

# A software-based fading channel simulator

by

Christopher H. Snow

Faculty Advisor: Dr. S. Primak

Electrical/Computer Engineering Project Report  
submitted in partial fulfillment of the degree of  
Bachelor of Engineering Science

Department of Electrical and Computer Engineering  
The University of Western Ontario  
London, Ontario, Canada  
March 31, 2003

© Christopher H. Snow 2003

All rights reserved

# Abstract

A software-controlled, hardware-based simulator of the mobile wireless multipath fading channel has been designed and implemented. Multipath fading occurs when several reflected versions of a transmitted radio signal arrive at the receiver. Movement of the receiver causes the number of multipath components, as well as their phases and propagation delays, to vary with time. The multiple received signals can add constructively or destructively, causing the power level at the receive antenna to fluctuate over a large dynamic range. This fluctuation causes an increased bit error rate relative to the classical additive white gaussian noise channel. The central focus of this project is to simulate the fading signal in a laboratory environment, for use in testing of mobile wireless devices such as cell phones and pagers. Currently available hardware simulators of the fading channel have various limitations, the most serious of which is the lack of flexibility in the fading model used. A proof-of-concept software-controlled fading simulator for a digital radio operating at 434 MHz has been developed, based on error generation at the bitstream of the transmitter. Measured bit error rates will be compared with the theoretical fading-channel performance of this radio. The benefits of the chosen design as well as suggestions for commercialization will be discussed.

**Keywords:** wireless communication, multipath fading channel, software-controlled hardware-based fading simulator, statistical multipath fading models, Nakagami- $m$  distribution, Markov chain

# Acknowledgements

A special thanks to the family and friends who have provided enlightening discussion, ideas, encouragement and support. Thanks go in particular to my supervisor and collaborator Dr. Serguei Primak, with whom I have worked on several research projects over the past two years. His support as ECE416 advisor has shaped the direction of this work. Credit is due to him for many of the ideas presented herein, while any faults are mine alone.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Programs</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Brief Technical Overview . . . . .	1
1.2 Motivation . . . . .	1
1.3 Design Considerations . . . . .	2
1.4 Outline of the Report . . . . .	2
<b>2 Literature Review</b>	<b>3</b>
2.1 Introduction . . . . .	3
2.2 Statistics of the fading channel . . . . .	3
2.2.1 Envelope distributions . . . . .	3
2.2.2 Correlation function . . . . .	4
2.3 Characterizing channel error sequences . . . . .	5
2.4 Channel simulators . . . . .	6
<b>3 Statement of Problem and Methodology of Solution</b>	<b>7</b>
3.1 Problem Statement . . . . .	7
3.2 General Design . . . . .	7
3.3 Design Constraints . . . . .	8
3.3.1 Fading simulated at the transmitter . . . . .	8
3.3.2 Minimum modification to transmitter hardware . . . . .	9
3.3.3 Software controllable . . . . .	9
3.3.4 Real-time simulation . . . . .	9
3.3.5 Transparent to equipment under test . . . . .	9
3.4 Possible Solutions . . . . .	10
3.4.1 Use samples from measurements of fading . . . . .	10
3.4.2 In-phase and quadrature AWGN channels . . . . .	10

3.4.3	Sum-of-sinusoids circuit . . . . .	10
3.4.4	Mismatch circuits at the transmit antenna . . . . .	11
3.4.5	Voltage-controllable RF amplifier . . . . .	11
3.4.6	Interface at the bitstream of the transmitter . . . . .	11
3.5	Chosen Design . . . . .	12
<b>4</b>	<b>Design of the Fading Channel Simulator</b>	<b>13</b>
4.1	Introduction . . . . .	13
4.2	Hardware . . . . .	14
4.2.1	Transmitter . . . . .	14
4.2.2	Receiver . . . . .	15
4.2.3	Error Generator . . . . .	15
4.2.4	Licensing in the 434 MHz Band . . . . .	16
4.3	Software . . . . .	16
4.3.1	Numerical Calculation of Fading Model Parameters . . . . .	17
4.3.2	Hardware Interface Software . . . . .	18
4.3.3	Transmitter and Receiver . . . . .	19
<b>5</b>	<b>Results and Discussion</b>	<b>20</b>
5.1	Introduction . . . . .	20
5.2	Results . . . . .	20
5.3	Discussion . . . . .	21
5.3.1	Benefits of this solution . . . . .	21
5.3.2	Drawbacks to this solution . . . . .	22
<b>6</b>	<b>Conclusions</b>	<b>23</b>
6.1	Future Work . . . . .	23
6.2	Conclusions . . . . .	23
<b>A</b>	<b>Code Listings — Hardware Interface</b>	<b>24</b>
A.1	Error Controller — <code>Controller.c</code> . . . . .	24
A.2	Transmitter — <code>Transmitter.c</code> . . . . .	26
A.3	Receiver — <code>Receiver.c</code> . . . . .	27
<b>B</b>	<b>Code Listings — Parameter Calculation</b>	<b>30</b>
B.1	Calculate the Parameters of the Markov Model . . . . .	30
B.2	$P_{err}$ for ASK Modulation in Nakagami- $m$ Fading . . . . .	32
B.3	$P_{err}$ for ASK Modulation in AWGN . . . . .	32
B.4	Nakagami- $m$ PDF of the SNR . . . . .	33
B.5	The Gaussian Q function . . . . .	34
<b>C</b>	<b>Circuit Diagrams</b>	<b>35</b>
C.1	Transmitter . . . . .	35
C.2	Receiver . . . . .	36
C.3	Error Generator . . . . .	37

<b>D Selected Product Data Sheets</b>	<b>38</b>
D.1 TLP-434 Transmitter and RLP-434 Receiver . . . . .	39
D.2 Maxim MAX232 RS-232 Level Converter . . . . .	40
D.3 Fairchild 74LS04 Hex Inverter . . . . .	42
D.4 MC14067B Analog Multiplexer/Demultiplexer . . . . .	43
<b>Bibliography</b>	<b>45</b>
<b>Vita</b>	<b>47</b>

# List of Tables

4.1	Components for the Transmitter . . . . .	14
4.2	Components for the Receiver . . . . .	15
4.3	Components for the Error Generator . . . . .	16

# List of Figures

2.1	Two-state Markov Model . . . . .	5
3.1	General design of a Fading Simulator . . . . .	8
4.1	Block Diagram of the Software-controlled Fading Simulator . . . . .	13
4.2	Block Diagram of the Transmitter . . . . .	14
4.3	Block Diagram of the Receiver . . . . .	15
4.4	Block Diagram of the Error Generator . . . . .	16
5.1	Theoretical and Simulated BER for $m = 2$ , $\bar{\gamma} = 10\text{dB}$ , $f_D = 10\text{ Hz}$ . . . . .	21
C.1	Circuit Diagram of the Transmitter Level Converter . . . . .	35
C.2	Circuit Diagram of the TLP-434 ASK Transmitter . . . . .	35
C.3	Circuit Diagram of the RLP-434 ASK Receiver . . . . .	36
C.4	Circuit Diagram of the Receiver Level Converter . . . . .	36
C.5	Circuit Diagram of the Transmitter Error Generator Interface . . . . .	37
C.6	Circuit Diagram of the Inverter . . . . .	37
D.1	Data Sheet for TLP-434 Transmitter and RLP-434 Receiver . . . . .	39
D.2	Data Sheet for MAX232 RS-232 Level Converter (page 1 of 2) . . . . .	40
D.3	Data Sheet for MAX232 RS-232 Level Converter (page 2 of 2) . . . . .	41
D.4	Data Sheet for Fairchild 74LS04 Hex Inverter . . . . .	42
D.5	Data Sheet for MC14067B Multiplexer/Demultiplexer (page 1 of 2) . . . . .	43
D.6	Data Sheet for MC14067B Multiplexer/Demultiplexer (page 2 of 2) . . . . .	44



# List of Programs

A.1	Controller.c . . . . .	24
A.2	Transmitter.c . . . . .	26
A.3	Receiver.c . . . . .	27
B.1	CalcMarkovParams.m . . . . .	30
B.2	askNakagami.m . . . . .	32
B.3	askBER.m . . . . .	32
B.4	NakagamiSNR.m . . . . .	33
B.5	gaussianQ.m . . . . .	34

# Chapter 1

## Introduction

### 1.1 Brief Technical Overview

The goal of this project is to create a hardware-based, software-controlled simulator of *multipath fading*, a phenomenon which exists on the mobile wireless radio channel. Multipath fading occurs when mobile devices receive several reflected versions (called *multipath components*) of a transmitted radio signal. Movement of the receiver causes the number of multipath components, as well as their phases and propagation delays, to vary with time. The multiple signals can add constructively or destructively, causing the power level, and hence the signal-to-noise ratio (SNR), at the receive antenna to fluctuate over a large dynamic range. This fluctuation causes an increased bit error rate (BER) relative to the classical additive white gaussian noise channel.

Such a simulator could be used in a laboratory environment for testing of mobile wireless devices such as cell phones and pagers.

### 1.2 Motivation

Devices which simulate the conditions of the fading channel as described in the previous section do exist in the commercial market. However, all such devices suffer from various drawbacks, the most relevant of which are high cost (tens of thousands of dollars) and a lack of flexibility with regards to the model of the fading channel.

There is by no means one perfect model of the mobile fading channel. One extremely important area of research in wireless communication is the experimental and

statistical modeling of the wireless channel. Many researchers have proposed different models for the fading channel, which will be discussed in more detail in Chapter 2.

## 1.3 Design Considerations

Given that the goal of the project is to decrease the cost and increase the flexibility of the simulator as compared to the currently available commercial models, several design considerations can be set:

- Low cost of hardware components
- Use of software control for increased flexibility
- User-controllable software parameters for easy reconfiguration

It will be seen that these considerations will greatly influence the final design of the fading simulator.

## 1.4 Outline of the Report

The report will continue in Chapter 2 with a review of some important literature and results related to modeling of the mobile fading channel. Specifically, the areas addressed will be: statistical characterization of fading, modeling of channel error sequences, and simulation of the fading channel. Chapter 3 will introduce further design constraints and outline possible solutions. Details of the chosen design will be given in Chapter 4. Chapter 5 will provide some performance results of the simulator, and analyze the benefits and drawbacks of the chosen solution. Future work and conclusions will be presented in Chapter 6. Several appendices contain the circuit diagrams and code listings for the fading simulator presented in this report, as well as the relevant product data sheets.

# Chapter 2

## Literature Review

### 2.1 Introduction

Because many channels (such as the 800, 900, 1800, and 1900 MHz bands used in digital cellular communication) suffer from multipath fading, there has been a great deal of interest in the mathematical modeling and simulation of the fading channel [1–4]. A review of the literature has been conducted, focusing on three important areas: the statistical characterization of fading, the modeling of channel error sequences, and simulation of the fading channel.

### 2.2 Statistics of the fading channel

#### 2.2.1 Envelope distributions

The probability density function (PDF) of the received signal envelope is an important part of fading channel modeling. This distribution is required for the calculation of the average probability of error and, along with the joint PDF, can be used to calculate the average duration of fades as well as other important statistics [1]. Several classical PDFs are often applied in the study of the fading channel envelope: the Rayleigh, Rician, and Nakagami- $m$  PDFs.

The Rayleigh PDF for the envelope is given in [5] as

$$p_R(r) = \frac{1}{2\sigma^2} e^{-\frac{r^2}{2\sigma^2}} \quad (2.1)$$

It applies in a situation where many diffuse waves (and no direct ray) arrive at the receiver. In the case where there is a strong direct ray, the envelope PDF is better

characterized by the Rician distribution [5]

$$p_R(r) = \frac{r}{\sigma^2} e^{-(r^2+s^2)/2\sigma^2} I_0\left(\frac{rs}{\sigma^2}\right), \quad r \geq 0 \quad (2.2)$$

where  $I_0(\cdot)$  is a modified Bessel function of the first kind of order zero [6].

The Nakagami- $m$  PDF, given by

$$p_R(r) = \frac{2}{\Gamma(m)} \left(\frac{m}{\Omega}\right)^m r^{2m-1} e^{-mr^2/\Omega} \quad (2.3)$$

where  $\Gamma(\cdot)$  is the Gamma function [6] and  $\Omega = E\{R^2\}$ , is a generalized probability density function with parameter  $m$  which can be varied to represent different amounts of contribution from direct rays and from diffuse components. The parameter  $m$  is also related to the *severity* of the fading. We can identify three regions for the parameter  $m$  and associate physical meaning to each one:

$m > 1$  : Some line of sight component, tends to Rician as  $m \rightarrow \infty$

$m = 1$  : Rayleigh-distributed fading — many diffuse components from all angles

$m < 1$  : Sub-Rayleigh fading (more severe than Rayleigh)

Using these PDFs, one can characterize the channel fading in terms of the probability that the received SNR is at a certain level. Other statistics that are useful include the joint PDFs of Equations 2.1, 2.2, and 2.3, which allow for the calculation of the *transitional* probabilities (the probability of being at a certain SNR level  $\gamma_1$  and transitioning to a second SNR level  $\gamma_2$ ). The reader is referred to [1] for further details.

## 2.2.2 Correlation function

The classic work of Jakes [7] established that the correlation function of a mobile receiver is given by

$$R(\tau) = J_0(2\pi f_D \tau) \quad (2.4)$$

where  $J_0(\cdot)$  is the Bessel function of the first kind [6] and  $f_D$  is the maximum Doppler frequency of the mobile, given by

$$f_D = \frac{v_m}{\lambda_c} \quad (2.5)$$

with  $v_m$  being the velocity of the mobile and  $\lambda_c$  being the carrier frequency. Typical Doppler frequencies may be on the order of  $f_D = 10\text{Hz}$  for pedestrian users, and  $f_D = 100\text{Hz}$  for vehicular users [7].

This result is important because any fading channel simulator should attempt to match the statistics of the real fading channel as closely as possible. Many of the more mathematically tractable models do not replicate Jake’s correlation function [8], and some academic debate has been waged over the issue of how much accuracy is necessary in modeling the correlation function [9,10].

### 2.3 Characterizing channel error sequences

An overview of models for channels with memory is given by the classic paper of Kanal and Sastry [11]. The models given are useful for the characterization of the error process of the channel. These models can be used to generate bit error patterns which resemble those of a real channel. More recent work includes the use of Finite-State Markov Modeling [8,12] and Hidden Markov Modeling [13]. A graphical representation of a two-state Markov model (the so-called “Gilbert-Elliot” model) is given in Figure 2.1. An error sequence generator using this model will transition between the two states according to the transition probabilities  $P_{1\rightarrow 1}$ ,  $P_{1\rightarrow 2}$ ,  $P_{2\rightarrow 1}$  and  $P_{2\rightarrow 2}$ . In state 1 it will generate more errors (due to the high  $P_{\text{err}}$ ), while in state 2 it will generate fewer errors. The resultant error sequence will contain *bursts* of errors due to the non-uniform distribution introduced by the Markov model.

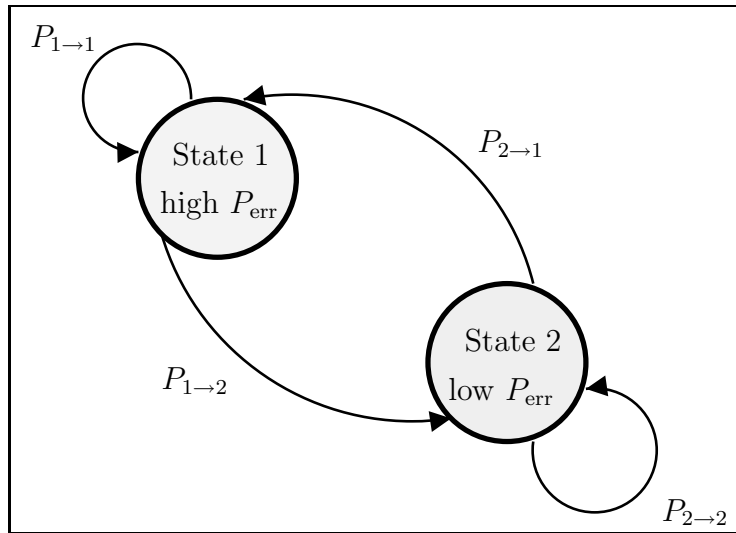


Figure 2.1: Two-state Markov Model

## 2.4 Channel simulators

Because of the ever-present development and deployment of new communication technologies, there is great interest in the simulation of mobile radio channels in software [14,15] as well as hardware [16]. Generally speaking, some type of algorithm is devised for approximating the error conditions on the fading channel. The simulator can operate at the signal level (describing the instantaneous SNR), the bit level (making an error/no error decision for each bit), or the block level (making a decision about blocks at a time). The appropriate level of modeling depends on the type of study being performed. From the point of view of this project, signal and bit level channel models [2,10,12] are the most relevant.

# Chapter 3

## Statement of Problem and Methodology of Solution

### 3.1 Problem Statement

The purpose of this project is to design a hardware device which, when controlled by an external software program, will provide an accurate simulation of the multipath fading channel.

### 3.2 General Design

Before detailing the specific design chosen for this project, it is instructive to examine the general design of a hardware fading simulator, and to look at the constraints imposed on the solution.

The general design for a software-controlled fading simulator is given in Figure 3.1 on page 8. Generally, the device under test in a fading simulator would be set up as the receiver. The details of the error generator in Figure 3.1 are not specified, but in general the error generator is some device that interfaces with the transmitter.

From this general design, we can impose several constraints on the solution, which are discussed below.



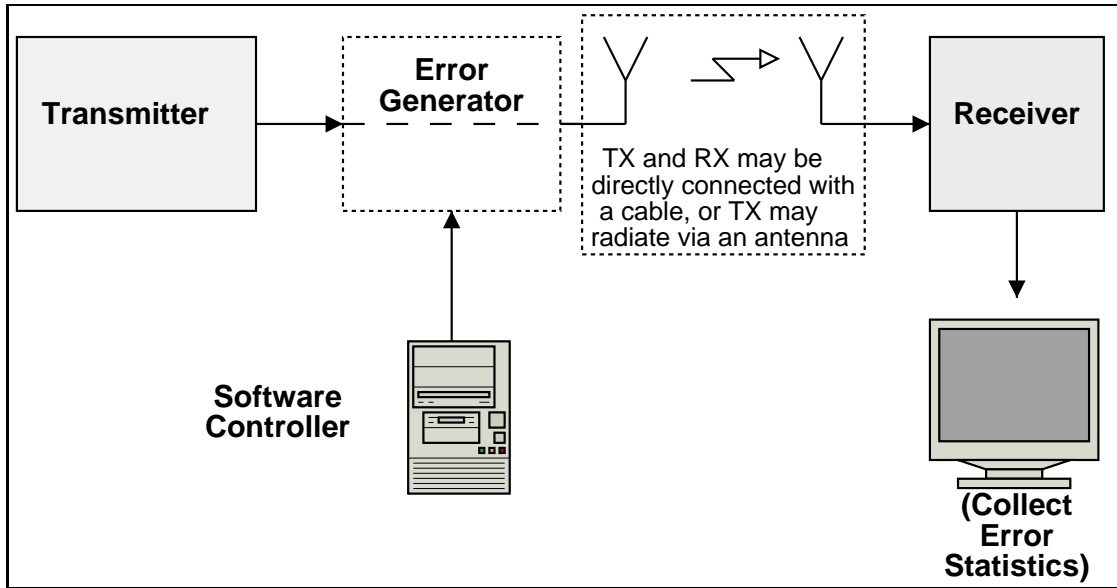


Figure 3.1: General design of a Fading Simulator

### 3.3 Design Constraints

There are a number of constraints we can place on this problem, in order to limit the possible solution-space:

1. Fading should be simulated at the transmitter
2. Minimum modification to transmitter hardware
3. Software controllable
4. Real-time, “online” simulation
5. Should be transparent to equipment under test

We will consider each constraint in turn, both in order to understand its origin and to see what limitations it poses on the design process.

#### 3.3.1 Fading simulated at the transmitter

A hardware fading simulator is often used for testing new types of communication hardware (including cell phones, pagers, wireless network cards, and other mobile wireless devices). Testing is often performed by using custom transmitter hardware to send data to the device under test (acting as the receiver). Thus, we would like to

have the fading simulator at the transmitter side while making minimal modifications to the receiver.

### **3.3.2 Minimum modification to transmitter hardware**

Although the transmitter hardware may be modified equipment, we would like to require as little change as possible to it. The optimal design for our fading simulator would be a black box which could be connected between the output of the transmitter and the receiving device, and which would not require any modification to the transmitter.

### **3.3.3 Software controllable**

The main problem with current hardware fading simulators is a lack of flexibility in the models used to simulate fading. Most commercial fading simulators are pure hardware, at most allowing for a varying average SNR. We would like to incorporate a software control element to the fading simulator, in order to allow complete flexibility in the fading model. Software control allows the user to modify the fading model should their testing requirements change.

### **3.3.4 Real-time simulation**

In order to provide a realistic testing environment, the device under test should be operated in the same way as it would be in real life. For this reason, the fading simulator should provide “on-line” simulation of fading.

### **3.3.5 Transparent to equipment under test**

Another important aspect of a realistic testing environment is to have the fading simulator transparent to the equipment under test — the fading simulator should simulate the real-life fading as accurately as possible, so accurate behaviours of the device under test can be measured.

## 3.4 Possible Solutions

Hardware fading simulators are not a new idea. In this section we will enumerate several possible designs of the fading simulator, and discuss the benefits and drawbacks of each technique. We must always keep in mind the design constraints as given in Section 3.3, in order to weigh each possible solution.

### 3.4.1 Use samples from measurements of fading

Early fading simulators were implemented by simply using recordings of actual fading taken from experimental measurements. The main limitation of this technique is that implementing a new fading model requires a series of experimental measurements in order to get the necessary fading samples. Some fading conditions may be hard to replicate exactly in the field, but can easily be characterized statistically. For this reason, we would prefer a method that relies on the statistical characterization of multipath fading.

### 3.4.2 In-phase and quadrature AWGN channels

It is well-known that if independent and identically distributed (i.i.d.) zero-mean Gaussian random variables are combined in an in-phase and quadrature manner, the resulting signal will have a Rayleigh-distributed envelope. This result can be used to generate samples of Rayleigh fading. The disadvantage of this technique is that other fading models cannot be easily obtained, limiting the flexibility of such an approach.

### 3.4.3 Sum-of-sinusoids circuit

The well-known “Jakes” sum-of-sinusoids fading simulator [7] consists of the sum of several sinusoids with random phases and magnitudes. Such a circuit can be implemented using phase shifters and amplifiers, and is known to produce a Rayleigh-distributed signal envelope. The drawback of this type of device is that a large number of high-frequency amplifiers and phase-shift circuits are required, making implementation costly.

### 3.4.4 Mismatch circuits at the transmit antenna

Another possibility for simulating fading at the transmitter is to divert some power destined for the antenna. This can be done by having a number of mismatched transmission line stubs in parallel with the antenna connection. By controlling which stubs are connected at any time, the amount of power transmitted (and thus the received SNR) can be controlled.

The main problem with such a technique is the requirement for RF microelectronics, specifically transmission line stubs and high-frequency transistor switches. Such a design is also dependent on the operating frequency of the wireless hardware under test.

### 3.4.5 Voltage-controllable RF amplifier

By using a controllable RF amplifier, the signal power transmitted (and hence received) can be controlled. If the control signals can be provided by a computer, then software-controlled fading can be implemented. The drawback to this type of circuit is that the amplifier used must be very wideband (to operate at RF frequencies). Alternatively, the RF signal can be converted to digital form (by sampling) and the amplification can be done digitally. Either of these approaches requires costly hardware.

### 3.4.6 Interface at the bitstream of the transmitter

Instead of adapting the transmitter power (which in turn will cause changes in the bit error rate), we can introduce errors directly into the transmitter bitstream. If the error rate is adapted over time, the receiving device will see a changing bit error rate (which is the end result of a fading channel). Hence we can say that modifying the bit error rate is equivalent to changing the signal power.

The problem with this solution is that it violates the constraint of Section 3.3.2: “minimum modification to transmitter hardware”. However, the reduced cost and ease of implementation of this solution (specifically, not needing additional RF electronics) makes it an attractive choice for implementing a software-controlled hardware-based fading simulator.

## 3.5 Chosen Design

The design chosen for the software-controlled hardware fading simulator is the one given in Section 3.4.6: the interface at the bitstream of the transmitter. The design of both the hardware and software for this solution is explained in detail in Chapter 4. It is worth noting here that the software and hardware portions of the device are relatively independent. The software controller has no knowledge of what hardware is present: it only knows how to manipulate the control lines in order to adjust the fading (and thus the bit error rate). For this reason, it would be relatively simple to implement one of the other hardware solutions from Section 3.4 while keeping the same software component.

# Chapter 4

## Design of the Fading Channel Simulator

### 4.1 Introduction

The chosen solution implements the software-controlled fading simulator by generating errors at the bitstream level in the transmitter circuit. This type of implementation requires a hardware error generator and a software control program.

A block diagram of the solution is shown in Figure 4.1. The three computer units are each software programs running on IBM-compatible computers with the Linux operating system. The transmitter and receiver blocks are both hardware devices.

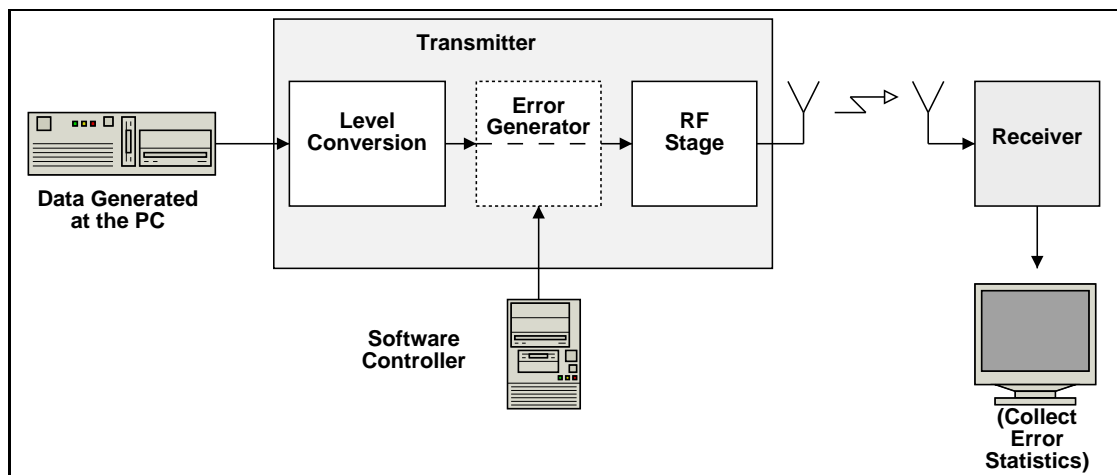


Figure 4.1: Block Diagram of the Software-controlled Fading Simulator

## 4.2 Hardware

In this section we will discuss the hardware component of the fading simulator. Since this simulator has been implemented as a proof-of-concept with a transmitter/receiver pair, we will also discuss the implementation of the transmitter and the receiver.

The data sheets for all the major components used are listed in Appendix D, on pages 38 to 44.

### 4.2.1 Transmitter

The role of the transmitter portion of the hardware is to take data from the serial port of a computer, modulate it as an amplitude-shift keyed (ASK) signal in the 434 MHz band (see Section 4.2.4 for details about the restrictions on the use of this frequency band), and radiate the modulated signal via an antenna.

A block diagram of the transmitter circuit is given in Figure 4.2. The full circuit diagrams can be found in Appendix C.1, on page 35. A list of components required to construct the transmitter is given in Table 4.1.

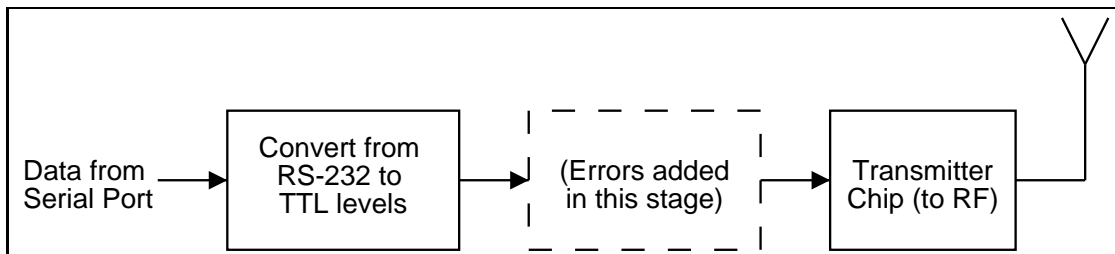


Figure 4.2: Block Diagram of the Transmitter

Component	Quantity
Laipac TLP-434 ASK Transmitter (434 MHz)	1
Maxim MAX-232 Level Converter	1
0.1 $\mu$ F Capacitor	5
Serial Port Connector and Shroud	1
Three-Conductor Shielded Cable	6 ft

Table 4.1: Components for the Transmitter

## 4.2.2 Receiver

The receiver unit must detect and demodulate the ASK signal sent from the transmitter, and convert the data into RS-232 levels suitable for input to the serial port of a computer.

A block diagram of the receiver circuit is given in Figure 4.3. The full circuit diagrams can be found in Appendix C.2, on page 36. A list of components required to construct the receiver is given in Table 4.2.

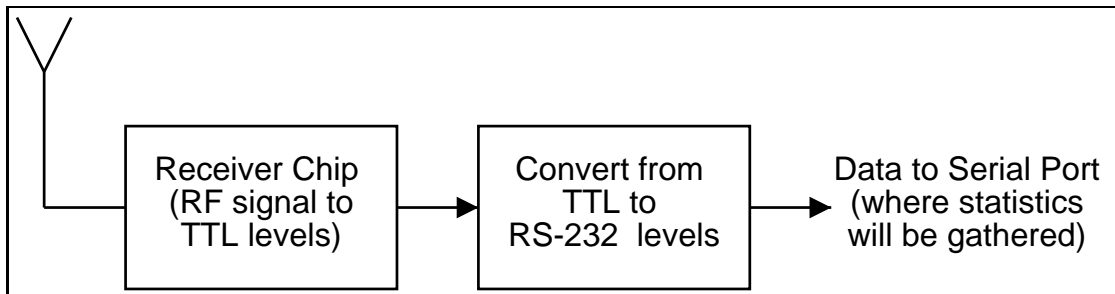


Figure 4.3: Block Diagram of the Receiver

Component	Quantity
Laipac RLP-434 ASK Receiver (434 MHz)	1
Maxim MAX-232 Level Converter	1
0.1 $\mu$ F Capacitor	5
Serial Port Connector and Shroud	1
Three-Conductor Shielded Cable	6 ft

Table 4.2: Components for the Receiver

## 4.2.3 Error Generator

The error generator chooses either the correct or erroneous bit to pass to the RF unit of the transmitter, based on a control signal from the parallel port of a computer. The error generator does this by using a multiplexer fed by both the correct and incorrect digital waveforms, and connecting the control pin of the multiplexer to one of the parallel port output pins.

A block diagram of the error generator circuit is given in Figure 4.4. The full circuit diagrams can be found in Appendix C.3, on page 37. A list of components required to construct the error generator circuit is given in Table 4.3.



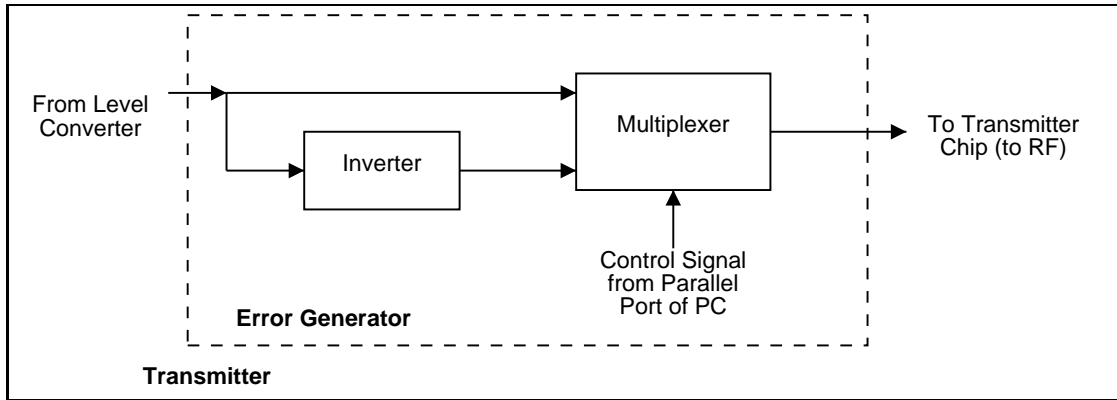


Figure 4.4: Block Diagram of the Error Generator

Component	Quantity
74LS04 Hex Inverter	1
MC14067B Multiplexer	1
Parallel Port Connector and Shroud	1
Three-Conductor Shielded Cable	12 ft

Table 4.3: Components for the Error Generator

#### 4.2.4 Licensing in the 434 MHz Band

The frequency band 433.05—434.79 MHz is licensed in Canada as an Industrial, Scientific, and Medical (ISM) band.<sup>1</sup> The licensing allows users of low-power devices (such as the TLP-434/RLP-434 transmitter/receiver pair used in this project) free access to the frequency band. The International Telecommunication Union (ITU) sets specifications for radiation limits and spectral masks in the ISM frequency bands.<sup>2</sup>

### 4.3 Software

The software portion of the solution consists of three sections: the calculation of the fading model parameters (written in Matlab), control software for the hardware portion of the device (written in C), and the transmitter and receiver software (written in C). Details of these components are given below.

<sup>1</sup>see <http://strategis.ic.gc.ca/SSI/sf/cane.pdf>, section S5.138

<sup>2</sup>see <http://www.itu.int/> for details

### 4.3.1 Numerical Calculation of Fading Model Parameters

The purpose of the numerical calculation routines (given in Appendix B) is to generate a finite-state, discrete-time approximation to the statistical model of the fading signal-to-noise ratio. To allow for the most general case, the channel has been modeled by the Nakagami- $m$  distribution, given in Equation 2.3. The software constructs a Markov model approximation of the Nakagami- $m$  distribution, following the method proposed by Wang and Moayeri [8].

The creation of the Markov model involves several steps:

1. Divide the envelope PDF into  $n$  distinct regions
2. Calculate the *average* BER for each region
3. Calculate the *transition probability* from every region to every other region
4. Use the calculated values to build the Markov model

Choosing the number of levels  $n$  is fairly subjective, depending on the desired accuracy of approximation to the exact PDF and the amount of memory available. The size of  $n$  is not an issue when a PC is used to control the fading simulator. However, if the controller is implemented on an embedded microprocessor, the memory requirements for storing the parameters of a model for large values of  $n$  may be a consideration. The algorithms using the Markov model run in  $O(n)$  time, so computational complexity is not an issue in choosing  $n$ .

Determining the optimal threshold for each of the  $n$  levels (to minimize the error introduced when generating the discrete approximation) is a complex optimization problem. We use the sub-optimal method of manually attempting to equate the *residency probabilities*

$$P_{\text{residency},i} = \int_{\gamma_i}^{\gamma_{i+1}} p_{\text{Nakagami}}(\gamma, m, \bar{\gamma}) d\gamma \quad (4.1)$$

where  $p_{\text{Nakagami}}(\gamma, m, \bar{\gamma})$  is the Nakagami- $m$  PDF, given in Equation 2.3.

The average BER for each level  $i$  is given by

$$h_i = \frac{\int_{\gamma_i}^{\gamma_{i+1}} P_{\text{err}}(\gamma) P_{\text{SNR}}(\gamma, m, \bar{\gamma}) d\gamma}{\int_{\gamma_i}^{\gamma_{i+1}} P_{\text{SNR}}(\gamma, m, \bar{\gamma}) d\gamma} \quad (4.2)$$

where  $P_{SNR}(\gamma, m, \bar{\gamma})$  is the PDF of the SNR, given by

$$P_{SNR}(\gamma, m, \bar{\gamma}) = \frac{m^m \gamma^{m-1}}{\bar{\gamma}^m \Gamma(m)} \exp\left(-\frac{m\gamma}{\bar{\gamma}}\right) \quad (4.3)$$

(with  $m$  as the parameter of the Nakagami distribution,  $\gamma$  as the SNR,  $\bar{\gamma}$  as the mean SNR, and  $\Gamma(\cdot)$  as the Gamma function [6]) and  $P_{err}(\gamma)$  is the probability of error for the modulation being used — in this case, ASK modulation given by

$$P_{err}(\gamma) = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \quad (4.4)$$

with  $E_b/N_0$  being the SNR and  $Q(\cdot)$  being the Gaussian-Q function [5]. The transition probability between any two levels  $i$  and  $j$  is given by

$$\tau_{i,j} = \int_{\gamma_i}^{\gamma_{i+1}} \int_{\gamma_j}^{\gamma_{j+1}} P_{SNR,joint}(\alpha, \bar{\gamma}, \beta, \bar{\gamma}|m, \rho) d\alpha d\beta \quad (4.5)$$

where  $\rho$  is the correlation, and  $P_{SNR,joint}(\cdot)$  is the joint PDF of two Nakagami- $m$  random variables [1]. The correlation  $\rho$  is dependent on the velocity of the mobile device, and is generally assumed to be in the form given in Equation 2.4 [7]. The integral in Equation 4.5 can be calculated numerically to arrive at values for  $\tau_{i,j}$ .

### 4.3.2 Hardware Interface Software

The hardware control program (see `Controller.c`, Section A.1 on page 24) uses the Markov model created by the method described in Section 4.3.1 to control the errors that are artificially generated by the hardware of Section 4.2. The program runs in a loop that is timed to be the duration of several bits (so-called *slow* fading). Each time through the loop, the program decides what state it should be in (and hence what the error rate is) by transitioning from the previous state in the Markov chain. Because the Markov chain has completely described probabilities  $\tau_{i,j}$  for transition between any two states, determining the next state is a simple matter of generating a uniformly distributed random number between 0 and 1, and choosing the state based on the cumulative sum of probabilities.

Once the current state has been chosen, a decision for “error/no error” is made based on the effective BER for that state. If the error state is chosen, the software sends a control signal (via the parallel port) to the hardware error generator, which will select the erroneous data to be transmitted.

### 4.3.3 Transmitter and Receiver

The transmitter program (see `Transmitter.c`, Section A.2 on page 26) sends a fixed set of data to the radio transmitter unit via the serial port interface. The program must first set the parameters of the serial port (specifically, the baud rate and parity setting), and then transmits a stream of known data.

The receiver program (see `Receiver.c`, Section A.3 on page 27) receives data from the radio receiver (via the serial port). It compares this data with the expected results and calculates the resultant bit error rate of the transmission.

# Chapter 5

## Results and Discussion

### 5.1 Introduction

The fading simulator discussed in Chapter 4 has been built and tested. Some representative experimental results are shown below to illustrate the performance of the simulator, and in Section 5.3 a discussion of the benefits and drawbacks of this solution will be presented.

### 5.2 Results

The software simulator can be configured for any values of the Nakagami parameter  $m$ , the mean SNR  $\bar{\gamma}$ , and the Doppler frequency  $f_D$  (which is related to the mobile device's velocity and the carrier frequency according to Equation 2.5). A representative BER vs. SNR plot for  $m = 2$ ,  $\bar{\gamma} = 10$  dB,  $f_D = 10$  Hz, and  $n = 5$  states in the Markov chain is given in Figure 5.1.

Figure 5.1 shows the theoretical performance of ASK modulation in Nakagami- $m$  fading (the solid line), as well as the measured experimental points (the dots). It is clear from this figure that the experimental results are slightly worse than the theory predicts. The reason for the extra errors in comparison to the theoretical data is that errors sometimes occur even when the error generator selects to have no error. These extra errors are due to outside factors such as noise. However, the experimental results are within a reasonable range of the theoretical results.

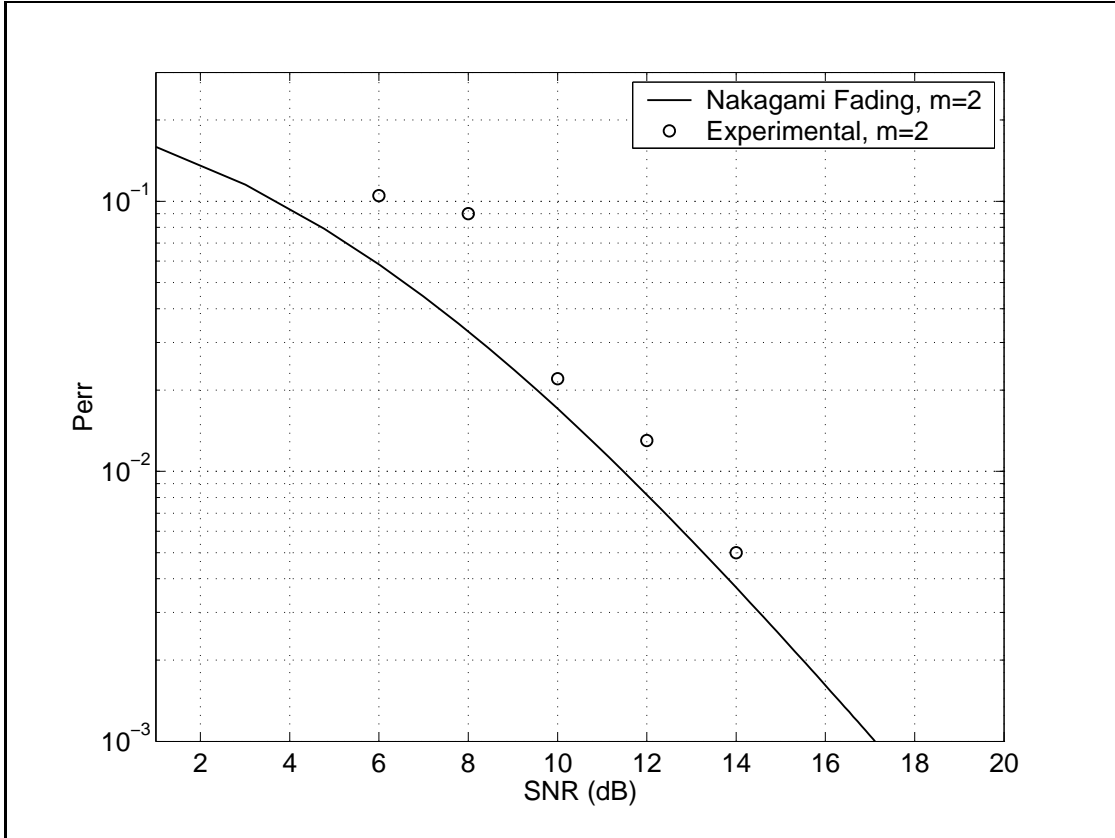


Figure 5.1: Theoretical and Simulated BER for  $m = 2$ ,  $\bar{\gamma} = 10\text{dB}$ ,  $f_D = 10\text{ Hz}$

## 5.3 Discussion

As with any solution, the particular hardware and software described in Chapter 4 has some advantages and some drawbacks. We will attempt to cover the most pertinent of both categories here.

### 5.3.1 Benefits of this solution

The most obvious benefit of the solution presented in this report, in comparison to current commercial fading channel simulators, is the low cost of the device. The cost of all components, including the transmitter/receiver pair, was less than \$75. Obviously additional costs would be incurred in creating a device that was suitable for use in a laboratory (such as RF shielding and creation of a printed circuit board). However, the total cost will still be considerably less than that of commercial products.

Another important benefit of this solution is its inherent flexibility. The software and hardware have both been designed in such a way that they are independent of

each other. Should the user decide to use a different model of the mobile fading channel, the software portion of the device can easily be modified to accommodate a new model. Similarly, should the user decide to implement one of the other hardware solutions presented in Section 3.4, the same control and modeling software can be used.

### **5.3.2 Drawbacks to this solution**

The main drawback of the design of the fading channel simulator is the requirement for modification of the transmitting hardware. Depending on the device that is to be tested, modifying the transmitter may be difficult (especially in a situation where the transmitter is built by an outside party). In the case where modification of the transmitter proves difficult, it is possible to use the same control and modeling software while adapting the hardware portion of the device to use one of the alternative solutions presented in Section 3.4.

# Chapter 6

## Conclusions

### 6.1 Future Work

The flexibility in the design of this project allows for several possible areas of future work. The software can easily be modified in order to study the effect of different channel models. These changes can be as simple as changing the PDF used for modeling the signal envelope, to something more complex such as incorporating real-time generation of the model parameters. The latter would allow for the easy implementation of dynamic fading conditions (where the fading severity changes during the simulation), which is possible but difficult to implement using the current software design.

There are also many extensions that can be done on the hardware side of the project. If facilities are available for high-frequency circuit design, one major project that can be undertaken is to implement one of the other possible hardware solutions of Section 3.4. This hardware could be coupled with the existing software to create a true “black box” software-controlled, hardware-based fading simulator.

### 6.2 Conclusions

The software-controlled, hardware-based mobile fading simulator presented in this report is an excellent low-cost alternative to currently available fading channel simulators. In addition, the inherent flexibility of the design allows the user to adapt the device to a variety of channel models, and thus to study the performance of wireless devices under a wide range of fading channel conditions.



# Appendix A

## Code Listings — Hardware Interface

Code in this section is used for interface with the hardware layer of the project. It is written in the C programming language.

### A.1 Error Controller — Controller.c

Listing A.1: Controller.c

```
1 // For details on parallel port programming in Linux, see
2 // http://www.tldp.org/HOWTO/mini/IO-Port-Programming-2.html#ss2.1
3
4 #include <stdio.h>
5 #define extern static
6 #include <asm/io.h>
7 #include <stdlib.h>
8 #include <time.h>
9
10 #define DELAY 500000 /* delay in microseconds */
11 #define PPORT 0x378 /* address of the parallel port */
12
13 #define BIT0 0x01 // actual data line D0 (pin 2)
14
15 // the bit rate of the transmitter (bps)
16 #define BITRATE 2400
17
18 // the number of bits over which to keep fade duration constant
19 #define NUMBITS 50
20
21 // number of microseconds in one second
22 #define MICROSEC 1000000
23
24 #define TRUE 1
25 #define FALSE 0
26
27 // Note: The 'cumsum' and 'h' table definitions were
```

```

28 // removed to save space.
29
30 int
31 genNextState(int currState)
32 {
33     double rnd;
34     int next = 0;
35
36     // generate a random number between [0,1]
37     rnd = (double)rand() / RAND_MAX;
38
39     // search in the cummulative transition table to
40     // find the next state
41     while(rnd > cumsum[next][currState]) {
42         next++;
43     }
44     printf("Rnd:_%f\tNext_State:_%d\n",rnd,next);
45
46     return next;
47 }
48
49 int
50 determineError(int currState)
51 {
52     double rnd;
53
54     // generate a random number between [0,1]
55     rnd = (double)rand() / RAND_MAX;
56
57     if (rnd <= h[currState]) // an error
58         return TRUE;
59     else
60         return FALSE;
61 }
62
63 int
64 main(int argc, char* argv[])
65 {
66     int i;
67     int pin;
68     int data;
69     int old;
70     int nbits = 0;
71     int nerr = 0;
72     int currState = 2; // start in the middle
73     int error;
74
75     data = BIT0;
76
77     // seed the random number generator
78     srand((unsigned int)time(NULL));
79
80     // get permissions on the ports
81     ioperm(PPORT, NPORTS, 1);

```

```

82
83     while (1) {
84
85         // determine the state transition
86         currState = genNextState(currState);
87
88         // determine if we have an error or not
89         error = determineError(currState);
90
91         // cause an error
92         if (error) {
93             printf("Error\n");
94             nerr++;
95             outb(data, PPORT); // turn on the pin
96         }
97         else {
98             printf("No Error\n");
99         }
100        nbits++;
101        printf("Sample Perr: %f\n", (float)nerr/nbits);
102        printf("Delay for %d microseconds\n",
103              MICROSEC*NUMBITS/BITRATE);
104        usleep(MICROSEC*NUMBITS/BITRATE);
105
106        // turn off the pin
107        outb(0, PPORT);
108    }
109
110    // we are done
111    return 0;
112 }

```

## A.2 Transmitter — Transmitter.c

Listing A.2: Transmitter.c

```

1 // Based on code from:
2 // http://en.tldp.org/HOWTO/Serial-Programming-HOWTO/x115.html
3 #include <sys/types.h>
4 #include <sys/stat.h>
5 #include <fcntl.h>
6 #include <termios.h>
7 #include <stdio.h>
8
9 #include <stdlib.h>
10
11 /* baudrate settings are defined in <asm/termbits.h>, which is
12    included by <termios.h> */
13 #define BAUDRATE B2400
14 #define MODEMDEVICE "/dev/ttyS0"
15 #define _POSIX_SOURCE 1 /* POSIX compliant source */
16 #define FALSE 0

```

```

17 #define TRUE 1
18
19 #define DATA "q"
20 #define REPEAT 10000
21 #define MAXDELAY 100 /* microseconds */
22 volatile int STOP=FALSE;
23
24
25 void
26 main()
27 {
28     int fd,c, res;
29     int delay;
30     int i;
31     struct termios oldtio,newtio;
32     char buf[255];
33     int good = 0, bad = 0;
34
35     fd = open(MODEMDEVICE, O_RDWR | O_NOCTTY );
36     if (fd <0) {perror(MODEMDEVICE); exit(-1); }
37
38     tcgetattr(fd,&oldtio); /* save current port settings */
39
40     bzero(&newtio, sizeof(newtio));
41     newtio.c_cflag = BAUDRATE | CS7 | CSTOPB | CLOCAL | CREAD;
42     newtio.c_iflag = IGNPAR;
43     newtio.c_oflag = 0;
44
45     /* set input mode (non-canonical, no echo,...) */
46     newtio.c_lflag = 0;
47
48     newtio.c_cc[VTIME]      = 0;
49     newtio.c_cc[VMIN]      = 1;
50
51     tcflush(fd, TCIFLUSH);
52     tcsetattr(fd,TCSANOW,&newtio);
53
54     // write data
55     while(1) {
56         res = write(fd,DATA,1);
57         delay = 1000000/1200;
58         printf("%d\n",delay);
59         usleep(delay);
60     }
61     tcsetattr(fd,TCSANOW,&oldtio);
62 }

```

## A.3 Receiver — Receiver.c

Listing A.3: Receiver.c

```

1 // Based on Code from

```

```

2 // http://en.tldp.org/HOWTO/Serial-Programming-HOWTO/x115.html
3 #include <sys/types.h>
4 #include <sys/stat.h>
5 #include <fcntl.h>
6 #include <termios.h>
7 #include <stdio.h>
8 #include <math.h>
9
10 /* baudrate settings are defined in <asm/termbits.h>, which is
11    included by <termios.h> */
12 #define BAUDRATE B2400
13 #define MODEMDEVICE "/dev/ttyS0"
14 #define _POSIX_SOURCE 1 /* POSIX compliant source */
15 #define FALSE 0
16 #define TRUE 1
17
18 #define DATA 'q'
19 volatile int STOP=FALSE;
20
21 main()
22 {
23     int fd,c, res;
24     struct termios oldtio,newtio;
25     unsigned char buf[255];
26     unsigned char diff;
27     int i;
28     int good = 0, bad = 0;
29
30     fd = open(MODEMDEVICE, O_RDWR | O_NOCTTY );
31     if (fd <0) {perror(MODEMDEVICE); exit(-1); }
32
33     tcgetattr(fd,&oldtio); /* save current port settings */
34
35     bzero(&newtio, sizeof(newtio));
36     newtio.c_cflag = BAUDRATE | CS7 | CSTOPB | CLOCAL | CREAD;
37     newtio.c_iflag = IGNPAR;
38     newtio.c_oflag = 0;
39
40     /* set input mode (non-canonical, no echo,...) */
41     newtio.c_lflag = 0;
42
43     newtio.c_cc[VTIME]      = 0;
44     newtio.c_cc[VMIN]      = 1;
45
46     tcflush(fd, TCIFLUSH);
47     tcsetattr(fd,TCSANOW,&newtio);
48
49
50     while (STOP==FALSE) {          /* loop for input */
51         res = read(fd,buf,255);
52         if (buf[0]==DATA) {
53             good++;
54             printf(".");
55             fflush(stdout);

```

```

56     }
57     else {
58         // check each bit of the received data
59         // for an error
60         diff = (unsigned char)(buf[0] ^ DATA);
61         for(i=0; i<=7; i++){
62             if (((diff) &
63                 (unsigned char)(pow(2,i))) != 0)
64                 bad++;
65         }
66         printf("\nBad char: %c\tNumeric: 0x%hx\n",
67             buf[0], buf[0]);
68         printf("Good: %d\tBad: %d\tPerr: %f\n",
69             good, bad, (float)bad/(good+bad));
70     }
71 }
72 tcsetattr(fd, TCSANOW, &oldtio);
73 }

```

# Appendix B

## Code Listings — Parameter Calculation

Code in this section is used for calculating the parameters of the Markov model approximation to the Nakagami- $m$  fading channel. It is written as Matlab scripts and functions.

### B.1 Calculate the Parameters of the Markov Model

Listing B.1: CalcMarkovParams.m

```
1 % Created:      CS 2003-02-15
2
3 clear all; close all;
4
5 % calculate the correlation coefficient
6 dopplerFreq = 10;
7 bitDuration = 1/2400;
8 rho = besselj(0,2*pi*dopplerFreq*bitDuration)
9
10 % the m value for the Nakagami-m model
11 m = 2
12
13 % the mean SNR value ->
14 meanSNR = 10 ^ (10 ./ 10)
15
16 % number of regions
17 N = 5;
18
19 % minimum SNR value
20 SNRs(1) = 0.1;
21
22 % actual transition values
23 SNRs(2) = 8;
24 SNRs(3) = 18;
25 SNRs(4) = 35;
26 SNRs(5) = 50;
```

```

27
28 % maximum SNR value
29 % if you make this too much higher than the highest SNR value,
30 % the numerical integrations later which complain... but if it
31 % is too low, you won't integrate the entire non-zero range for
32 % the Nakagami PDF
33 SNRs(N+1) = 5*meanSNR;
34
35 % store the 'h' values
36 h = zeros(1,N);
37
38 % store the probability of residence values
39 Pres = zeros(1,N);
40
41 % now integrate over the region, to find the 'h' values
42 for i=1:N,
43     h(i) = quad8('askNakagami',SNRs(i),SNRs(i+1),[],[],m,meanSNR);
44 end
45
46 % display the SNR and h values
47 SNRs
48 h
49
50 % now we have to find transition probabilities. To do this,
51 % we'll have to do double integration of the joint PDF,
52 % with one variable ranging over the 'from' region and the
53 % other ranging over the 'to' region
54
55
56 % create an inline function we can pass to 'dblquad'
57 func = inline(sprintf('JointNakagamiSNR(x,%f,y,%f,%f,%f)',
58     meanSNR,meanSNR,m,rho),'x','y');
59 % another inline function to check results
60 func2 = inline(sprintf('NakagamiSNR(x,%f,%f)',m,meanSNR),'x');
61
62
63 transitions = zeros(N,N);
64 sums = zeros(1,N);
65
66 disp('Calculating transition probabilities...')
67 for i=1:N,
68     disp(sprintf('Starting row %d...',i))
69     for j=1:N,
70         transitions(i,j) = dblquad(func, SNRs(i), SNRs(i+1),
71             SNRs(j), SNRs(j+1));
72     end
73     Pres(i) = quadl(func2, SNRs(i), SNRs(i+1));
74 end
75
76 transitions
77
78 for i=1:N,
79     transitions(:,i) = transitions(:,i) ./ sum(transitions(:,i));
80 end

```



```

81 transitions
82
83 Pres
84
85 avgPerr = quad8('askNakagami',SNRs(1),SNRs(N+1),[],[],m,meanSNR);
86
87 cs = cumsum(transitions);
88
89 for i=1:N,
90     disp(sprintf('%f, %f, %f, %f, %f', cs(i,1), cs(i,2),
91                                     cs(i,3), cs(i,4), cs(i,5)));
92 end
93
94 disp(sprintf('\nh[5] = %f, %f, %f, %f, %f;', h(1), h(2),
95                                     h(3), h(4), h(5)));

```

## B.2 $P_{err}$ for ASK Modulation in Nakagami- $m$ Fading

Listing B.2: askNakagami.m

```

1 % evaluate the product askBER(...) .* NakagamiSNR(...)
2 function y = askNakagami(SNRvals, m, meanSNR)
3
4 y = askBER(SNRvals) .* NakagamiSNR(SNRvals, m, meanSNR);
5
6 return

```

## B.3 $P_{err}$ for ASK Modulation in AWGN

Listing B.3: askBER.m

```

1 % function BER = askBER(SNRvals)
2 %
3 % calculate the BER of ASK modulation over an AWGN channel
4 % with ideal coherent detection, for the given SNR per bit
5 % value(s)
6 %
7 % Parameters:
8 %
9 % SNRvals      the value(s) of SNR per bit at which to calculate
10 %              the value of the BER
11 %
12 % Created:     CS 2003-03-03
13 %
14 % Modified:
15 %
16 % Reference:   "Digital Communications" John G. Proakis, 1995,
17 %              pg. 271, Eq. 5-2-57

```

```

18 %
19 % Adapted from bpskBER.m. CS.
20
21 function BER = askBER(SNRvals)
22
23 BER = gaussianQ( sqrt(SNRvals) );
24
25 return

```

## B.4 Nakagami- $m$ PDF of the SNR

Listing B.4: NakagamiSNR.m

```

1 % function P = NakagamiSNR(SNR, m, meanSNR)
2 %
3 % calculate the Nakagami-m PDF of the SNR for the
4 % given value(s) of SNR
5 %
6 % Parameters:
7 %
8 % SNR          a vector of SNR values
9 % m           the Nakagami m parameter ( 0.5 <= m <= inf)
10 % avgSNR      the mean SNR for the signal
11 %
12 % Created:    CS 2003-01-12
13 %
14 % Reference:
15 %           "Digital Communication over Fading Channels: A Unified
16 %           Approach to Performance Analysis", Marvin K. Simon &
17 %           Mohamed-Slim Alouini, 2000, pg. 22, Eq. 2.21
18 %
19 % Verification:
20 %           the file TestNakagamiSNR.m will calculate and plot
21 %           the function for 2 methods - direct calculation, and
22 %           substitution into the Nakagami PDF (the Nakagami PDF
23 %           has been verified against the figure in Simon & Alouini)
24 %
25 %           The two figures come out the same.
26 %
27 %           Furthermore, TestNakagamiSNR will integrate the PDF
28 %           along most of its length, and display the total area
29 %           (which should of course be equal to 1)
30
31
32 function P = NakagamiSNR(SNR, m, meanSNR)
33
34 first = (m^m)*(SNR.^(m-1)) / ( (meanSNR ^ m) * gamma(m) );
35
36 P = first .* exp( -m.*SNR/meanSNR );
37
38 return

```

## B.5 The Gaussian Q function

Listing B.5: gaussianQ.m

```
1  % function Q = gaussianQ(x)
2  %
3  % Gaussian-Q function
4  %
5  % calculate the gaussian-Q function for the
6  % given value(s) of x
7  %
8  %  $Q(x) = 1/2 * \text{erfc}(x/\text{sqrt}(2))$ 
9  %
10 % Parameters:
11 %
12 % x           the values at which to calculate
13 %             the function
14 %
15 % Created:    CS 2002-06-07
16 %
17 % Reference:  "Digital Communications"
18 %             John G. Proakis, 1995,
19 %             pg. 40, Eq. 2-1-98
20 %
21 % Verification:
22 %             check the values of  $Q(0)$ ,  $Q(\text{inf})$ 
23 %              $Q(0)$  should be 0.5
24 %              $Q(\text{inf})$  should be 0
25
26 function Q = gaussianQ(x)
27
28 Q = (1/2) * erfc( x ./ sqrt(2) );
29
30 return
```

# Appendix C

## Circuit Diagrams

### C.1 Transmitter

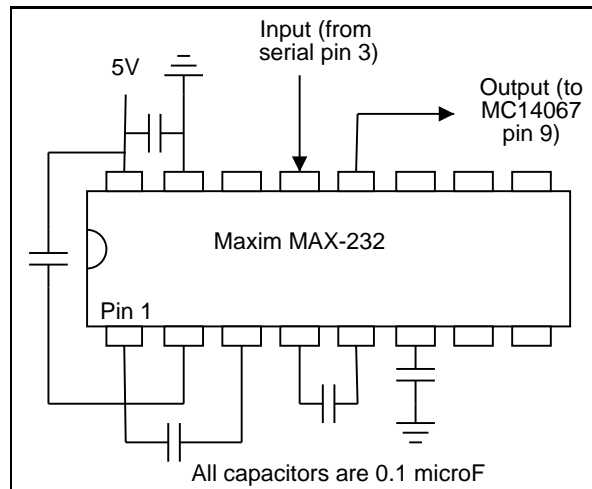


Figure C.1: Circuit Diagram of the Transmitter Level Converter

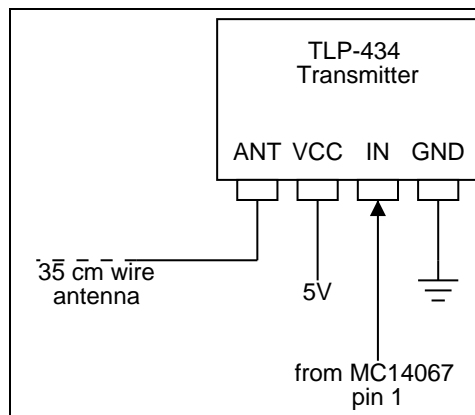


Figure C.2: Circuit Diagram of the TLP-434 ASK Transmitter



### C.3 Error Generator

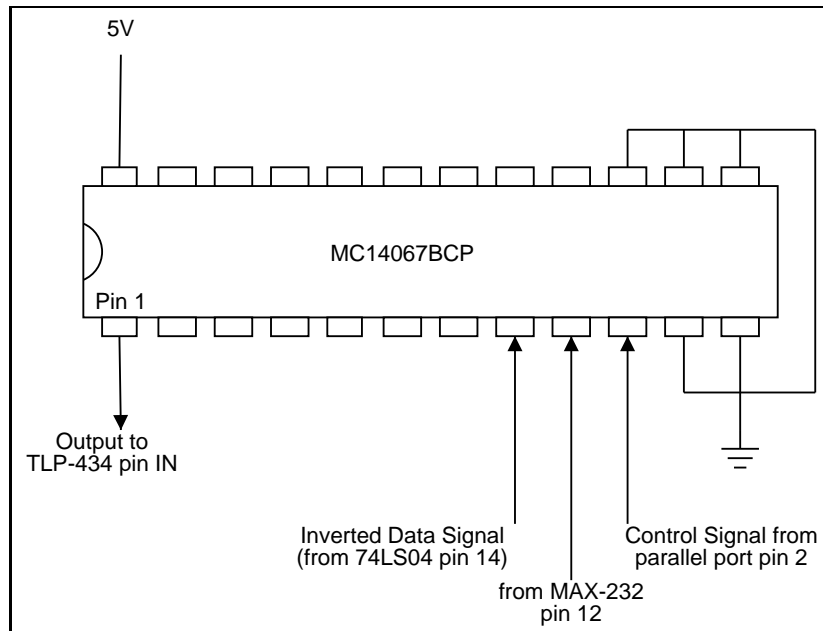


Figure C.5: Circuit Diagram of the Transmitter Error Generator Interface

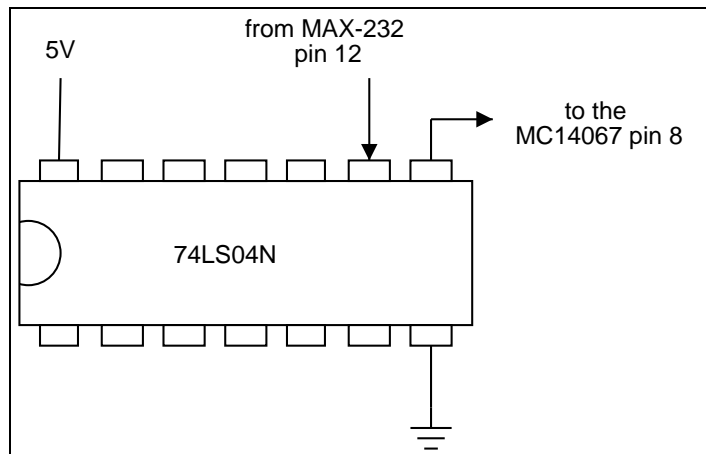


Figure C.6: Circuit Diagram of the Inverter

# Appendix D

## Selected Product Data Sheets

Selected pages from data sheets for the major components used are contained on pages 39 to 44. The websites for each product are listed along with the data sheets.

# D.1 TLP-434 Transmitter and RLP-434 Receiver

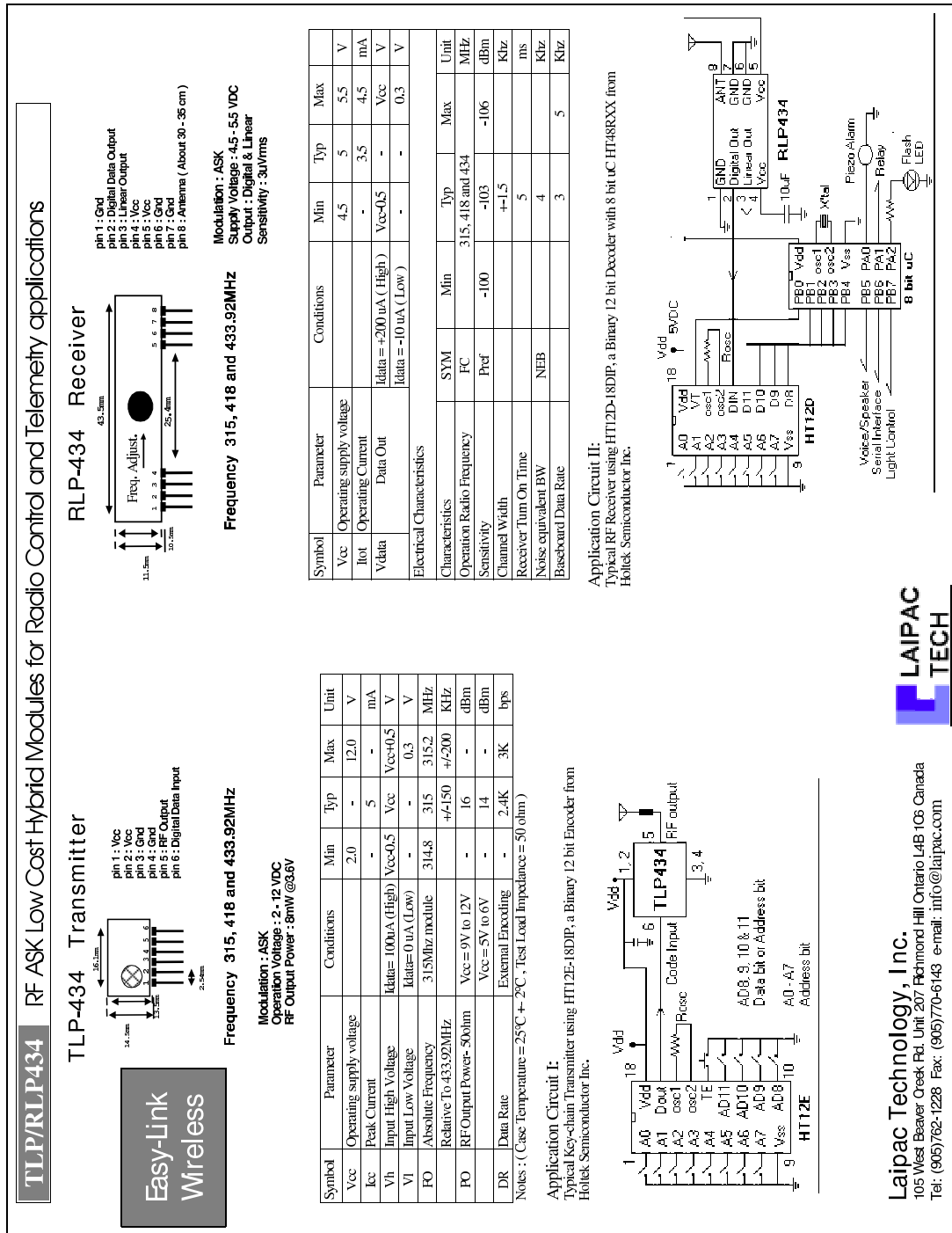


Figure D.1: Data Sheet for TLP-434 and RLP-434  
 Manufacturer: Laipac Technologies  
 URL: [http://www.laipac.com/easy\\_434\\_eng.htm](http://www.laipac.com/easy_434_eng.htm)

**LAIPAC  
TECH**

**Laipac Technology, Inc.**  
 105 West Beaver Creek Rd. Unit 207 Richmond Hill Ontario L4B 1G6 Canada  
 Tel: (905)762-1228 Fax: (905)770-6143 e-mail: info@laipac.com



## D.2 Maxim MAX232 RS-232 Level Converter

19-4323; Rev 10; 8/01

### +5V-Powered, Multichannel RS-232 Drivers/Receivers

#### General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where  $\pm 12V$  is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 $\mu$ W. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

#### Features

- ◆ Superior to Bipolar
- ◆ Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)
- ◆ Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- ◆ Meet All EIA/TIA-232E and V.28 Specifications
- ◆ Multiple Drivers and Receivers
- ◆ 3-State Driver and Receiver Outputs
- ◆ Open-Line Detection (MAX243)

#### Applications

- Portable Computers
- Low-Power Modems
- Interface Translation
- Battery-Powered RS-232 Systems
- Multidrop RS-232 Networks

#### Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

*Ordering information continued at end of data sheet.  
\*Contact factory for dice specifications.*

#### Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value ( $\mu$ F)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.1	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package

Maxim Integrated Products 1

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at [www.maxim-ic.com](http://www.maxim-ic.com).

Figure D.2: Data Sheet for MAX232 RS-232 Level Converter

Page 1 of 2

Manufacturer: Maxim Integrated Products

URL: [http://www.maxim-ic.com/quick\\_view2.cfm/qv\\_pk/1798/1n/en](http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1798/1n/en)

## +5V-Powered, Multichannel RS-232 Drivers/Receivers

**MAX220-MAX249**

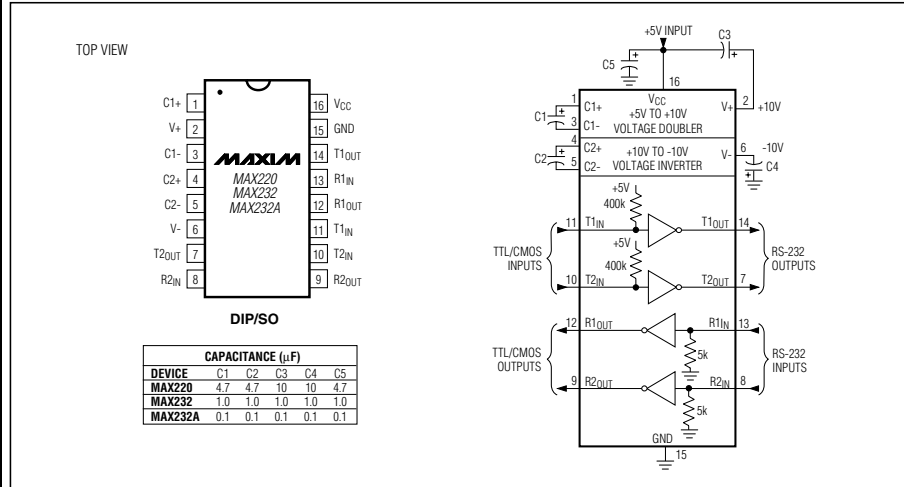


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

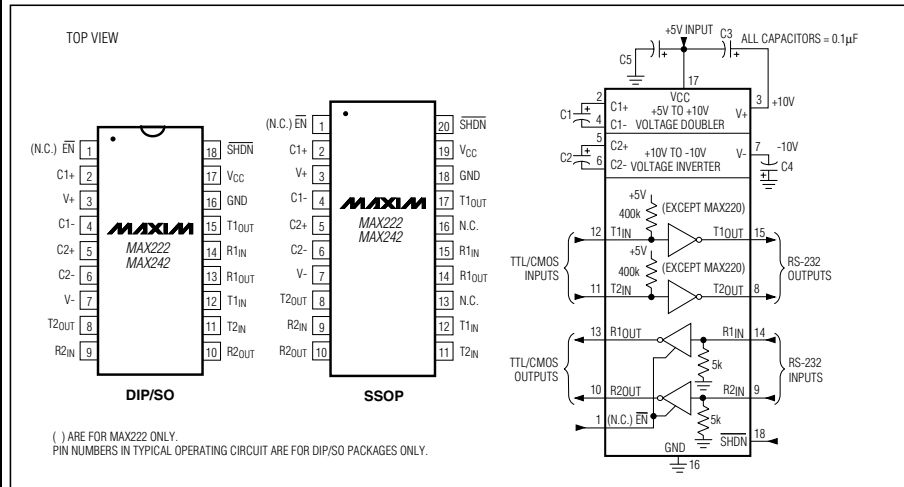


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

**MAXIM**

17

Figure D.3: Data Sheet for MAX232 RS-232 Level Converter

Page 2 of 2

Manufacturer: Maxim Integrated Products

URL: [http://www.maxim-ic.com/quick\\_view2.cfm/qv\\_pk/1798/ln/en](http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1798/ln/en)

## D.3 Fairchild 74LS04 Hex Inverter

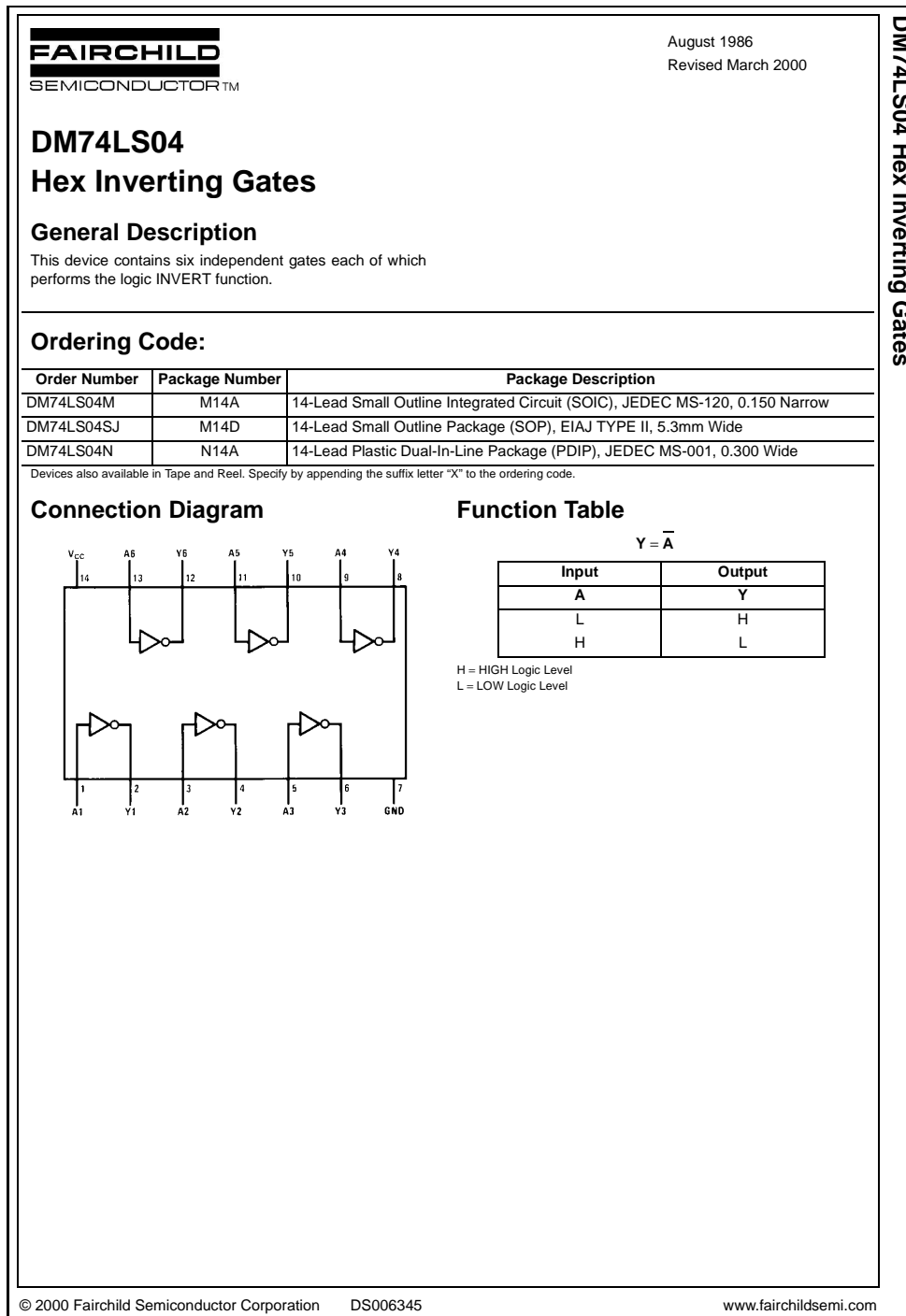


Figure D.4: Data Sheet for Fairchild 74LS04 Hex Inverter  
 Manufacturer: Fairchild Semiconductor  
 URL: <http://www.fairchildsemi.com/pf/DM/DM74LS04.html>

## D.4 MC14067B Analog Multiplexer/Demultiplexer

### MC14067B

#### Analog Multiplexers / Demultiplexers

The MC14067 multiplexer/demultiplexer is a digitally controlled analog switch featuring low ON resistance and very low leakage current. This device can be used in either digital or analog applications.

The MC14067 is a 16-channel multiplexer/demultiplexer with an inhibit and four binary control inputs A, B, C, and D. These control inputs select 1-of-16 channels by turning ON the appropriate analog switch (see MC14067 truth table.)


- Low OFF Leakage Current
- Matched Channel Resistance
- Low Quiescent Power Consumption
- Low Crosstalk Between Channels
- Wide Operating Voltage Range: 3 to 18 V
- Low Noise
- Pin for Pin Replacement for CD4067B

**MAXIMUM RATINGS** (Voltages Referenced to  $V_{SS}$ ) (Note 1.)

Symbol	Parameter	Value	Unit
$V_{DD}$	DC Supply Voltage Range	- 0.5 to + 18.0	V
$V_{in}, V_{out}$	Input or Output Voltage Range (DC or Transient)	- 0.5 to $V_{DD} + 0.5$	V
$I_{in}$	Input Current (DC or Transient), per Control Pin	$\pm 10$	mA
$I_{sw}$	Switch Through Current	$\pm 25$	mA
$P_D$	Power Dissipation, per Package (Note 2.)	500	mW
$T_A$	Ambient Temperature Range	- 55 to + 125	$^{\circ}C$
$T_{stg}$	Storage Temperature Range	- 65 to + 150	$^{\circ}C$
$T_L$	Lead Temperature (8-Second Soldering)	260	$^{\circ}C$

1. Maximum Ratings are those values beyond which damage to the device may occur.  
 2. Temperature Derating:  
 Plastic "P and D/DW" Packages: - 7.0 mW/ $^{\circ}C$  From 65 $^{\circ}C$  To 125 $^{\circ}C$

This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation,  $V_{in}$  and  $V_{out}$  should be constrained to the range  $V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{DD}$ .  
 Unused inputs must always be tied to an appropriate logic voltage level (e.g., either  $V_{SS}$  or  $V_{DD}$ ). Unused outputs must be left open.

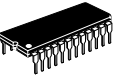


**ON Semiconductor**

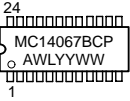
<http://onsemi.com>

---


**MARKING DIAGRAMS**




**PDIP-24**  
P SUFFIX  
CASE 709



MC14067BCP  
AWLYYWW



**SOIC-24**  
DW SUFFIX  
CASE 751E



14067B  
AWLYYWW

A = Assembly Location  
 WL, L = Wafer Lot  
 YY, Y = Year  
 WW, W = Work Week

---

**ORDERING INFORMATION**

Device	Package	Shipping
MC14067BCP	PDIP-24	15/Rail
MC14067BDW	SOIC-24	30/Rail
MC14067BDWR2	SOIC-24	1000/Tape & Reel

© Semiconductor Components Industries, LLC, 2000  
August, 2000 - Rev. 4

1

Publication Order Number:  
MC14067B/D

Figure D.5: Data Sheet for MC14067B Analog Multiplexer/Demultiplexer

Page 1 of 2

Manufacturer: OnSemi Inc.

URL: <http://www.onsemi.com/productSummary/0,4317,MC14067B,00.html>

**MC14067B**

**MC14067 TRUTH TABLE**

Control Inputs					Selected Channel
A	B	C	D	Inh	
X	X	X	X	1	None
0	0	0	0	0	X0
1	0	0	0	0	X1
0	1	0	0	0	X2
1	1	0	0	0	X3
0	0	1	0	0	X4
1	0	1	0	0	X5
0	1	1	0	0	X6
1	1	1	0	0	X7
0	0	0	1	0	X8
1	0	0	1	0	X9
0	1	0	1	0	X10
1	1	0	1	0	X11
0	0	1	1	0	X12
1	0	1	1	0	X13
0	1	1	1	0	X14
1	1	1	1	0	X15

**MC14067B  
PIN ASSIGNMENT**

X	1	24	V <sub>DD</sub>
X7	2	23	X8
X6	3	22	X9
X5	4	21	X10
X4	5	20	X11
X3	6	19	X12
X2	7	18	X13
X1	8	17	X14
X0	9	16	X15
A	10	15	INHIBIT
B	11	14	C
V <sub>SS</sub>	12	13	D

<http://onsemi.com>

2

Figure D.6: Data Sheet for MC14067B Analog Multiplexer/Demultiplexer  
Page 2 of 2

Manufacturer: OnSemi Inc.

URL: <http://www.onsemi.com/productSummary/0,4317,MC14067B,00.html>

# Bibliography

- [1] M. Simon and M.-S. Alouini, *Digital Communication over Fading Channels: A Unified Approach to Performance Analysis*. New York: Wiley-Interscience, 2000.
- [2] M. Zorzi, R. Rao, and L. Milstein, “Error statistics in data transmission over fading channels,” *IEEE Trans. Commun.*, vol. 46, pp. 1468–1477, Nov. 1998.
- [3] R. H. Clarke, “A statistical theory of mobile radio reception,” *Bell Syst. Tech. J.*, vol. 47, pp. 957–1000, 1968.
- [4] H. Suzuki, “A statistical model for urban radio propagation,” *IEEE Trans. Commun.*, vol. COM-25, pp. 673–860, July 1977.
- [5] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, third ed., 1995.
- [6] M. Abramowitz and I. A. Stegun, eds., *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. New York: Dover, 1965.
- [7] W. C. Jakes, ed., *Microwave mobile communications*. New York: Wiley, 1974.
- [8] H. S. Wang and N. Moayeri, “Finite-State Markov Channel — a useful model for radio communication channels,” *IEEE Trans. Veh. Technol.*, vol. 44, pp. 163–171, Feb. 1995.
- [9] C. Tan and N. Beaulieu, “First-order Markov modeling for the Rayleigh fading channel,” in *GLOBECOM 1998*, vol. 6, (Sydney, NSW, Australia), pp. 3669–3674, Nov. 1998.
- [10] C. Tan and N. Beaulieu, “On first-order Markov modeling for the Rayleigh fading channel,” *IEEE Trans. Commun.*, vol. 48, pp. 2032–2040, Dec. 2000.
- [11] L. Kanal and A. Sastry, “Models for channels with memory and their applications to error control,” *Proc. IEEE*, vol. 66, pp. 724–744, July 1978.
- [12] F. Babich and G. Lombardi, “A Markov model for the mobile propagation channel,” *IEEE Trans. Veh. Technol.*, vol. 49, pp. 63–73, Jan. 2000.
- [13] W. Turin and R. van Nobelen, “Hidden Markov modeling of flat fading channels,” *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1809–1817, Dec. 1998.

- [14] U. Dersch, R. Ruegg, H. Kaufmann, and R. Rufener, “Modelling and simulation of indoor radio channels,” in *Proc. IEEE ICC 93*, vol. 3, (Geneva), pp. 1970–1974, May 1993.
- [15] C. Snow and S. L. Primak, “On a filter based simulator of Isotropic Sattering Omnidirectional Receiving Antenna (ISORA) fading channel.” Internal Note I, October 2001.
- [16] E. Casas and C. Leung, “A simple digital fading simulator for mobile radio,” *IEEE Trans. Veh. Technol.*, vol. 39, pp. 205–212, Aug. 1990.

# Vita

NAME: Christopher Snow  
PLACE OF BIRTH: Winnipeg, Manitoba  
YEAR OF BIRTH: 1979  
SECONDARY EDUCATION: Sir Frederick Banting S.S. (1993-1998)  
HONOURS and AWARDS: First Prize, Electrical Engineering Design Day 2003  
2002 IEEE Canada Vehicular Technologies Travel Bursary  
2002/03 IEEE Inc. Award  
2001/02 Hydro One Scholarship  
NSERC Undergraduate Research Awards (1999, 2000, 2001)  
Dean's Honour List (1999, 2000, 2001, 2002)  
UWO Entrance Scholarship (1998)