COMPACT FIBERID READER USING OPTICAL FREQUENCY DOMAIN REFLECTOMETRY (OFDR)

McKensie Sherlock

Follow this and additional works at: https://digitalcommons.uri.edu/theses
COMPACT FIBERID READER USING OPTICAL FREQUENCY DOMAIN REFLECTOMETRY (OFDR)

BY

MCKENSIE SHERLOCK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND
2023
MASTER OF SCIENCE THESIS
OF
MCKENSIE SHERLOCK

APPROVED:
Thesis Committee:

Major Professor    Tao Wei
Kaushallya Adhikari
James Miller
Brenton DeBoef
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND
2023
ABSTRACT

As devices grow increasingly interconnected, the need to identify them has grown. While current identification technology like barcodes, QR codes, digital magnetic strips, and RFID exist, they all share the shortcoming of reproducibility, something undesired for a unique identifier. Posing as an alternative, physical unclonable functions (PUF) rely on random physical properties of materials generated when manufactured, patterns that are unique and can’t be reproduced, like a fingerprint. Optical fibers can be used as PUFs given that they produce a unique reflection pattern as a function of distance, a Rayleigh scatter pattern. Therefore, this research aims to build a reader, recorder, and comparator of this unique fiber identification information, a fiber ID reader, while specifically focusing on improving the form factor of the data acquisition.

A prototype fiber ID reader was designed using a computer controlled oscilloscope and optical frequency domain reflectometry (OFDR). The data acquisition system was minimized by using an FPGA system on chip (SoC), the Red Pitaya. RoC plots of the system were taken while testing different lengths of test fiber, with the smallest length having an EER of 10%.
ACKNOWLEDGMENTS

I would like to thank several people for their support and help—without them this research would not have been possible. First, I would like to thank my major professor Dr. Tao Wei for his mentorship. He first invited me to work in the Next Lab as an undergraduate student, and remained a guiding figure throughout my time in graduate school. I’m also grateful to my thesis committee and URI’s ECBE department.

I would also like to thank my fellow Next Lab researchers. Whenever I needed help with new lab equipment or came across unusual errors, they were there to give tips and bounce ideas off of. Special thanks to Zhenyu Xu and Thomas Mauldin, who helped me greatly in the beginning stages while I was learning the basics of FPGAs and optical setups.

Lastly, I would like to thank my family for supporting me now and always.
TABLE OF CONTENTS

ABSTRACT ................................................................. ii

ACKNOWLEDGMENTS ................................................... iii

TABLE OF CONTENTS ................................................... iv

LIST OF FIGURES ....................................................... vii

LIST OF TABLES ........................................................ xi

CHAPTER

1 Introduction .......................................................... 1
  1.1 Fiber as a Physical Unclonable Function (PUF) ............... 1
  1.2 Optical Reflectometry and OFDR ............................... 2
  1.3 SoC and FPGA Basics ........................................... 3

2 Prototype Design ..................................................... 5
  2.1 Materials, Devices, and Software ................................ 5
  2.2 Iteration 1: Basic Reflection Collector ......................... 7
    2.2.1 Circuit Layout ........................................... 7
    2.2.2 MZI Math ............................................... 9
    2.2.3 Data Acquisition ....................................... 11
    2.2.4 Zero-Crossing Resampling .............................. 12
    2.2.5 Signal Domain Transformation ........................... 14
  2.3 Iteration 2: Balanced Photo Detection (BPD) Reflection Collector 15
    2.3.1 Vertical Resolution and High Pass Filter ............... 16
2.3.2 Polarization Detector Amplification and Laser Power . . 16
2.3.3 Reviewing Power Levels Throughout the System . . . . . 18
2.3.4 Implementing Balanced Photodetector . . . . . . . . . . 21
2.4 Iteration 3: Polarization-Diverse Reflection Collector . . . . 22
  2.4.1 The Measuring Path and Polarization Controller Math . 23
  2.4.2 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
2.5 Iteration 4: Fiber Comparison . . . . . . . . . . . . . . . 27
3 SoC Design . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
  3.1 FPGA Training . . . . . . . . . . . . . . . . . . . . . . . . . 30
    3.1.1 Verilog and Vivado . . . . . . . . . . . . . . . . . . . . 30
    3.1.2 PYNQ . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
    3.1.3 RedPitaya . . . . . . . . . . . . . . . . . . . . . . . . . 31
  3.2 Zero-Crossing Detector (ZCD) . . . . . . . . . . . . . . . . 32
    3.2.1 Design 1 . . . . . . . . . . . . . . . . . . . . . . . . . . 32
    3.2.2 Design 2 . . . . . . . . . . . . . . . . . . . . . . . . . . 34
    3.2.3 Design 3 . . . . . . . . . . . . . . . . . . . . . . . . . . 35
    3.2.4 In-lab ZCD Board . . . . . . . . . . . . . . . . . . . . . 36
  3.3 Reflection Collector on Red Pitaya . . . . . . . . . . . . . . 36
    3.3.1 DAC and Trigger Signal . . . . . . . . . . . . . . . . . 40
    3.3.2 ADC and Data Repackager . . . . . . . . . . . . . . . . 40
    3.3.3 ZCD and Hysteresis . . . . . . . . . . . . . . . . . . . . 41
  3.4 Similarity and Identification with Cross-correlation . . . . . 43
    3.4.1 Xcorr IP block . . . . . . . . . . . . . . . . . . . . . . 43
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4.2</td>
<td>Replicating the Similarity Function</td>
<td>44</td>
</tr>
<tr>
<td>4</td>
<td>System Evaluation</td>
<td>46</td>
</tr>
<tr>
<td>5</td>
<td>Discussion</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>LIST OF REFERENCES</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>BIBLIOGRAPHY</td>
<td>54</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Diagram of Iteration 1 Optical Circuit</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Photo of Iteration 1 Setup</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>Voltage vs Time Plot of Measuring Path (A) and Resampling Path (B)</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>Short-time Fourier transform of the resampling path’s signal. The range between the two red dots corresponds with the range of data we want to use. However, the frequency is not adequately linear even within this range.</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>Truncated Plot of Resampling data with Zero Crossing Points Marked</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>Reflection vs Distance Plot of Measuring Path Reflections using a 30m DUT</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>Input Voltage Data of Measuring Path (A) and Resampling Path (B) using 1kHz filter and 6.5 dBm TLS Power</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>Reflection vs Distance Plot of Measuring Path Reflections using 1kHz Filter and 6.5 dBm TLS Power</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>Reflection vs Distance Plot of Measuring Path Reflections using 3.5 dBm TLS Power and 10 dB PD Amplification</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>Reflection vs Distance Plot of Measuring Path Reflections using 3.5 dBm TLS Power and 20 dB PD Amplification</td>
<td>19</td>
</tr>
<tr>
<td>11</td>
<td>Reflection vs Distance Plot of Measuring Path Reflections using 6.5 dBm TLS Power and 10 dB PD Amplification</td>
<td>19</td>
</tr>
<tr>
<td>12</td>
<td>Reflection vs Distance Plot of Measuring Path Reflections using 6.5 dBm TLS Power and 20 dB PD Amplification</td>
<td>19</td>
</tr>
<tr>
<td>13</td>
<td>Diagram of Expected Power Levels throughout the System Assuming 3.5 dBm TLS Power</td>
<td>20</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Diagram of Measured Power Levels throughout the System Given 3.5 dBm TLS Power (Measured in Black and Estimated from Measured in Purple)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Diagram of Measured Power Levels throughout the System Given 3.5 dBm TLS Power after Replacing Problematic APC Connector</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Reflection vs Distance Plot of System using a BPD</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Diagram of Iteration 3 Optical Circuit</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Simplified Measuring Path MZI with Polarization Plots of the Signal on Both Arms</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Voltage vs Time Plot of Signals on the Measurement Paths S (A) and P (B), and Resampling Path (C)</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Reflection Level vs Distance Plots of Vertical and Horizontal Polarizations, X-axis Zoomed in to 10 m</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Voltage vs Time Plot of Signals on the Measurement Paths S (A) and P (B), and Resampling Path (C), After Making Adjustments</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Reflection Level vs Distance Plot of Combined Measurement Path of System with 3 m DUT Connected (Blue) and no DUT connected (Orange)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Reflection vs Distance Plot of Target Slice and Search Area of Interest with Corresponding Similarity vs Distance Plot</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Circuit diagram of ZCD design #1: using TVL021</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Picture of ZCD design #1 implemented on a protoboard</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Output of ZCD design #1 as measured by an oscilloscope</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Oscilloscope measurement of rising and falling time delays of ZCD design #1</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Picture of ZCD design #2 implemented on a protoboard</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Oscilloscope measurement of rising and falling time delays of ZCD design #2</td>
<td></td>
</tr>
</tbody>
</table>
Figure 30 Picture of ZCD design #2.2 implemented on a protoboard . . . 38
Figure 31 Oscilloscope measurement of rising and falling time delays of ZCD design #2.2 ........................................ 38
Figure 32 Picture of ZCD design #3 implemented on a protoboard . . . 39
Figure 33 Oscilloscope measurement of rising and falling time delays of ZCD design #3 ........................................ 39
Figure 34 Block Diagram of FPGA Design for Data Collection ........ 40
Figure 35 Setup for Testing the ZCD Board and Red Pitaya with Function Generators .............................................. 42
Figure 36 Plot of 100 kHz Signal Sampled at 1 MSps .................... 42
Figure 37 Screenshot of FPGA Debugger Showing ZCD Board Output and Valid Signal .............................................. 43
Figure 38 Block Diagram of FPGA Design for Cross Correlation IP ... 44
Figure 39 Block Diagram of FPGA Design for Cross Correlation IP ... 44
Figure 40 Reflection vs distance data, with the calibration region colored in yellow and different sections of test fiber represented in other colors .............................................. 45
Figure 41 ROC plot comparing hysteresis length settings for 30 cm DUT . 47
Figure 42 ROC plot comparing hysteresis length settings for 10 cm DUT . 47
Figure 43 ROC plot comparing hysteresis length settings for 3cm DUT . 47
Figure 44 ROC plot comparing hysteresis length settings for 30 cm DUT, zoomed in to a 5% EER ........................................ 48
Figure 45 ROC plot comparing hysteresis length settings for 10 cm DUT, zoomed in to a 5% EER ........................................ 48
Figure 46 ROC plot comparing hysteresis length settings for 3 cm DUT, zoomed in to a 5% EER ........................................ 48
Figure 47 ROC plot comparing collections from different days for 30 cm DUT ...................................................... 49
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>ROC plot comparing collections from different days for 10 cm DUT</td>
</tr>
<tr>
<td>49</td>
<td>ROC plot comparing collections from different days for 3 cm DUT</td>
</tr>
<tr>
<td>50</td>
<td>ROC plot comparing collections from different days for 30 cm DUT, zoomed in to a 5% EER</td>
</tr>
<tr>
<td>51</td>
<td>ROC plot comparing collections from different days for 10 cm DUT, zoomed in to a 5% EER</td>
</tr>
<tr>
<td>52</td>
<td>ROC plot comparing collections from different days for 3 cm DUT, zoomed in to a 5% EER</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
</tr>
</tbody>
</table>

1. Table of Iteration 1 Optical Circuit Settings
2. Table of Iteration 1 Data Acquisition Settings
CHAPTER 1

Introduction

The modern world is increasingly more interconnected. More and more essential transactions of everyday life, like shopping, banking, and communicating both personally and professionally, all occur more frequently online. Additionally, once physical-only security tools like keys for buildings and vehicles are now digitally and wirelessly checked, like hotel key cards and car key fobs. Adding another layer of interconnectivity is the internet of things (IoT), which connects everyday household devices like ovens and refrigerators, to larger systems like cameras for personal home surveillance.

Security measures need to be in place to allow only designated devices to connect and communicate. To determine which device is which, we use unique device identifiers, or UIDs. Current examples of popular UIDs include barcodes, QR codes, digital magnetic stripes and RFID systems. However, these ID technologies all fall short in one key area, which is the reproducibility of their algorithmically generated identification data. One way to remedy this issue is to use a physical unclonable function (PUF) as an identifier instead, with one such PUF type being optical fiber [1].

1.1 Fiber as a Physical Unclonable Function (PUF)

Physical unclonable functions, or physical one-way functions, work like the one-way functions modern cryptology relies on. These one-way functions can be made or read in one direction, but are hard to decipher in the other direction [2]. Physical unclonable functions use the properties of a physical medium or transmission line to serve as the one-way function. For example, a human fingerprint can be thought of as a function of groove depth vs spatial area of the finger.
two fingerprints are identical, even fingerprints from the same person, they make
great, non-reproducible physical identifiers.

Another example of a PUF, and the target type for this research, is optical
fiber. While fingerprints have varying groove depths, optical fibers have variable
indices of refraction in their core. When light moves forward through a fiber, these
small and random fluctuations cause scattering of light backwards through the
fiber, which is known as Rayleigh scatter [3]. These random refraction patterns
are the result of imperfections generated in the fiber’s manufacturing process, be-
coming a reliable, unchanging feature of the fiber [4]. Therefore, fibers can be
thought of as functions of variable reflection vs distance.

1.2 Optical Reflectometry and OFDR

For PUFs to be used as identifiers, they must first be read and stored. While
a fingerprint’s grooves can be read and recorded with a scanner or paper and
ink, light through optical fibers are read using optical reflectometry. Two major
methods for performing optical reflectometry are optical time domain reflectometry
(OTDR) and optical frequency domain reflectometry (OFDR). The older of the two
methods, OTDR, works by sending a pulse of light through the fiber and recording
its echoes, or reflections. Since the speed of light through a single medium is always
constant, the delays when the echoes caused by reflection points are received can
be translated to the reflection point locations along the fiber.

Conversely, the reflection’s location along the fiber isn’t measured as directly
using OFDR as it is using OTDR. Instead of releasing a short tone, the laser emits
light that sweeps continuously through a predetermined range of wavelengths. The
light intensity level is translated to voltage, and voltage vs time data is recorded.
To view the data as reflection level vs distance, the data needs to be transformed.
Since the signal was taken using a frequency sweep, it can be viewed as a frequency
domain signal in addition to being a real time domain signal. This signal can then be transformed using an FFT into the reflection vs time domain, like one is familiar with in OTDR. Finally, time domain can be translated to distance, once again using the speed of light as a constant relationship between the two.

Examples of OFDR setups can be seen in [5] and [3]. While these configurations work well in a lab setting, setups need to be smaller and more cost-effective to be used in industry. To take steps in compacting the FiberID setup in [1], work will be done on the DAQ side, so that data collection, data transformation, and post-processing can all be done on one device. In this way, one can replace the setup of a computer, MATLAB, and auxiliary oscilloscope with a single FPGA System on Chip (SoC).

1.3 SoC and FPGA Basics

A system on chip includes all the hardware for inputs and outputs, as well as the computing resources for controlling and processing, all on one chip. This is contrasted with devices like personal computers, which are systems of multiple components connected to a motherboard. Since everything needed for the system is on one chip for SoCs, this means they help designs to be compact and small form factor.

FPGAs stand for Field Programmable Gate Arrays. FPGAs are useful hardware development tools since one can modify a design as errors are encountered. They work as a system of lookup tables to simulate gates. For example, to simulate an AND gate, a lookup table can be used which outputs ”1” for an input of ”11,” and outputs ”0” for the three other input combos. More complex circuits are then made by combining these lookup tables, in addition to using other resources like BRAM, VRAM, digital signal processors (DSP) and flip-flops (FF) [6].

A designer doesn’t need to describe the circuit at the gate level, however.
Hardware description language, like Verilog and VHDL, grant designers the ability to write C-like instructions that describe the circuit, which is later interpreted and generates a bitstream file. The bitstream tells the FPGA the configuration it needs for your design. One other way to describe a design is by using high-level synthesis, or HLS. Here, one can use actual C to describe the design, while also prompting the interpreter to fit designs within time and resource utilization specifications.

One particularly powerful combination is SoC FPGAs, where the computing power of a microprocessor can be combined with the programmable hardware potential of FPGAs. FPGAs have great parallelization potential but not a lot of memory, while processors are smart and easier to interface with for users, but are better at processing serially. By computing repeatable functions in parallel using the programmable logic (PL), and coordinating and controlling the PL’s function calls using the processing system (PS), the pros of both kinds of computing can be achieved.
CHAPTER 2
Prototype Design

The following chapter describes the development of the prototype fiber ID reader. The optical circuit side of the fiber ID reader was built in iterations. In the first iteration, a bare-bones reflection collector was assembled. Iterations two and three experiment with setups that use different types of photo detection technology, as well as adjust some design parameters like laser output power level and interferometer length. Lastly, the similarity calculation to compare different fiber IDs, cross correlation, is added in the fourth iteration, where the system now functions as a working prototype fiber ID reader.

2.1 Materials, Devices, and Software

Numerous optical devices, measurement hardware and software were used in the development of the prototype fiber ID reader. The following descriptions introduce each component, as well as briefly explain its importance in building the system.

Tunable Laser Source (TLS)

The laser source used for all iterations of the system is a Santec TSL-710\textsuperscript{1} tunable semiconductor laser. It’s a high performance laser with a large wavelength range and high signal-to-noise ratio, and is a popular tool for sensing and interferometry. In this project, it provides the sweeping frequency light source needed to perform OFDR.

\textsuperscript{1}https://www.santec.com/dcms_media/other/instruments_TSL-710-C-E.pdf
**Photodetector (PD)**

Photodetectors convert an optical signal into an electrical signal. When light hits the diode within the photodetector, electrons are knocked from their positions and moved, causing a current to flow. The starting power of light in watts to the resulting current in amps is about 1:1. The current is then translated to voltage using a transimpedance amplifier.

The types of photodetectors used during the development of this system are as follows: two Thorlabs PDA10CS\(^2\) amplified detectors, one Koheron PD100\(^3\) detector, and one Koheron PD100B\(^4\) balanced detector. All photodetectors were resources already available in the lab.

**Polarization Controller (PC)**

Polarization controllers can induce a desired polarization state by manipulating the position and tension on a fiber. The controller used in developing the prototype, the Thorlabs FPC562\(^5\), has three disks with fiber looped around each. These three disks act as two quarter-wave plates and one half-wave plate, which together can bring about a desired polarization state by changing the tilt on the disks.

**Polarization Beam Splitter (PBS)**

Polarization beam splitters separate light into parallel and perpendicular components. By passing light through a prism with specific refractive properties, the light can be separated into just the transmission and reflection components. These components are often denoted S, for perpendicular or reflected, and P for parallel or transmitted. The PBS used during prototype development is Thorlabs’

---

\(^2\)https://www.thorlabs.com/thorproduct.cfm?partnumber=PDA10CS  
\(^3\)https://www.koheron.com/photonics/pd100-photodetection  
\(^4\)https://www.koheron.com/photonics/pd100b-photodetection  
\(^5\)https://www.thorlabs.com/thorproduct.cfm?partnumber=FPC562
PicoScope

PicoScopes are oscilloscope peripherals designed to be controlled by an external computer, instead of being their own standalone system. Today, PicoScope has a wide library of devices to choose from, with varying number of ports, bit resolution, and storage capacity depending on a project’s needs. Over the course of the prototype’s development, three different PicoScopes were used, two being from the 5000’s family and one from the 6000’s.

In addition to the default PicoScope GUI, they have also developed support packages or toolboxes that allow for controlling the scope via a different program. For the development of this system, the PicoScope’s corresponding Instrument Toolbox for MATLAB was used to easily take data and perform post-processing on it in one script. Further simplifying this Toolbox is the library PicoScope.m written by Zhenyu Xu, which was used in the making of the MATLAB data collection scripts.

2.2 Iteration 1: Basic Reflection Collector

In this first iteration, a basic OFDR setup was made. This included splicing together an optical interferometer circuit, writing a MATLAB script to control data collection using the PicoScope, and transforming the test fiber’s reflection data into the desired reflection vs distance form.

2.2.1 Circuit Layout

For the first design iteration, we needed an initial setup to begin building off of. We first referenced the work of Soller et al. and Froggatt [5, 3], keystone papers in the field of OFDR, but used only the most essential components for a working

---

OFDR circuit. A block diagram of the system is depicted below in Figure 1, and a photo of the setup is shown in Figure 2.

The circuit begins at the TLS. Light from the TLS travels thorough a 95/5 coupler, with 95% of the light passing through the measurement path on top and 5% of light traveling down the resampling path at the bottom. The outputs of both paths are collected at amplified photodetectors, which are then connected to the PicoScope for measurement. A table detailing the parameter settings for the optical circuit setup can be seen below in Table 1 below.

<table>
<thead>
<tr>
<th>TLS</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sweep Range</td>
<td>1540-1560 nm</td>
</tr>
<tr>
<td>Sweep Rate</td>
<td>50 nm/s</td>
</tr>
<tr>
<td>Power</td>
<td>3.5 dBm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resampling MZI</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difference in Length $\Delta L$</td>
<td>20 m</td>
</tr>
<tr>
<td>Time Delay $\Delta \tau$</td>
<td>200 ns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PD</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 dB</td>
</tr>
</tbody>
</table>

Table 1. Table of Iteration 1 Optical Circuit Settings

Both paths are made up of Mach–Zehnder interferometers (MZI). MZIs are a type of interferometer where a light source is split down two arms, and their inter-
ference is observed when the signals from both arms are mixed back together. The MZI on the measuring path measures the interference between the local oscillator arm (the lower arm), and the reflections caused by the fiber under test (FUT) as they scatter back through the couplers. The MZI on the resampling path measures the time delay between the same signal traveling along arms of different length. More information about how MZIs work using the resampling MZI as an example is provided in the next subsection.

2.2.2 MZI Math

Light on the resampling path passes through the first 50/50 coupler of an MZI, dropping 3dB in power, mixing with itself at the output of the second 50/50 coupler. The two arms of the MZI are optical fibers of differing length, meaning that one copy of the signal is delayed when mixed together at the output of the MZI. The AC components of the signal and delayed signal on the output can be described with the following equations:
\[ E_1(t) = A_1 e^{j2\pi f(t - \tau_1)} \]
\[ E_2(t) = A_2 e^{j2\pi f(t - \tau_2)} \]

with \( \tau_1 \) and \( \tau_2 \) being the delays for the input of a frequency \( f \) to reach the output coupler of the MZI and \( A_1 \) and \( A_2 \) being the amplitude of both inputs. The output signal \( E \) can then be found to be

\[ E(t) = A_1 e^{j2\pi f(t - \tau_1)} + A_2 e^{j2\pi f(t - \tau_2)} \]

Note that the intensity of the light is proportional to the output electric field

\[ I(t) \propto |E(t)|^2 \]

After substituting for \( E(t) \) and simplifying using Euler’s formula, the above equation can be rewritten as

\[ I(t) \propto A_1^2 + A_2^2 + 2A_1A_2 \cos(2\pi f(\tau_2 - \tau_1)) \]

We can substitute \( \Delta \tau \) for the difference of the two delays \( \tau_1 \) and \( \tau_2 \). We can also represent the frequency \( f \) as the product of the laser’s frequency sweep rate \( \gamma \) and time \( t \). Finally, we get the representation of the intensity of the mixed signals at the MZI output

\[ I(t) \propto A_1^2 + A_2^2 + 2A_1A_2 \cos(2\pi \gamma \Delta \tau t) \]

Therefore, the final output intensity of the resampling MZI is proportional to the sum of a DC element and a cosine of constant frequency, represented more explicitly below
\[ I_{MZI}(t) \propto A_1^2 + A_2^2 + 2A_1A_2 \cos(2\pi f_{MZI}t) \]  

(1)

\[ f_{MZI} = \gamma \Delta \tau \]  

(2)

The constant frequency \( f_{MZI} \) is then determined by the laser’s sweep speed and \( \Delta \tau \), a physical property of the MZI. It can be found by taking the difference between the lengths of the two arms and dividing by the speed of light, like so

\[ \Delta \tau = \frac{L_1 - L_0}{c} \]  

(3)

### 2.2.3 Data Acquisition

Now that the optical circuit for this first iteration was complete, the fundamental control for the data acquisition was created. As stated previously, the data was collected using a joint PicoScope and MATLAB setup. A collection script was written, which sets the sampling frequency and duration of the PicoScope, the vertical bit resolution and coupling type of the inputs, as well as generates the TTL signal that triggers the laser to begin its sweep. The measuring path and resampling path were connected to inputs A and B, respectively. A table noting the specs of the DAQ setup is shown below in Table 2, and plots of the raw voltage data for the measuring path and resampling path received by the photodetectors is shown in Figure 3.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ch A - Vertical Range</td>
<td>20 mV</td>
</tr>
<tr>
<td>Ch B - Vertical Range</td>
<td>1 V</td>
</tr>
<tr>
<td>Ch A - Coupling Type</td>
<td>AC</td>
</tr>
<tr>
<td>Ch B - Coupling Type</td>
<td>AC</td>
</tr>
<tr>
<td>Sampling Frequency</td>
<td>50 MSps</td>
</tr>
<tr>
<td>Sampling Duration</td>
<td>0.7 s</td>
</tr>
</tbody>
</table>

Table 2. Table of Iteration 1 Data Acquisition Settings
Figure 3. Voltage vs Time Plot of Measuring Path (A) and Resampling Path (B)

With the data acquisition set up, it was almost time to transform the data. But first, the measurement data needed to appear like it underwent a near-perfectly linear frequency sweep. The next subsection describes how to resample the measurement path data using zero crossings.

### 2.2.4 Zero-Crossing Resampling

Before transforming the data, the linearity of the laser’s frequency sweep has to be clean. An initial look at the laser sweep’s linearity is shown below in a short-time Fourier transform (STFT) in Figure 4. If the sweep is perfectly linear which is the ideal case, then the output of the resampling path should be a single constant frequency, and the STFT would reflect this constant frequency region as a flat, horizontal line. As can be seen in the STFT from Figure 4 below, this region is not adequately linear.

To fix this and achieve a more linear frequency sweep, we can focus only on the data corresponding to the zero crossings of the resampling path’s signal. For any constant frequency signal, points corresponding to the same phase between 0
Figure 4. Short-time Fourier transform of the resampling path’s signal. The range between the two red dots corresponds with the range of data we want to use. However, the frequency is not adequately linear even within this range.

and $2\pi$ should be equidistant in time from all other previous and following points of the same phase. If this signal has a close but not quite constant frequency, this distance between same phase points will vary.

Zero-crossings for constant frequency sinusoids always occur twice per $2\pi$ interval, meaning each zero-crossing should always have a $\pi$ phase difference between them. Therefore, by only considering data corresponding to the sampling path’s zero crossings, the reflection data acts as if it was generated using a constant-frequency sweep sampled at twice the frequency of the resampling MZI’s output sinusoid.
2.2.5 Signal Domain Transformation

Now that the reflection data has been collected and resampled, it can be transformed from voltage vs time data to reflection level vs distance.

When taking the Fourier transform, customarily one transforms time-domain data to frequency-domain data. Then, the transformed domain’s axis, the frequency domain, can be determined from the original sampling frequency. In this research, reflection data was taken with a linear frequency sweep, meaning one can also view the signals as frequency-domain data. In this case, the transformed domain’s x-axis, the time domain, can be calculated from a sampling period.

From the section 2.2.2. on MZI’s, the MZI output equation can be rewritten as a function of frequency by combining \( t \) with the frequency sweep speed \( \gamma \).

\[
I(t) \propto A_1^2 + A_2^2 + 2A_1A_2 \cos(2\pi \Delta \tau f)
\]

Here, the difference in frequency \( df \) between two similar points of the resam-
pling signal’s sinusoid is $\frac{1}{\Delta\tau}$. However, since the signal is sampled at every zero crossing point, the difference in frequency between zero crossings is $\frac{1}{2\Delta\tau}$. Therefore, the sampling period needed to determine the time axis of the transformed signal is the inverse of $df$, $2\Delta\tau$. Lastly, one can convert from time-domain to distance-domain by multiplying the time axis by the speed of light in glass.

The reflection vs time plot of the transformed measuring path data can be seen in Figure 6, where a 30 m DUT made up of 10 3 m APC-connected fibers is measured. While the peaks in reflection level corresponding to where the 3 m fibers are connected are visible, the reflection level in between those peaks isn’t visible above the noise floor. In order to see the Rayleigh back scattering needed to perform fiber ID, the noise floor must first be lowered.

![Figure 6. Reflection vs Distance Plot of Measuring Path Reflections using a 30m DUT](image)

### 2.3 Iteration 2: Balanced Photo Detection (BPD) Reflection Collector

This next iteration focused on lowering the noise floor of the system. Efforts were focused on four main areas: adjusting the vertical resolution of the DAQ.
and implementing a high pass filter, raising the amplification level of the photodetectors, reviewing the power levels throughout the system, and implementing a balanced photodetector (BPD).

### 2.3.1 Vertical Resolution and High Pass Filter

Referring back to Figure 3 in Section 2.2.3, one can see the vertical resolution is not well utilized. Because of the low frequency wave within the signal, the resolution cannot be smaller than 20 mV. This is unfortunate, since the higher frequency data traveling along that wave is what actually contains the reflection data corresponding to the device under test. To remove this unwanted low frequency data, a high-pass filter was added. It was determined that when implementing a high-pass filter for this system, the passband should be no higher than 27.4 kHz. Out of the filters already available in the lab, a 1kHz high pass filter was chosen.

After implementing the 1kHz high pass filter, the reflection path signal no longer included that large wave, as seen in Figure 7. This meant the vertical resolution could be lowered to zoom in on that higher frequency data. A reflection vs distance plot of the system with a 3 m DUT can be seen in Figure 8. Rayleigh backscattering reflections between the two connections of the DUT is now visible.

### 2.3.2 Polarization Detector Amplification and Laser Power

Another possible method to lower the noise floor is by raising the power of the signal. This can be done in two ways, one by increasing the photodetector’s amplification setting and two by raising the output power of the laser itself.

Raising the photodetector’s amplification setting shrinks the detector’s bandwidth, so knowledge of your signal’s bandwidth is needed to determine the photodetector’s maximum amplification setting. Since the reflection signal will be resampled at twice the frequency of the resampling signal, 5-6 MSps, the Nyquist
Figure 7. Input Voltage Data of Measuring Path (A) and Resampling Path (B) using 1kHz filter and 6.5 dBm TLS Power

Figure 8. Reflection vs Distance Plot of Measuring Path Reflections using 1kHz Filter and 6.5 dBm TLS Power
frequency of the signal is between 2.5-3MHz. After referring to the datasheet of
the photodetector, the measuring path signal’s photodetector amplification level
was raised from 10 dB to 20 dB.

The laser power output can be easily changed via its menu. The two output
levels tested were 3.5 dBm and 6.5 dBm. All four combinations of photodetector
amplification and laser output power can be seen in Figures 9-12.

2.3.3 Reviewing Power Levels Throughout the System

In addition to adjusting the output power and amplification of the tunable
laser and photodetector, the power level at different locations of the system was
reviewed and compared with the expected power level values.

When given a laser power level, the power levels after all couplers and connec-
tions can be estimated. Possible bad connections or splices can then be found and
corrected if there is a sudden large drop between estimated and measured values.
A diagram detailing the estimated power levels throughout the system given an
input power level of 3.5 dBm is shown in Figure 13, and a diagram showing the
actual measured values of the system is shown in Figure 14. Unmarked gray ovals
represent 50/50 couplers, and green circles indicate APC connections.

It was discovered that there was a suspicious drop in power after the APC
connector leaving the 90% fiber of the 90/10 coupler. When the APC connector
itself was connected to an optical power meter, the meter read 2.8 dBm as expected.
However, when this APC connector was connected to a 3 m fiber and then measured
with the power meter, the meter measured -7.8 dBm instead of the expected 2.06
dBm. After the suspicious connector was replaced with a new APC connector,
power levels were measured once again, this time matching much closer to the
expected values. The fixed system’s power levels are shown in Figure 15.
Figure 9. Reflection vs Distance Plot of Measuring Path Reflections using 3.5 dBm TLS Power and 10 dB PD Amplification

Figure 10. Reflection vs Distance Plot of Measuring Path Reflections using 3.5 dBm TLS Power and 20 dB PD Amplification

Figure 11. Reflection vs Distance Plot of Measuring Path Reflections using 6.5 dBm TLS Power and 10 dB PD Amplification

Figure 12. Reflection vs Distance Plot of Measuring Path Reflections using 6.5 dBm TLS Power and 20 dB PD Amplification
Figure 13. Diagram of Expected Power Levels throughout the System Assuming 3.5 dBm TLS Power

Figure 14. Diagram of Measured Power Levels throughout the System Given 3.5 dBm TLS Power (Measured in Black and Estimated from Measured in Purple)
2.3.4 Implementing Balanced Photodetector

The last attempt to lower the noise floor in this iteration was switching out the amplified photodetector on the measuring path for a balanced photodetector. Balanced photodetectors work by subtracting two copies of the same input that have a 90-degree phase difference between them. Since the two signal copies have complete destructive interference at this phase difference, subtracting the two amplifies the signal while simultaneously negating noise common to both copies of the signal. This works similarly to differential signaling, where the signal of interest is amplified and the common mode noise is canceled out.

Since a balanced photodetector requires two inputs of the same signal, a new APC connector was spliced on to the other output of the last 50/50 coupler on the measurement path and reflection vs. distance data was recorded. Figure 16 shows the reflection vs distance plot of the system with a 3 m fiber connected, acting as the DUT. The Rayleigh backscattering of the DUT is highlighted in red between the two APC connection peaks around 2 m and 4.5 m, and a difference in noise floor is clearly visible.

Figure 15. Diagram of Measured Power Levels throughout the System Given 3.5 dBm TLS Power after Replacing Problematic APC Connector
2.4 Iteration 3: Polarization-Diverse Reflection Collector

One other technique in lowering the noise floor is by reducing the effect of signal fading caused by different polarization states between a measuring arm and local arm destructively interfering with each other. To remedy this, we implemented a polarization-diverse reflection collection system in this iteration.

First, modifications to the optical circuit needed to be made. A diagram of the new circuit is shown in Figure 17. A polarization beam splitter was added to one output end of the 50/50 coupler of the measurement MZI. The two outputs of the PBS were connected to both amplified photodetectors. A new non-amplified photodetector was used for capturing the resampling path’s signal. A new PicoScope with four input signal ports replaced the old two input port PicoScope, now that there were three inputs total. Lastly, a polarization controller was added to the measurement MZI’s local oscillator arm.

Unlike the other modifications made to the system, it may not be immediately clear why a polarization controller on the local oscillator arm is a necessary
addition. The next subsection reviews the reason why in detail.

2.4.1 The Measuring Path and Polarization Controller Math

The new measuring path consists of an MZI and a polarization beam splitter (PBS). In the MZI, one arm is connected to the FUT and the other arm contains a polarization controller (PC). In this project, we collect both horizontal and vertical polarization components of the FUT’s reflection signal separately, then recombine them later. To do so accurately and avoid the FUT’s reflections on one arm interfering unequally with the light on the other arm, a polarization controller set to 45/45 is required on the other arm. The reason why is demonstrated below.

The following Figure 18 depicts a simple version of the MZI of the measuring path with one arm containing a polarization controller, as well as plots of the polarization of the signals $E_1$ and $E_2$.

$E_1$ and $E_2$ can be expressed as their composite horizontal and vertical polarization components, as is done below

\[
E_1^P = E_1 \cos(\theta_1) \quad E_1^S = E_1 \sin(\theta_1)
\]
\[
E_2^P = E_2 \cos(\theta_2) \quad E_2^S = E_2 \sin(\theta_2)
\]

After $E_1$ and $E_2$ pass through the measurement path’s MZI, the intensity of
the vertical and horizontal polarization components is as follows

\[ I^P \propto E_1E_2 \cos(\theta_1) \cos(\theta_2) \]

\[ I^S \propto E_1E_2 \sin(\theta_1) \sin(\theta_2) \]

Using the Pythagorean theorem, we can combine both components to find the expression of the output intensity \( I \) of the MZI

\[ I \propto \sqrt{I^{P2} + I^{S2}} \]

\[ I \propto E_1E_2 \sqrt{\cos^2(\theta_1) \cos^2(\theta_2) + \sin^2(\theta_1) \sin^2(\theta_2)} \]

Here we can see that the intensity of the two signals depends on their polarization states, even when their magnitudes remain the same. One can try to account for this, but determining the polarization state for both arms of every part of the signal would be difficult.
Instead, we can manipulate the polarization of one arm using the polarization controller to make the output intensity no longer dependent on both arm’s polarization states. This can be done by setting $E_1$’s polarization state to 45 degrees.

$$\theta_1 = \frac{\pi}{4}$$

$$I \propto E_1 E_2 \sqrt{\cos^2\left(\frac{\pi}{4}\right) \cos^2(\theta_2) + \sin^2\left(\frac{\pi}{4}\right) \sin^2(\theta_2)}$$

$$I \propto \frac{\sqrt{2}}{2} E_1 E_2 \sqrt{\cos^2(\theta_2) + \sin^2(\theta_2)}$$

$$I \propto \frac{\sqrt{2}}{2} E_1 E_2$$

### 2.4.2 Results

The polarization controller on the optical circuit’s local oscillator arm was calibrated to 45 degrees by monitoring its output power levels after passing through a PBS. The plates of the polarization controller were adjusted until both power levels were about equal.

After all modifications to the system were completed, the PC was calibrated to 45/45 using the optical power meter, and adjustments to the MATLAB script to accommodate the new PicoScope model and its extra inputs were made, the initial data was collected. Plots of the voltage vs time data of the FUT’s reflections for both horizontal and vertical polarization components can be seen in Figure 19. The reflection level vs distance FFT plot of both polarization directions can be seen in Figure 20.

For the sake of both inputs being balanced, the 1kHz filter on the single measuring path signal was removed, since there are now two inputs for each po-
Figure 19. Voltage vs Time Plot of Signals on the Measurement Paths S (A) and P (B), and Resampling Path (C)

Figure 20. Reflection Level vs Distance Plots of Vertical and Horizontal Polarizations, X-axis Zoomed in to 10 m
larization direction, but only one available and viable filter in the lab. This is why the low frequency wave pattern is back on the voltage vs time plots. Therefore, the vertical resolution had to be widened again to accommodate for the low frequency wave.

The FFT plot is not as sharp as it was in previous iterations. While FFTs like this one were encountered when the photodetectors were over saturated, it was probably not the cause of the low quality FFT, since the same FFT issue occurred even after testing the new PicoScope setup with older iterations optical setups, which should produce sharp enough reflection vs distance FFT plots. It was discovered that the new PicoScope, while including two new input slots and retaining a similar size memory to the old PicoScope’s, has a maximum bitrate that is less than the old PicoScope’s.

The only viable PicoScope in the lab now to use was a 4-input, high bitrate but low memory scope. The following actions were taken to fix the issues above. To accommodate the decreased memory, the length of the resampling MZI was halved from 85 m to 40 m, which in turn halved the sampling frequency. Additionally, the wavelength range sweep was decreased and the sweeping frequency was increased. To account for having only one usable filter, two Thorlabs 2 kHz high pass filters were ordered and added to the two measuring signal paths. The new voltage vs time and reflection vs distance data plots taken after making the changes are shown in Figures 21 and 22.

2.5 Iteration 4: Fiber Comparison

As a last step for the prototype design, the fiber is identified. The similarity between two sections of fiber is calculated using the cross correlation function. Normalizing the result gives the comparison a similarity score between zero and one. The prototype’s ID worked by the user inputting a range of test fiber and
Figure 21. Voltage vs Time Plot of Signals on the Measurement Paths S (A) and P (B), and Resampling Path (C), After Making Adjustments

Figure 22. Reflection Level vs Distance Plot of Combined Measurement Path of System with 3 m DUT Connected (Blue) and no DUT connected (Orange)
searching that range for a user determined signature. If the fiber signature of interest did exist in that range, a high score corresponding to the distance location of the signature along the fiber under test would appear. If the signature did not exist within that range, then a peak of high similarity score would not be present.

An example of an identification resulting in a peak of high similarity is illustrated in Figure 23 below. In the top plot, reflection data of the device was taken 10 times, a slice of the device under test (DUT) from one collection, shown in yellow, was compared along the length of the entire DUT from another collection as the area of interest to search, shown in orange. In the bottom plot, similarity was found by taking the sliding dot product of the yellow slice along the orange area of interest of the DUT. The spike of similarity along the DUT in the bottom plot occurs at the starting location of the yellow slice in the top plot, at 4.87 m.

Figure 23. Reflection vs Distance Plot of Target Slice and Search Area of Interest with Corresponding Similarity vs Distance Plot
CHAPTER 3
SoC Design

This next chapter covers the building of the FiberID reader using a system on chip. First, training in FPGA development was conducted in Xilinx’s Vivado using Verilog, followed by practice using PYNQ and the Red Pitaya board. Then, to support the transition from the PicoScope to the Red Pitaya, several designs of a zero-crossing detector (ZCD) board were made. After a working ZCD board had been acquired, the reflection collection and fiber ID features were recreated on the Red Pitaya in both hardware and software using PYNQ.

3.1 FPGA Training

Before beginning FPGA development using the Red Pitaya, I needed to complete a brief period of training. This section briefly introduces Verilog and Vivado, PYNQ, and Red Pitaya, as well as a summary of the training underwent for each.

3.1.1 Verilog and Vivado

As mentioned in the FPGA Basics section, Verilog is a hardware description language that one uses to describe circuits for FPGAs. As I had only a bit of experience using VHDL, another hardware description language, I completed the exercises on HDL Bits, a website for practicing Verilog programming, learning best programming practices for describing state machines and calling Verilog’s equivalent of functions. After completing the HDL Bits exercises, I gained experience in AMD Xilinx’s Vivado, a design suite for programmable hardware. Using the ZedBoard and PYNQ board, I completed tutorials covering AXI streams, the streaming communication method used on AMD devices.
3.1.2 PYNQ

PYNQ is a Xilinx project that aims to use Python to interface with FPGAs. Instead of the usual method of communicating with FPGAs using C, PYNQ connects with FPGAs using Python and Jupyter. This eases programming in the processor side, since PYNQ’s libraries simplifies the use of IPs. These benefits are increased when coupled with the fact that signal processing using other popular python libraries like pandas, NumPy, and SciPy, making PYNQ an extremely useful tool in development. While Xilinx provides two boards specifically designed for use with PYNQ, any board can be controlled with PYNQ with a custom PYNQ image.

Later tutorials for Vivado were completed using PYNQ and the PYNQ board. In addition to gaining experience communicating with FGPAs using PYNQ, additional experience using common IPs, like FIFOs, and Direct Memory Access (DMA) blocks was gained.

3.1.3 RedPitaya

The board chosen to act as the DAQ for the SoC design stage was the Red Pitaya. The Red Pitaya is a combination processor and FPGA board that includes two pairs of 125 MSps, 14-bit ADC and DAC inputs and outputs, making it an ideal tool for this project.

To get started using the Red Pitaya, a PYNQ custom board image was first created. Then, five sample projects\(^1\) for the Red Pitaya written by Anton Potočnik were completed. Project 4, “Frequency Counter,” was of particular help, where his IPs “AXI4-Stream Red Pitaya DAC” and “AXI4-Stream Red Pitaya ADC” were reused throughout the FPGA design of the FiberID reader.

\(^1\)http://antonpotocnik.com/?cat=29
3.2 Zero-Crossing Detector (ZCD)

Once the tutorials on the tools needed to develop the SoC design were completed, work on implementing the prototype design onto the Red Pitaya could begin.

The first roadblock to transitioning from the PicoScope to the Red Pitaya as a DAQ was that the Red Pitaya only has two available inputs, where the Prototype design requires three; two for the horizontal and vertical polarization components of the reflection signal, and one for the resampling signal. Since both reflection signals needed to be received by high performance ADCs, we reevaluated how we received the resampling signal to deal with the smaller number of ADCs the Red Pitaya offers when compared to the previous design.

In the prototype design, all data samples were collected, and only after all samples were collected were the samples not associated with zero-crossings of the resampling signal deemed unusable. These unused samples were a majority of all samples collected, which wastes memory as well. Instead of collecting all data and throwing away the majority, it would be more efficient if the resampling signal could be translated to a digital signal via a zero-crossing detector (ZCD) that could trigger the sampling of the reflection data. This way, the Red Pitaya could read the resampling data as a digital input, not requiring a third ADC input, as well as freeing up memory by only collecting usable samples.

In the following section, three attempts of ZCD protoboard designs are introduced, as well as a fourth ZCD design by Zheyi Yao which was used in the final design for the fiber ID reader.

3.2.1 Design 1

To translate the signal from analog to digital, we thought about using an op amp circuit with a Zener diode. The Zener diode keeps all negative amplitude
portions of the resampling signal at zero, and the op amp drives all non-zero signal values to its maximum. This will create the desired square wave digital signal.

The first chip considered for this design was the Texas Instruments TLV021 comparator chip, whose data sheet even lists zero-crossing detectors as a possible application. The circuit design for the zero-crossing detector for this chip provided by Texas Instruments is shown below in Figure 24.

The design was implemented using a protoboard. The chip was soldered to a test PCB, which in turn was soldered to the protoboard along with an SMA connector, a 100 nF capacitor, and a 3.3 kOhm resistor. A picture of the protoboard is shown below in Figure 25.

The board was given 2.8 V power, similar to the 3.3 V power it could eventually receive from the Red Pitaya. We tested the board with a 200 mV, 0.5 MHz signal, and measured its input and output signals with an oscilloscope. A picture of the oscilloscope output is shown below in Figure 26. It is worth noting that the time delay between the blue input signal experiencing a zero crossing and the yellow output signal rising or falling is significant, which can be better seen in Figure 27. This time delay is also noted in the TLV021’s data sheet, which in hindsight,
3.2.2 Design 2

The next ZCD circuit was designed with the Texas Instruments LT1116. This chip is a comparator similar to the TLV021, but has a much smaller propagation delay of 10-14 ns. With the resampling signal’s frequency observed to be 1.3 MHz, this delay should now only 1.3-1.8% of the sampling period, instead of 27.3-44.2% like in the first design.

A picture of design 2’s protoboard is shown below in Figure 28. The TLV021 chip is attached to a testing board, which is soldered to the protoboard along with a Zener diode and two capacitors, one electrolytic and one non-polarized.

In the oscilloscope outputs shown in Figure 29 below, a marked improvement in delay times is shown. The only thing needing tweaking is the voltage of the output level, which should be closer to the 3.3V that the Red Pitaya takes as an input.

To attempt to fix the voltage level, a voltage divider was added to step down from about 4.8 V to 3.3 V. The newly edited circuit is shown below in Figure 30. The next figure, Figure 31, is a picture of the oscilloscope output of the revised
Figure 26. Output of ZCD design #1 as measured by an oscilloscope protoboard. Immediately apparent is the new shape of the output, which seems to indicate added capacitance somewhere. This could be coming from the overlapping wires on the underside of the protoboard. A better wiring layout could fix this problem, or a design using a different chip designed to output at 3.3 V like the Red Pitaya requires.

3.2.3 Design 3

Since it seemed that in the last design the most glaring issue was that the output was not compatible for the Red Pitaya, for this next design attempt, we chose a chip with the Red Pitaya’s output in mind. This design uses the LTC6752 chip, a 280 MHz, 2.9 s delay comparator with a CMOS output. The protoboard made with this design is shown in Figure 32.

Testing of the protoboard produced the oscilloscope results shown in Figure 33. Unfortunately, there appears to be some DC signal on the input. The zero crossings are now different from one taking an all-AC signal, making this detector
3.2.4 In-lab ZCD Board

Ultimately, in the interests of time, we decided to continue with an already-designed and fabricated zero-crossing detector board designed by Zheyi Yao, a previous researcher in the NEXT Lab. This design uses the LTC6957-3 CMOS output logic converter chip. The board can take the resampling input via an SMA connector, and interfaces with the Red Pitaya using the male version of its extension connector.

3.3 Reflection Collector on Red Pitaya

Once a viable ZCD was obtained, work could begin on the reflection collector functionality for the Red Pitaya. There are two aspects to consider when designing an FPGA SoC, the programmable logic side and the processing system side. The hardware was configured using Verilog, whereas the control of that hardware, as well as the processing of data collected from it, is handled in Jupyter. The hardware design used for data collection can be seen in the block diagram in Figure 34.

Figure 27. Oscilloscope measurement of rising and falling time delays of ZCD design #1

inaccurate. The next step for improving this design would possibly be adding a coupling capacitor right after the SMA connector to remove the DC level.
Figure 28. Picture of ZCD design #2 implemented on a protoboard

Figure 29. Oscilloscope measurement of rising and falling time delays of ZCD design #2
Figure 30. Picture of ZCD design #2.2 implemented on a protoboard

Figure 31. Oscilloscope measurement of rising and falling time delays of ZCD design #2.2
Figure 32. Picture of ZCD design #3 implemented on a protoboard

Figure 33. Oscilloscope measurement of rising and falling time delays of ZCD design #3
3.3.1 DAC and Trigger Signal

The timing of the laser and the Red Pitaya needed to be synchronized. Therefore, a trigger signal is needed. The trigger could occur in two ways: either the laser triggers the Red Pitaya, or the Red Pitaya triggers the laser. Since the inputs for the Red Pitaya are already limited, the Red Pitaya triggering the laser was deemed easier to implement. The laser is triggered when it receives a TTL rising edge after being manually switched to a trigger-awaiting mode. A simple Verilog block was written that sets the output of the Red Pitaya’s DAC channel A to either HIGH or LOW, depending on a GPIO input being either 1 or 0, respectively. It is worth noting that by default, the Red Pitaya’s DAC output range is -1V to 1V. An already modified Red Pitaya from the Next Lab was used in this design, raising the DAC output range to 0V to 2V.

3.3.2 ADC and Data Repackager

The “Red Pitaya ADC IP” written by Anton Potočnik, mentioned earlier in section 3.1.3, was used as the basis of the data acquisition section. The key changes to be made were (1) the ADC should only pass along samples corresponding to
zero crossings, and (2) the ADC data needs to be passed to the processor.

To accomplish the first goal, properties of the AXI stream protocol were taken advantage of. AXI is the data stream type for AMD processors, needing tags "ready" and "valid" to be high to pass along data. Since only data corresponding to zero crossings are valid for this research’s purposes, the signal "valid" was programmed to rise only when the ZCD signal reader block reads a rise or fall from the ZCD board’s output, post hysteresis (more detail on the ZCD board and reader block can be found in the following section).

While "ready" and "valid" tags are the only tags required to trigger an AXI stream in general, some IP blocks require more signals to function. The DMA IP block stands for direct memory access, and is what allows data to write to and read from the processor’s memory. Since the memory needs to know how much data there is to be written in order to store it properly, the DMA requires a "last" signal as well, to denote the end of a data stream or packet of data. The Data Repackager section of the FPGA design, in addition to splitting both ADC inputs into different streams, is responsible for generating a "last" signal after streaming 131,072 samples. This number is a power of 2, which simplifies counting using circuits, as well as being similar in length to the number of zero-crossing samples used in the prototype design.

3.3.3 ZCD and Hysteresis

The ZCD board was first tested with simple sinusoid signals generated by a function generator. One signal would go into both ADC inputs on the Red Pitaya, and the other signal would serve as the input to the zero crossing detector. This ZCD input signal would then sample the ADC signal at a sampling rate twice that of its frequency (since zero crossings appear twice per one period). For example, a 1 MHz ZCD input would sample at 2 MSps. A picture of the testing setup is
At first there were mixed results depending on the ADC and ZCD input signals. When the sampling rate was above 2 MSps, the recorded signals appeared to be clean sinusoid signals. But when the sampling rate was lower due to a lower frequency ZCD input, suddenly the signals recorded by the ADCs were choppy and no longer looked like clean sinusoids. A zoomed-up plot of the faulty signals can be seen in Figure 36, depicting the signal being accurate for some lengths of time (in blue) and inaccurate for others (in red).

The cause of the faulty looking signal was found using the FPGA’s system debugger. The ZCD board’s output would occasionally flicker between high and
These glitches completely ruin the sampling rate’s fidelity, which is needed to eventually transform reflection data. To remedy this, a variable hysteresis setting was added to the ZCD Signal Reader section of the FPGA design. Hysteresis is a small threshold or delay between states that must be met before recording a state change. After implementing a programmable hysteresis delay that waits for a Jupyter-defined number of samples after a rise or fall to ensure a zero crossing has really occurred, this issue was fixed and choppy signals were no longer recorded.

### 3.4 Similarity and Identification with Cross-correlation

In this section, the similarity calculation and comparison functionality from the prototype design is reproduced on the SoC. Part of this transition includes moving the cross-correlation function from the processor to the programmable logic, while the rest is taken care of in Jupyter.

#### 3.4.1 Xcorr IP block

Once the FPGA was programmed, we took advantage of the parallelization potential of the cross-correlation function to build a cross-correlation block with Vivado HLS using systolic arrays. This implementation functions like a dot product with a buffer region—that way, when input ranges don’t line up while attempting to compare two fiber signatures, the xcorr block still leaves 13 samples of room to still successfully identify the fiber, given it is a match. To incorporate both even and odd shifts, two arrays stream the input signals in at both ends, whose structure...
is illustrated in Figure 38. Developing the xcorr block allowed for optimizing the performance to be as parallel and fast as possible. This second FPGA design is loaded onto the FPGA by prompting from the Jupyter notebook to upload the next bitstream, and is displayed in Figure 39.

3.4.2 Replicating the Similarity Function

One last addition was made to the embedded system version of the design. While collecting data sets for testing, it was found that certain collections would have a poor performance when compared to the average collection. Meaning, even when comparing two identical sections of fiber, the system would fail to identify them as a match. To combat this, we also compared an additional region, called
Figure 40. Reflection vs distance data, with the calibration region colored in yellow and different sections of test fiber represented in other colors.

The calibration region. This region falls on the lead connecting the whole system to the fiber under test. This region is always present, meaning two different readings should always return a high similarity score, regardless of the fiber under test. If comparing calibration regions returns a low score, we know not to consider that set’s identification scores and to treat it as an outlier. This region and the sections of fiber to be compared is illustrated in Figure 40.
CHAPTER 4
System Evaluation

With the embedded system design completed, we chose to test the accuracy of the system as a fiber identifier by generating ROC curves. A receiver operation characteristic (ROC) curve is a common measure of the accuracy of a system, particularly in areas of identification. It does so by comparing the system’s true positive and false positive rates, with a perfect identifier appearing as a sharp right-angled curve outlining the top left corner of the plot, where the ideal is a false positive rate of 0 occurring at the same time as a true positive rate of 1.

Data collected for plotting the ROC curves was generated in the following manner: a group of 10 repeated readings of the same optical system setup is called a set. Nine sets were collected on different days and times while keeping all other parameters constant, as well as sets taken on the same day but using different hysteresis buffer lengths, all other parameters kept constant. After taking similarity comparisons of each set, the results are graphed as ROC curves below in Figures 41 through 52.

There is a general trend for both figures: the performance of the system will decrease as the length of test fibers being compared decreases. This makes sense, as there is less information of the fiber’s unique reflection pattern when the lengths are shorter. However, at all fiber test lengths, most ROC curves had an Equal Error Rate (EER) less than 5%.

Figures 41-43 (Figures 44-46 are the same but zoomed in to the top left corner) investigate the performance of the system with different hysteresis buffer lengths when comparing different section lengths of test fiber. The set with hysteresis buffer of 1 performed worse across all test fiber lengths. Since a hysteresis buffer
Figure 41. ROC plot comparing hysteresis length settings for 30 cm DUT

Figure 42. ROC plot comparing hysteresis length settings for 10 cm DUT

Figure 43. ROC plot comparing hysteresis length settings for 3 cm DUT
Figure 44. ROC plot comparing hysteresis length settings for 30 cm DUT, zoomed in to a 5% EER

Figure 45. ROC plot comparing hysteresis length settings for 10 cm DUT, zoomed in to a 5% EER

Figure 46. ROC plot comparing hysteresis length settings for 3 cm DUT, zoomed in to a 5% EER
Figure 47. ROC plot comparing collections from different days for 30 cm DUT

Figure 48. ROC plot comparing collections from different days for 10 cm DUT

Figure 49. ROC plot comparing collections from different days for 3 cm DUT
Figure 50. ROC plot comparing collections from different days for 30 cm DUT, zoomed in to a 5% EER

Figure 51. ROC plot comparing collections from different days for 10 cm DUT, zoomed in to a 5% EER

Figure 52. ROC plot comparing collections from different days for 3 cm DUT, zoomed in to a 5% EER
length of 1 is the same as having the hysteresis setting turned off, this implies that implementing this hysteresis buffer was necessary to ensuring good performance of the fiber ID system. From the results of the 3 cm plot, the best hysteresis setting of the system is a length of 2, which was used as a constant parameter in the next set of figures.

In Figures 47-49, (see Figures 50-52 for zoomed-in version), the performance of the system varying by time was measured. It can be seen that, excluding the set taken the morning of 2023-03-06, data collected on different days still performed similarly. This is reassuring, since it means sensitive parts of the system like the polarization of exposed fiber in the optical circuit are not changing significantly between days.
CHAPTER 5

Discussion

The goal of this research was to make a fiber ID reader with small form factor as a way to read and compare unique fiber signatures to support their role as a type of physical unclonable function. This goal was achieved for the data acquisition element of the system. The final system on chip implementation produced RoC curves with EER rates within 5%.

While the desired outcome of a fiber ID reader using a system on chip was achieved, there are some improvements to be made that could be beneficial when repeating the development process of this research. If given more time, work on the custom zero crossing detector board would continue till a new board is designed and manufactured, instead of moving on with the ZCD already available in the lab. Another change that could be made is revising the testing script, so it finishes faster. Dividing the DUT area into slices as it is done currently means there are more slices the shorter the slice length is. Instead, when replicating this research, the testing script should pick a fixed amount of test slices regardless of slice length. The speed could also be increased by implementing the FFT operation using the hardware of the FPGA, instead of computing on the processor side.

Lastly, regarding next steps for this research, the form factor of the overall fiber ID system could be made smaller. Focus would then be centered on reducing the size of the optical system, for example using a smaller and or cheaper laser system, or by reducing the length of the fibers in the optical circuit.
LIST OF REFERENCES


BIBLIOGRAPHY


