strategy, namely, SOVA (soft output viterbi algorithm) or MAP (Maximum A-Posteriori) algorithm. We will look into the choices in much greater depth in chapter 5.

2. Space-time encoder: Basically the idea behind this encoder is that information is being looked at in a matrix manner with rows and columns of symbols. The rows correspond to the spatial dimension of transmission and the columns correspond to the temporal dimension i.e. each column corresponds to a symbol interval.

This block has a major effect on the architecture since the kind of encoder chosen has a bearing on the complexity of the detection process. The options available have already been touched upon in the previous section i.e. Spatial Multiplexing or Space-Time Coded systems. The difference between the two schemes arises from whether diversity or rate maximization is being aimed at.

Space-Time Code systems combine the space-time-encoder with symbol mapping which leads to a space-time coded modulation scheme. In this case, we can think of the symbol assignment block as a sub-block of the space-time encoder.

Spatial Multiplexing aims to increase the rate of transmission, while space-time coding aims to increase transmit diversity. In the case of Spatial Multiplexing, the aim is to send as many independent signals as there are antennas, thus increasing the average data-rate over the MIMO channel. Each substeam corresponding to an antenna can be optionally channel coded (usually convolutional), as this is found to provide some coding gain too. The most common examples of spatial multiplexing are the BLAST algorithms i.e. V-BLAST, H-BLAST and D-BLAST. The difference between these implementations comes from the method of application of error control coding [30].

In order to protect the transmissions from the multiple antennas against channel fading, joint coding is done. This is the essence of Space-time coded systems. It is important to remember that if the level of joint coding between the sub-streams is increased, the independence is decreased. Joint coding is realised using Space-Time codes which allow adjusting and optimization of joint coding for multiple antennas. As stated, the space-time encoder works on a matrix of symbols. In case of Space-time codes, the focus is on the generation of a code matrix in a manner that would maximize coding gain, diversity gain and channel capacity. There are two major flavors to this in Space Time Block Coding (STBC) and Space Time Trellis Coding (STTC).

In case of STBC, blocks of incoming symbols are buffered and a symbol matrix is generated from these buffered symbols. Most Space-Time Block Codes aim to generate orthogonal streams of data using an orthogonal matrix. This enables easier design of the detection block by the use of linear maximum likelihood detection at the receiver, but have no coding gain.

In case of STTC, an encoding trellis is used similar in operation to those used in trellis code modulation. The number of states in the trellis and the structure of the trellis determine the coding gain. Thus, this works on one symbol at at time, encoding it into a symbol vector of length nT that will be transmitted. Space Time Trellis Codes provide coding gain along with the provision of diversity gain. Unfortunately they are difficult to design and also require a computationally intensive encoder and decoder. There has been a lot of research on the design of space time codes and means of combining the gains of tellis and block coding [20] [14] [29].

3. RF modulation/demodulation and transmission/reception:
This is a standard transmission system where the type of constellation chosen determines the modulation chosen. The transmit path is that of upconversion mixing to convert the baseband information to RF, optional filtering, power amplification and transmission over the antenna. It is to be remembered that there is a need for the

digital data to be converted to analog using a DAC, one for each antenna. Therefore, there needs to be an interface between the digital backend and the RF frontend. This process is reversed at the receiver end.
Downconversion can be done in one of two ways: Direct or using IF i.e. superheterodyne. In case of Direct upconversion/downconversion the need for extra hardware for the IF stage is taken out. The pros and cons of one over the other will be discussed later on along with greater detail on the structure of the RF system.

4. Decoding Strategies: Receivers for Spatial Multiplexing can be divided into three classes: the maximum likelihood receiver, the linear receivers and the successive interference cancellation receiver. The Maximum Likelihood (ML) receiver gives the best error performance, but is the most computationally intensive since the complexity i.e. the number of iterations, the power requirements etc, increases exponentially with the number of transmit antennas. Linear receivers are much simpler but display severe performance degradation. Some common examples of this type of receiver are the Zero Forcing (ZF) Receiver that just inverts the channel and the Minimum Mean Squared Error (MMSE) receiver that is a little more complex. The Successive Interference Cancellation receiver (used in V-BLAST) works by detecting the best transmit antenna channel available and then successively cancelling interference due to the detected signal. This basically amounts to an iterative manner of solving the matrix inversion problem [25] [34].
In case of STBC, the decoder is usually a very simple Maximum Likelihood decoder which has linear processing at the receiver. In case of STTC, there is a need for Maximum Likelihood sequence estimation. This is more computationally taxing than the STBC decoding requirements. In general, it is preferable that soft-values of bits are generated so as to enable easier channel decoding.

Thus, the system implemented begins with a bit stream that is channel encoded using turbo encoding with the encoders following the UMTS standard. The bits are then mapped onto the constellation of interest to give a stream of symbols. The resulting symbol stream is then demultiplexed into three branches. Each of the three

branches is assigned to an antenna where modulation and transmission is done using a direct-conversion architecture. The received symbols are then downconverted and demodulated into symbol vectors which are fed to the detector stage. The detector is being implemented using the Maximum-likelihood A Posteriori Probability (ML-APP) algorithm and the log-likelihood ratio (LLR) values that are generated are fed to the channel decoder, after being multiplexed to form the original stream of data in LLR format, to make a hard decision on the bits. The channel decoder uses the Log-MAP algorithm to perform iterative decoding of the incoming LLR sequence.

Chapter 3

Overview of the System

All the major blocks of the design are shown in the diagram in figure 3.1.

3.1 Blocks of Design

1. RF system:

The first block that the information encounters is the RF system. Direct down conversion is performed where the carrier frequency is 2400 MHz. The system uses reference board MAX2701 for the receiver while using MAX2750 as the VCO. There are three elements to the antenna and this results in the use of three receiver boards. The choice of the bandwidth will be explained in later chapters. At the transmit side three MAX2721 EVKITS are planned on being used.

2. Interface board (Analog to Digital Conversion):

The in-phase (I) and quadrature (Q) channel signals that are generated by the RF system would need to be sent through a bank of ADC's to enable digital baseband processing of the data. Since data recovery is dependent upon the use of the proper sampling instant, it is a general practice to oversample the data i.e. take samples at

multiples of the symbol rate. Each ADC performs encoding into 8 bits.
This leads to an interface generating :

3 antennas * (8 bits for I and 8 bits for Q) = 48 bits at 60 MHz.

The sampling of the data is done by a separate interface board which interfaces between the digital backend starting with the FPGA and the RF frontend. The clock for sampling is provided by the FPGA. The interface to the RF system is done using BNC connectors while the I/O interface with the FPGA board is done using the P-160 expansion slots that are present on the FPGA board for supporting plug-in modules for various applications.

3. Data Acquisition block:

The MIMO detection and the channel decoding functions are being done in a non-real-time manner. To obtain data for analysis, there is a need for trace files to be generated and stored. In the case of our design, the job of data acquisition and trace file generation will be done by the Xilinx XC2VP7 Development board. The XC2VP7 has 792 Kb of total Block SelectRAM on chip organized as 44 blocks of 18 Kbits each. The design of the trace file generation was done with an intention of filling this SelectRAM available. It is also important to note that there is 32MB of SDRAM memory on the development board. The SelectRAM can be used in conjunction with a SDRAM memory controller to fill the 32MB of memory. This will be looked into later based on the success of the present design.

Three functional sub-blocks form the data acquisition block : the trace formation block, the interface block and the UART core. The trace formation block performs the reading of the bits presented to its input from the P-160 expansion slots. The interface board provides inputs of 48 bits width which have to be written to memory to enable the data to be read out to a PC later. The other option of transfer onto a PC for offline analysis is the use of the PCI bus. But the use of this method required the building of a board with memory and hence memory control logic. Since this function can be done by an FPGA while giving us more options for future use, we chose the latter option. There are 44 SelectRAM blocks in the XC2VP7. The 18

Kb SelectRAM memory can be configured in multiple ways by instantiating different library primitives in the design. This will be explained in chapter 7. In the case of this design, the 18 Kb memory has been configured as 512 rows of 36 bits each. In each row there are 4 parity bits which are never used. Thus, effectively we use 512 * 32 bits in a given SelectRAM block.

The module runs at twice the speed of the I/O interface. The reason for this will become clearer in Chapter 7. The basic idea is that of convenience in memory access where the idea is to write to memory in the form of 96 bit words while the interface runs at 48 bits. The module has been designed for easy debugging by stagewise flag use. This will become clearer in the coming chapters. The reading of the I/O interface contents only begins after the Start signal coming in from a data detection block. Also, once the data is filled, there is going to be a need for the data to be read out and sent to be worked upon by detection and decoding algorithms.

The means of communicating with a peripheral unit to process the information in the traces is done using the RS-232 port on the FPGA board. To use the RS-232 port for serial communication, there is a need for a UART core to be designed. The read from the RS-232 port can be done using a simple C program on the destination PC.

A small complexity that needs to be accounted for is the interfacing of the module that writes to memory using 96 bits as a word, to the UART core that works with 8bit input. Moreover, each module works at a different clock speeds. This is done using an interface between the selectRAM writing module and the UART. This sort of interface calls for the use of handshaking. This implementation was done using a complicated FSM based design. The acquisition block will be explained in detail in chapter 7.

4. Maximum Likelihood detection:
This is a technique of detection wherein all possible candidates for a given received

vector are iteratively checked to determine the closest match. This is an exhaustive search that presents the greatest complexity in implementation. The decision on the closest candidate is done on the basis of Mean Squared Error (MSE). One of the major positives of this algorithm is the fact that the output of this algorithm is in the form of Log Likelihood Ratios (LLR). LLR represents the probability of a given bit being a 0 or a 1 by means of a ratio of their probabilities. Thus a negative value of LLR implies that the bit is more likely to be a 0 while a positive value indicates a greater probability of a 1. The greatest use of the LLRs come in their applicability for channel decoding. The LLR encompasses in it soft-values that represent a-priori information about the channel already known which is used by the channel decoders to make the process of decoding more robust by getting more accurate estimates. These works especially well with turbo decoding.

5.Turbo coding

Turbo coding is a Forward Error Correcting (FEC) channel coding algorithm that has shown an error performance much higher than other channel coding techniques like convolutional coding and block coding. This technique of coding generally takes two convolutional codes and concatenates them in either a serial or parallel manner with an interleaver between them. The interleaver randomizes the ordering of bits within a block of information carrying bits ensuring that there is an element of randomness in the design. This randomness ensures the presence of minimal low weight codes which is the characteristic of a good coding algorithm. Generally there are two convolutional encoders whose outputs are multiplexed. The decoding of Turbo Codes requires the use of soft inputs which are usually in LLR form. The use of soft inputs leads to the use of algorithms like SOVA (Soft Output Viterbi Algorithm) and the MAP (Maximum A-posteriori Probability). These iterative solutions where soft-inputs (SI) are used along with a-priori information (information obtained along with the input that aid the decoding process) to generate soft-outputs (SO) are called SISO algorithms. In a general solution to Turbo-decoding, there will be as many SISO processors as there are encoders that operate together. The SISO processors share information with each other where the soft-outputs of one decoder will be used to

generate the input to the other decoder and vice-versa. This is illustrated in figure 3.2 The design of the interleaver and the SISO processors is what defines the complexity of the decoding structure. There are different types of interleavers such as the block interleaver, pseudo-random interleaver, circular interleaver, semi-random interleaver design and the optimal (near-optimal) interleaver [19]. Also, the choices for the implementation of the SISO processor would be variants of the MAP algorithm, like the log-MAP, max-log-MAP, constant-log-MAP and linear-log-MAP [32], or SOVA. In our implementation, we perform a log-MAP implementation on the LLR outputs of the ML-APP detector stage. The intricacies of this implementation and more about turbo codes will be dealt with in Chapter 5.

3.1.1 Present Implementation Status

At present, we have implemented and tested the working of the channel coder/decoder, the ML-APP detection block, the data acquisition block and have a preliminary design but untested version of the interface board. The RF system and the relevant preprocessing is yet to be implemented along with channel estimation. At present oversampling is being employed in place of timing synchronization so as to enable a correlation based approach to determining the best sampling time.

3.1 Blocks of Design

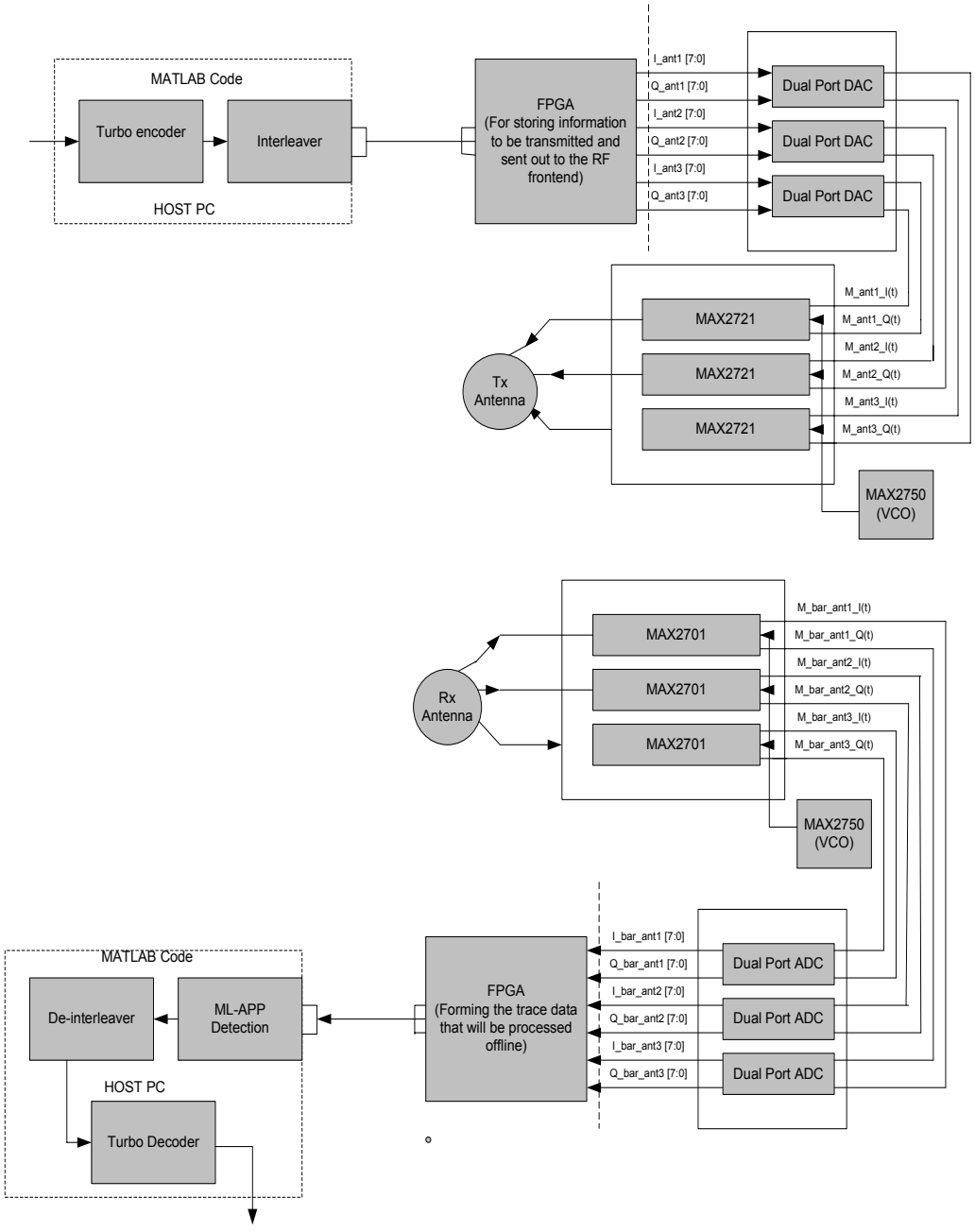

Figure 3.1: The Block diagram of the System that is being used

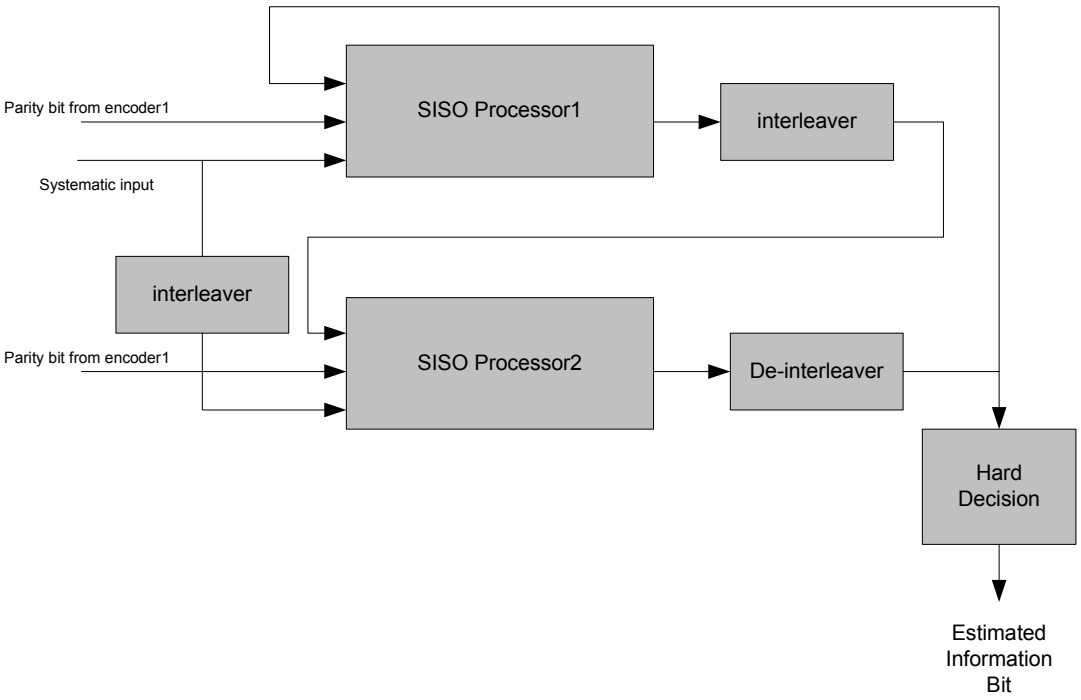

Figure 3.2: Decoder Structure for the case where there are 2 encoders used

Chapter 4

ML-APP Algorithm Implementation

Let us consider the system model already talked about in chapter 2:

$$\mathbf{r} = \mathbf{Hs} + \mathbf{n};$$

where $\mathbf{H}$ is the complex channel matrix, while the signals at the transmitter and receiver are complex column vectors of symbols, $\mathbf{s}$ and $\mathbf{r}$.

The function of the MIMO detection stage is to form estimates of the transmitted symbol vector $\mathbf{s}$ from the received vector $\mathbf{r}$. This block generally is the most demanding in terms of complexity in a given MIMO system since most of the complex signal processing functions are called upon for the detection process. The complexity of this decision process is dependent upon the algorithm of choice for space time coding.

The system that is being built here uses spatial multiplexing. In such a system, the main gains available to be exploited are high data rates and greater reliability due to the presence of diversity. An ideal detection algorithm for spatial multiplexing

schemes would exploit the antenna diversity provided by the use of multiple antennas. The main algorithms of choice are suboptimal equalization-based detection schemes [25], including nulling-and-cancelling schemes and optimal the maximum-likelihood (ML) detection algorithm.

Another way of classification is in terms of the linearity of the algorithm[8]. Equalization-based detection algorithms, such as *Zero Forcing* and *MMSE-based* receivers, are linear in that the complexity of the algorithms increases linearly with the number of antennas being considered. The nulling-and-cancelling and maximum-likelihood (ML) detection schemes are both non-linear detection schemes. The linear detection schemes are sub-optimal and do not fully exploit the available diversity, but are far less complex to implement. The same holds for the nulling and cancelling approach. The maximum-likelihood detection algorithm can exploit all the available diversity but is computationally intensive. Thus, there is a stong demand for computationally efficient, generally sub-optimal detection schemes that can exploit a large part of the available diversity. The final decision on the usage of a given algorithm is based on the mode of implementation (software vs hardware) and the application being aimed at.

4.1 Detection Methods

4.1.1 Linear Equalization Based Detection

In this case, an estimate of the transmitted data vector $\mathbf{d}$ is calculated as $\mathbf{y} = \mathbf{Gr}$, where $\mathbf{G}$ is an equalization matrix. The detected data vector is then obtained as $\overline{\mathbf{d}} = \mathrm{Q}\{\mathbf{y}\}$
where Q. represents the component-wise quantization according to the symbol alphabet used. The type of equalization matrix used enables classification into two popular implementations of this type of detection scheme: *Zero Forcing* and *MMSE*. Conceptually, each substream in turn is considered to be the desired signal, and the

remainder are considered "interferers".

The Zero-Forcing equalizer uses the pseudo-inverse of $\mathbf{H}$ which is given by

$$\mathbf{G} = (\mathbf{H^H H})^{-1}\mathbf{H^H}$$

The core implementation issue here, is finding the inverse of the channel matrix. The algorithm has its major folly in not considering the noise in the system for its detection process. The resulting detection estimate is of the form

$$\mathbf{y_{ZF}} = (\mathbf{H^H H})^{-1}\mathbf{H^H r} = \mathbf{d} + \overline{\mathbf{w}}$$

which basically corresponds to the data vector $\mathbf{d}$ corrupted by the transformed noise $\overline{\mathbf{w}}$.

The minimum mean-squared error (MMSE) equalizer is given by

$$\mathbf{G} = (\mathbf{H^H H} + \mathrm{N_0}\mathbf{I})^{-1}\mathbf{H^H}$$

which minimizes the mean-squared error $\mathrm{E}\{||\mathbf{y} - \mathbf{d}||^2\}$ [8]. The major constraint in the design of such a system is that there is a need for an accurate estimate of the amount of noise in the system to provide the designer of the detection algorithm the value of N_0 which characterizes the noise in the system. It is to be noted that if $\mathrm{N}_0 = 0$, then the MMSE algorithm becomes equivalent to the zero-forcing algorithm.

The major problem with both the methods in linear equalization based detection is that they both perform very poorly under conditions where the channel matrix becomes very ill-conditioned, which usually happens when there is a significant line of sight (LOS) component and when there is correlated fading in the channel. This results in the reduction of "invertibility" of the channel which is a must, especially for detection algorithms like ZF and MMSE. These algorithms, though simple, have been found to be quite unusable for systems with low BER targets because the rank of the matrix tends to 1 [7, 23].

4.1.2 Nulling and Cancelling/VBLAST

These set of iterative non-linear algorithms provide a middle path between linear equalization methods and the brute-force Maximum likelihood approach, thereby providing a better BER performance at fair complexity. This approach uses a serial decision feedback approach i.e. an iterative methodology, to detect each layer separately [25]. When a layer is detected, an estimate of the corresponding layer's contribution to the received vector $\mathbf{r}$ is subtracted from $\mathbf{r}$, and the result is then used to detect the next layer. Thus once a layer has been detected, it is considered as interference for the other layers and is hence cancelled out. This is where this algorithm differs from the previous approach where only nulling was performed. This continues iteratively until all the layers are detected. To detect a given layer, the layers that have not been detected are nulled out or equalized. The choice of the equalization leads to ZF-VBLAST and MMSE-VBLAST as the most popular choices for the nulling-and-cancelling approach.

Let us consider the general case of V-BLAST detection [26].

There would be a particular order of evaluation, say S_{OPT}, which determines the order in which the symbols from the transmitted vector are extracted. The S_{OPT} is ordered such that the component with the best SNR is chosen at each stage of iteration to be cancelled [26]. These are the steps that are followed:
Step 1: Using nulling vector $\mathbf{w_{k_i}}$ $1 \leq i \leq nT$, form the decision statistic $\mathbf{y_{ki}}$ by performing a linear combination of the components of r:

$$\mathbf{y_{k_i}} = \mathbf{w_{k_i}^T r}$$

Step 2: Slice $\mathbf{y_{k_i}}$ to obtain

$$\overline{\mathbf{a}}_{\mathbf{k_i}} = \mathbf{Q}(\mathbf{y_{k_i}})$$

where Q depends upon the constellation being used
Step 3: Assuming $\overline{\mathbf{a}}_{\mathbf{k_i}} = \mathbf{a_{k_i}}$, i.e., the decoded symbol is error free, remove the effect of

this symbol on the received vector thereby reducing, in an indirect sense, the number of antennas in the system. This is done by:

$$\mathbf{r_2} = \mathbf{r_1} - \overline{\mathbf{a}}_{\mathbf{k_i}}(\mathbf{H})_{\mathbf{k_i}}$$

where $(\mathbf{H})_{\mathbf{k_i}}$ is the k_i^{th} column of $\mathbf{H}$.
The ordering of k_i is given by $S_{OPT} = \{k_1, k_2, \ldots k_{nT}\}$.
In the above, the ordering of k_i is of critical importance since the detection procedure is susceptible to error propagation. The importance of ordering comes from the fact that it determines the subset of the rows of the channel matrix that the nulling vector needs to be constrained by. This does not exist in simple nulling since the decisions are always based on nT *-1* rows of the $\mathbf{H}$ matrix since there is no cancellation.

[26] and [25] deal with the issues of optimal ordering in greater detail. Let us consider the case of ZF-VBLAST to analyze the numerical complexity issues.

Initialization Phase.

1. $i \leftarrow 1$
2. $G_1 = H^+$ where H is the channel matrix and H^+ represents the pseudo-inverse of the matrix.
3. Find the row in $G_1 = H^+$ with the minimum norm, i.e., $k_1 = \text{argmin}_j ||(G_1)_j||^2$

Recursion Phase.

1. $w_{k_i} = G_i(k_i;:)$, i.e., the $k_i{}^{th}$ row of the G matrix is taken as nulling vector.
2. $y_{k_i} = w_{k_i}^T r$, i.e., the decision vector is created by simple matrix multiplication.
3. $\overline{a}_{k_i} = Q(y_{k_i})$; obtain the estimated symbol by quantization.
4. $r_{i+1} = ri - \overline{a}_{k_i} H(k_i;:)$; remove the effects of the decoded symbol from the received vector.
5. Replace the $k_i{}^{th}$ column of H with zeros i.e. cancellation is done.
6. $G_{i+1} = H^+$
7. Among the non-zero rows of $G_{i+1} = H^+$, find the row with the minimum norm i.e. $k_{i+1} = \text{argmin}_{j \notin \{k_1 \ldots k_i\}} ||(G_1)_j||^2$.
8. $i \leftarrow i + 1$

As can be seen, the algorithm involves the recursive computation of the pseudo-inverse (also called the Moore - Penrose Matrix inverse) to perform the computation of the nulling vectors. The total number of iterations equals the number of transmit antennas present, and in each iteration there are two matrix multiplications - one computation of the pseudoinverse of a progressively smaller matrix (during each iteration one column of the H matrix is zeroed out) and the computation of the norm of a decrementing number of columns. The difference between MMSE-VBLAST and ZF-VBLAST lies in the computation of the nulling vector.

4.1.3 Maximum-Likelihood detection

This is the optimum decoding method from a mathematical point of view. A search is performed over all possible symbols and the most likely one is chosen. No noise enhancement takes place and numerical issues are virtually not present since this method calls for no matrix inversion or divisions. This also makes it more robust in the case when LOS (line-of-sight) components are present.

It has been shown by numerical analysis [34] that the maximum likelihood detection algorithm has a significant advantage in SNR over other detection methods because of a high diversity that equals the number of receive antennas, independent of the

number of transmit antennas. What this means is that the SNR curves of this method have a greater rate of drop when compared to other methods that have the same system configuration. Another observation made was that this method also works when the number of transmit antennas is greater than the number of receive antennas, which is not possible for techniques which have already been discussed that use the pseudo-inversion of **H**. Hence, theoretically, the data rate can be increased by using a larger number of transmit antennas, though there will also be an increase in the SNR requirement. The combined effect of these advantages makes this scheme a very appealing one.

4.1.4 Complexity of Hardware Implementation

There is an obvious disadvantage of this method in the exponential increase in the complexity of the algorithm both with the number of transmit antennas and the size of the symbol alphabet. Let us look into this in greater detail.

As stated before in case of ML detection, all candidate transmit vectors need to be considered. The transmit vectors are nothing but a row of symbols. If the constellation under use has 2^Q constellation points, then each symbol encodes Q bits. Thus, if there are $nT = nR =$ N transmit antennas, each transmit vector would be of length N*Q bits. Since we need to cycle through all the 2^{NQ} bit combinations, there is a need for a counter of size N*Q bits to be present. Let us call the transmit vector **x**. Also let us denote the MSE as **J**. In order to make a hard decision on the transmitted symbols, the detector would have to find that **x** which gives the minimal value for the expression:

$$\mathbf{J} = ||\mathbf{r} - \mathbf{Hx}||^2 = \mathbf{real}^2 + \mathbf{imag}^2$$

One of the problems with the MSE approach is the square operation is costly to perform in hardware. The algorithm can be made relatively simpler by the use of approximations for the norm calculations which reduce the number of multiplications needed. One approach is to use a MAE detector that uses the magnitude of the error

vector to generate **J**, instead of the MSE. This is shown below

$$\mathbf{J} = ||\mathbf{r} - \mathbf{Hx}|| = |\mathbf{read} + \mathbf{j} * \mathbf{imag}| = \sqrt{\mathbf{real^2} + \mathbf{imag^2}}$$

Again, square root is also better done without. This is helped by the use of approximations like the approximate MAE

$$|\mathbf{read} + \mathbf{j} * \mathbf{imag}| = \sqrt{\mathbf{real^2} + \mathbf{imag^2}} \approx \{\mathbf{max}(|\mathbf{real}|, |\mathbf{imag}|) + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{min}(|\mathbf{real}|, |\mathbf{imag}|)\}$$

The use of approximate MAE also helps reduce the complexity of the MSE calculations by reducing the number of square operations from 2 to 1, thus giving rise to the approximate MSE

$$|\mathbf{read} + \mathbf{j} * \mathbf{imag}|^\mathbf{2} = \mathbf{real^2} + \mathbf{imag^2} \approx \{\mathbf{max}(|\mathbf{real}|, |\mathbf{imag}|) + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{min}(|\mathbf{real}|, |\mathbf{imag}|)\}^\mathbf{2}$$

Thought this looks like a very complex algorithm to implement due to its brute force nature what helps is the fact that there is a lot of inherent regularity in the design. Moreover, there is no need for performing floating-point operations. This manifests itself in the ability to perform the detection on more than one received symbol vector at a time by making use of more than one LLR checking unit at a time. The brute force checking algorithm ensures that there is minimal control signaling required since once started, the unit for computing the LLR's for a given received vector will have to go through all the candidates for it irrespective of the starting vector (as generated by a counter) and hence has a predefined and constant number of iterations that need to be done. This ensures autonomous working. Moreover, the 2's complement representation of numbers ensures that the total number of iterations can be reduced by computing more than one **J** at a time. It has been found that for systems with a small number of antennas (4 X 4 or lesser) the complexity of ML detection algorithm turns out to be comparable or even lesser than the ZF-VBLAST algorithm [7].

While the hard decision would give the best estimate on the transmitted bits, it is also important to generate log likelihood ratio (LLR) information on each bit because it aids in soft input channel decoding further down in the receiver chain. In order to generate LLRs, the ML-APP detector generates the value of **J** for the case where

each bit position is in turn 1 and a 0. What this is basically trying to implement is the estimated LLR for each transmitted bit, bi(x), where i is the bit index of the NQ transmitted bits. That is,

$$\text{LLR(i)} = \min_{b_i(\mathbf{X})=0)}(J) - \min_{b_i(\mathbf{X})=1)}(J)$$

What this implies is that for each bit position in the NQ bits in the candidate vectors, the lowest possible J value was calculated when that bit was a 0 and a 1 irrespective of the values of the other bits. The difference between the values for 0 and 1 for each bit position then gave the LLR value for that bit position.

A note of gratitude is due to Dr David Garrett and Dr Linda Davis of Bell Labs Research, Sydney, Australia, for their help in providing information on the ML-APP design.

Chapter 5

Turbo Coding Implementation

5.1 Forward Error Correcting Codes

Forward Error Correcting [FEC] codes are channel codes which are used to improve the quality of transmission over a wireless communication channel. This improvement in efficiency is attained by the use of redundancy wherein redundant bits (non-information carrying parity bits) are transmitted along with the information bits. At the receiver, the FEC decoders present are designed to utilize this redundancy to provide tolerance to a certain number of channel errors. This increased handling of errors enables operation at lower transmit power, transmission over greater distances, transmission at higher bit rates, and the ability to work under environments with greater interference. In the case of a general FEC encoder, K input bits would result in a output, called the *codeword* of N bits, where N > K. Of the possible 2^N combinations of output bits only 2^K are valid codewords. The ratio K/N is called the *code rate* denoted by r. It is important to note that low rate codes have a better *energy* efficiency while higher rate codes have a better *bandwidth* utilization. Thus an optimal solution between bandwidth and power is one of the important design decisions for the turbo decoder implementation.

5.2 Basic FEC codes

Block codes work in a block wise manner where, in our example, a block is of K bits and a matrix multiplication is performed to it to generate N bits. This matrix is called the generator matrix **G**. If the block of information bits is **m**, then the output code word **C** would be

$$\mathbf{C} = \mathbf{mG}$$

At the receiver side the length **N** noisy codeword will be multiplied by a parity check matrix **H** which provides the syndrome vector. The information provided by this vector is used to determine bit error positions of the error bits in the received vector. There are some rare patterns that lead to no error being displayed.

In case of convolutional encoders there is no need for the data to be partitioned into fixed size messages in convolutional codes; the data streams can be encoded in a semi-infinite stream. This is advantageous because the decoder does not have to wait for an entire block to be received before decoding can commence thus providing better latency characteristics.

Convolutional encoders, unlike block encoders, have memory. For a general encoder taking in k-bits at a time this means that every k-bit input affects the value of L_c n-bit outputs of the encoder, where L_c is a code parameter called the *constraint length*. In many cases, such as the UMTS encoder being used in this implementation, the incoming bits are worked upon one bit at a time i.e. k = 1. This could be a very inefficient implementation if n > 2. A method to obtain a rate of 1/2 is to systematically delete bits from the stream of output bits which is called *puncturing*. This reduces the effectiveness of the code and is left as a later option in this implementation.

Both of the above are popular solutions to channel encoding needs for wireless systems that do not need a very low BER. The implementation of the encoder and decoder are simple (relative to the Turbo decoder) with look-up table based solutions or algebraic decoding solutions for block coding and Viterbi Algorithms based Maximum Likelihood Sequence Estimation in case of convolutional coding.

5.3 Turbo coding basics

A turbo code differs from coding techniques already discussed in that it is not just a single code but a combination of two (or more) codes called *constituent codes*. Based on whether the constituent codes are running in a parallel manner or in a serial manner turbo codes can be distinguished as *Parallel Concatenated Convolutional Codes* or *Serial Concatenated Convolutional Codes*. Each constituent code may be a FEC code, which is generally the case, and usually the same code is repeated twice. The major difference between turbo codes and other codes is the use of an *interleaver* between the data feeding the two encoders. The general structure of a Turbo encoder is as seen in the figure 5.1

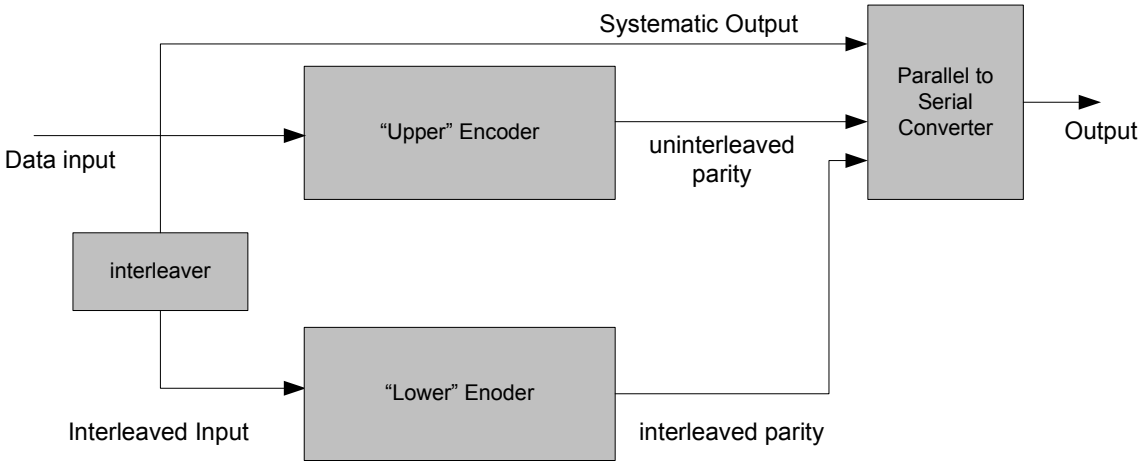

Figure 5.1: Generic turbo encoder as per 3GPP

Consider the Parallel Concatenated Convolutional Codes. As can be seen the same set of inputs is given to the two encoders, Upper and Lower Encoders, but they differ in their order. The interleaver performs the bit randomization in a prescribed (usually as per a standard) but irregular manner. The outputs of the two encoders are serialized and sent along with the data. It is to be noted that, because of the interleaver, the output of the second encoder would be very different from that of the first encoder. It is this property of randomness and difference that makes the turbo

coding technique effective. This is explained below.

A good linear code is one that has a high percentage of high-weight code words, where the weight of a code is the total number of 1's in it. Though there are bound to be a few low weight, codes their frequency has to be minimized. It is this property that is satisfied by turbo codes. The weight of the turbo code is the sum of the weights of the input and the parity outputs of the two constituent encoders. Thus even if one of the parity outputs is of low weight, the other could compensate for it. Since the second encoder works on the scrambled version of the input data the probability of both the parities being of low weight is small. This characteristic is what makes turbo codes effective.

In this implementation we follow the Universal Mobile Telecommunications System (UMTS) standard for the encoder and decoder specifications [32]. The details of their design will be discussed next.

5.4 Encoder and Decoder Details

5.4.1 UMTS Encoder

The structure of the UMTS encoder is as shown in figure 5.2 As can be seen in the figure the encoder has 2 constraint length 4 RSC encoders used in a parallel concatenation manner. RSC encoders are a popular choice because they are *systematic*, which implies that the data is sent unchanged along with the encoded bits which provide redundancy. Turbo coding scheme needs data to be present along with the parity bits. The feedforward and feedback generators are 15 and 13 respectively, both in octal. The number of bits in the input data block,K, can be anywhere between 40 and 5114. This is one of the constraints of design that needs to be decided. The value used by us and the reason for the given choice will be described later. The data is encoded by the upper encoder in the natural order while its scrambled version is

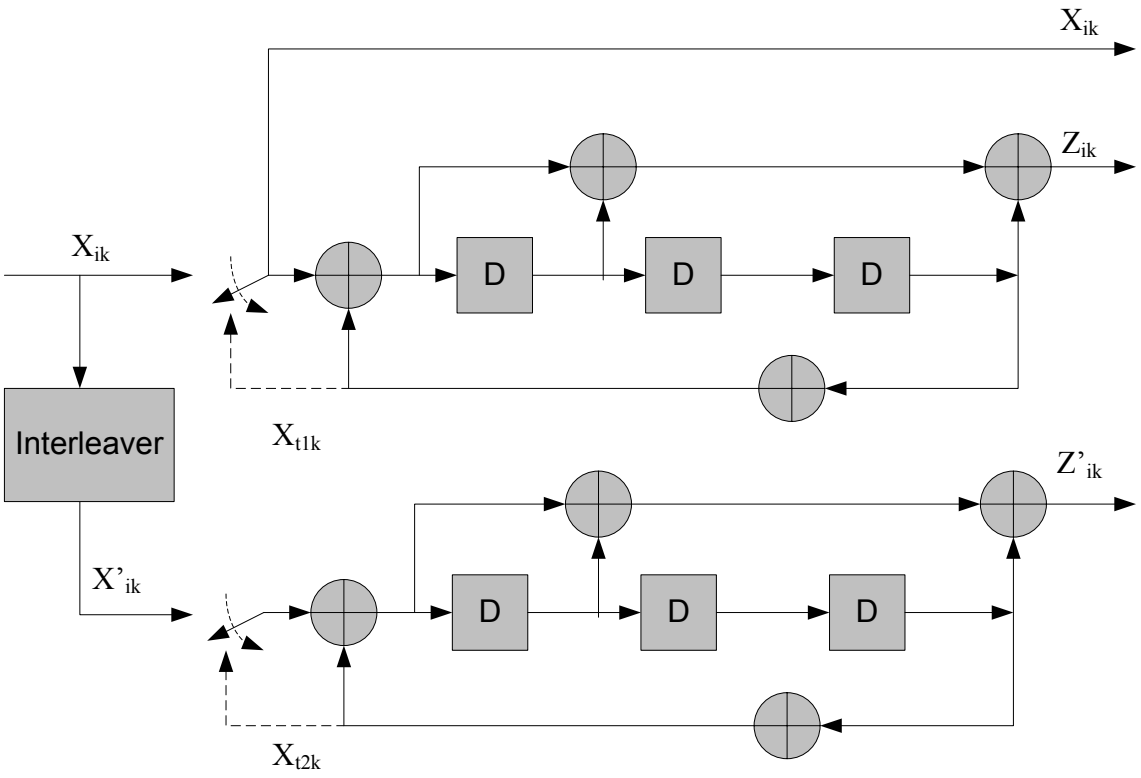

Figure 5.2: UMTS Turbo Encoder

used by the lower encoder after being interleaved. Initially, the first 2 switches are in the up position wherein the parity being generated by the data coming in is sent out.

The data bits are transmitted along with the parity bits generated by the two encoders. It is to be remembered that the code also generated tail bits after all the input data has been encoded. Tail bits are used to terminated the trellis in a known state since turbo decoding algorithms work best when they know both the initial and the final state. The reason for this will become clear when we deal with the details of the turbo decoding algorithm. If the feedback bit were to be used as the encoder input, then the XOR of these two bits will be zero, and thus the encoder will return to the all-zeros state after three clock cycles (i.e., the output of the leftmost XOR gate in the figure will be zero, since the two inputs are the same). Therefore, the encoder can

be brought back to the all-zeros state by inputting the three feedback bits generated immediately after the K-bit code word has been encoded. The termination is done by using the feedback as the input to the encoder. Thus the format of the output from the encoder is of the form:

$$
\begin{aligned}
\text{for k} = 0 \text{ to } K\text{ - }1 & \\
X_{3k+1} &= Xi_k\text{; input data} \\
X_{3k+2} &= Zi_k\text{; parity bit from the first encoder} \\
X_{3k+3} &= Z^{'}i_k\text{; parity bit from the second encoder} \\
\text{for k} = K \text{ to } K + 2 & \\
X_{3k+1} &= X_{t1k}\text{; feedback input for encoder 1} \\
X_{3k+2} &= Z^{'}_{t1k}\text{; parity for encoder 1} \\
\text{for k} = K + 3 \text{ to } K + 5 & \\
X_{3k+1} &= X_{t2k}\text{; feedback input for encoder 2} \\
X_{3k+2} &= Z^{'}_{t2k}\text{; parity for encoder 2}
\end{aligned}
$$

5.4.2 UMTS Decoder

After encoding, the entire n-bit turbo code word is assembled into a frame, modulated, transmitted over the channel, demodulated and detected. If T_i is the transmitted bit and the decoded bit is R_i, it has to be noted that while T_i can only take the values 1 and 0 the received bit R_i can take on any value between 0 and 1 based on the scheme decoding the bits. In our case the ML-APP detection algorithm provides us with the latter which is called a *soft* value in contrast to a *hard* value. The most often used representation of the soft value is the Log-Likelihood Ratio (LLR) which can mathematically be represented as:

$$LLR(T_i) = \ln \frac{P(R_i|(T_i = 1))}{P(R_i|(T_i = 0))}$$

In this expression $P(R_i|(T_i = j))$ represents the conditional probability of receiving R_i given that the code bit $T_i = j$ was transmitted. A high level abstraction of the UMTS decoder is as shown in the figure 5.3 As indicated by the presence of a feedback

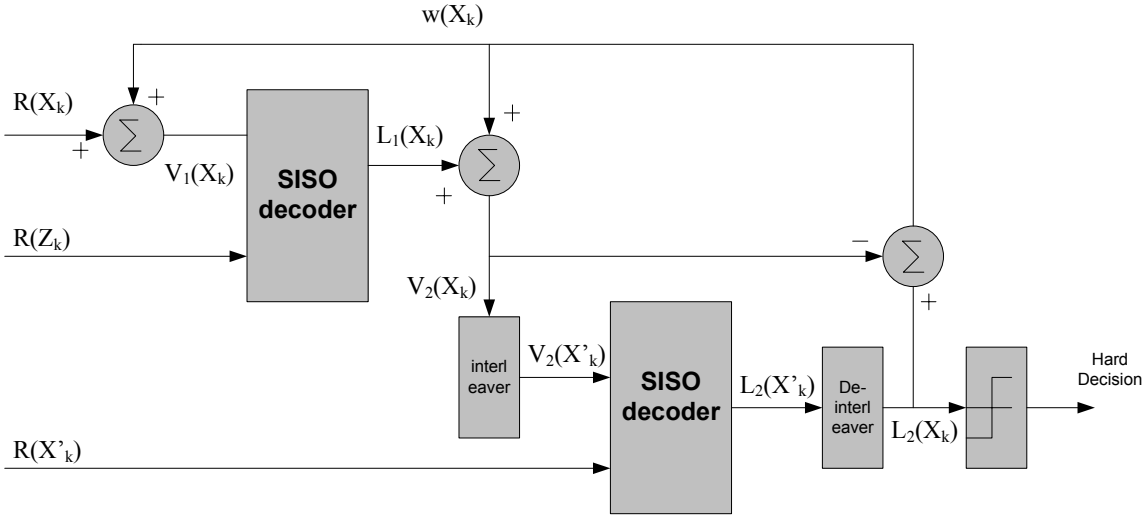

Figure 5.3: UMTS Turbo Decoder

path, the decoder operates in an iterative manner. Each full iteration consists of two half-iterations, one for each constituent RSC code. The timing of the decoder is such that SISO decoder2 operates during the first half-iteration, and SISO decoder1 operates during the second half-iteration. The function of a SISO decoder is to take in a-priori information and generate a-posteriori information ,usually in LLR form.

The first SISO block makes an estimate of the probability for each data bit as to whether it is a 1 or a 0 by operating on the received data and parity bits (soft values from demodulator) produced by the first constituent encoder. This estimate is then sent to the second SISO processor as an additional information block along with the interleaved received data and the parity bits (soft values from demodulator) produced by the second constituent encoder. The information generated by the second SISO decoder is used as additional information by the first SISO decoder for redecoding the first code. The information generated by each SISO decoder acts as the correction term called extrinsic information for the other SISO decoder which uses it as the

a-priori information. This process of two passes through the decoding algorithm is considered to be one iteration and is repeated for a fixed number of iterations, or until some external mechanism determines that no further iterations will improve the BER for that frame. After all iterations are complete, an estimate of the data bits sent is made by making a hard decision on the LLR output of the second SISO decoder. In this implementation the extrinsic information that is generated by the second decoder ,denoted by $w(X_k)$, $1 \leq k \leq K$, is fed back to the first encoder which adds it to the systematic LLR input, $R(X_k)$, which forms a new variable, denoted by $V_1(X_k)$. For $1 \leq k \leq K$, $V_1(X_k)$ is equal to the combination of $R(X_k)$ and $w_1(X_k)$ after which it is just the tail bits for decoder 1. The LLR output of the first decoder is $L_1(X_k)$, where $1 \leq k \leq K$.

$w(X_k)$ is subtracted from $L_1(X_k)$ to form $V_2(X_k)$ which contains the sum of the systematic channel LLR and extrinsic information from decoder 1. This input is interleaved to get $V_2(X_k^{'})$ which forms the input to the second decoder along with the soft input values for the parity generated by the lower encoder. This is for $1 \leq k \leq K$ after which $V_2(X_k^{'})$ equals just the tail bits of the second encoder. The LLR output of the first decoder is $L_2(X_k^{'})$, where $1 \leq k \leq K$ which is deinterleaved to form $L_2(X_k)$. The final hard decisions on the bits are taken using $L_2(X_k)$.

As stated earlier the possible implementations for the SISO decoder are the SOVA and MAP algorithms. Here we use the MAP algorithm which does a bitwise estimation of the input. An essential difference between the MAP and the SOVA algorithm is the use of max* operator, also called Jacobi logarithm, and its different flavors i.e. log-MAP, max-log-MAP, constant-log-MAP and linear-log-MAP. All of these work with logarithmic data. The log-MAP is the most complicated but with the best BER performance and will be used in this project wherein the Jacobi logarithm is computed exactly using

$$\begin{aligned} \max^*(\mathrm{x}\ ,\ \mathrm{y}) &= \ln\left(e^x + e^y\right) \\ &= max(x,y) + \ln\left(1 + e^{-|y-x|}\right) \\ &= max(x,y) + f_c(|y-x|) \end{aligned}$$

where $f_c(|y-x|)$ is the correction function implementable using logarithm and exponential functions in software.

5.5 Decoding Algorithm

The decoding algorithm followed here is taken from [32]. The decoding for each SISO decoder has 2 phases of operation wherein a sweep is done in the forward and reverse directions, not necessarily in that order. During each sweep partial path metrics are computed using the max* operator. During the second sweep the LLR values are computed. LLR values output during the forward sweep are in the right order and hence we perform the reverse sweep first and then the forward sweep.

5.5.1 UMTS Trellis and Branch Metrics

The trellis of the RSC encoders used in the UMTS turbo encoder is shown below: 5.4

solid is 1 and dotted lines are used for an input of 0.

There are only four distinct branch metrics:

$$\begin{aligned} \gamma_0 &= 0 \\ \gamma_1 &= V(X_k) \\ \gamma_2 &= R(X_k) \\ \gamma_3 &= V(X_k) + R_(X_k) \end{aligned}$$

5.5.2 Backward and Forward sweep

The backward path metric for state S_i at trellis stage k is denoted by $\beta_k(S_i)$, with $1 \leq k \leq K+3$ and $0 \leq i \leq 7$. Initially, $\beta_{K+3}(S_0) = 0$ and $\beta_{K+3}(S_i) =$ *-infinity*, for $1 \leq i \leq 7$.

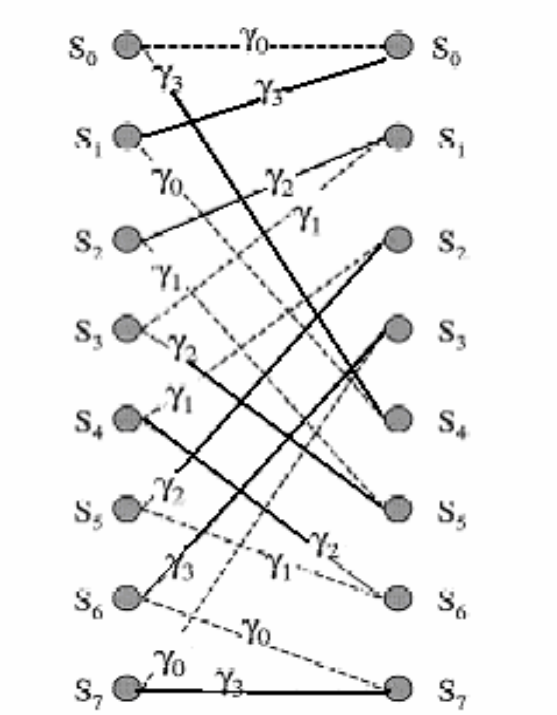

Figure 5.4: Trellis structure of RSC encoder used in UMTS encoder

For $k = K + 2$ downto $k = 1$, the partial path metrics are found using :

$$\beta_k^{'}(S_i) = \max{}^*(\beta_{k+1}(S_{j1}) + \gamma_{ij1}, \beta_{k+1}(S_{j2}) + \gamma_{ij2}) \tag{5.1}$$

where, $\beta_k^{'}(S_i)$ is the un-normalized backward path metric and S_{j1} and S_{j2} are the two states in stage *k+1* which are connected to the state S_i in stage k in the trellis. Normalization is done as per

$$\beta_k(S_i) = \beta_k^{'}(S_i) - \beta_k^{'}(S_0) \tag{5.2}$$

The forward path metric for state S_i at trellis stage k is denoted by $\alpha_k(S_i)$, with $0 \leq k \leq K$ and $0 \leq i \leq 7$. Initially, $\alpha_0(S_0) = 0$ and $\alpha_0(S_i) =$ *-infinity*, for $1 \leq i \leq 7$.

Beginning with stage $k = 1$ and proceeding through to $k = K$, the un-normalized path metrics are found according to

$$\alpha_k^{'}(S_j) = \max\ ^*(\alpha_{k-1}(S_{i1}) + \gamma_{i1j}, \alpha_{k-1}(S_{i2}) + \gamma_{i2j}) \tag{5.3}$$

where, S_{i1} and S_{i2} are the two states in stage *k-1* which are connected to the state S_j in stage k in the trellis. Normalization is done as per

$$\alpha_k(S_i) = \alpha_k^{'}(S_i) - \alpha_k^{'}(S_0) \tag{5.4}$$

The computations of the LLR is done along with the generation of the forward path metric. The likelihood of the branch connecting state S_i at stage *k -1* to state S_j at stage k is

$$\lambda_k(i,j) = \alpha_{k-1}(S_i) + \gamma_{ij} + \beta_k(S_j) \tag{5.5}$$

The LLR is then given as

$$L(X_k) = \max^*{}_{(S_i \to S_j):X_i=1}\lambda_k(i,j) - \max^*{}_{(S_i \to S_j):X_i=0}\lambda_k(i,j)$$

where the max* operator is computed recursively over the likelihoods of all data 1 branches $(S_i \to S_j) : X_i = 1$ or data 0 branches $(S_i \to S_j) : X_i = 0$.

5.5.3 Design Parameters

The main design parameters for a turbo code are the number of iterations, the interleaver size, interleaver design and constituent codes. In this implementation the interleaver design and the constituent codes are fixed since the UMTS standard is being used. UMTS also has provided a standard for the design of the interleaver algorithm. Also the interleaver size is constrained between 40 and 5114. Thus, the two parameters that need to be worked with generally are the number of iterations and the size of the interleaver. It has been found that the BER decreases with a increase in the frame size and therefore the interleaver size. This can be attributed to greater randomness being present in case of a larger set of data to work with. But if the size of the interleaver is too big it may lead to unacceptable latencies in the

system in terms of encoding and decoding delays and, moreover, there would be a need for a greater amount hardware to be implemented. Also, it is known that greater the number of iterations lesser is the BER. This is because with each iteration there is additional information being generated and hence this leads a better estimate of the sent information bit. An important point to be noted is that the improvement diminishes with every iteration and therefore there would be a "sweet-spot" where the we would achieve a fairly good compromise in terms of the BER. Another constraining factor in the decision on the number of iterations is the increase in latency of the system with each iteration. In this implementation we take a starting value of 10 iterations and a frame size of 1000 LLR's. This value can be finalized based on the SNR at the receiver with a lower number of iterations and a smaller interleaver size being used if the SNR is high enough.

Chapter 6

RF System Design

The RF frontend system for the transmitter and the receiver performs the translation to and from the RF frequency band to enable the transmission of information over a wireless channel. This translation of frequencies is done by the utilization of a number of components, either discrete or integrated, which work in cohesion to achieve the maximum possible reliability in transmission and reception. This block is going to be dealt with as a system blocks rather than as a collection of lower level entities to enable the exploration of choices and decisions at a macro-level rather than at the microlevel of sub-system discrete components. It is assumed here that the reader has the required knowledge about the general structure of a transceiver.

6.1 Receiver Architecture Decision

The most fundamental decision made here involved the choice between the two fundamental receiver architectures: Super-Heterodyne and Direct Conversion.

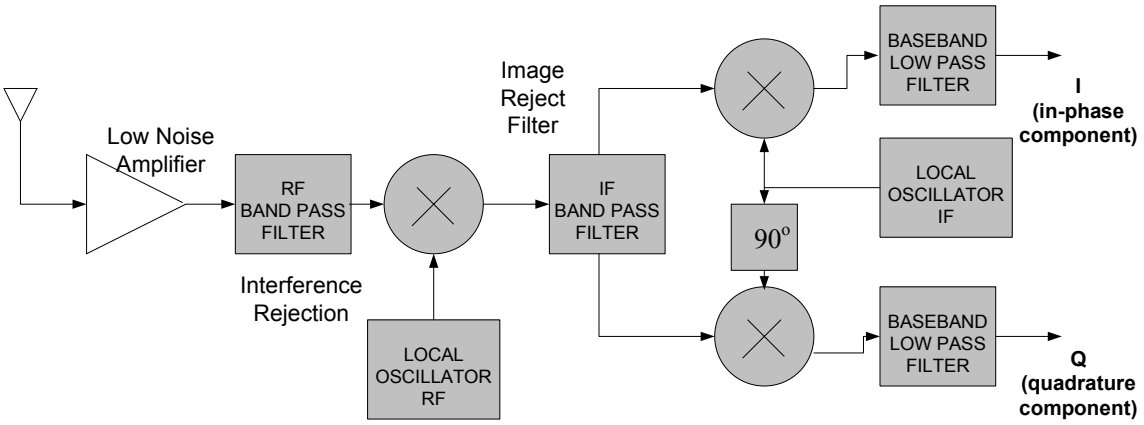

Figure 6.1: Superheterodyne Receiver

6.1.1 Superheterodyne architecture

The principle is to use of more than one mixer stage, to convert the radio frequency to the baseband. The architecture of a generic superheterodyne receiver is as shown in figure 6.1:

The RF signal is applied to a low-noise amplifier (LNA) and then goes through a bandpass filter that performs rejection of out of band interference. It then goes through a mixer with the LO tuned to the RF frequency to downconvert the signal to an intermediate frequency (IF). This is then succeeded by a image reject filter to reject the image of the desired signal and then is downconverted to the baseband frequency.

6.1.2 Direct Conversion Architecture

This is also called Zero-IF architecture and homodyne when the LO is synchronized in phase with the carrier frequency. The generic structure of this architecture is shown in figure 6.2. This architecture differs from the heterodyne concept by the fact that conversion is done directly from the RF to the baseband stage thus avoiding the IF stage. The quadrature channels are necessary in typical phase, frequency

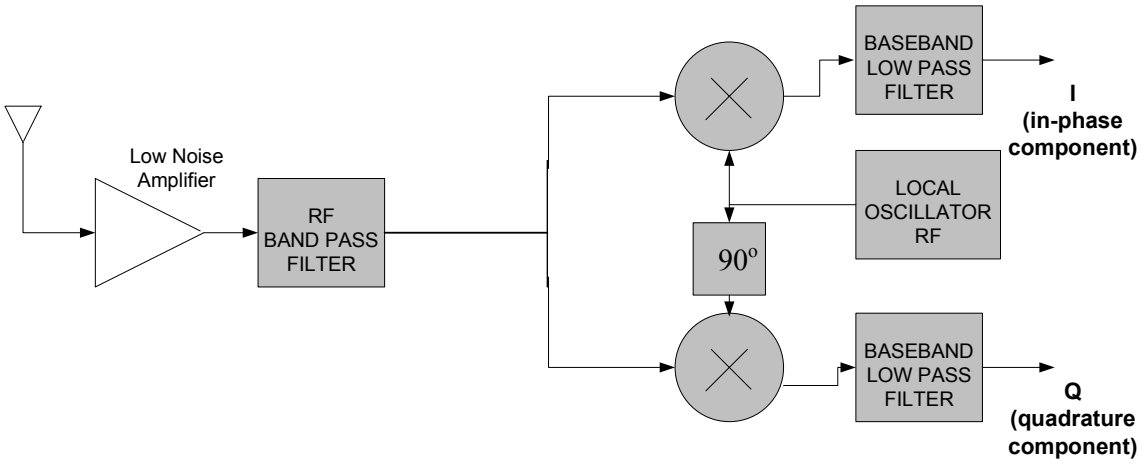

Figure 6.2: Direct-conversion Receiver

and higher order modulated signals because the two sidebands of the RF spectrum contain different information and result in irreversible corruption if they overlap each other without being separated into two phases.

As will be discussed in the next section the direct conversion from RF to baseband has its pros and cons. The most obvious advantage is that there is a reduction in the number of components required since there is no need for the image reject filters and moreover there is a reduction in the number mixers since the IF stage is done away with.

6.1.3 Direct Conversion Issues

Direct Conversion has a major advantage over other other architectures by virtue of the fact that the problem of image rejection is circumvented since the IF is zero [28]. Also the IF SAW filters needed to provide effective reduction of the image are replaced with less expensive Low Pass Filters and baseband amplifiers. The use of lesser number of components makes this scheme more amenable to higher degree of integration.

There are some issues that need to be considered before the use of Direct Conversion architecture:

1. LO leakage arises because a small fraction of the LO energy gets to the antenna through the LNA and the mixer, since these have a limited reverse isolation, which is then radiated out. This becomes in band interference for closely spaced receiver systems tuned to the same frequency. Thus it is always a good practice to use high reverse isolation LNA's and mixers in the receiver that would attenuate most of the leakage.

2. Distortion in the form of second and third order mixing is also an issue in direct conversion architectures since they lead to the translation of higher order closely spaced interferers down to the base-band frequency due to the presence of second and third order non-linearity terms in the LNA stage. But it has been shown that the use of balanced mixers can negate the effect of second order distortion [6]. The third order distortion still exists which, to a certain extent, can be catered to by the use of mixers with greater linearity.

3. DC offset is the most crucial of the issues that need to be addressed. Since the downconverted band extends to zero frequency, extraneous offset voltages can corrupt the signal and, more importantly, saturate the following stage. One of the reasons for the presence of offsets are component mismatches and the presence of LO leakage. The most important thing to ensure is that the DC-offset does not cause the saturation of the ADC's in the system or reduce the dynamic range by a large margin. It can be removed using DSP and the receiver[24]. Another technique to deal with offsets is the use of capacitive-coupling which is useful for wideband systems such as ours [27].

In general, due to the reduced component number and the fact that the off-chip image-reject filter can be removed, the direct-conversion architecture is suitable for designs where cost and level of integration or area are the central motivators. There are a number of single chip solutions for down-conversion receivers. We use MAX2701 Evaluation Kit from MAXIM^{TM} for the reception and use MAX2721 for the transmission. These are matched for transmission and reception at 2.4 GHz i.e. the license-free

ISM band.

6.1.4 Comparison with Low IF

VLIF, or near zero-IF, architectures have many of the desirable properties of ZIF architectures, but without the DC offset problems. This is because they do not downconvert the signal down to the DC level but do so close to it. The RF image rejection performed by the I/Q down conversion mixers can be implemented as either quadrature down or double-quadrature down converters. The channel selection is performed by a polyphase filter, which also aids in image rejection of the final down conversion to DC, as well as relaxing the dynamic range requirements on the A/D converter. Even if these are considered, the fact still remains that there is a requirement for a greater number of components for its implementation which is a major driving force for its lack of use in this implementation since there is a need for portability and monetary considerations to be thought about. Also, as has been stated earlier most of the limitation of the direct conversion architecture can be negated by careful design practices. It is this reasoning that drives our choice for direct conversion receivers.

The MAX solution suite has been designed for use in wideband 2.4GHz ISM radios [3] with features like the use of double balanced mixers so that IIP2 can be maximized, low noise figure, high input IIP3, selectable gain for the LNA and the ability to evaluate the LNA, Mixer and VGA (in case of the Evaluation kits). The RF front end of the system is slated to be done after this document is submitted. The noise characterization and the subsequent system parameterization will be dealt with as a part of the future work in this context.

Chapter 7

Data Acquisition Module

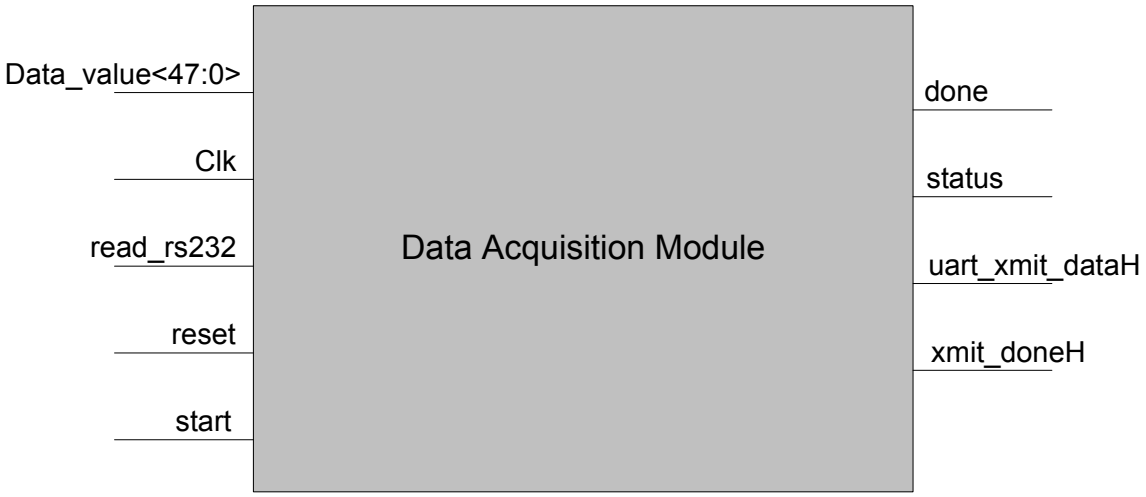

Figure 7.1: Top level block representation of acquisition module

The off-line manner of operation adopted in this implementation necessitates the formation of a memory trace of the data of interest which would then be worked upon using a software approach. This trace formation and readout is done using the module shown in figure 7.1.

The functionality of the different pins are:

data_value

48 bit input from the I/O interface i.e. the P-160 expansion slots. The input comes at the rate FPGA_Clk/2.

Clk

FPGA_Clk. (right now at 100MHz internal xtal oscillator but can be sourced by the Digital Clock Manager on the FPGA).

read_rs232

Signal that initiates the transfer from the internal memory to the RS-232 port. This can be connected to a push button switch on the FPGA to enable debugging and control flow.

reset

System reset.

start

This is the signal from a higher level module that indicates the presence of data at the I/O interface.

done

This signal indicates the completion of the transfer process to the RS-232 ports. Another debugging tool.

status

An active low signal that indicates the completion of the formation of a trace i.e. the entire memory has been filled up by the data from the I/O.

uart_XMIT_dataH

Serial data output by the UART block within the acquisition module. This is connected to the RS-232 port.

At its simplest the function of this module can be explained as reading the 48 bits at the input every second clock cycle and writing this data to memory on the FPGA. After the entire memory is full the trace data is sent out to the PC using a serial communication interface called RS-232 which is fed by *uart_XMIT_dataH* pin after being driven out by a UART core.

7.1 Lower Level of Abstraction of Design

The three blocks that constitute the acquisition block are shown in figure 7.2.

The modularity of the design reflects the division of labour between the different blocks. Lets look at each block in detail.

7.1.1 Trace Formation Block

This block is responsible for the formation of a trace in memory. It reads the I/O interface which runs at half the speed of the FPGAs clock. The FSM of the operation of this block is as shown in figure 7.3.

During two cycles of the controller clock there will be a constant input at the I/O interface. At every second clock of the controller the data from the I/O is clocked in to form one half of a 96 bit memory word. This memory word will then be written into a bank of 3 SelectRAMs, each of 32 bit memory word length. Since there are only 44 SelectRAM blocks we end up using only 42 of them.

There are two distinct stages of operation of this block.
1. On *start* the system begins to read from the I/O and fill up the memory. The timing of the operation will be as shown in figure 7.4.
2. Once the memory is filled *status* goes low and handshaking with the UART begins with the interface block between the two as the arbiter. A new word is read from memory every time *read_check* signal is provided by the interface block. The first read from memory also requires an explicit *read_rs232* signal to be provided to kick off the serial readout process.

7.1.2 Interface Block

This block ensures handshaking between the UART and the trace formation blocks. The FSM of the operation of this block is shown in figure 7.5.

The interface block quite simply provides the *read_check* signal to the trace formation block when a new 96 bit word needs to be read in, and provides the *xmitH* signal to the UART when 8 bits are ready to be read out serially. The 96 bit input therefore needs to be split up into 12 words of 8 bits each before being sent to the UART.

7.1.3 UART core

The UART core performs the transmission of the data in a serial manner through the RS-232 port. The RS-232 port on the FPGA board is configured as shown below figure 7.6. The code was a modification of the code created by Jeung Joon Lee [22].

The Virtex-II Pro development board provides a DB-9 connection for a simple RS232 port. The board utilizes the TI MAX3223 device for driving the RD and TD signals. The RTS and CTS signals are not connected to the DB-9 connector, but are provided if needed for future use. A UART core is needed to use this RS-232 port connection. The different blocks of the UART core are as shown in figure 7.7.

The *baud_clk* signal is related to the system clock "ClK" by

$$baud\_clk = Clk/(BAUDRATE * 16)$$

where $BAUDRATE$ is the rate of transfer of information bits i.e. data bits + start bit + parity bit (optional) + stop bit(s). The $BAUDRATE$ for transmission is fixed by the protocol being used for RS-232 communication.

The structure of the frame used for serial communication is as shown in figure 7.8 The protocol used for this implementation uses one start bit, one stop bit and sends 8 bits of data at a time.

The FSM of the transmission is as shown in figure 7.9.

xmit_dataH pin in the UART is connected to the RXD pin in the RS-232 Driver on the FPGA.

7.1.4 Reading from Serial Port

Use of polling based reading from the port. 2 phases of reading.

1.Configuration of control registers:

1.1) Set interrupt options.
1.2) Line Control Register configuration. Set DLAB on and setup for start bit, stop bit and no parity.
1.3) Line Status Register configuration.
1.4) Set Baud rate - Divisor Latch High Byte and Divisor Latch Low Byte.
1.5) FIFO control register configuration.

2. Reading from the port:
The reading from the RS-232 port is done in a polling manner. The C-program continuously polls the MSB of the LSR control register which indicates the presence of data. When a 1 is seen this implies a byte of data is ready to be read from the port. This is then stored in a file to be processed by the Detection and Turbo Decoding algorithms.

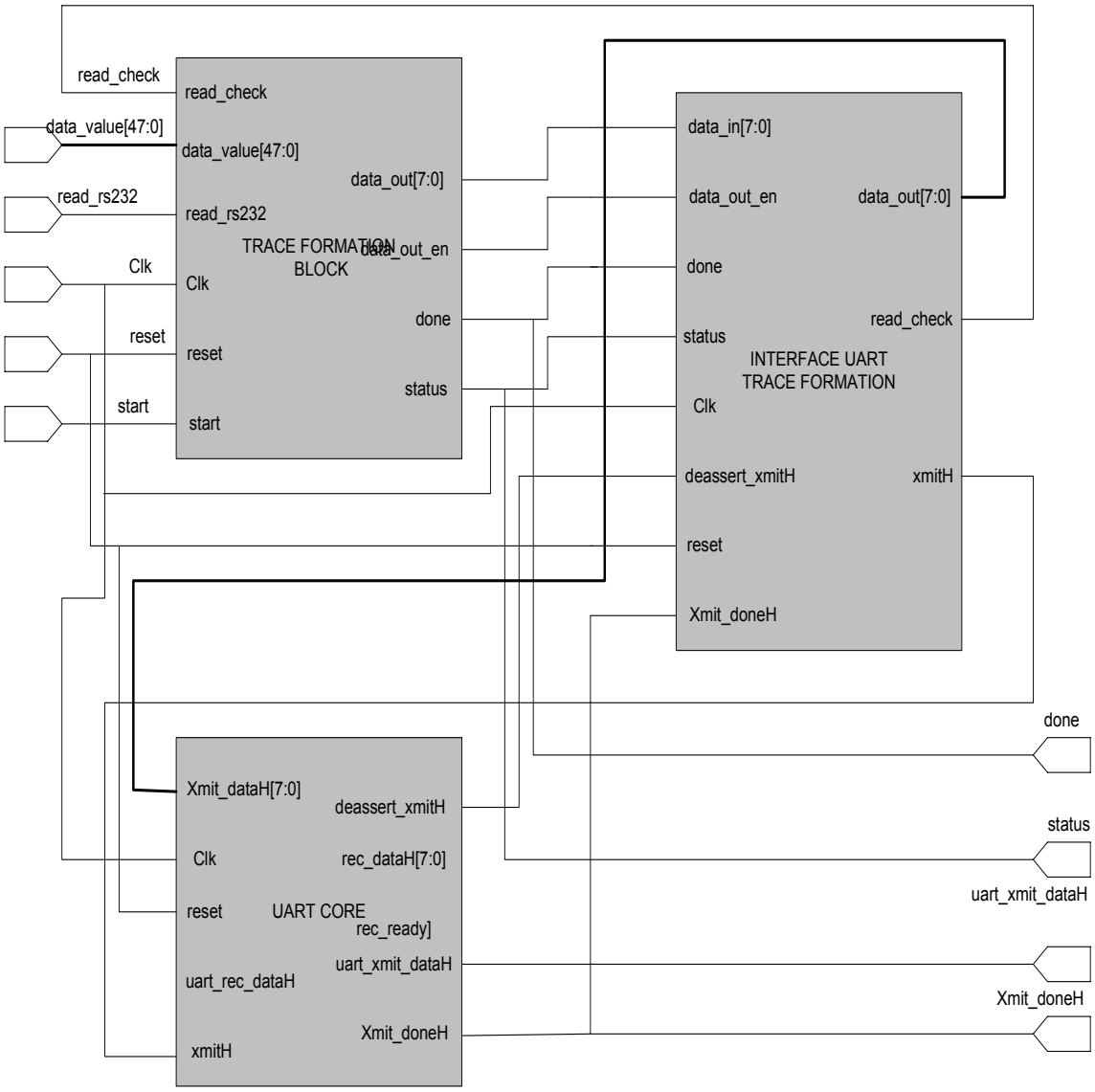

Figure 7.2: Lower level block representation of acquisition module

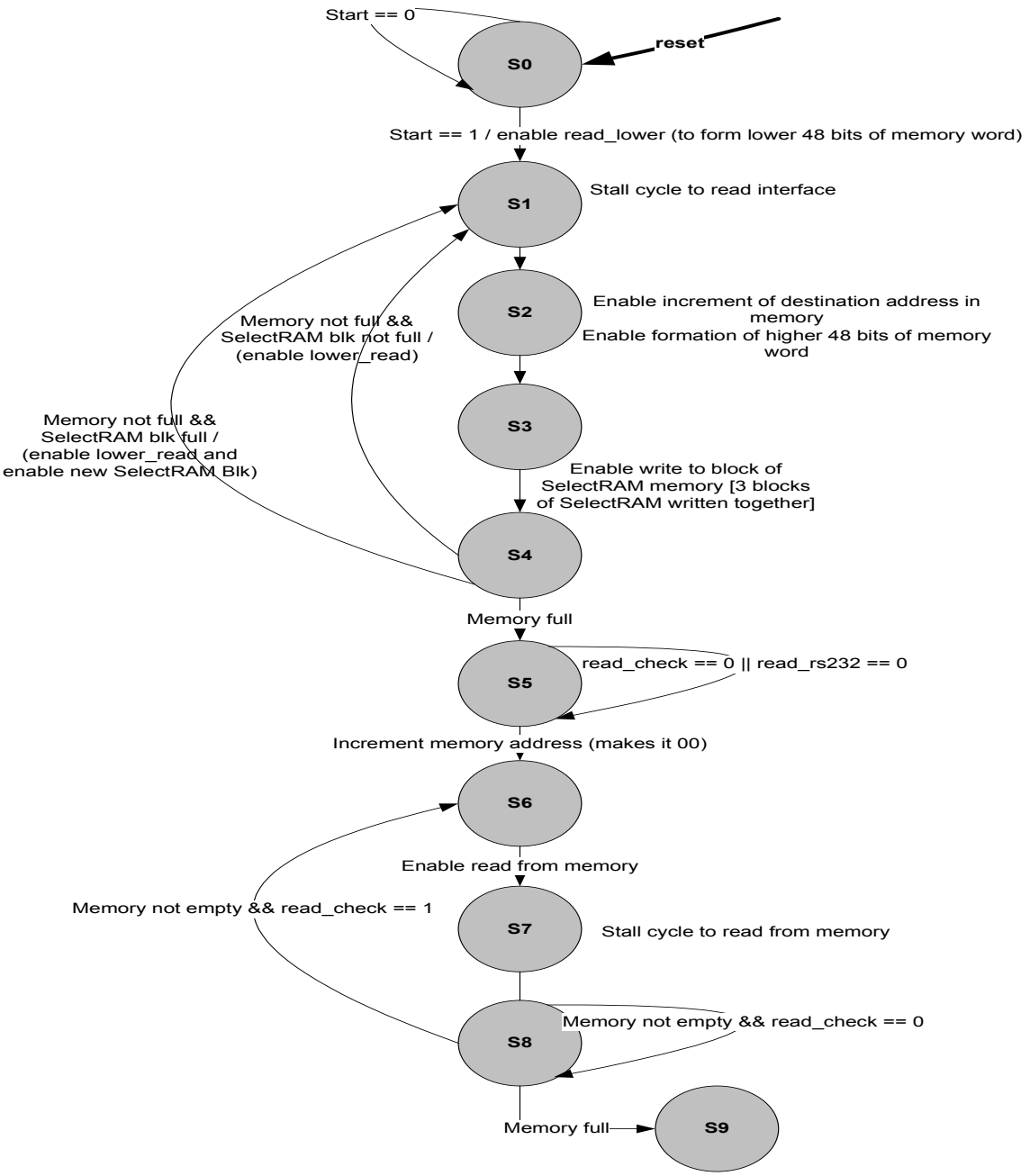

Figure 7.3: FSM of Trace Formation Block in the Acquisition module

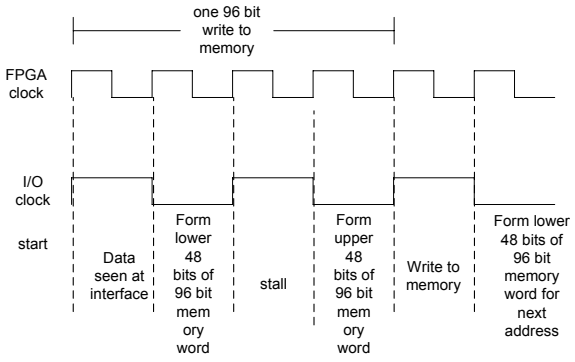

Figure 7.4: The timing of the read from the IO pins on the FPGA

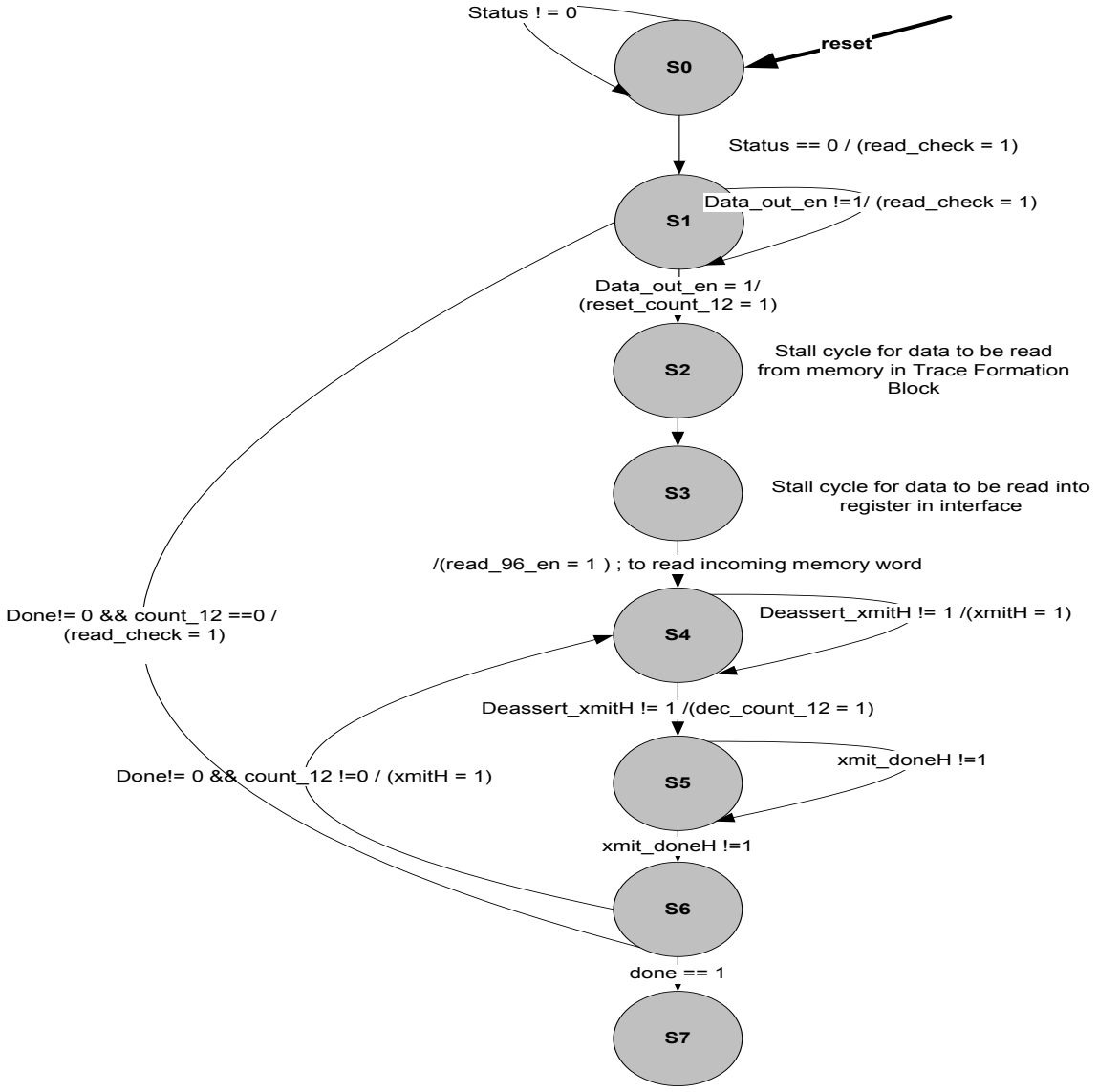

Figure 7.5: FSM for operation of the interface between trace formation block and UART

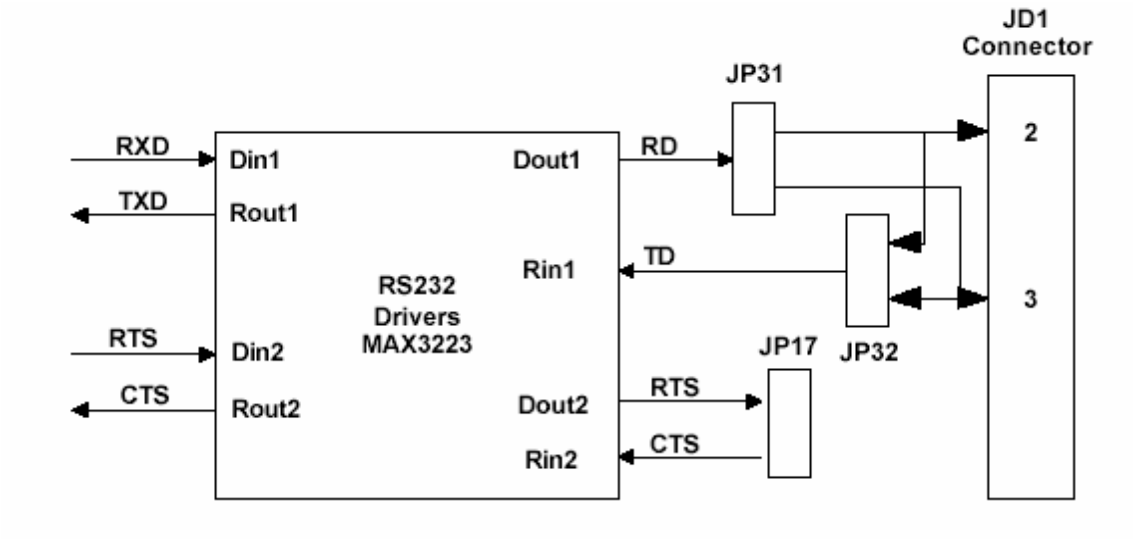

Figure 7.6: The RS-232 Connector available on the XC2VP7 Development Board

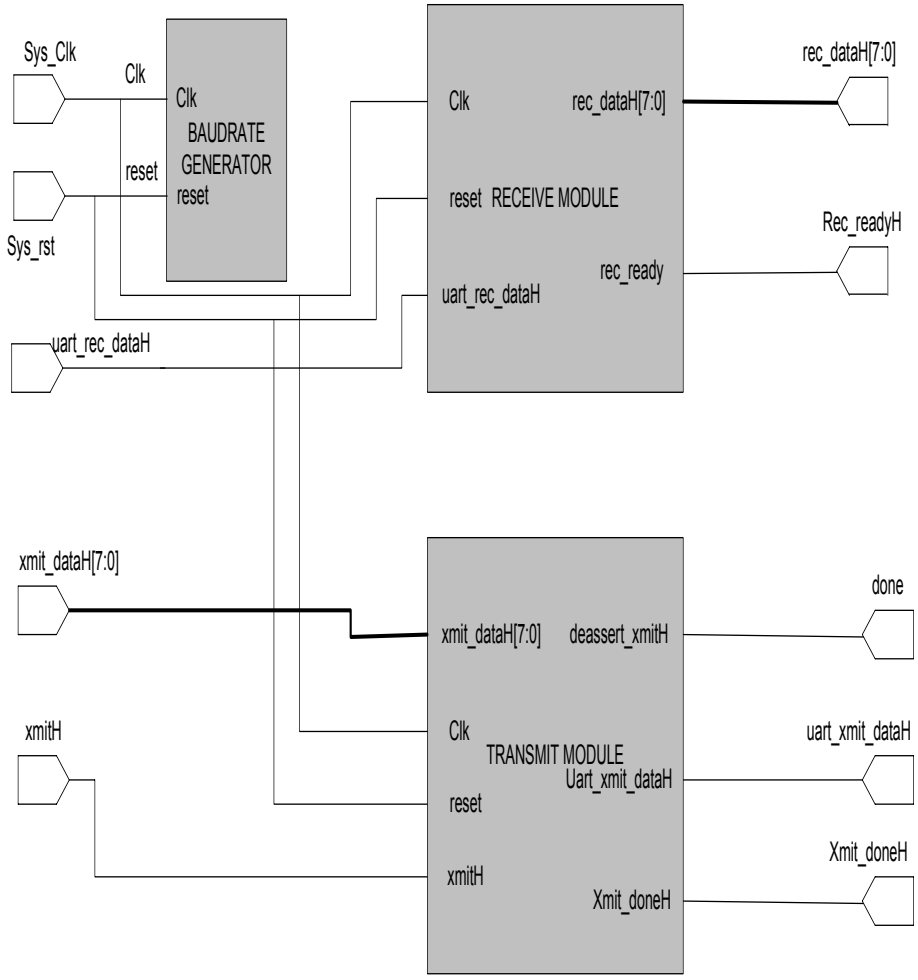

Figure 7.7: Block level representation of the units that form the UART

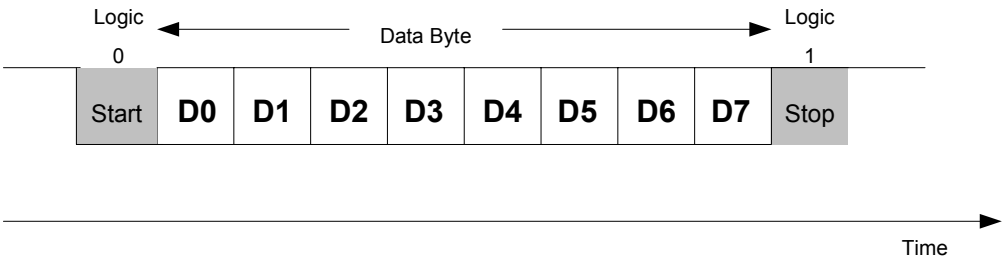

Figure 7.8: Structure of a frame transmitted by the UART

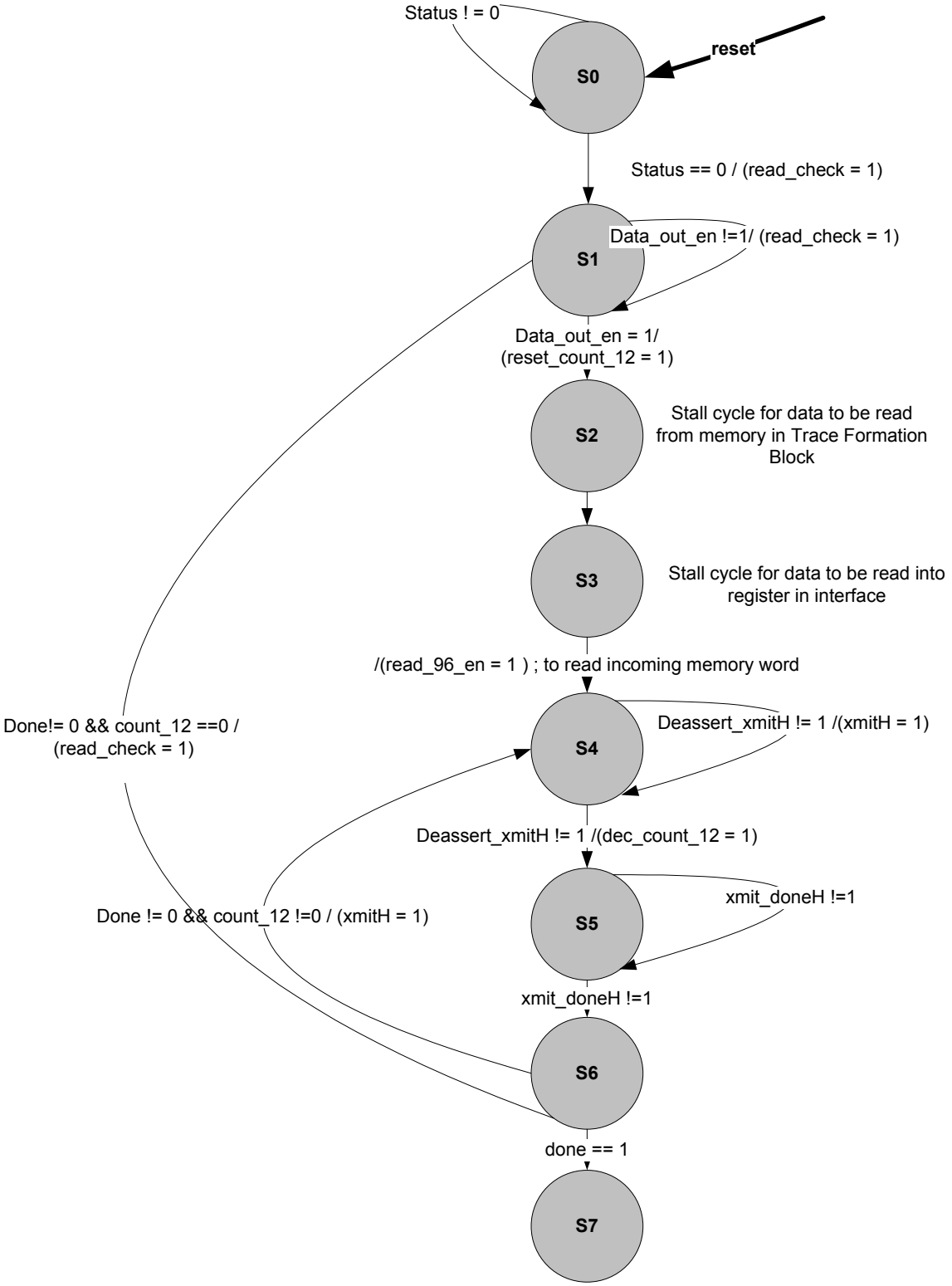

Figure 7.9: FSM of the UART transmission

Chapter 8

Testing and Results

This chapter will provide a quick overview of the means used, assumptions made and results obtained for testing the various modules in the design.

8.1 Data Acquisition Test

Testing of the different modules that constitute this block were done independently and then a comprehensive test was done on the entire module with the formation of a trace and transfer of the trace over the RS-232.

8.1.1 Trace Formation Test

This test involved the validation of the write and read from memory at 100MHz. This was done using the push button switches and LEDs present on the Memec XC2VP7 reference board [5]. The push buttons were used to simulate the signals coming in from the peripheral modules and the LEDs were used to show the status at different phases of operation. In this case, data was simulated by using a data generation module that provided an incremental input given a particular seed which

can be replicated as and when needed. Hitting the different push button switches caused the transition to different holding states which were used to indicate successful completion of certain phases of the design, like waiting for *start*, completion of trace formation and success in checking. The successful completion of a given stage was indicated by use of the LEDs. Final success/ failure was indicated by glowing another LED.

8.1.2 Comprehensive Test

This is done to test the complete data acquisition module and is done in much the same way as the previous test. In this case the phases of operation are the filling of the memory and the next is the transfer of data serially over RS-232. Each phase is triggered using the push button switches. The data that is read out using a C program and written to a trace file is "diffed" with the expected data to determine the presence of errors.

8.1.3 Performance and Results

The transfer of data was found to be successful using the approach above. It is also important to note the final synthesis results for this block as given by XILINX^{TM} Project Navigator:

Maximum frequency of Operation: 132.7 MHz (Conservative Estimate)
Number of slices used in FPGA: 628 out of 5440 (11%)

From the above we can see that this system is still open for utilization since 89% of the slices are still not used. The best case performance of the system using the above frequency estimate will be discussed in the next chapter.

8.2 ML-APP Detection Test

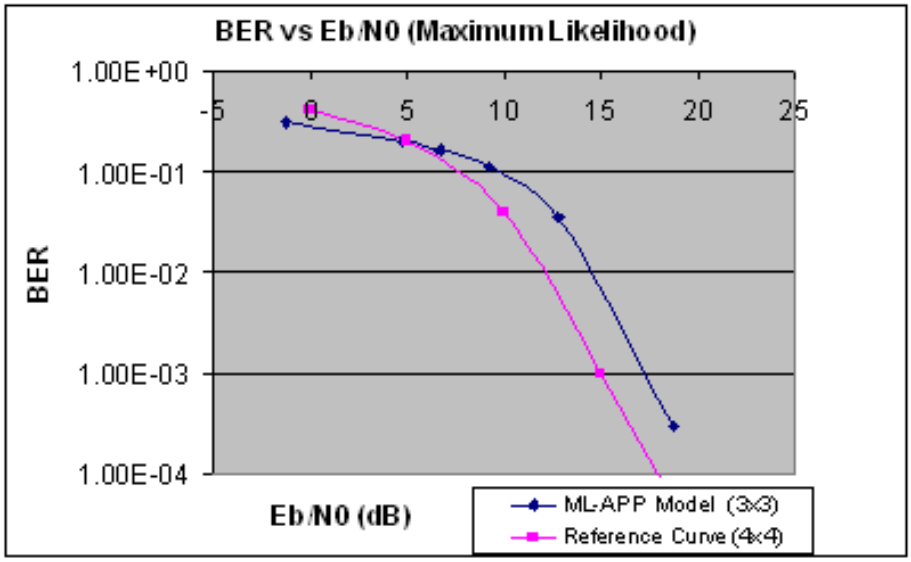

Figure 8.1: BER versus Eb/N0 for ML-APP detector using 16 QAM for a 3x3 system and its comparison with a system of 4x4 configuration

It is important to note that the channel model used for these simulations is that of flat and slow fading in keeping with the assumption that the simulations are being done in an indoor office environment. The flat fading assumption states that the spectral characteristics of the signal are not affected by the channel i.e. the all the frequencies in the signal undergo the same proportion of fading. This enables the representation of the channel coefficient as a complex value. Also, the slow fading channel assumption ensures that the same channel estimate can be used over multiple symbols. Under this assumption, Rayleigh fading was used where the **H** matrix was assumed to be constant for the entire duration of the test. For a given **H**, the real and imaginary parts of the coefficients are modeled as having zero mean and a variance of 1. This is implemented using the *randn()* function in MATLAB^{TM}. Also, the noise is modeled as having a zero mean but with a variance of *sigma*. The average signal power for the constellation used is 1. Thus the energy per bit is 1/4 for 16-QAM. Simulations were run to get the BER versus SNR graph shown in figure 8.1 which also contains the plot with points taken from [8] for verification purposes. The two

curves differ in respect of the number of antennas being present. While the test model has a 3x3 configuration the reference has a 4x4. Also the 4-QAM constellation, used in the reference design, has a better SNR characteristic. This explains the improved performance of the reference curve. It can be seen that the reference curve has a 3 dB improvement over the curve for the 3 x 3 design for a BER of 10^{-2}.

8.3 Turbo Decoding Test

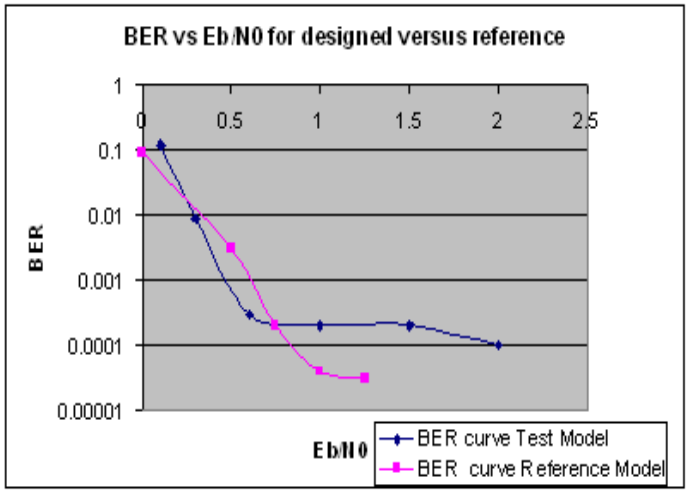

Figure 8.2: BER versus Eb/N0, 4 iteration and a block size of 10000; comparison with reference

A lot of documentation has been presented in the case of performance using BPSK modulation in AWGN channels using matched filters at the receiver. This is the model that will be utilized here too to enable ease of validation with present results already published. More about the model can be found at [33]. The resulting graph from simulation of BER versus Eb/N0 is as shown in figure 8.2 and for verification has been compared against the reference graph published in [31]. The difference can be justified on the basis of the number of data points taken. In the case of the reference the total number of samples is of the order of a million while in the case of the model being tested it is 10000. The reduced size of the data set was taken because the

time for execution would otherwise have been very high. This is the reason behind the flattening of the curve prematurely. It is to be noted that the formation of the statistics for the 10000 sample curve shown took about three hours to run and if this test had to be run for the requisite million samples the test would take about 2 weeks. This statistic is the reason for the shortened test we have run.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

The thesis has presented a general overview of the options that are present in designing a multiple antenna system. It also describes the design idea for the characterization of a MIMO system that will be used for the characterization of a co-polarized and co-located antenna structure. The system at present has a tested digital backend with acquisition being done using an FPGA based design while all the signal processing is done in using MATLAB. Let us look at the specifications of the system:

As has been stated we use 42 of the 44 16Kbit usable blocks of SelectRAM on the FPGA for trace formation. This gives us 688 Kbits of sample data. Since each sample is of 48 bits the total number of samples that can be obtained is 14,336 samples. Since there is an oversampling of 4, we have 3548 samples of actual information samples. Considering the worst case of 500 samples for channel estimation, we can obtain a total of 3 Ksamples of information. We are interested in obtaining bit error rates (BER) in this implementation. Therefore we should look at the samples as a collection of 12 bits of information, i.e. 3 antennas each using 16 QAM constellation, giving us

a total of approximately 36,000 bits of trace information. This enables us to perform computations in BER up to 10^{-4}. It is generally seen in literature that most of the SNR versus BER curves are of the order of 10^{-4} [14, 10] and hence this testbed should provide sufficient information to characterize the performance of the antenna structure.

9.2 Future Work

On a short term basis, the aim is to perform the MATLAB implementation of synchronization and channel estimation. We will then move on to the design of the Interface board and perform the parameterization of the RF-frontend using noise and power level assumptions of the IEEE 802.11 standard. Also the usage of the on-board SDRAM in place of the SelectRAM is going to be looked into.

Over a long term the aim is to perform the study the effect of environment, space-time coding, channel coding and interleaver design on the capacity gains of the antenna. Also the aim is to port a major part of the design onto a hardware block such as the FPGA.

Bibliography

[1] Bell Labs scientists shatter limit on fixed wireless transmission: Lucent Bell Labs press release, 9th September 1998. See http://www.lucent.com/press/0998/980909.bla.html.

[2] www.sworld.com.au.

[3] 1.8GHz to 2.5GHz Direct Downconversion Recievers. available at http://pdfserv.maxim-ic.com/en/ds/MAX2700-MAX2701.pdf.

[4] Practical Aspects of Multiple Architectures for HSDPA. 3GPP Tech. Rep. TSGR1 16 (00)1219.

[5] Virtex-II Pro (P4/P7) Development Board user's guide. available from http://www.memec.co.jp/html/xilinx/eboard/docs/dskit2v/v2pro_board_users_guide_1_0.pdf.

[6] Asad A. Abidi. Design Considerations for Direct-Conversion Receivers. In *IEEE Journal of Solid-State Circuits*, Dec. 1995.

[7] A. Adjoudani, E. Beck, A. Burg, G. Djuknic, T. Gvoth, D. Haessig, S. Manji, M. Milbrodt, M. Rupp, D. Samardzija, A. Siegel, T. Sizer II, C. Tran, S. Walker, S. A. Wilkus, and Wolniansky. Prototype Experience for MIMO BLAST over Third Generation Wireless System. In *IEEE Journal on Selected Areas in Communication*, Apr. 2003.

[8] H. Artes, D. Seethaler, and F Hlawatsch. Efficient Detection Algorithms for MIMO Channels: A Geometrical Approach to Approximate ML Detection. In *IEEE Transactions on Signal Processing*, Nov 2003.

[9] Ernst Aschbacher, Sebastian Caban, Christian Mehlfuhrer, Georg Maier, , and Markus Rupp. Design of a Flexible and Scalable 4x4 MIMO Testbed.

[10] Inaki Berenguer and Xiaodong Wang. Coding and Signal Processing for MIMO Communications- A Primer. available from http://www-Ice.eng.cam.ac.uk/~ib226/cv/papers/MIMsurvery.pdf.

[11] A. Burr. Turbo-codes: The Ulmtimate Error Control Codes? In *Electronics and Communication Engineering Journal*, Aug. 2001.

[12] A. Burr. MIMO Wireless Systems: Principles, Potential, Problems and Prospects. In *COST273 Workshop in Helsinki*, May 2002.

[13] D. C. Garrett, L. M. Davis, and G. K. Woodward. 19.2 mbit/s 4 x 4 BLAST/MIMO detector with soft ML outputs. In *ELECTRONICS LETTERS*, 2003.

[14] David Gesbert, Mansoor Shafi, Da shan Shiu, Peter J. Smith, and Ayman Naguib. From Theory to Practice: An Overview of MIMO Space-Time Coded Wireless Systems. In *IEEE Journal On Selected Areas in Communications*, Apr. 2003.

[15] M. Guillaud, A. Burg, M. Rupp, E. Beck, and S. Das. Rapid Prototyping Design of a 4x4 BLAST-over-UMTS Downlink. In *Proc. of the 35th Asilomar Conference on Signals, Systems and Computers*, 2001.

[16] M. Guillaud, A. Burg, M. Rupp, E. Beck, D. Perels, and N. Felber adn W. Fichtner. FPGA Implementation of a MIMO Receiver Front-end for the UMTS Downlink. In *International Zurich Seminar on Broadband Communications*, 2002.

[17] Zhan Guo and Peter Nilsson. A 50 mbps 4/spl times/4 maximum likelihood decoder for multiple-input multiple-output systems with qpsk modulation. In

Proceedings of The 10th IEEE International Conference on Electronics, Circuits and Systems, Dec 2003.

[18] Zhan Guo and Peter Nilsson. A V-BLAST Detector Based on Square Root Algorithm. In *Proceedings of The 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*, 2003.

[19] Fu hua Huang. Evaluation of Soft Output Decoding for Turbo Codes. available at http://scholar.lib.vt.edu/theses/available/etd-71897-15815/unrestricted/etd.pdf.

[20] Robert W. Heath Jr, S. Sandh, and A.J. Paulraj. Space-Time Block Codes versus Space-Time Trellis Codes. In *Proceedings of IEEE International Conference on Communications*, June 2001.

[21] Anand Konanur, Keyoor Gosalia, Sandeep Krishnamurthy, Brian Hughes, and Gianluca Lazzi. Investigation of the Performance of Co-polarized, Co-located Electric and Magnetic Dipoles for Increasing Channel Capacity. In *IEEE Antennas and Propagation Society International Symposium*, June 2003.

[22] Jeung Joon Lee. Micro-UART: Synthesizable UART. available at http://www.cmosexod.com/ip/u-uart/micro-uart.PDF.

[23] Rupp M.and Guillaud M. and Das S. On MIMO Decoding Algorithms for UMTS. In *35th Asilomar Conference on Signals, Systems and Computers*, 2001.

[24] Tony Montalvo. Highly Integrated RF and Wireless Transceivers. Class Notes for ECE792X: IC for Wireless Communications.

[25] Wolniansky P.W., Foschini G.J., Golden G.D., and Valenzuela R.A. V-BLAST: an Architecture for Realizing Very High Data Rates over the Rich-Scattering Wireless Channel. In *Proceedings of IEEE 1998 URSI International Symposium on Signals, Systems, and Electronics*, Sept 1998.

[26] Wolniansky P.W., Foschini G.J., Golden G.D., and Valenzuela R.A. Detection Algorithm and Initial Laboratory Results using the V-BLAST Space-Time Communication Architecture. In *Electronics Letters Vol. 35*, Sept 1999.

[27] Behzad Razavi. Challenges in Portable RF Transciever Design. In *IEEE Circuits and Devices Magazine*, Sept. 1996.

[28] Behzad Razavi. Design Considerations for Direct-Conversion Receivers. In *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, June. 1997.

[29] Javier R. Fonollosa Roger Gaspa. Comparison of Different Transmit Diversity Space-Time Code Algorithms. available at http://gps-tsc.upc.es/comm/main/publications/papers_comm/p2000/roger_IST2000.pdf.

[30] Karen Su. Space-Time Coding: From Fundamentals to the Future. available at http://www.srcf.ucam.org/~ks349/passdoc.php3?name=Su03C&type=pdf.

[31] Jun Tan and Gordon Stuber. New SISO Decoding Algorithms. In *IEEE Transactions on Communications*, June. 2003.

[32] M. C. Valenti and J. Sun. The UMTS Turbo Code and an Efficient Decoder Implementation Suitable for Software-Defined Radios. In *International Journal of Wireless Information Networks*, Oct. 2001.

[33] Matthew Valenti. *Iterative Detection and Decoding for Wireless Communications*. PhD thesis, 1999.

[34] R. van Nee, A. van Zelst, and G. Awater. Maximum Likelihood Decoding in a Space Division Multiplexing System. In *Proceedings of IEEE Vehicular Technology Conference (VTC)*, 2000.

[35] Weidong Xiang, Deric Waters, Thomas Pratt, and John Barry. Implementation and Experimental Results of a Three-Transmitter Three-Receiver (3x3) OFDM/BLAST Testbed. In *submitted to Wireless Communications Magazine*, Nov. 2003.