A LABVIEW BASED SOFTWARE DEFINED RADIO FOR CARRIER AND SYMBOL TIMING EXTRACTION

BY

SERGIO LARA, B.S.

A thesis submitted to the Graduate School

in partial fulfillment of the requirements

for the degree

Master of Science

Major Subject: Electrical Engineering

New Mexico State University

Las Cruces New Mexico

April 2020

Sergio Lara

Candidate

Electrical Engineering

Major

This Thesis is approved on behalf of the faculty of New Mexico State University, and it is acceptable in quality and form for publication:

Approved by the thesis Committee:

Dr. Charles Creusere

Chairperson

Dr. Paul Furth

Committee Member

Dr. David Mitchell

Committee Member

DEDICATION

I dedicate this work to my parents, Graciela and Jesus, who love me unconditionally, and my brothers, Jesus and Christian, for their continuous support.

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my advisor, Dr. Charles Creusere, for his continuous support, encouragement, and patience. Besides my advisor, I would like to thank Mark Dunham for his expertise in the field. Lastly, my sincere thanks to Applied Technology Associates for providing me with the opportunity to work on this research project.

VITA

March 30, 1994	Born at El Paso, Texas
2012-2016	B.S., New Mexico State University, Las Cruces, NM
2014-2016	Computer Technician, New Mexico State University New Mexico State University
2018-2020	Research Assistant, Electrical Engineering Department New Mexico State University

Major Field: Electrical Engineering

ABSTRACT

A LABVIEW BASED SOFTWARE DEFINED RADIO

FOR CARRIER AND SYMBOL TIMING

BY

SERGIO LARA, B.S.

Master of Science

New Mexico State University

Las Cruces, New Mexico, 2020

Dr. Charles Creusere, Chair

In this project, we study LabVIEW and LabVIEW FPGA as a possible environment for software-defined radio development. In addition, a software defined radio receiver is built to extract data for possible use in satellite diagnostics.

LabVIEW and LabVIEW FPGA are National Instruments proprietary graphical programming languages which use the dataflow programming paradigm. Coupled with National Instruments' vector signal transceiver and FPGA module, the intricacies of developing an SDR which includes a phase-locked loop for carrier phase extraction and a symbol timing algorithm are explored. Additionally, the FPGA tools readily available through the integrated Xilinx intellectual property blocks are investigated and used in the developed software-defined radio.

The developed radio using LabVIEW contains two parts: a phase-locked loop and a symbol timing algorithm. Using a Weaver demodulator and a complex to complex mixer, the PLL was developed and the carrier phase was extracted. Furthermore, the symbol arrival time intervals were calculated through a window based detection algorithm. Lastly, a two-stage archiving protocol was designed, where data is temporarily stored in a buffer on the FPGA board and then streamed to the hard drive in an efficient fashion.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

LabVIEW is a language created by National Instruments used in the creation of test and measurement automations. With the introduction of LabVIEW FPGA and the introduction of the FlexRIO FPGA module, it is desired to determine whether LabVIEW is a viable option for designing software defined radios. Previous software defined radio development kits have not provided communications engineers with a well polished environment, with many systems still implemented using analog hardware. Advances in embedded systems have opened up the possibility for communications components to be implemented digitally. Various SDRs have been implemented on FPGA cards using languages such as VHDL and Verilog, but knowledge of script programming is required to use them. Using an FPGA card, it is desirable to determine whether LabVIEW is a suitable environment for software defined radios.

Communications systems are dependent on the crystal oscillators to properly generate the carrier sinusoid and the symbol timing intervals. Over time, the the oscillator crystal wears out, causing the frequency of the sinusoids to deviate from the desire value. In situation where the deviations are small, the small change in frequency can be ignored or possibly fixed, through a software update of the communications system. On the other hand, if the deviations become large enough or the deviations change sporadically, the communications system becomes unusable. Therefore, it is imperative to know the health of the crystal oscillator. However, determining the health of the oscillator crystal in a satellite in orbit is impossible. To estimate the health of the satellite, a software defined radio capable of tracking and logging the carrier frequency and the symbol timing deviations is desired.

Using such a system, it may be possible to determine the health of the crystal with post processing without the need to physically inspect the satellite.

2 BACKGROUND

2.1 FPGA

Field Programmable Gate Arrays (FPGAs) are programmable semiconductor devices that can be programmed to run highly complex algorithms [17]. Application Specific Integrated Circuits (ASICs), on the other hand, are custom manufactured chips designed to perform a task. Comparing the two, ASICs outperform the FPGAs in speed and power consumptions, but are not programmable; therefore, updates to the algorithm or to apply error fixes are impossible. Due to advances in technology, FPGA have become extremely powerful and are used in a wide range of applications including: aerospace & defense, audio, medical, wired communications, and, even, wireless communications [17]. Additionally, the programmability of the FPGA allows for faster development of systems. The longest step in developing a system with an FPGA is the compilation of the code. Even with a fast server, the compilation can take multiple hours to complete. In contrast, an ASIC requires weeks to design and manufacture. Lastly, the cost to manufacture an ASIC is high while multiple codes can be compiled and uploaded to the FPGA, allowing for different systems to be implemented at the cost of one FPGA.

2.2 Carrier Frequency and Doppler Effect

Satellites communicate to the earth station by modulating a digital signal with a carrier sinusoid. To generate the carrier of the signal, an oscillator is configured to run at a specific frequency. The signal is then transmitted to the earth station, where it is demodulated and decoded. In order for the communication systems to work, the satellite and the earth station must generate the exact same carrier. In other words, the carrier sinusoids should have the same frequency and the same phase. This posses a great problem in real world applications. As the satellite orbits the earth, the apparent carrier frequency of the signal will change as observed by the ground station. This effect is known as the doppler effect, which is the change in frequency produced by a moving source with respect to an observer [2]. Phase-locked loops can be implemented in the receiver to ensure that its generated carrier is identical to the modified carrier of the received waveform.

2.3 Digital Modulations

Two types of digital modulation types are commonly used for satellite earth station communications: binary phase shift keying (BPSK) and binary frequency shift keying (BFSK). With BPSK, the phase of the carrier is determine by the bit being transmitted (i.e., a "0" or a "1"). Any combination of phases can be used, although, to obtain optimum performance, the phases should be separated by 180° [5]. For example, to send a "1", the output carrier would have a phase of 0° and to send a "0", the output carrier would have a phase of 180°. BFSK, on the other hand, changes the frequency of the carrier signal based on the bit transmitted. The best performance for BFSK is obtained when the two frequencies are orthogonal to each other, with the average of the two frequency being the desired center frequency. To achieve orthogonality, the frequency should be separated by $\frac{1}{T}$, where T is the bit period or interval. For instance, when a "1" is transmitted the frequency of the signal is $2.271716GHz$; when a "0" is transmitted, the frequency of the signal would be $2.271684GHz$. The average of the two frequency is the center frequency of $2.2717GHz$