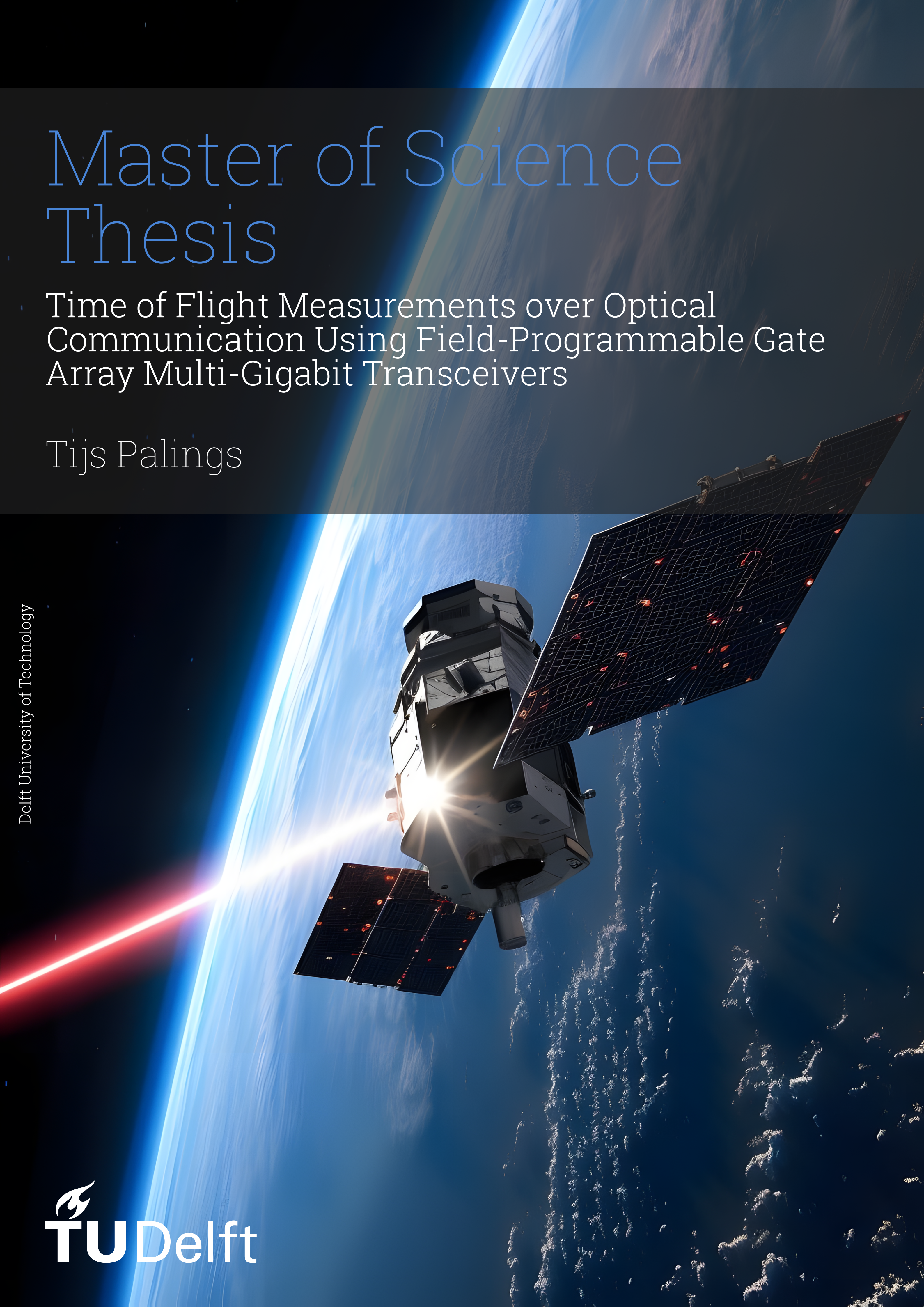


Master of Science Thesis

Time of Flight Measurements over Optical Communication Using Field-Programmable Gate Array Multi-Gigabit Transceivers

Tijs Palings

Delft University of Technology

Master of Science Thesis

Time of Flight Measurements over Optical Communication Using Field-Programmable Gate Array Multi-Gigabit Transceivers

by

Tijs Palings

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on 30th of November 2023.

Student number:	4657098	
Project duration:	March 23, 2023 – November 30, 2023	
Thesis committee:	Dr. J. Guo,	TU Delft, Chairman
	Dr. S. Speretta,	TU Delft, Daily Supervisor
	Dr. Ir. J. Bouwmeester	TU Delft, Examiner

Cover: laser satellite communication concept art by TS2 space

Preface

This thesis report represents the final deliverable in order to obtain a Master of Science Degree in Aerospace Engineering with as specialization Space Engineering at Delft University of Technology. The reports contains the design, tests and analysis of several optical communication systems. They concern systems built with a field-programmable gate array (FPGA) for communication and signal processing and acquisition. These systems aim to incrementally increase in complexity such that limits can be identified as the thesis progresses and points of improvement can be implemented or identified.

It is the culmination of over 6 years worth of material and skills acquired at the TU Delft. Before the thesis I had little to no practical experience with FPGAs, printed circuit boards or laser communication for that matter. But with the set of skills I've managed to acquire over the years, I've stepped out of my comfort zone and approached the problem head on from the beginning. I was a bit overwhelmed at first, but managed to break down the problem to tackle it bit by bit. Which is what a good engineer should be able to do in my opinion, and I feel like I've managed to do exactly that in this thesis.

I would like to thank my dad for always supporting me whenever and wherever I needed, Marceline for all the support and kindness she provided me day and night, making the sun shine just a little brighter.

A special thanks to TNO for finding time to let us do measurements with their test setup in Scheveningen.

Most of all I would like to thank Stefano Speretta and Rashika Jain for their critical feedback, the weekly meetings and their time and effort they put in me and my work during the thesis. I genuinely enjoyed the collaboration and appreciate the opportunity you have provided to me. I'd approached you with the wish to do a more practical thesis and practical it was. Thank you for the opportunity.

Tijs Palings
Delft, November 2023

Summary

Since the early ’80s, continuous efforts to realize laser communication for deep-space communication began. Due to the immature and inefficient nature of these then-new technologies, they were barely implemented. Despite this, researchers and engineers recognized their potential, leading to a surge in research activity over the years. This effort has since yielded significant progress in satellite communications, with some systems proven in flight.

Laser communication systems achieve orders of magnitude higher data rates compared to their Radio Frequency (RF) counterpart, while being more compact and requiring less power to function [1] [2]–[4]. These high data rates are necessary for modern, data-heavy systems in deep space, such as synthetic aperture radar, multispectral/hyperspectral imaging, and high-definition video transmission [5]. These systems require between 1 Mbps and 1 Gbps for acceptable operations [5].

Part of the communication subsystem is performing range measurements for orbital determination. Normally this is done using multiple RF frequencies, where a dedicated ranging signal is sent for a period of time, in which several range measurements are obtained. This means that the normal data stream is set to a halt for the duration of this period. For optical systems of the other hand this is not the case. The necessary information required to perform range measurements can be embedded into the normal data structure [6] instead.

In terms of optical communication, there exist Commercial Off-The-Shelf (COTS) components that are often used in networking that could be used to mimic, test and parameterize satellite laser communication. With the use of an Field Programmable Gate Array (FPGA), its on-board Multi-Gigabit Transceiver (MGT) and Small Form-factor Pluggable (SFP) modules, a satellite laser communication system can be reconstructed of which the performance can be parameterized that are comparable to what would be achievable in orbit. Furthermore it provides the opportunity to research constraints, outside influences and parameters that influence the accuracy and precision of such systems. This thesis aims to study these exact features by incrementally increasing complexity, starting from a relatively basic system to a more complex system that fully embeds optical communication, identifying limits and constraints as complexity increases.

To start, a free-space laser ranging experiment is conducted with a test setup provided by and in collaboration with Toegepast-Natuurwetenschappelijk Onderzoek (TNO). It is important to conduct this experiment as it shows the effects of free-space on the Signal to Noise Ratio (SNR) and what could be expected when performing range measurements though an atmosphere. This is a passive laser ranging experiment as no real communication is involved, rather a set of 40 pulses lasting a total of 8 µs is transmitted towards a retro-reflector. The reflection is captured and based on the time of arrival and the time of transmission, the Time of Flight (ToF) can be measured. The retro-reflector is located at a distance of roughly 2442 m from the transmitter and receiver. From the ToF measurements a distance between 2459.73 m and 2463 m were observed, the standard deviation of these measurements was 1.26 m and a measurement uncertainty of 6 m. For this test, there was no time for proper calibration, hence the large residual on the range measurement.

This test demonstrated the basic constraints in retrieving a range measurement. As no communication is embedded, the accuracy depends mainly on the accuracy by which the time of arrival and transmission of photons can be registered. To improve this measurement accuracy, an FPGA was programmed that is capable of much higher transmission speeds and sampling frequencies. The first constraint from the previous test however is tackled by conducting a calibration of the internal delay of the system. An internal delay of 15.83 ns was measured with a standard deviation of 275.5 ps with the use of Digital Dual Mixer Time Difference (DDMTD). This is remarkable, as this allows for a significantly higher measurement resolution of 0.01 ns, compared to the 5 ns sampling period. During the execution of the test, it could quickly be concluded that these values do not accurately represent the internal delay of the MGT.

Although the General Purpose Input Output (GPIO) from the previous experiment has proven to be stable and easy to control, the experiment that follows must include the MGT firmware. Thus the next steps are aimed at quantifying the internal delay caused by the MGT as well as perform range measurements on optic fiber cables with the use of FPGA transceivers. It was shown that the internal delay of the MGT was not constant, rather it consists of a deterministic and stochastic part, due to a bug embedded in the firmware. Due to the time limits imposed on the thesis, the bug could not be resolved in time. Nonetheless it could be quantified and it was determined that it is dependent on the line rate of the FPGA. Range measurements are conducted in 5 different optic fiber lengths, with three different line rates. The results of these tests are displayed in table 1.

Table 1: Summary of the range measurements results as a function of line rate

r_{target}(m)	**μ(m) @ 500 Mbps**	**μ (m) @ 1 Gbps**	**μ (m) @ 1.25 Gbps**	**σ_{fiber} (m) @ 500 Mbps**	**σ_{fiber} (m) @ 1 Gbps**	**σ_{fiber} (m) @ 1.25 Gbsp**	**σ(ns) @ 500 Mbps**	**σ (ns) @ 1 Gbps**	**σ (ns) @ 1.25 Gbps**
10	9.927	8.758	9.804	3.228	1.591	1.146	15.804	7.791	5.611
20	19.652	18.802	19.891	3.199	1.588	1.203	15.666	7.775	5.893
30	29.847	28.962	29.778	3.243	1.585	1.217	15.884	7.759	5.957
40	39.968	38.911	40.024	3.331	1.594	1.289	16.311	7.807	6.312
50	50.064	50.178	50.001	3.268	1.657	1.275	16.003	8.116	6.245

It could be concluded that the average standard deviation on a range measurement is dependent on the line rate, which is 3.235 m for a line rate of 500 Mbps, 1.618 m for a line rate of 1 Gbps and 1.226 m for a line rate of 1.25 Gbps. These standard deviations are a direct result of the bug within the firmware likely due to misconfiguration of the MGT firmware, causing occasional resets in the Gigabit Transceiver Type-H (GTH), randomizing the phase difference between the transmitter and the receiver clock. It is expected that these standard deviations drop significantly whenever the phase lock is constant during a transmission.

These systems are simulated to fly on Lunar Meteoroid Impact Observer (LUMIO) and Lunar Pathfinder using an orbital simulation made by [7] using Tudatpy kernels. The aforementioned standard deviations are substituted as noise values in in the simulation and the positional error estimates were recorded. The limitation of the simulation were however that the high measurement rates of the FPGA configurations, which are 25, 50 and 62.5 million measurements per second with the current data stream, were not attainable due to the discretization errors that occur on the floating point values of the small time step values required. To avoid this, the standard deviation is divided by the square root of the number of samples per measurement time step. For a measurement times step of 5 min and a measurement rate of 10 Hz, the standard deviations were divided by $\sqrt{3000}$. An integration time step of 5 s. The positional error converged to below 10 m after roughly 10 days for a line rate of 500 Mbps and 9 days for the line rates of 1 and 1.25 Gbps.

The main constraint that has been identified, besides the obvious system bug, is the rate at which the signals can be observed within the FPGA. The second experiment trivially circumvents this by using DDMTD to allow for a resolution smaller than the sampling clock permits. Ideally this is implemented into the system where the MGT is used for communication. A proposal for this implementation is made to finalize the system to create a system of two FPGA terminals capable of performing synchronous laser communication ranging measurements. A DDMTD is implemented on both terminals responsible for the sub-picosecond timing distribution. This device, often used by setups requiring picosecond or lower levels of phase stability are often used by Conseil Européen pour la Recherche Nucléaire (CERN) [8]–[10]. Implementing this Printed Circuit Board (PCB) within the design, under the assumption that the current system as is is fully debugged, accurate phase comparisons can be conducted and the timing of the system can be properly distributed to measure the range more accurately.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AXI	Advanced eXtensible Interface	38
BER	Bit Error Rate	77
BRAM	Block Random Allocated Memory	39
CCSDS	The Consultative Committee for Space Data Systems	9
CCR	Corner-Cube Reflector	vii
CDC	Clock Domain Crossing	viii
CDR	Clock Data Recovery	68
CERN	Conseil Européen pour la Recherche Nucléaire	iii
COTS	Commercial Off-The-Shelf	ii
CPU	Central Processing Unit	41
CSM	Codeword Synchronization Marker	10
DAQ	Data Acquisition	38
DDMTD	Digital Dual Mixer Time Difference	ii
DMTD	Dual Mixer Time Difference	11
DFF	D-type Flip Flop	12
DRP	Dynamic Reconfiguration Port	64
EDFA	Erbium-Doped Fiber Amplifiers	20
FID	Frame Identifier	73
FIFO	First In First Out	66
FMC	FPGA Mezzanine Card	38
FPGA	Field Programmable Gate Array	ii
FSM	Fine Steering Mirror	20
GPIO	General Purpose Input Output	iii
GRACE-FO	Gravity Recovery and Climate Experiment Follow-On	8
GTH	Gigabit Transceiver Type-H	iii
HDL	Hardware Design Language	37
HPE	High Photon Efficiency	9
ICW	Interleaved Codeword	10
I/O	Input/Output	3
IP	Intellectual Property	xi
JTAG	Joint Test Action Group	37
KNN	K-Nearest Neighbors	34
LHC	Large Hadron Collider	110
LLCD	Lunar Laser Communication Demonstration	10
LLR	Lunar Laser Ranging	8
LPC	Low-Pin Count	64
LPF	Lunar Pathfinder	x
LUMIO	Lunar Meteoroid Impact Observer	iii
LSB	Least Significant Bit	87
MGT	Multi-Gigabit Transceiver	ii
MMCM	Mixed-Mode Clock Manager	40
MSB	Most Significant Bit	86
MUX	Multiplexer	65
NASA	National Aeronautics and Space Administration	110
OOK	On-Off Keying	63
OPLL	Optical Phase-Locked Loop	2
PCS	Physical Coding Sublayer	17

PCB	Printed Circuit Board	iii
PCIE	Peripheral Component Interconnect Express	64
PDU	Protocol Data Unit	10
PISO	Parallel Input Serial Output	63
PLL	Phase-Locked Loop	viii
PMOD	Peripheral Module	viii
PRBS	Pseudo Random Bit Sequence	69
PMA	Physical Medium Attachment	17
PPM	Pulse Position Modulation	10
QPLL	Quad Phase-Locked Loop	67
RA	Ranging Ambiguity	6
RCID	Ranging Codeword Identifier	ix
RCW	Ranging Codeword	vii
RD	Running Disparity	69
RF	Radio Frequency	ii
RR	Ranging Resolution	6
RSM	Range Synchronization Marker	10
RTLT	Round Trip Light Time	6
SC	Spacecraft	71
SFP	Small Form-factor Pluggable	ii
SIPO	Serial Input Parallel Output	63
SLP	Satellite License Plate	3
SLR	Satellite Laser Ranging	8
SMA	SubMiniature version A	64
SMCW	Synchronization Marked Codeword	vii
SNR	Signal to Noise Ratio	ii
TIE	Time Interval Error	11
ToF	Time of Flight	ii
TNO	Toegepast-Natuurwetenschappelijk Onderzoek	ii
UART	Universal Asynchronous Receiver/Transmitter	61
VHDL	Virtual Hardware Design Language	4
VIO	Virtual Input Output	74
XCLK	PMA Parallel Clock	66

Symbols

Symbol	Definition	Unit
B	Bandwidth	[Hz]
B_n	Code loop bandwidth	[Hz]
c	Speed of Light	[m/s]
C	Capacity	[bit/s]
d	Distance/Range	[m]
e	Partial pressure of water vapor	[mbar]
f	Frequency	[Hz]
f_c	Carrier frequency	[Hz]
R_{line}	Line Rate	[bps]
S	Symbol length	[bits]
T	Period	[s]
τ	Round Trip Light Time Delay	[s]
ψ	Phase	[rad]

I

General Introduction

1 Introduction

In this chapter, first the background and the rationale for this thesis topic will be presented in section 1.1. Next the research questions and objectives will be introduced in section 1.2. In section 1.3 the hardware and software are briefly described. Finally in section 1.4 the outline of the report structure will be presented.

1.1. Background and rationale

In the early 80s the first continuous efforts towards realizing laser communication for deep-space communication were made. Due to the immature and inefficient nature of those then new technologies, they were barely implemented. Despite this, researchers and engineers recognized its potential, leading to a surge in research activity over the years. This effort has since then yielded significant progress in the field of satellite communications to where some systems have been proven in flight.

Laser communication systems achieve orders of magnitude higher data rates compared to their RF counterpart, while simultaneously being more compact, requiring less power and having a narrow beam divergence allowing for longer distance communication [1] [2]–[4]. These benefits, especially a higher data rate, are required to take more modern and data heavy systems to deep-space, such as synthetic aperture radar, multispectral/hyperspectral imaging and high-definition video transmission [5]. These systems require between 1 Mbps and 1 Gbps for acceptable operations [5].

The inclusion of a RF telecommunication system has been a standard practice for every spacecraft in existence. This system holds significant importance in facilitating mission-critical functions such as the transfer of scientific data and operational system information. Moreover, it plays a pivotal role in navigation and trajectory planning by providing an approximate radial distance measurement between the ground station transmitter and the spacecraft receiver, thereby aiding orbital estimations.

The prevailing direction in deep-space communication involves a gradual shift from RF technology to optical technology. Consequently, optical systems must possess the capability to perform ranging measurements with equal or greater accuracy compared to RF systems. Fortunately, optical communication offers the advantage of higher data rates, enabling enhanced ranging precision. Additionally, with optical systems no separate ranging signal can be multiplexed with the regular data stream as with RF ranging. Instead the data required is embedded in the data stream itself.

Optical systems continue to present challenges due to their inherent complexity, often necessitating dedicated hardware components like an Optical Phase-Locked Loop (OPLL), which can be susceptible to various sources of noise [11]. Furthermore, the inclusion of an OPLL imposes additional hardware constraints, such as stringent thermal requirements. In light of these considerations, it becomes advantageous to explore avenues for enhancing performance without escalating system complexity. One approach involves delving deeper into the electronics responsible for ToF measurements. With these systems such as FPGAs it may become possible to achieve high ranging accuracy without relying on coherent optical systems typically required for millimeter-level accuracy.

In standard commercial networking practices the use of optical communication is being standardized with devices such as SFP modules, which contain all the hardware required to transmit and receiver laser signals. These modules are used in standardized MGT slots that are used to drive these modules, creating an easy and robust way of to approach the complex problem that is laser communication.

To test the laser communication ranging performance of these COTS components an FPGA can be utilised, creating a easy to operate system to test various data protocols and data rates. Moreover the it is a scalable system through its variety of inputs and outputs for which components can be bought or designed.

This thesis endeavors to construct a surrogate optical communication system akin to a satellite by utilizing an FPGA. The primary objective is to scrutinize the ToF measurements generated by this system, evaluating its accuracy and precision. Additionally, the study seeks to discern the system parameters that exert influence on the measurements and explore avenues in to improving the measurements.

The final deliverable for the thesis is a validated system with a documented description of its design, that has the potential to be used in future test setups through its standardized Input/Output (I/O). Moreover, points of improvement are listed and recommendations for future work are provided.

1.2. Research Questions and Objective

This section will provide an overview of the problem at hand. Firstly, the primary research objective will be outlined, followed by the main research question. Additionally, a series of sub-questions will be presented, designed to guide the research process by systematically exploring system complexity, quantifying timing delays, and examining system properties.
Research Objective:

The objective of this study is to investigate optical laser ranging measurements of various systems such that certain system parameters can be correlated to a degree of timing delay.

Research Question:

What ranging performance can be achieved in a satellite laser communication system, with the use of commercial off-the-shelf Multi-Gigabit Transceivers on an Field Programmable Gate Array?

This question will be deconstructed into the following sub-questions to aid in the answering the main research question:

1. How can laser ranging be achieved using a communication system?
2. What methods can be utilized to generate a stable and consistent signal on an FPGA?
3. How can the internal propagation delay of the FPGA be quantified?
4. How does the line rate of the communication influence the range measurements in the FPGA?
5. How can the performance of the system be improved to millimeter-level?

1.3. Hardware & Software

In order to answer the research questions listed in section 1.2 two types of hardware are used. To answer the first sub-question, the test setup from TNO is used. The remaining sub-questions are answered by tests using the AMD Kintex UltraScale FPGA KCU105 Evaluation Kit operated by custom firmware.

The test in Part II is conducted using a test setup from TNO [1]. This setup is built to validate the Satellite License Plate (SLP) identification method in a free-space environment. An SLP is passive component that can be attached to any satellite. It is a wavelength-selective retro-reflecting tag that has a unique spectral signature. From a ground station, multiple laser beams can be transmitted into the vacuum of space where one or more satellites are expected to be and based on the retroreflected signal the satellites can be identified. This technology is especially useful with the current miniaturization of space as satellites are becoming smaller and are being released in swarms.

[1] `https://www.tno.nl/en/newsroom/insights/2023/08/successful-ground-test-shows-potential/`, [accessed on 1-11-2023]

This system is not built for the purpose of determining range. Nonetheless it can be used for doing exactly this. In basic terms this system consists of two parts, a retro-reflector and a transmitter/receiver terminal separated by a distance in free-space. By observing the transmission and reception of a laser signal the distance can be computed using software written in Python. The setup and methodology of this test is described in more detail in chapter 4.

For the tests in Part III the AMD Kintex UltraScale FPGA KCU105 Evaluation Kit is used. The FPGA board is illustrated in figure 1.1.

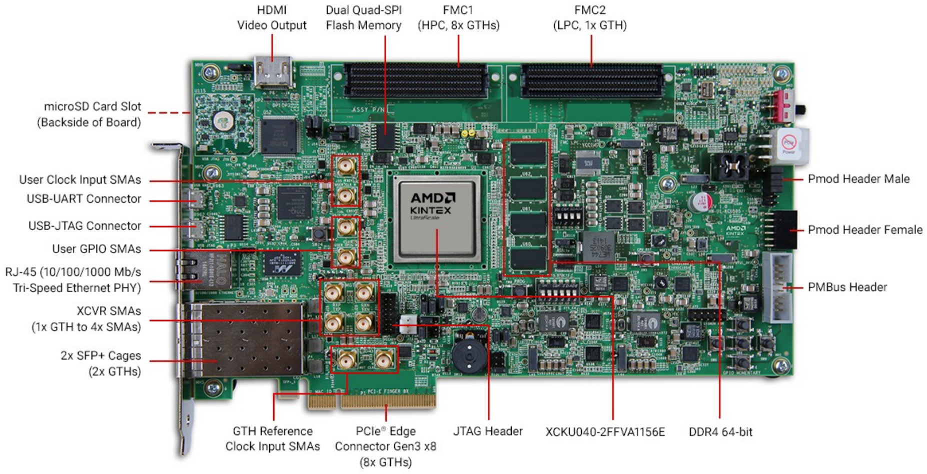


Figure 1.1: AMD Kintex UltraScale FPGA KCU105 Evaluation Kit layout [2]

The software used to program the FPGA is the included software Vivado ML edition 2021.2. The software includes pre-coded firmware pieces called Xilinx IP as well as other Virtual Hardware Design Language (VHDL) and Verilog syntax packages. Throughout the thesis a wide variety of IP will be used to construct the firmware for the FPGA. It is common practice to so so as these firmware blocks are sophisticated and easy to operate.

These IPs include clocking managers and Multi-Gigabit Transceiver firmware as an example. The creation and usage of IP is not limited to Xilinx, MATLAB has add-ons that can be used to create MATLAB IP that can be implemented as VHDL or Verilog code inside of Vivado. IPs created by MATLAB are also used to capture signal data from the FPGA.

As the name suggests, IP is property that belonging to the party that has created it, even when used in designs made by students, companies and institutions. The blatant distribution of this IP is strictly prohibited and protected by law. It can be regenerated using scripts that are only functional when the party possesses a valid license to regenerate the IP. Hence the firmware will not be included in this thesis work but merely a description of its inner workings. Furthermore it will be stored on an internal database from the TU Delft that can only be accessed by members within the institution.

1.4. Report structure

In order to structure the thesis accordingly it is separated into three parts. Part I pertains to the general introduction and subject matter discussed in this thesis. Part II pertains to everything related to the free-space laser ranging measurements made with the test setup from TNO. Part III pertains to all matter to do with the design and tests with the FPGA.

The tests are listed in the chronological order of events. Each have been assigned its own chapter that more or less follows the same structure where applicable. The findings of each test are taken

[2] https://www.xilinx.com/products/boards-and-kits/kcu105.html, accessed on [8-6-2023]

into consideration and expanded upon in the test that follows, incrementally adding to the system complexity. Hence each chapter has been provided its own conclusion, discussion and recommendation section. The secondary reason for structuring it this way is to avoid one large conclusion with all important findings scattered throughout such a chapter. Instead everything important can be found in the individual chapters.

First in Part I the introduction is presented in chapter 1, as well as the research questions, used hardware and software and this report structure.

Next in Part II the free-space laser ranging experiment using the test setup from TNO is presented. In chapter 4 first the objectives and goals of the test are discussed. Next the system design is discussed, followed by a detailed description of the method used in obtaining and analysing the test results. Next the verification and validation of the processing software is presented, and the chapter is finalized by the presentation of the results, discussion and recommendations.

In Part III experiments conducted with laser communication ranging using an FPGA. In chapter 5 the experiment aimed at familiarizing with the FPGA and measuring its internal delay is presented. Following the same structure, first the system design is presented followed by the method, verification, validation, results, conclusion, discussion and recommendations. The next chapter in the same part is chapter 6, which aims at performing the first range measurements using optical fiber loopbacks and analysing the accuracy and noise of these measurements. It adheres to the same structure as mentioned before.

Finally under the same part a proposal is made on a full end-to-end two terminal communication system with the implementation of an DDMTD in chapter 7. Because only the PCB design and proposals are made it deviates from the aforementioned structure. First the objective and goals of the chapter are listed followed by the system design. The chapter is then concluded by a conclusion and recommendations section.

The thesis is concluded with a general conclusion in chapter 8 that provides a high level overview of the findings as well as concretely summarize the answers to the research questions.

2

Literature Review

In this chapter, parts of the literature study [12] are presented to include all necessary parts to understand the material discussed in this thesis. First in section 2.1, the prerequisites to understanding laser communication ranging are presented. Next, passive optical ranging systems are discussed in section 2.2, followed by active optical ranging systems in section 2.3.

2.1. Optical telemetry ranging prerequisites

The general driving principle behind laser communication ranging is based on the definition that photons travel from point A to point B with a constant speed of c which is equal to 299 792 458 m/s in an ideal vacuum, Round Trip Light Time (RTLT). If the time delay between transmitting and receiving a photon can be determined accurately enough, one is able to determine the range using the following trivial equation:

$$d = \frac{c \cdot t}{2} \tag{2.1}$$

Where t is the total time it takes for a photon to travel from point A to B and back to A in some fashion.Equation 2.1 only refers to the most trivial form of measuring distance between two objects. It primarily assumes that the distance between the two objects is near constant and that light travels in a straight path through space-time, neglecting relativistic effects.

Naturally, this assumption does not always hold well for space-based applications as spacecraft move with velocities upward of several kilometers per second relative to the observer in case of Earth-probe communication. The change in distance with respect to time over the communication time cannot be neglected. Furthermore gravitational, atmospheric and effects related to spacecraft dynamics influence accuracy as well as clock frequency jitter for example.

2.1.1. Range Ambiguity and Range Resolution

These are two important concepts that apply for ranging applications in general. The Ranging Ambiguity (RA) is the minimum distance that cannot be distinguished from multiples of that distance by the ranging system. Ranging Resolution (RR) refers to the minimum distinguishable unit of measurement of a particular ranging system.

RA exists in periodic signals. These signals repeat the same value after a fixed time interval, which can be the case for optical telemetry ranging. An example of such a signal is further elaborated upon in subsection 2.3.1. Here an identifier is embedded in the data structure of which the order of progression is known, such as a counter that progresses in increments of one. There are however only a limited number of integers that can be used as the identifier, after an amount of time these values are bound to be repeated. If such a repetition occurs on the transmitter cycle before the identifier of the same value is received, there is RA. This directly translates to a RTLT that needs to be smaller than the total repetition period of the signal.

This may may have to be resolved depending on the communication principles of a system as well as total RTLT. The following is assumed:

- The communication between two terminals is continuous and the signal has a certain repetition frequency f_p.
- The period of the repetition frequency is smaller than the RTLT.

From [6] the following is described as the range ambiguity:

$$d_{RA} = \frac{c}{2 \cdot f_c} \tag{2.2}$$

This equation uses the carrier frequency f_c in its calculation assuming that only the optical carrier is transmitted and re-transmitted. In practice it is unreasonable to assume this as transmitting in this manner is wasteful to the bandwidth and no system information is telemetered. Hence the same function can be applied to a practical signal consisting of blocks of information telemetered at a specific repetition frequency f_p by substituting f_c from Equation 2.2 with this frequency.

Usually the ephemeris can be used to infer the likely range to resolve the ambiguity. However there is little research dealing with this ambiguity problem for laser communication ranging [11] and it is difficult to do so. In cases where the RTLT is sufficiently small such that the repetition period is smaller, no ambiguity exists.

In some systems there exist technical limitations that have an impact on the RR including [6]:

1. Maximum instrument sampling rate
2. Analog-to-digital converter timing jitter
3. Group delay effects in transmission lines caused by temperature changes

Most importantly, for deep-space missions, it is the noise that ultimately determines the limits of the ranging system [6].

2.1.2. Phase definition

For most laser communication ranging systems, the phase of the signal is what ultimately determines the RTLT measurement. From the basic principles of quantum mechanics, light is a superposition of classical particles and a traveling wave and may be interpreted as such depending on the application and its limits. Generally the phase of a periodic signal, $x(t)$ is defined by the following[6]:

$$\psi(t) = 2\pi \frac{t - t_0}{T} \qquad \mod 2\pi, \ t \geq t_0 \tag{2.3}$$

Where t_0 is an arbitrary point in time associated with the start of a new periodic cycle. This definition of phase will be applicable for most of the current ranging technologies.

2.2. Passive optical ranging

The primary technology behind passive optical ranging are so called retro-reflectors, also called corner-cube prisms or CCR. These reflect incoming light back comparable to a mirror, however in the case of the mirror light is reflected depending on the incident angle. For CCRs, the light is reflected back the direction from which it originated irrespective of the incident angle of the incoming light. Any laser signal sent to a retro-reflector will therefore travel the approximately the same path back to the transmitter. This has the benefit of only requiring one transmitter and one receiver, meaning only one accompanying system will have to be calibrated.

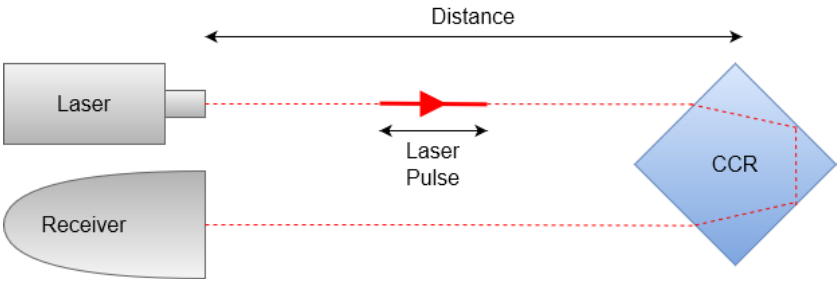


Figure 2.1: Simplified schematic of a passive optical ranging system, CCR is a corner-cube reflector.

A basic illustration of such a system is presented in figure 2.1. Here a laser is responsible for generating laser pulses at a particular frequency with a set duty cycle. A laser pulse is transmitted after which there is a period of silence where a reflected response is expected. Based on the known time of transmission and the time of arrival a ToF can be computed

This type of system is relatively simplistic and useful for a varying set of applications. The APOLLO mission placed several retro-reflectors on the surface of the moon for the Lunar Laser Ranging experiment, these reflector panels contained several CCRs, as illustrated in figure 2.2. The current achieved accuracy range for the retro-reflectors on the Moon is 1.1 mm[13]. The limit to this accuracy is the stability of the clock used to measure the time of arrival and time of transmission of the photons. Moreover retro-reflectors are being used for the Gravity Recovery and Climate Experiment Follow-On (GRACE-FO) mission for both as part of its inter-satellite ranging system as well as ranging with the Satellite Laser Ranging (SLR) global ground station[14].

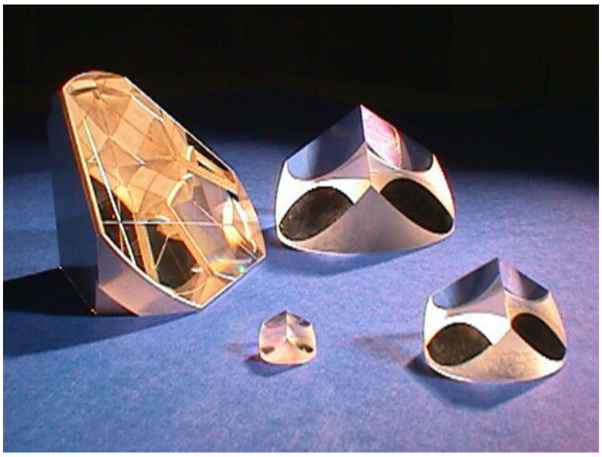


Figure 2.2: LLR retro-reflectors or corner-cube-prisms used for the APOLLO mission[1].

2.2.1. Signal loss

The use of retro-reflectors is a robust method to measure range for a relatively stable target, i.e. the Moon. However this method is prone to a variety of uncertainties that are introduced by the workings of the method itself. The main challenge for this method, before laser ranging can be accomplished in the first place is the detection of returning photons sent from the laser pulse itself.

A laser beam has a certain divergence ranging a several arc seconds depending on the laser an telescope hardware. This results in a photon spread of several tens of meters to several kilometers depending on the distance of the target. The retro-reflector of the target reflects incident beam with a certain divergence as well, causing further spread of the photons to a spot tens of kilometers on Earth. For the Lunar Laser Ranging (LLR) experiment, the resulting spot on the Moon has a diameter of approximately 7 km, photons hitting the retro-reflectors are reflected back to Earth with a divergence of roughly 10 arcsec.

[1] https://tmurphy.physics.ucsd.edu/apollo/lrrr.html

This results in a spot of 20 km diameter and, depending on the receiver size, can only catch a fraction of the photons. A 1 m diameter telescope would only be able to receive $2 \cdot 10^{-9}$ of the photons [15]. In addition to other losses such as quantum efficiency, mirror reflectance, optical performance under thermal stress, and velocity aberration, result in a signal loss of roughly 10^{-21} [15].

The massive signal loss that occurs makes it impractical to use retro-reflectors for precise and dependable laser ranging from Earth to probe for distances beyond the Moon.

2.3. Active optical ranging

Instead of having a passive component reflecting the transmitted laser signal back to the source, the optical communication link itself can also be utilised to perform ranging. It can be inferred that the system does not require a specific uplink ranging signal to function[6]. The The Consultative Committee for Space Data Systems (CCSDS) is considering standardizing two methods for optical ranging, termed *Synchronous* and *Asynchronous* modes [16], or more commonly known as *Echo* and *Asynchronous*.

The aforementioned modes indicate whether or not the data clock of the spacecraft have to be synchronized or not in order for the optical ranging to properly function. These modes determine the structure of the laser communication systems themselves as they may require dedicated hardware that may or may not be present in the way the communication system is designed. The modes of operation are illustrated in figure 2.3.

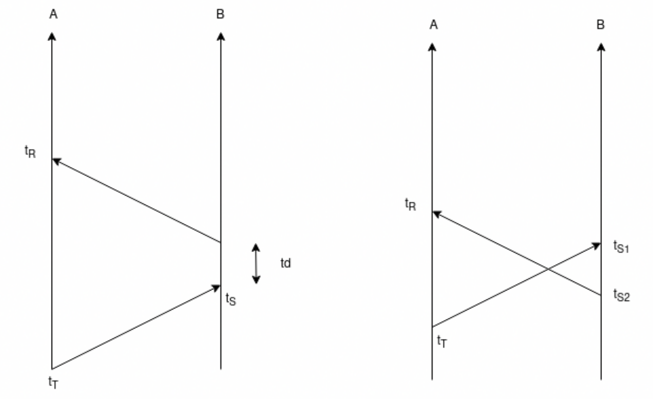


Figure 2.3: Synchronous or Echo communication ranging left, Asynchronous communication ranging right

For a synchronous transponder the spacecraft delay would need to be calibrated as accurately as possible in order to get a proper time estimation. Furthermore the signal return rate at terminal A is equal to the fire rate of the laser multiplied by the joint probability that pulses are detected at both ends of the link [17]. This works best when the RTLT is relatively short and there is a high probability of detection at both ends of the link. For interplanetary links asynchronous transponders must be considered, because the probability of detection at both sides is small.

As illustrated in figure 2.3, an asynchronous transponders fire independently at known frequencies regardless of the received signal. Data clocks do not need to be synchronized.

2.3.1. CCSDS data protocol

Before being able to understand both synchronous and asynchronous ranging methods, the data protocol stack currently under development by the CCSDS must be understood, as the methods that will be highlighted rely on this structure.

The CCSDS is an organization that discusses and develops standards for space data and information systems with governmental and quasi-governmental space agencies.

Firstly, clock synchronization is achieved by embedded markers in the code stack of the transmission. The CCSDS has defined a protocol stack for optical communication standards. This standard definition is currently only available for High Photon Efficiency (HPE) systems[18][19]. Because of this, only protocols are defined for Pulse Position Modulation (PPM). In an optical channel, a PPM symbol consists of M slots, however in practice, these M slots are followed by P guard slots, which never contain any data. This is done to accommodate physical requirements and to assist the receiver in synchronization [16].

The CCSDS has standardized two of these HPE signalling types: HPE telemetry, HPE beacon and optional accompanying data signalling[19]. The differences in these types are depicted in table 2.1.

Table 2.1: Standardized HPE signalling types[19]

Signalling type	**M**	**P**	**T$_s$**	**Data rate**
HPE telemetry	{4, 8, 16, 32, 64, 128, 256}	$\frac{M}{4}$	{0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512} ns	485bps - 2.1Gbps
HPE beacon	2	2	65536 ns	18bps - 2kbps

CCSDS standard defines Protocol Data Unit (PDU) to transform information into a collection of PPM symbols and guard slots. These steps may include processes such as randomization, encoding interleaving and adding synchronization markers[19]. For the ToF system to function properly, only the synchronization markers are of relevance. The following PDU are of relevance for understanding ToF ranging.

- A codeword: a set of PPM symbols after encoding.
- Interleaved Codeword (ICW): processed version of a codeword by taking one or more codewords and running them through an interleaver.
- SMCW: a concatenated version of Codeword Synchronization Marker (CSM) and an ICW.
- A repeated SMCW: several copies (Q) of the same SMCW concatenated to one and other to increase the availability of the signal.

A system can synchronize its clocks in theory by identifying the SMCWs. The general protocol structure that is created through this convention is illustrated in figure 2.4.

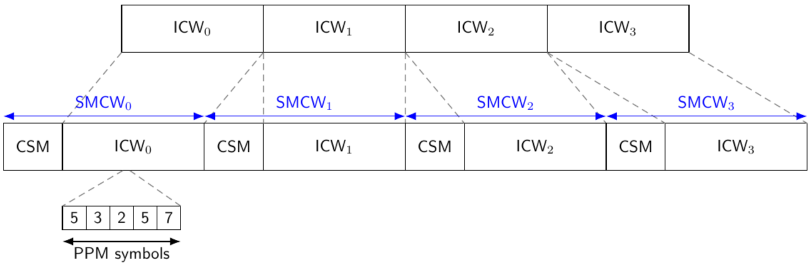


Figure 2.4: General SMCW Structure using 8-PPM symbols from [16].

In the work of Marc Sanchez Net [16] an addition to the CCSDS protocol stack is proposed. This would append a RCW to a SMCW creating a Range Synchronization Marker (RSM). Which serves as a special marker for the system to execute ranging to some capacity. This RSM can be followed by N regular SMCW to form a RCW, illustrated in figure 2.5. It must be noted however that this is not a convention used for past missions such as the Lunar Laser Communication Demonstration (LLCD).

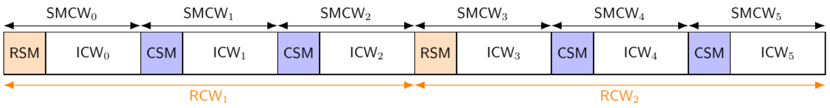


Figure 2.5: General RCW structure with N=3 SMCW from [16].

2.4. Digital Dual Mixer Time Difference

A DDMTD circuit is a digital variant of the analog Dual Mixer Time Difference (DMTD) circuit. Traditionally a DMTD is used to measure the time difference between two events with high precision. Naturally hardware such as FPGAs are digital in nature, hence a digital version of this device can be developed capable of measuring time difference between two digital clocks with a sub-picosecond resolution using a relatively low frequency counter, acting as a digital phase detector with femtosecond time resolution [20].

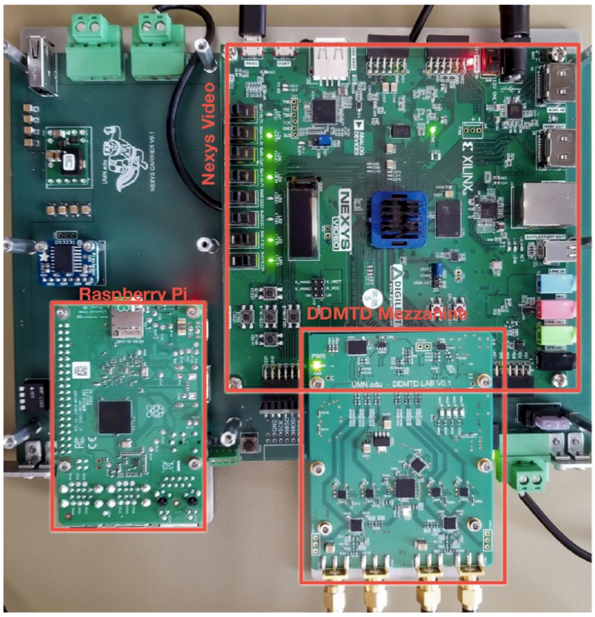


Figure 2.6: DDMTD Nexys Board [8]

A DDMTD is a hardware that is separate from a computing unit (figure 2.6), in this case an FPGA is considered. This allows for the system to be more accurate in phase measurements than the on board clock permits. This system has not been implemented for space applications but is instead used at CERN to measure drifts in the reference clock used for high energy physics experiments [8]. The DDMTD has been developed as part of a Ph.D thesis from Pedro Moreira [10].

The circuit can be used to compare signal data clocks instead, potentially also with femtosecond accuracy. In [8] a standard deviation of 100 fs was achieved on the noise measurement of the clock drift, this would theoretically translate to a ranging accuracy of roughly 30 µm solely by timing data clock phase difference.

2.5. Principle of operation

The DDMTD is a digital circuit that comprises a PLL and two flip-flops. It is designed to enable the precise comparison of Time Interval Error (TIE) between two clocks. There are two inputs to the DDMTD, each with a respective clock, u_1 and u_2, each with their own frequency f_1 and f_2. To track variations in phase between the two signals a PLL creates a new clock that is phase-locked to u_1 called u_{ddmtd}, with a slight offset frequency from f_1. A schematic of this principle is illustrated in figure 2.7.

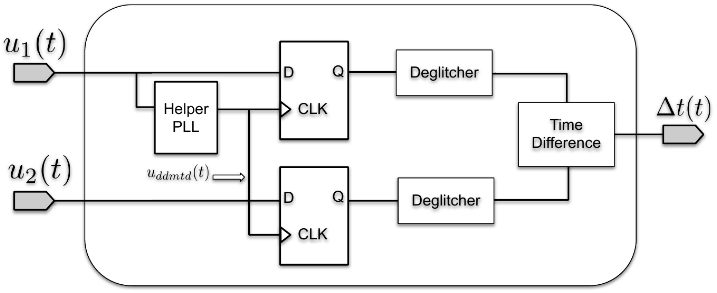


Figure 2.7: DDMTD schematic[20]

The aforementioned offset is carefully chosen according to Equation 2.4 [8].

$$f_{\text{ddmtd}} = \frac{N}{N+1} \cdot f_1 \tag{2.4}$$

Where N is an integer that determines the number of input clock cycles required for a full phase cycle of the heterodyned signal [8]. The higher N is chosen, the closer f_{ddmtd} to f_1 will be, resulting in a smaller f_{beat} [10].

Starting from the left of figure 2.7, both signal clocks are then sampled by D-type Flip Flop (DFF). If the phase between u_{ddmtd} and the input clock changes by some integer multiple of π, the output of the flip-flops will change state, tracking the integrated time difference over a period of N/2 cycles of u_{ddmtd} clock [8]. The resulting output will form a clock with the following beat frequency [8]:

$$f_{\text{beat}} = \frac{1}{N} \cdot f_{\text{ddmtd}} = \frac{1}{N+1} \cdot f_1 \tag{2.5}$$

A diagram illustrating this phenomenon is illustrated in figure 2.8.

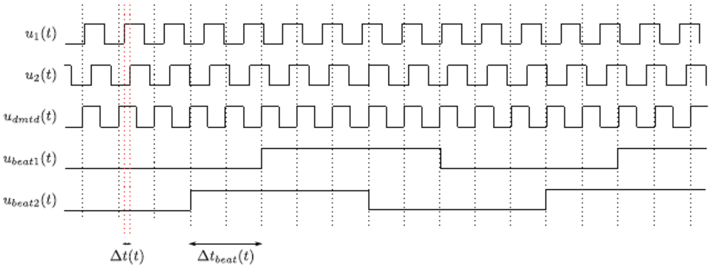


Figure 2.8: DDMTD time diagram for N = 5 [10]

In this figure $N = 5$ is arbitrarily chosen as an example. After N/2 input clock cycles, it is possible to measure the integrated time difference between two clock edges with a minimum precision given by Equation 2.6.

$$\Delta t_{min} = \frac{1}{f_{ddmtd}} \cdot \frac{f_{beat}}{f_1} = \frac{T_{ddmtd}}{N+1} = \frac{T_1}{N} \tag{2.6}$$

Where T_x denotes the period of the sub-scripted clock signal. The eventual phase difference between u_1 and u_2 can be measured with a precision given by the following equation [9]:

$$\Delta t = \Delta t_{\text{beat}} \cdot \frac{f_{\text{beat}}}{f_1} = \frac{\Delta t_{\text{beat}}}{N+1} \tag{2.7}$$

Δt_{beat} is the time difference between the transitions of the two beat clocks and Δt is the time difference between the input clocks [8]. The highest frequency that can be measured is given by the following equation [8]:

$$f_{\text{max}} < \frac{f_1}{N+1} \tag{2.8}$$

The output of the DDMTD circuit are the two beat clocks. These can be used as input in the FPGA where these signals can be processed further. In the PhD thesis from Pedro Moreira [10] a digital counter is implemented that counts the time difference between the two beat signals. This counter is then fed into a computer that correlates this count with the operating frequency of the counter to obtain Δt. The resolution of this counter influences the uncertainty of the phase measurement.

To increase the resolution of the time difference measured by the DDMTD, N can be increased, resulting in a low beat frequency according to Equation 2.5. However this comes at a penalty of noise in the form of glitches in the edge transitions of the beat signals [10].

The presence of glitches can be primarily attributed to timing jitter in the input clock signal, as well as the metastability behavior of the digital components [10]. The phase noise of both the input clock and the Helper PLL clock is sampled by the DDMTD DFFs during the digital down-conversion process [10]. When there is an increase in phase noise in the input clocks of the DDMTD, it leads to an extended transitory period during which glitches tend to occur [10].

The presence of these glitches must be minimised, hence the presence of the de-glitching modules in figure 2.7, after the outputs of the DFFs. The duration of a glitch is limited to $1/f_{\mathrm{ddmtd}}$, the sampling period of the flip-flops [10].

3

Methodology

This chapter presents the overarching methodology employed in conducting the tests throughout this thesis. It serves as the cohesive framework that integrates the rationale behind the tests and the underlying theories with respect to the research questions and objectives. This is achieved by walking through a high-level description of the thesis progress, discoveries, design decisions and methods used.

3.1. Requirements

In this section top-level system requirements are be listed in table 3.1 of all the systems in this thesis. The general requirements are denoted by `GEN`, requirements related to decisions made by TNO are denoted by `TNO` and finally requirements belonging to the systems in chapter 5 and chapter 6 are marked by `SYS-1` and `SYS-2` respectively.

These requirements are derived from hardware availability, which impose limits on what can be used during the thesis. Design choices that are set by the system by TNO, and finally some requirements imposed by design choices made during the thesis.

Table 3.1: Top-level system requirements.

Requirement ID	Requirement	Origin
GEN-A-001	All tests utilising an FPGA shall be using the KCU105.	Hardware availability
GEN-A-002	All firmware shall be programmed using Vivado ML 2021.2	Hardware availability
GEN-A-003	All system firmware for the FPGA shall be validated using post-implementation functional simulations executed using Vivado ML 2021.2.	Hardware availability
TNO-A-001	The measurement rate of the free-space ranging experiment shall be 50 MS/s	Hardware limitations
TNO-A-002	The amount of pules transmitted during one pulse train shall be equal to 40	Design choice TNO
TNO-A-002	The amount of pules transmitted during one pulse train shall be equal to 40	Design choice TNO
TNO-A-003	The pulse train shall have a frequency of 5 MHz	Design choice TNO
TNO-A-004	The pulse train shall have a duty cycle of 25%	Design choice TNO
SYS-1-A-001	The transmitted signal shall have a frequency of 10 MHz	Hardware limitations
SYS-1-A-002	The transmitted signal shall be a square sine wave.	Design choice
SYS-1-A-003	The system shall use DDMTD for phase frequency detection.	Design choice

Continued on next page

Table 3.1 – continued from previous page

Requirement ID	Requirement	Origin
SYS-2-A-001	The system shall make use of the commercial MGT for communication.	Design choice
SYS-2-A-002	The system shall use SFP modules for laser transmission and reception	Design choice
SYS-2-A-003	The optic fiber cable shall be single-mode $9/125\ \mu m$	Hardware availability
SYS-2-A-004	The SFP shall transmit at a wavelength of 1550 nm	Hardware availability

3.2. Methodical Approach

The main research question and objective have been broken down into five sub-questions as described in section 1.2. These sub-questions will in turn be answered through three separate tests, each test aiming to improve the overall accuracy of the measurement as well as system complexity in terms of laser communication ranging. This section aims to provide a overview of the general methods used for the experiments and the relation they have with one and other. The experiments increase in complexity as the thesis progresses to identify constraints and limits that can be addressed and identified in each system that follows.

The underlying principles and effects of a laser communication system can be analysed by using its predecessor, laser ranging. Such a system is nearly identical to a laser communication system, it lacks however an intrinsic data structure. Nonetheless, the methods of transmitting and receiving photons remains identical, while simultaneously revealing constraints and points of improvement that can be made to increase the accuracy for the systems that follow. Hence this test is suited to answer the first sub-question.

Hence the first test that is considered thus, is a free-space laser ranging experiment further elaborated upon in chapter 4. The test setup belongs to TNO and is not made in particular to recover range the range between the reflector and terminal, instead it is made identify the reflective coating on the retro-reflectors as part of a different experiment explained in section 1.3. This has one major implication for the purposes of this thesis, one being that the system is not calibrated for ranging, which is necessary in order to attempt to obtain the real range between transmitter and retro-reflector. This value is usually expected to be a near constant value of latency that can be subtracted from the obtained ToF measurement.

The setup is similar to the illustration from the previous chapter depicted in figure 2.1. Here multiple laser pulses are transmitted at a frequency f and a duty cycle τ. After about 40 pulses are transmitted there is a moment of silence where the reflection is expected.

The goals of this test is to:

- Obtain a range measurement based on the ToF of a laser pulse.
- Quantify the uncertainties with respect to the range measurement.
- Identify constraints in obtaining a range measurement through free-space.

As mentioned, this test did not involve any range measurement based on optical communication, but merely on the ToF of the photons. This ToF method can be used on a data stream however, only then the ToF of particular data packets are measured instead of the photons. Hence the next tests should be improved by implementing this exact data link to start. The major constraint in terms of accuracy, was the inability to measure the accuracy based on the fact that the exact distance remains unknown and could deviate from the approximation by several meters, in addition to the fact that no calibration steps with respect to ranging were taken. Furthermore the large resolution created by the slow measurement rate of the oscilloscope is bound to play a role in terms of accuracy.

The uncertainties that have been identified are therefore the unknown actual distance, improper calibration and the slow measurement rate. Resolving these uncertainties further, will result in a more accurate system, hence this is done in the test that follows. Moreover a proper data protocol is required to be implemented such that ranging can be done through the communication protocol.

The test that follows aims to resolve the uncertainties as well as prepare for the implementation of proper communication, by answering the second and third sub-question. Hence the hardware is switched from the TNO setup to an FPGA. These devices are capable of operating at much higher frequencies, are exceptionally versatile and most importantly are equipped with COTS optical networking transceivers that can be used to establish communication. The addition of the FPGA however introduces a steep addition in terms of system complexity. Not only does signal data have to be measured and analysed, but also the system firmware surrounding the test has to be developed from the ground up, including clocking mechanisms and other attributes. From this, the following goals that are to be reached from this test can be identified:

- Familiarize programming an FPGA.
- Instantiate the proper internal clocking mechanisms.
- Create basic signals using FPGA hardware.
- Extract signal data from the FPGA.
- Perform post-implementation functional simulations using Vivado ML.

First the system has to be calibrated by measuring the internal delay. This is done by programming firmware that routes a square wave signal to a GPIO pin on-board of the FPGA that serves as a reference point for what would be the point where the transmitter would be located. This is physically bridged by the means of a wire to another GPIO pin, resembling the receiver. This creates an internal delay that is ought to represent the delay a signal experiences travelling through the FPGA, which can be measured and theoretically calibrated for.

In order to achieve this, the first four goals are to be accomplished. Measuring this delay however is non-trivial, as first of all the signal itself needs to be measured from within the FPGA silicon, and direct access to these signals is complex. Luckily MATLAB IP firmware can be used to access this signal data. This has proven to be not the most optimal solution, however due to its ease of implementation and use it outweighs a more optimal solution in the short term.

This method does however have several limitations with regards to data capture. As with every component on the FPGA, it is driven by a clock with a specific frequency, meaning that this frequency is directly the sampling rate of the signals. Naturally, a high as possible sampling rate is desired as this results in the highest resolution. This data capture method is a large improvement over the acquisition speed from TNO, it quadruples the rate at which signals are sampled.

The progression of the signal through the system and back will create a phase difference. This phase difference will be directly related to the total time delay of the signal. In order to measure this phase difference however, phase frequency detectors can be used. These take the form of digital circuits that aim to put out meaningful information with respect to the phase. It must be understood that this information with regards to the phase in the FPGA does not take the form of an integer of floating number, rather it takes shape as one or more binary signals.

With the current described setup, one would be able to discern a phase difference between the transmitted and received signal with a resolution of 5 ns. A solution to this problem exists, which is often used in high precision network solutions [8]. A DDMTD can be implemented which is able to measure the phase difference with a better resolution than the clock permits. This circuit is described in more detail in chapter 2.

Once all the firmware is in place a post-implementation functional simulation can be made using a testbench. A testbench is a VHDL or Verilog module that virtually provides the appropriate inputs to the firmware under test, such that the simulate can propagate the inputs. These simulations are catered to the specific hardware used in the design and can be considered to be extremely accurate and close to reality. This is done to save time, as the process of programming an FPGA is time consuming and depending on the size of the firmware and the amount of changes made per iteration can take several hours. Once all is confirmed to work, the tests can take place and the results can be recorded.

The previous test has proven to be a good baseline for the next test in terms of managing and programming the FPGA, however the results of the internal delay were found to be not representable for the internal delay a signal would experience if it had been transmitted over the dedicated MGT on the FPGA, as the signal from the previous test does not cross into this hardware domain. An MGT is standardized hardware used for nearly all networking applications ranging from consumer goods to professional equipment, including networking over glass fiber with the use of laser terminals.

Thus the next test shall cater to this shortcoming as well as progress to actual communication over an optical link. As mentioned, the MGT would be a perfect solution to this problem as these hardware components are perfectly capable of this type of communication. Implementing this hardware further increases the level of complexity of the system. Hence the goals of this test are the following:

- Create a signal transmissible over the MGT of the FPGA.
- Re-calibrate to compensate for MGT internal delays.
- Obtain range measurements through optical telemetry ranging.
- Create different firmware versions operating at different line rates.
- Identify components introducing uncertainty in the range measurements.
- Demonstrate the potential in-orbit system performance.

The MGT can be interfaced with by Xilinx IP firmware, to configure standardized communication with the use of SFP modules. These are standard modules used for networking applications that contain both a laser transmitter and receiver that can be attached to optic fiber cables. These optic fibre cables are a suitable medium to test laser communication ranging through data protocol.

The next step is to re-calibrate the system to compensate for the added internal delays due to the use of the MGT. This can be done in two ways, either bridge the Physical Medium Attachment (PMA) layer or Physical Coding Sublayer (PCS) internally through firmware changes, or to attach a short 1 to 2 cm piece of optic fiber to the transmitter and receiver in loopback formation. Due to the unexpected behaviour of the MGT internals, a straightforward approach to simply bridge the SFP externally was chosen.

Next, multiple single-mode optic fiber cables with a length of 10 m are ordered in multiple configurations to create fibre lengths ranging from 10 to 50 m. With these configurations the system performance can be analysed as the target range and internal delay measurements are known. These tests can be repeated for different line rates to demonstrate the current influence of this parameter on the existing system. The line rate is terminology used by Xilinx in their user guides. It is defined as the data rate that exists in the medium over which it is transmitted, hence the name line rate as data is usually transmitted over a cable or line. This distinction is made in order to easily separate different data rates in the system as the internal data rate may not be equal to the data rate that exists over the line due to the type of encoding for instance.

Finally the components in the system causing jitter and other uncertainties can be identified and the overall ranging performance can be analysed. Possible improvements to this system are then discussed in chapter 7, where the implementation of DDMTD is discussed to be embedded in the GTH such that the ToF can be measured more accurately.

II

Free-space laser ranging

4 Free-Space Laser Ranging Measurements

In this chapter the free-space laser ranging experiment conducted in collaboration with TNO is presented. Starting with the objective and goals of the experiment presented in section 4.1. Next, the system setup is detailed in section 4.2. Following is the method employed to conduct the experiment and measurements in section 4.3. The results in section 4.4, followed by the conclusion of the experiment in section 4.5 and a discussion of the results and future recommendations in section 4.6.

4.1. Objective and Goals

The main objective of this test is to obtain a range measurement with the use of a basic laser ranging system as precursor to employing actual communication. This is one of the basic forms of laser ranging, not considered to be ranging through its telemetry. It provides an opportunity to establish a baseline of what can be achieved with minimal complexity in terms of ranging.

The objective of this chapter is to first sub-question:

1. How can laser ranging be achieved using a communication system?

The goals resulting from this objective are:

- Obtain a range measurement based on the ToF of a laser pulse.
- Quantify the uncertainties with respect to the range measurement.
- Identify constraints in obtaining a range measurement through free-space.

4.2. System Set-up

A top-level sketch of the test setup is illustrated in figure 4.1. In this figure the parts of the set-up relevant to making a range measurement is illustrated.

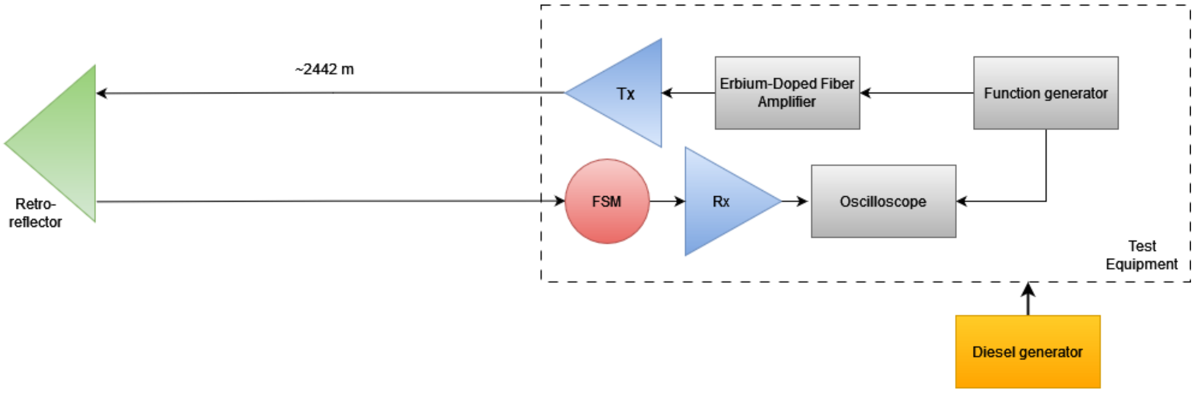


Figure 4.1: Top-level sketch of the test setup from TNO

The measurement equipment is powered by a mobile diesel generator, due to the absence of power from the national power grid at the measurement location. The major implication of this in terms of the range measurements is that the generator creates electromagnetic distortions on the power lines, influencing devices downstream and in turn the measurements.

To add to this, it must also be noted that, for a future repetition of this experiment, the diesel generator could destroy devices such as the oscilloscope. This warning was provided by TNO as they were suspecting, although not certain, that the oscilloscope used in the setup was unusable for accurate measurements after the test was done due to the unstable power source. Hence it would be unwise to connect an oscilloscope that one is not willing to potentially destroy.

Starting with the function generator, here the desired signal is generated in its electrical form. The function generator is wired to both the Erbium-Doped Fiber Amplifiers (EDFA) and the oscilloscope in parallel in order to properly record the transmitted signals. The EDFA is an important component for these free-space setups, it amplifies the optical signal increasing the SNR at transmission.

Once the signal is amplified it is transmitted to the retro-reflector. In this setup there are multiple retro-reflectors, but based on the transmitted wavelength, one of the reflectors will respond with specific spectral properties as part of the SLP experiment. An image of the array is displayed in figure 4.2. For ranging, it does not matter which of the reflectors is chosen.

When the laser is reflected back it bounces off a Fine Steering Mirror (FSM). An FSM is a mirror connected to several motors and a control system allowing for fine movement of the mirror to point the laser onto the receiver, maximizing the received power and ensuring that the photons reach their destination. The signal is then converted from the optical to electronic domain in the receiver, which is then measured by the oscilloscope.

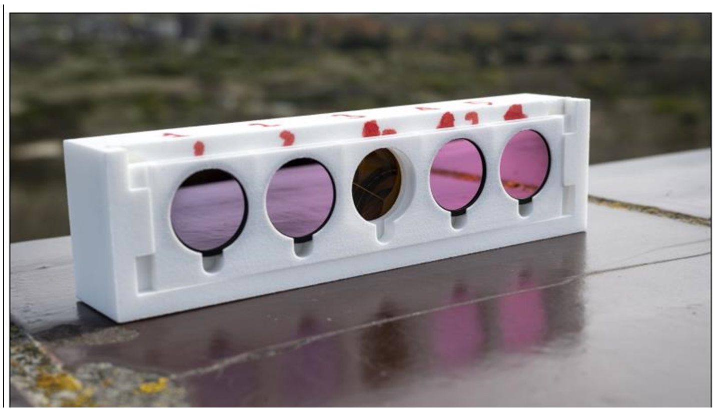


Figure 4.2: Retro-reflector array used in the test setup from TNO[1].

As mentioned in section 1.3, this setup is not made with laser ranging in mind, affecting the final accuracy due to the exact distance being unknown.

The function generator is programmed to transmit a pulse train. The pulse train is defined as N amount of 1-bit pulses with duty cycle τ and frequency f. The pulse trains transmitted consisted of $N = 40$ pulses with a frequency of $f = 5\text{MHz}$ and a duty cycle of $\tau = 25\%$, a pulse train lasted in total 8 µs, after which, transmitter sends no signal for 20 µs. Within this period, the reflected pulse train is expected as the total RTLT is expected to be around 16.28 µs.

At first an issue existed where the system had self-reflection, this meant that somewhere within the 20 µs there existed a ghost signal as a direct result of this. With some adjustment, this reflection has been eliminated.

The capture window lasting from the start of transmission to end of reception is defined as a trace. An example illustration of such a trace is illustrated in figure 4.3.

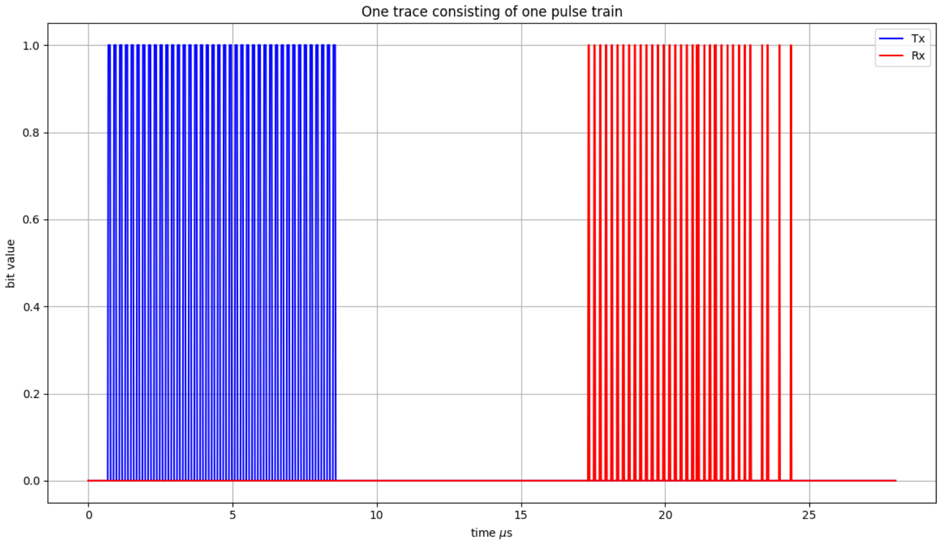


Figure 4.3: Illustration of one trace captured by the oscilloscope in the TNO test setup, operating at 50 MS/s, after processing the raw voltage measurements. Left most pulse train is from the transmission signal, right pulse train is from the receiver signal

The pulse train generated by the function generator is then converted into an optical signal and pre-amplified using an EDFA.

The optical signal then travels to a retro-reflector located on the roof of the TNO building in The Hague, Oude Waalsdoperweg. This retro-reflector is designed to reflect the light approximately the same path, back to the receiver, which is located on the Meteotoren in Scheveningen. This situation is illustrated by a screenshot taken from Google Maps depicted in figure 4.4. The distance between the retro-reflector and the receiver is provided to be roughly 2442 m and 2430 m according to Google Maps. These were rough approximations, the exact distance is unknown as range measurements were not the primary objective of TNO. This uncertainty will greatly impact the quantification of the accuracy of the system. The precision however will remain quantifiable as this is independent of the true distance. Hence with this test set-up, the precision of the system will be quantified.

[1] https://www.tno.nl/en/newsroom/insights/2023/08/successful-ground-test-shows-potential/, [accessed on 6-11-2023]

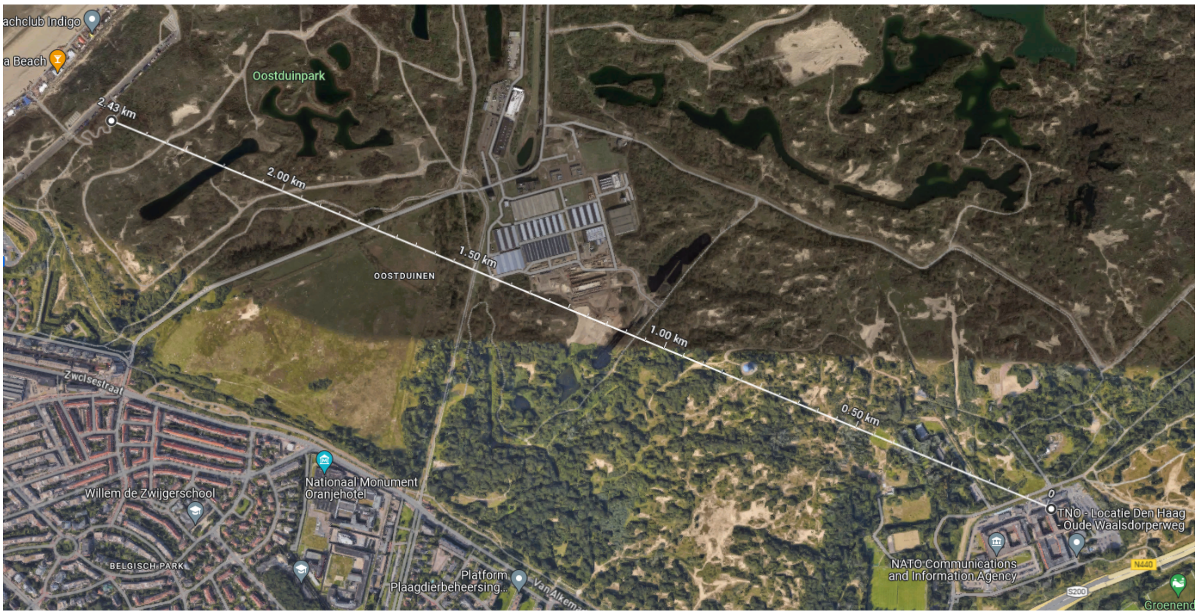


Figure 4.4: Google Maps screenshot of the distance between the Meteotoren and the TNO building

At the receiver, the photons arrive to a photo-diode where they will be translated from the optical domain to the electrical domain, after which they will be registered by an oscilloscope. It has a sampling rate of 50 MS/s, translating into a time resolution of 20 ns or 6 m. The resolution of the oscilloscope is the limiting factor with regards to the test setup, limiting the uncertainty of the range to 6 m.

4.3. Method

The oscilloscope from section 4.2 is used to capture 100 traces. Each train consists of a transmission pulse train and, ideally, an exact replica of the transmission on the receiver. In order to recover the range from these traces, the start of a pulse train, on both the transmitter and receiver needs to be identified. In turn, these time measurements can be subtracted by one and other to determine the ToF of the photons. Before this measurement can be done, the data need to be post-processed [2], as the circumstances surrounding the measurements made the results noisy.

4.3.1. Post-Processing

Initially, it is essential to explore the data set that requires analysis. The measurement obtained from the oscilloscope looks as illustrated in figure 4.5. This sample of the data encapsulates the general behaviour of the readings and will be taken as an example whenever the data is discussed.

[2] https://github.com/tpalings1998/laser_ranging_test.git

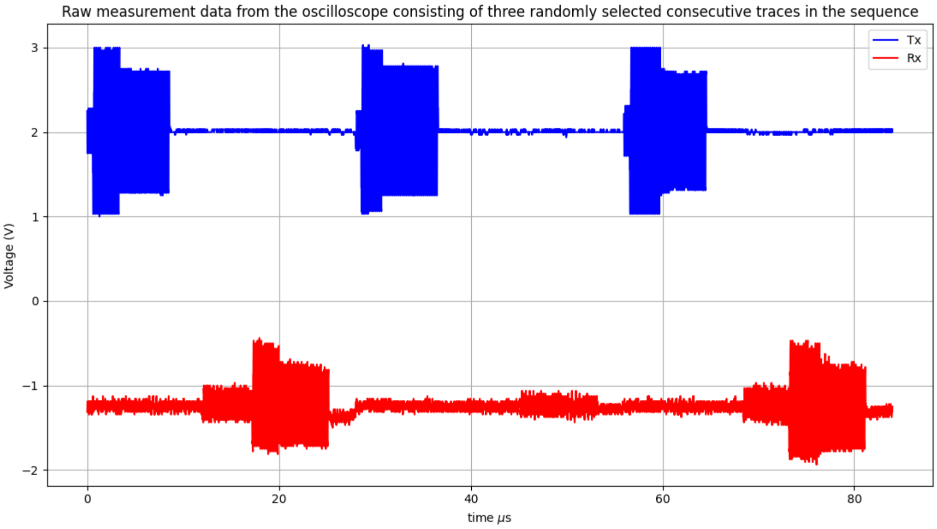


Figure 4.5: Three randomly selected consecutive traces from the oscilloscope, showcasing the behaviour of the raw measurements. The top curve represents the transmitter signal, the bottom curve represents the receiver signal

Both the transmitter and the receiver readings are encapsulated by noise, naturally the receiver side more so compared to the transmitter. This is due to background light, the position of the FSM at the time of measurement, scattering effects due to the atmosphere and the phase noise of the diesel generator at the Meteotoren for example.

Starting with the transmitter data, a relatively simple Gaussian distribution can be applied to calculate the mean and standard deviation in the data. Values above or below a threshold according to this distribution are then identified as one or zero bits respectively. Due to the relative little noise on the transmitter this threshold is easily identifiable.

Before identifying this threshold however, closer attention to the transmitted pulse train must be paid. Taking a closer look at the first pulse trace from figure 4.5 as an example illustrated in figure 4.6, it can be considered to consist of 3 parts. The first part is a small part lasting from approximately 0 to 0.56 µs, the second part is high in amplitude lasting from about 0.56 to 3.37 µs, and the third and final part is a large flat part lasting for the remainder of the pulse train. It is known from the function generator that the amount of peaks shall be exactly 40. For the remainder of this chapter, when referring to different parts of a trace, these specific parts are meant. Counting these peaks however results in more than 40 peaks. Meaning that some noise artifacts are picked up higher leading up to a pulse train, or remain after the pulse train is finished.

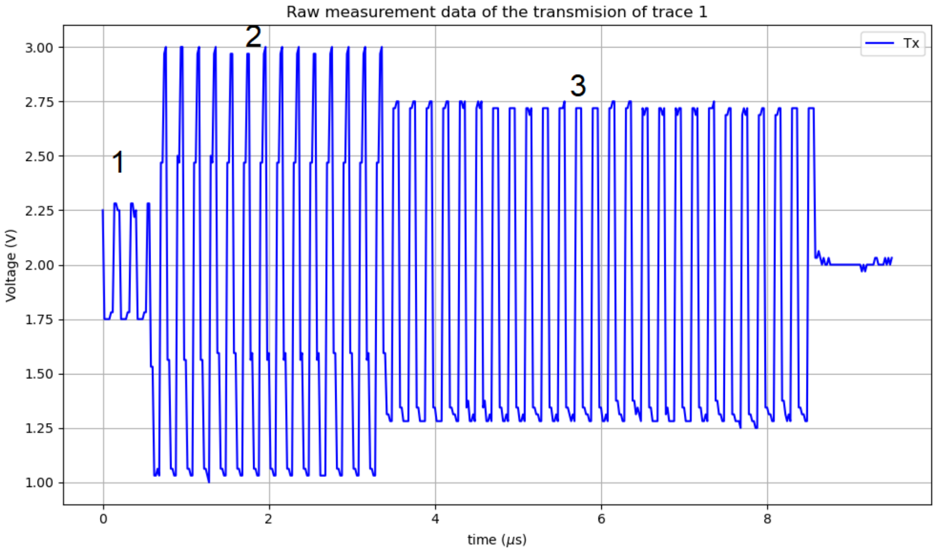


Figure 4.6: Magnification of the transmission data of the first trace from figure 4.5.

To distinguish between noise artifacts and valid signals, a reflection analysis can be applied on the receiver side. The receiver pulse reading can be divided into three segments. When comparing the pulse counts in each part of the receiver with their counterparts in the transmitter, it becomes evident that the number of peaks and valleys in the last two parts of the receiver corresponds to those in the last two parts of the transmitter. However, the first part exhibits a higher count in the receiver compared to the transmitter, indicating that it represents residual noise within the system.

This holds for all traces with the exception of some, such as the middle trace in figure 4.5. It can be seen that there is a faint pulse train on the receiver that is observable, but bears little to no resemblance to the original transmitted pulse (figure 4.6) other than the corresponding amount of peaks and valleys. This may likely be due to periodic external perturbations like a period of inaccuracy in the FSM slight vibrations due to the wind or power fluctuations in the diesel generator. The exact cause of this is not known.

In conclusion the start of a pulse train can be identified by a set of large in amplitude voltage spikes (part 2) followed by a flat, less high in amplitude plateau (part 3), combining to a total of exactly 40 peaks.

To filter out the noise of the transmitter, the values of the 100 traces are combined into a large array, from which the parameters of a corresponding Gaussian distribution can be calculated. Calculating the mean and standard deviation leads to the characteristics presented in Equation 4.1.

$$X_{Tx} \sim N(\mu, \sigma^2) \sim N(1.96, 0.155) \tag{4.1}$$

As the data is so distinguishable from the background noise, taking all values outside ±1σ on each side of the distribution as being the signal is sufficient. For the sake of convenience for further analysis, all the values that are identified by applying this filter are set to 1 when they are positive and -1 if negative. All other components outside of the filter are considered noise and set to 0. The signal from the oscilloscope is differential, hence the negative voltage measurements shall be taken into account as well. This translates the measurements to a more processable variant without noise, illustrated in figure 4.7.

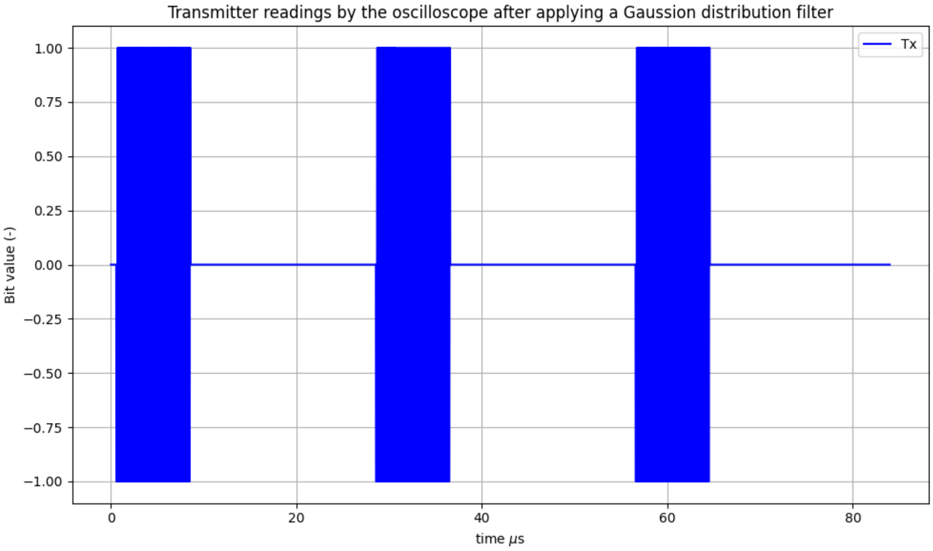


Figure 4.7: Transmitter readings after applying a Gaussian distribution filter, taking all values outside of±1σ

The same principle can be applied to the receiver data. However due to the large noise components it must be approached differently. When calculating the average and standard deviation of the receiver reading, it takes into account the accumulated noise over 100 traces, widening the Gaussian distribution significantly, making it difficult to differentiate between pulses and noise such as with the second trace illustrated in figure 4.5. Here the pulses are so faint that overgeneralizing becomes an issue, especially as the SNR in general is relatively large.

Taking the first trace from the receiver in figure 4.5, illustrated in detail in figure 4.8 as example to start the analysis. The first part is sufficiently high to potentially be detected as being part of the pulse. This could be avoided as done with the transmission, however at the trailing edge, after the third part between 25 to 28 µs, there is a dip in the signal that is equal to magnitude of the first part. This is due to the phase noise of the diesel generator, however it is sufficiently high to throw off the same algorithm that has been developed for the transmission, creating the potential to misidentifying it as a signal.

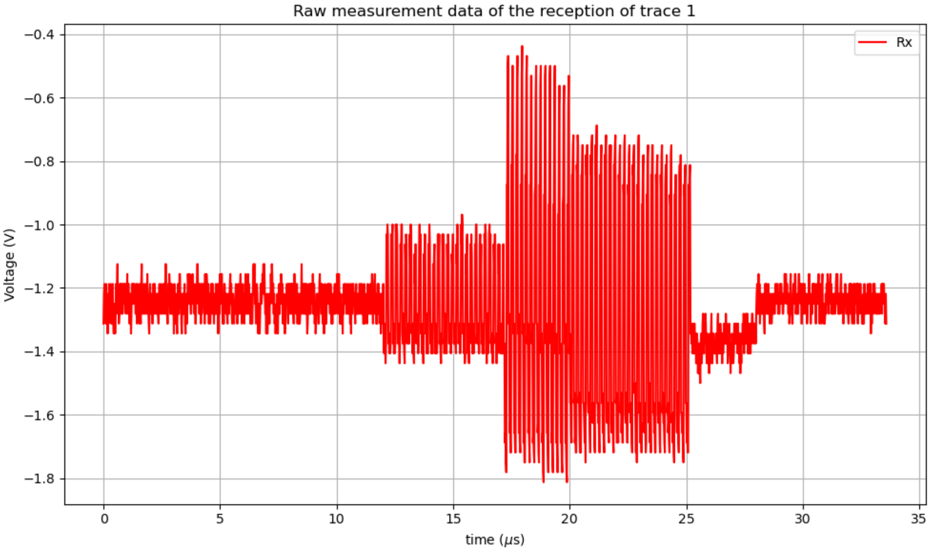


Figure 4.8: Magnified version of the receiver data of the first trace from figure 4.5.

In order to mitigate these effects, the receiver traces are processed on an individual basis rather than over the whole array as done with the transmitter, reducing the accumulation of noise, making even faint traces such as the second trace identifiable. Each trace is processed in the same way as the transmitter measurements, resulting in 100 individual Gaussian distributions. A table presenting the values are listed in Appendix A. For the receiver, it was found that taking values outside of ±1.7σ as the signal recovered most of the pulses. Naturally not all pulses can be recovered as some are lost due to noise or the algorithm does not identify them as such because the SNR is too low on that particular peak. To demonstrate the algorithm it is applied to the receiver data from figure 4.5, resulting in figure 4.9.

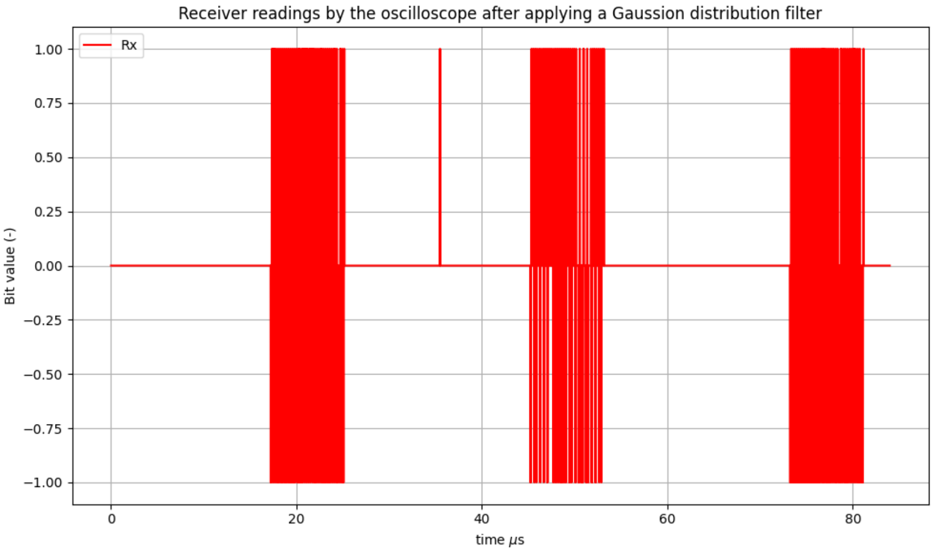


Figure 4.9: Receiver readings after applying a Gaussian distribution filter, taking all values outside of ±1.7σ

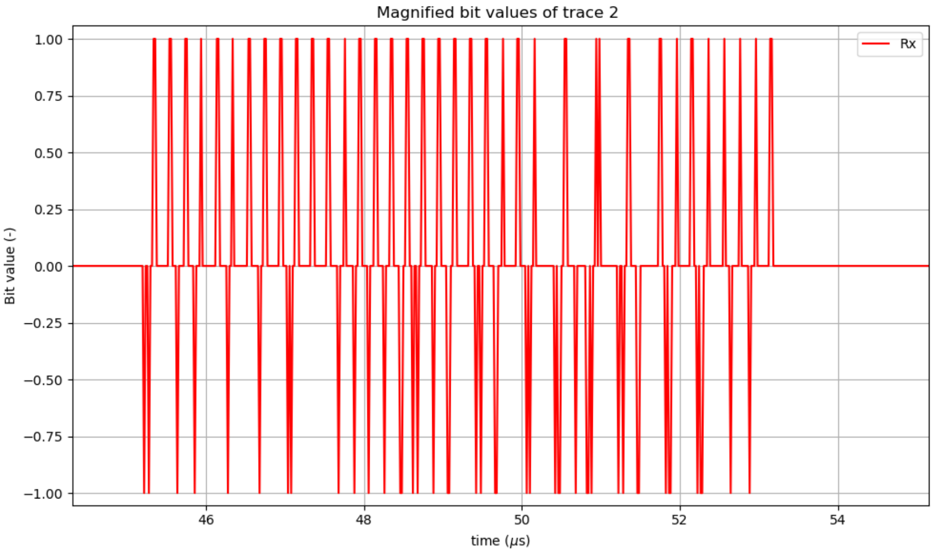


Figure 4.10: Magnified receiver readings of the second trace after applying a Gaussian distribution filter, taking all values outside of ±1.7σ

It can be observed from figure 4.10 that there are some gaps in between the data, as well as a peak outside of the data. This is especially the case for the middle trace, as can be seen in figure 4.5, the SNR of this particular trace is not high and therefore the method may have a difficult time distinguishing between the noise floor and the signal itself. For cases such as these, manual intervention may be required. This is less than optimal because, for a growing number of traces, the amount of manual detection required is labor intensive. As the measurement consists only of 100 traces and the vast majority is detectable through the filter, it is deemed to be sufficient. Nonetheless, this signal contains little noise and makes it more processable. After post-processing has been done, the range measurements can be made.

4.3.2. Range Calculation

Consider the pulse train to be one solid block instead of a set of pulses. This block is described as two parameters, the time at which its rising edge occurs with respect to the start of the measurement t_{rise} and the time at which its falling edge occurs with respect to the start of the measurement t_{fall}. Both the transmitter and the receiver have these attributes. These attributes are illustrated in figure 4.11.

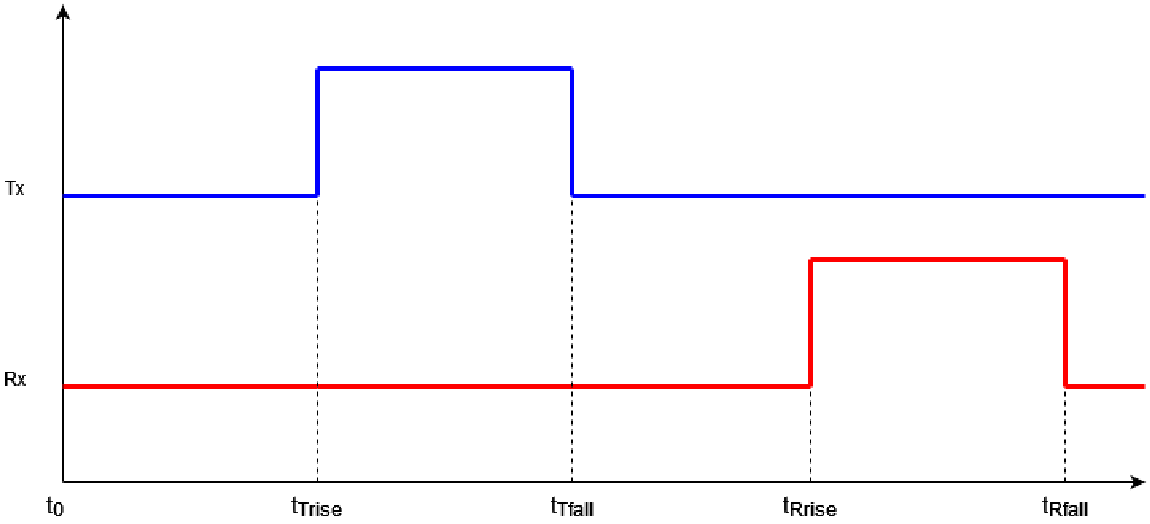


Figure 4.11: Visualization of the parameters associated with a pulse train

Then the total round-trip light time can be recovered using Equation 4.2 or Equation 4.3.

$$\Delta t_{total} = t_{R_{rise}} - t_{T_{rise}} = \Delta t_{rise} \tag{4.2}$$

$$\Delta t_{total} = t_{R_{fall}} - t_{T_{fall}} = \Delta t_{fall} \tag{4.3}$$

Equation 4.2 determines the total time delay between a received pulse train and transmitted pulse train with respect to the start of the pulse trains, Equation 4.3 with respect to the end of a pulse train. In the ideal scenario, the outcome of both equations shall be equal. In reality this may not necessarily be the case due to the trigger mechanisms in the oscilloscope. This effect will be discussed later in this section. These rising and falling edges can be detected using the following equation for each point in the transmitter and receiver data.

$$y[i] = x[i+1] - x[i] \tag{4.4}$$

Where i is the index of each point on the transmitter and receiver data. If $y[i]$ is equal to 1, then it is considered a rising edge, if it is -1 it is considered to be a falling edge. These index values are stored until the data set is evaluated. Then these index values are subtracted by one and other and multiplied by the sampling period of oscilloscope, resulting in a calculation for Δt_{total}.

This RTLT value is the total delay experienced by the system. As mentioned previously, the objective of TNO was not to be able to precisely determine the range. For this, calibration is required that will map the additional delays experienced by the signal in its trajectory. The known delays are listed in table 4.1. In addition to these known delays, there are other unknown delays such as the propagation through installed electronic circuits and other embedded systems. As these values remain unmapped, the final range computation can only be compensated by what is known.

Table 4.1: Gaussian distribution parameters.

Attribute	**Length (m)**	**c/c$_{\text{vac}}$**
RG58 coaxial cabling	~10	0.66
Optic fiber cabling incl. EDFA	~24-29	0.68

The EDFA tends to have 10-15 m of optical fiber inside. Connecting the equipment in between the floors of the Meteotoren is done using coaxial cable, which amounts to roughly 10 m. The time delay caused by the components can be calculated using the values from table 4.1. Calculating these time delays yield the following:

$$\Delta t_{RG58} = \frac{\ell_{RG58}}{0.66 \cdot c} = 50.51\ ns \tag{4.5}$$

$$\Delta t_{fiber} = \frac{\ell_{fiber}}{0.68 \cdot c} = [117.65, 142.15]\ ns \tag{4.6}$$

In conclusion, the target RTLT measurement is the total RTLT delay subtracted by the components adding unwanted delay.

$$\Delta t = \Delta t_{total} - \Delta t_{RG58} - \Delta t_{fiber} \tag{4.7}$$

As the distance travelled by the photons is twice the distance to the retro-reflector, the eventual distance shall be divided by a factor of two. The range can be recovered by using the equation Equation 4.8.

$$\Delta r = \frac{c \cdot \Delta t}{2} \tag{4.8}$$

The limit to this range calculation is oscilloscope sampling rate. As mentioned earlier, a trigger position is set on the oscilloscope. This position is set on the first rising edge of the transmission pulse train, or t_{Trise} in figure 4.11. This causes the uncertainty of the start of the transmitted trace to be negligibly small. It does however introduce uncertainty with respect to the t_{Tfall}, t_{Rrise} and t_{Rfall}, which is exactly one clock cycle or 6 m expressed in terms of range for each of the aforementioned attributes.

Thus, if the range were to be calculated from the falling edges, the uncertainty would essentially double to 12 m compared to 6 m when measuring it from the falling edges, hence it is opted to calculate the range with respect to the rising edges in order to minimize the uncertainty.

4.3.3. Atmospheric Correction

The photons from the signal travel through the atmosphere causing slight scattering and additional delays introduced by the atmosphere, because of the varying refractive index of the air curving the path of the photons. There exist several models that have been developed over the years that apply to atmospheric corrections related to a laser terminal at higher elevation angles, pointing outward to space [21]. In these models, photons travel through the entirety of the atmosphere and back and depending on the elevation can encounter several tens of kilometers of atmosphere.

However, with this test setup, the total distance photons travel through the atmosphere is less than 5 km in total, a fraction of what photons experience according to the model of [21]. Hence a different model can be applied to this particular situation.

There is a model that uses the average air temperature along the trajectory to correct for atmospheric effects [22]. This model computes the mean integral value of the air refractive index along the trajectory. The disadvantage of this method is that the accuracy is limited to the number of points along the trajectory the temperature is measured.

This method will be briefly described as part of the attempt to correct for atmospheric influence.

Applying the theory from [22] to the test setup, consider the trajectory as illustrated in figure 4.1. Generally, the following holds for the length of the trajectory [22]

$$\mathcal{L} = \frac{c \cdot \tau}{\bar{n}} \tag{4.9}$$

In this equation $\mathcal{L}$ is the length of the photon trajectory due to the refraction in a non-homogeneous atmosphere, c is the speed of light in vacuum, τ is the signal travel time and $\bar{n}$ is the mean integral of the air refractive index along the trajectory.

Given the calculated signal time delay from Equation 4.7, if the air refractive index integral can be evaluated along the trajectory, the correction for atmospheric propagation can be applied.

The experiment was conducted the 1st of May 2023 between 12:00 and 13:00. The atmospheric properties recorded during this time at the Meteotoren are displayed in table 4.2.

Pressure	1016.8 hPa
Relative humidity	0.93
Temperature	284.95 K

Table 4.2: Atmospheric properties measured during the experiment according to

The contribution of the error due to the atmosphere is overwhelmed by the large uncertainty due to the slow sampling speed. Correcting for these effects therefore currently has no meaning. Still it is important that his is mentioned as ultimately, it may be the final limit if the uncertainty is decrease significantly down to millimeter level.

4.4. Results

The 100 traces are evaluated and the methodology from section 4.3 is employed to calculate the range. These traces are indexed starting from 0 to 99. First, the data is post-processed and the distance of each transmitting and receiving pulse train is observed. There are two methods to quantify the time delay between pulses as mentioned in the previous section, either by looking at the starting time or end time of each pulse train. Because the data may be incomplete, both are observed simultaneously and the results compared.

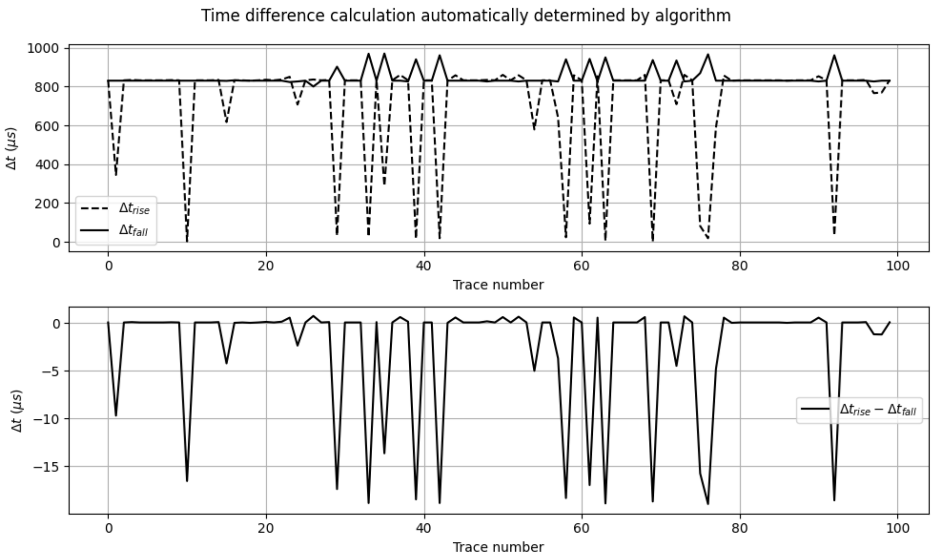


Figure 4.12: Automated ToF calculations per oscilloscope trace

When running the Python script to automatically detect these time intervals, the results are shown in figure 4.12. It reveals that the start and end times of the pulse trains in each trace do not align perfectly.

It is apparent that using the pulse train start as a reference consistently yields a greater range calculation in comparison to employing the pulse train conclusion as a reference, with the exception of the evident outliers. In most traces, the difference is two oscilloscope samples (40 ns).

Outside of the outliers, it is difficult to differentiate between the two measurements which one is the most representative of the ToF. The results differ due to the way the data is processed. Looking at the figure, it may be concluded however that the ToF calculation from $\Delta t_{fall}$is more stable as it has significantly less outliers, and the outliers it has are of lower amplitude hinting at the fact that this may be a better frame of reference to take for the range measurement.

To confirm this, the z-value from section 4.3 is reduced allowing larger noise spikes to be identified as a signal. This will have the effect that the ToF measurements will consists of an increasing amount of outliers in the parts of the data where the SNR is lowest, while in parts where the SNR is sufficiently high more unidentified signal is added to the data set. In these high SNR regions of the data, the ToF values of both Δt_{rise} and Δt_{fall} shall converge to the value that represents the ToF, until too much noise is incorporated such that the data only consists of outliers.

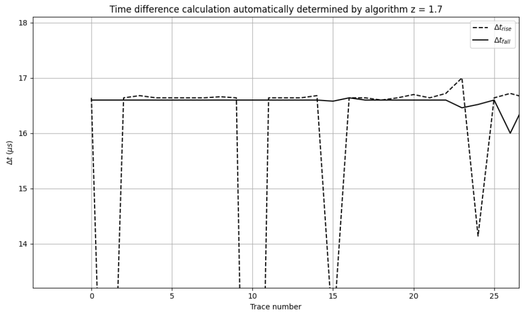


Figure 4.13: Difference in Δt_{rise} and Δt_{fall} when applying a Gaussian distribution filter with $z = 1.7$

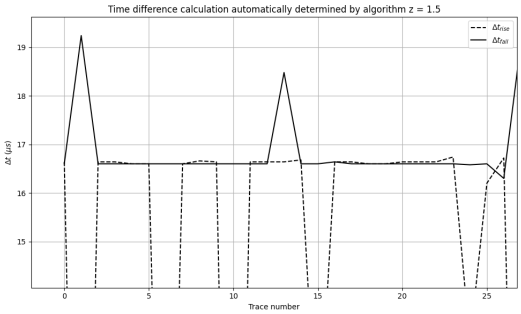


Figure 4.14: Difference in Δt_{rise} and Δt_{fall} when applying a Gaussian distribution filter with $z = 1.7$

Doing this results in most parts of the Δt_{fall} curve to remain stable for the most part, while the Δt_{rise} curve converges to Δt_{fall}. This is illustrated in figure 4.13 and figure 4.14. Manual confirmation, done by manually observing the time difference in the data of randomly selected traces confirms that Δt_{fall} is the most representative for the ToF measurements.

Outliers are manually evaluated and corrected when applicable. In some traces, the SNR is sufficiently low that the signal is indistinguishable from the surrounding noise. This concerns trace 29, 42, 58 and 76. These traces are illustrated in figure 4.15, figure 4.16, figure 4.17 and figure 4.18. It is evident from these figures that the signal has not been correctly received, potentially due to the FSM being temporarily out of focus, therefore these traces are omitted from the analysis. A comparison of the ToF measurements before and after manual correction are depicted in figure 4.19.

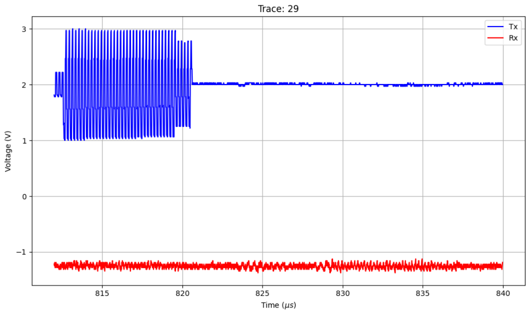


Figure 4.15: Trace 29 of the oscilloscope measurements, top representing the transmitter, bottom representing the receiver.

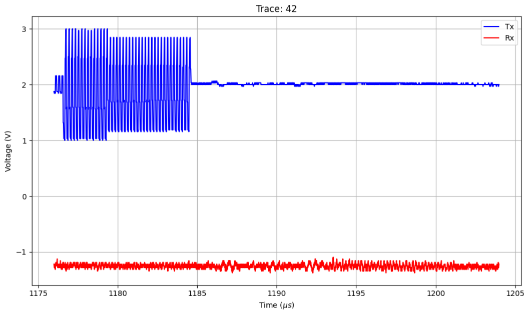


Figure 4.16: Trace 42 of the oscilloscope measurements, top representing the transmitter, bottom representing the receiver

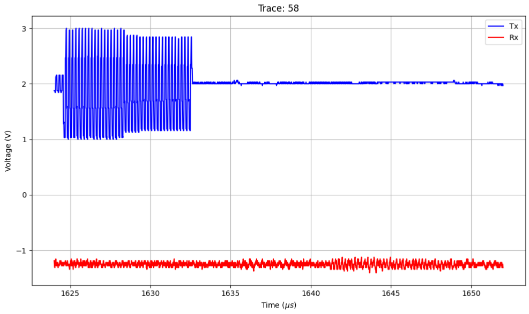


Figure 4.17: Trace 58 of the oscilloscope measurements, top representing the transmitter, bottom representing the receiver

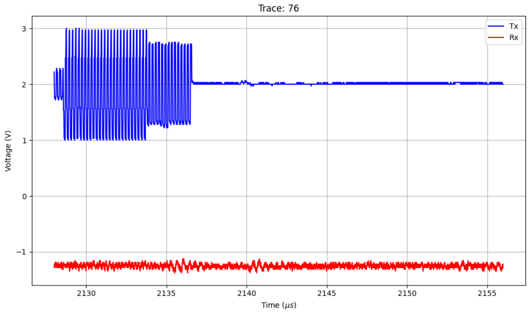


Figure 4.18: Trace 76 of the oscilloscope measurements, top representing the transmitter, bottom representing the receiver

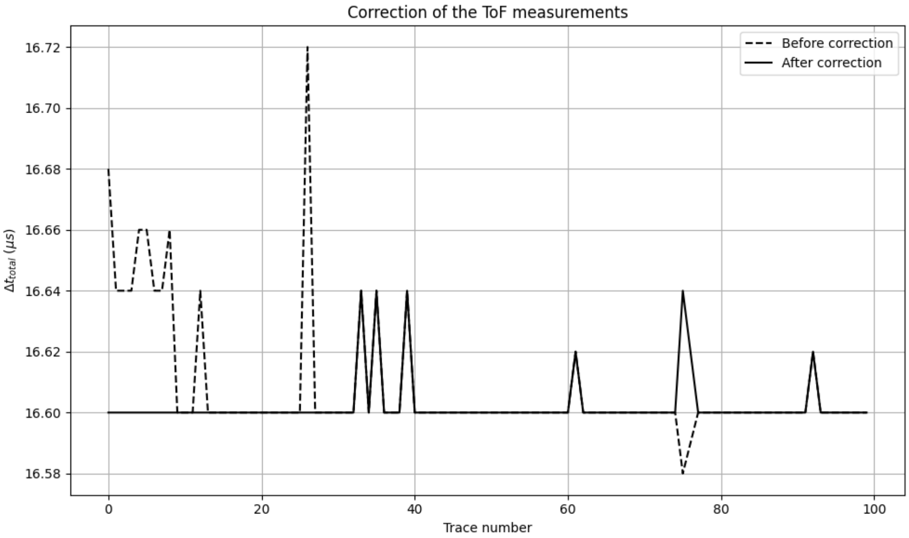


Figure 4.19: Comparison of ToF measurements before and after manual correction of the outliers

The number of outliers is considerably reduced after correction, increasing the accuracy of the ToF measurement. The results of this process are then substituted into Equation 4.7 for Δt_{total} to calculate the calibrated time of flight, which can be converted into the measured range using Equation 4.8, resulting in the plot from figure 4.20.

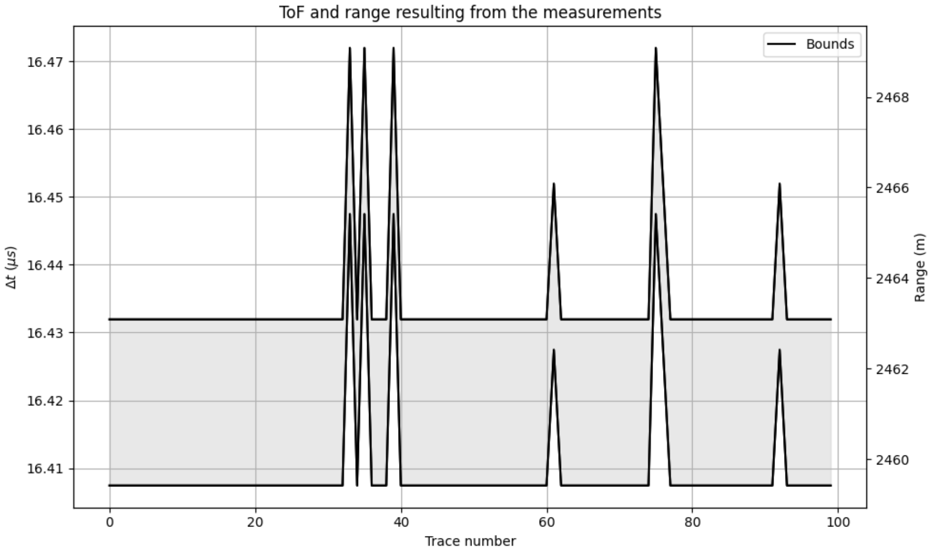


Figure 4.20: The two-way ToF and the corresponding range calculation resulting from the measurement. The shaded area in between the curves are the possible distances due to the unknown exact length of the optic fiber

These calculations result in an average between 2459.73 m and 2463.40 m with a standard deviation of 1.26 m. This is approximately 20 m from the target range of 2442 m. Assuming that these delays can only occur from the system electronics, of which the propagation speed is roughly $0.7c$, the approximate time that is unaccounted for is 95.3 ns or 28 m.

4.5. Conclusion

A free-space laser ranging experiment was conducted in collaboration with TNO. A laser terminal and receiver were placed on the Meteotoren in Scheveningen, pointing at a retro-reflector located on top of the TNO building in The Hague, Oude Waalsdoperweg. The primary objective of TNO with the set-up was not to perform any ranging, hence the measurements made may not be precise due to lack of calibration required for range measurements. A top-level sketch of the test set-up is illustrated in figure 4.1.

The laser transmits a set of pulses referred to as pulse trains, generated by a function generator. These pulse trains consist of exactly 40 pulses each, at a frequency of 5 MHz with a duty cycle of 25%, the total duration of the pulse train is 8 μs after which nothing is transmitted for 20 μs. Within these 20 μs a reflection of the pulse train is expected to arrive on the receiver. A pair of transmitted and received waveforms is called a trace. In total the experiment consists of 100 such traces. An oscilloscope with a sampling rate of 50 MS/s is attached to the function generator and to the receiver capture these traces.

Naturally the receiver data is encapsulated by a large amount of noise due to background lighting, while the transmitter data is less noisy. Because of this, the data needs to be filtered before analysing the results. Several methods were tried, such as a Butterworth to filter out the components attributing to the noise, doing so however stretches the data beyond recognition. Instead, a Gaussian distribution filter is used on both signals.

For the transmitter signal, all the values are arranged into an large array from which the average and standard deviation are calculated. Due to the high SNR, a smaller z-value ($z = 1$) can be taken to distinguish between signal and noise. The receiver signal is approached differently as compiling all data points in an array subjects the average and standard deviation to over-average the noise and signal components, making it increasingly difficult to distinguish fainter reflections on the receiver. Hence, each trace is evaluated individually, resulting in 100 separate Gaussian distributions. For each Gaussian distribution, a z-value of $z = 1.7$ was found to be sufficient to differentiate between signal and noise on the receiver.

After applying the filter, it was found that several outliers remain in the data and Δt_{rise} and Δt_{fall} do not coincide as is desired. Some manual intervention was required to correct for the low SNR in the places where these computations were performed. Four traces in total were unrecognizable and discarded for the analysis. The main reason for the misalignment between the rising and falling edges is primarily due to the trigger position of the oscilloscope being set on the first rising edge of the transmitted pulse.

There are several known attributes that contribute unwanted delay to the signals trajectory, such as the EDFA and various coaxial and optic fiber cabling in the system. A rough estimate on these values is known, but an exact value on what is required for calibration is not. The total added delay by the cabling and EDFA ranges from 168.13 to 192.66 ns.

In general, the computed range is between 2459.42 m and 2463.09 m, roughly 20 m from the target range of 2442 m. Measurements have a standard deviation of 1.26 m, making them relatively accurate. The precision of the system however is poor as was anticipated as its primary purpose was not to perform ranging.

Hence no steps were taken to calibrate it exactly this purpose, nor was there the time to do exactly this. The time residuals that is unaccounted for is 95.3 ns. This can likely be attributed to the internal delay caused by the system electronics. For the tests that follow, these will be the first steps that are important towards obtaining an accurate range measurement. Furthermore a major limiting factor that needs to be improved upon for a repetition of this test and the test in this thesis that follows is the increase of measurement speed. Currently it is set to 50 MS/s, this has proven to be limited. Which can be improved upon by employing an oscilloscope with a larger bandwidth in this setup, or by switching to an FPGA which is capable of handling data at much higher rates as well as allowing for communication to occur.

This chapter aims to answer the first sub-question:

How can laser ranging be achieved using a communication system?

The answer to this sub-question is that, obtaining trivial readings from a laser over free-space involves a lot of equipment that allow for this to happen. Such as an FSM, EDFA and the variety of equipment and wiring TNO has put in place that are not elaborated upon in detail.

The ranging itself can be done by simply looking at the arrival times of the transmission and reception pulse trains. When this is done, the measured range is relatively constant with only a standard deviation of 1.26 m caused by the different starting and end times of the pulse train. The residual of the measurement however is 20 m due to improper calibration due to the lack of time scheduled for the test.

4.6. Discussion

Overall this test shows promising results. The SNR of was exceptionally high in 97 out of 100 cases, the other cases where the SNR is too low could be attributed due to the FSM being out of focus at just the right moment, although this is speculative as there are many variables involved in free-space ranging, where even the slightest vibration could cause the system to go out of focus.

The measurements are stable and rather deterministic with only 6 outliers out of the remaining 97 traces. Taking a closer look at these traces, they concern cases where the SNR is low, making the start of a received pulse train difficult to distinguish, even with manual human intervention. Which influences the results, but is part of reality.

There is roughly 20 m of range that is not resolved for, which is reasonable to assume that this can be attributed to the internal electronics of all the devices involved in making the measurements, re-iterating the importance of calibration.

One obvious improvement of this test would be to swap the oscilloscope with one having a larger bandwidth. However considering the unstable power source could potentially destroy this sensitive equipment, it may become extremely costly. One way to mitigate this, is to use an uninterruptible power supply in between the diesel generator and the equipment to protect such sensitive equipment against large fluctuations in power. These devices are often used in countries where the power grid is unstable, and sensitive electronics need to be protected from power surges, brown outs, or black outs.

Although this test does not demonstrate laser communication, it does demonstrate the order of magnitude accuracy that can be achieved without any form of communication. It therefore highlights basic constraints that are in place for an optical link, namely the measurement rate and calibration. Furthermore it demonstrates that in order for having a good SNR over on the receiving end takes a lot of complex equipment, the inner workings of this are outside of the scope of this thesis.

To continue to implement laser communication ranging instead, as well as achieve higher sampling rates, switching to an FPGA as test equipment would be the next step.

4.7. Recommendations

The following recommendations are made to improve a repeated experiment with this test set-up or something similar to conduct laser ranging.

- **Calibration**
  Proper calibration is required to get a more precise range measurement. This can be done by placing a mirror, something comparable, relatively close (~10-20 cm) to the transmitter and receiver. This way the trajectory of the photons in free-space is negligibly small compared to trajectory towards the retro-reflector. The value obtained can then be subtracted from all future measurements to recover the actual RTLT. It also removes the assumptions that need to be made on the refractive index of all the optic fiber cabling as well as the propagation speed of all coaxial cables.
- **Increase sampling rate**
  The sampling rate of the oscilloscope needs to increase. The accuracy of the measurement is ultimately limited by the uncertainty due to the sampling period of the device used to observe the signals. The uncertainty is different from the standard deviation of the range measurement. Taking the results from section 4.4 as an example. The experiment is in essence repeated a total of 100 times, resulting in a standard deviation of 1.26 m. This value essentially deceases if the experiment is repeated. The uncertainty remains roughly 6 m unless a higher sampling frequency is acquired.
- **Movable retro-reflector**
  It is recommended the retro-reflector is movable by more than $\frac{c}{f_s}$, preferably on a rail of which the velocity can be accurately controlled by for example a stepper motor. The system in essence can measure the range in with a resolution of $\frac{c}{f_s}$, if the retro-reflector moves with towards or away from the source with a magnitude larger than this value, it will be distinguishable from the measurements. This would add scientific relevance as now the range rate can be computed.
- **K-Nearest Neighbors detection algorithm**
  The current detection algorithm works fine in general, but has troubles identifying edge cases. Moreover the final result has to be revisited by the user to manually correct outliers. K-Nearest Neighbors (KNN) is a non-parameteric supervised learning classifier algorithm. This machine learning algorithm can be trained to automatically detect pulses with a potentially exceptional accuracy. For the current data set this is impractical due to the relative small sample size, however in the future if the test were to be repeated it is recommended to spend additional time in developing such an algorithm as it can prove to be an efficient and fast way of accurately determining the ToF.
- **Realize an optical communication data stream**
  To make the test more functional with respect to testing satellite laser communication ranging performance, instead of transmitting pulses, a communication protocol could be established to analyse the effects of free-space on the data structure. Currently, the start and end times of pulse trains are relevant in computing the ToF. These pulse trains experience losses and variable SNR, which may influence the amount of ToF measurements that can be extracted from a data stream. This could be realized by employing an FPGA to take care of the optical signalling.

III

FPGA Laser Communication Ranging Experiments

5 Internal FPGA Signal Delay Measurements

From chapter 4, the main points of improvements were to properly calibrate the system, increase the measurement rate and implement communication with the use of an FPGA. This chapter aims to calibrate an FPGA for range measurements by measuring the internal delay, as well as increase the measurement rate. This will be a precursor to the implementation of a communication stream.

First the objective and goals of this experiment are presented in section 5.1. Next in section 5.2, the system setup and design is presented. This is followed by a detailed explanation of the methods used in section 5.3. In section 5.4 the system verification is presented where the system as a whole is verified and the first performance metrics resulting from a post-implementation functional simulation are displayed. After which in section 5.5 the system is validated and the performance metrics are further analysed on the physical hardware. The results of the tests are presented in section 5.6. This is followed by a conclusion, discussion and recommendations for future work in section 5.7, section 5.8 and section 5.9 respectively.

5.1. Objective and Goals

From the free-space laser ranging test presented in chapter 4 several recommendations were mentioned and points of improvements were discussed throughout the chapter. The three points that are taken into consideration in this chapter are to, first of all employ an FPGA for signal generation, secondly calibrate the system for ranging and increase the measurement rate.

Hence the experiments conducted with this system in this chapter aim to fulfill these points and to provide a stable basis for the experiment that follows in chapter 6.

The objective of this chapter is to attempt to answer the second and third research questions listed in section 1.2.

1. What methods can be utilized to generate a stable and consistent signal on an FPGA?
2. How can the internal signal propagation delay within the FPGA be quantified?

The goals resulting from the objectives are to:

- Familiarize programming an FPGA.
- Instantiate the proper internal clocking mechanisms.
- Create basic signals using FPGA hardware.
- Extract signal data from the FPGA.
- Perform post-implementation functional simulations using Vivado ML.

5.2. System Design

In this section, the system design and the steps taken to create the design are discussed. First, a small part is dedicated to defining an FPGA in subsection 5.2.1. This is followed by the system model in subsection 5.2.2. In order to conduct tests, a method of acquiring data must be devised, this is described in subsection 5.2.3. In order to measure a delay DDMTD is employed and discussed in subsection 5.2.4. Finally the clocks driving the system are presented in subsection 5.2.5

5.2.1. Field-Programmable Gate Array

As mentioned an FPGA is used for conducting this experiment and the experiments that follow. An FPGA is a semiconductor device that stands at the intersection between hardware and software in digital electronics. It provides a clean slate consisting of an array of programmable logic gates and interconnects after manufacturing. This provides the user with the ability to program custom digital systems and functions in the field, effectively redefining its purpose even after deployment. This capability has made FPGAs a vital component in numerous applications, such as telecommunication, aerospace, medical devices, and more.

FPGAs are primarily utilized in scenarios where adaptability and rapid prototyping are essential. They serve as an ideal platform for designing and implementing complex digital systems, as their functionality can be altered by configuring the internal logic gates using a Hardware Design Language (HDL) such as VHDL or Verilog. The ability to rewrite these configurations without the need for extensive hardware modifications or redesigns makes FPGAs indispensable for tasks like real-time signal processing, image and data manipulation, and rapid development of custom digital solutions.

5.2.2. System Model

Figure 5.1 illustrates a top level system model diagram of the first experiment with the FPGA that is conducted. Starting with the clock mechanisms in the green section, the PLL is responsible for generating the simple square clock signal by phase-locking it to a system clock. This signal is then simultaneously routed to the PMOD pins, a helper PLL and the DDMTD firmware.

Routing the square wave signal to one of the PMOD pins introduces a delay to the signal due to the path it needs to travel. The pins are then bridged using a wire allowing for the signal to travel to the other pin, which is in turn routed back into the FPGA. The phase between the transmitted signal and the received signal can then be compared by a phase frequency detector as the phase difference is directly related to the delay the signal experiences. A benefit of using these PMOD pins is that they belong to the GPIO group of the FPGA board, meaning that they are directly accessible, providing the user with direct control over what happens to the signals.

The phase frequency detector technique that is used is DDMTD which is independent of the system clock and yields highly accurate results. Hence both the signals are routed to this module in the firmware, allowing for this phase measurement.

The helper PLL is responsible creating a phase-locked signal with a lower frequency, but close to the original clock signal, in turn provided to the DDMTD. The DDMTD outputs two slow beating clocks that are provided to the data capture unit. The data capture unit is connected to a PC through a Joint Test Action Group (JTAG) connection. The two beating clocks can be analysed and the phase difference can be calculated, which directly corresponds to the internal delay.

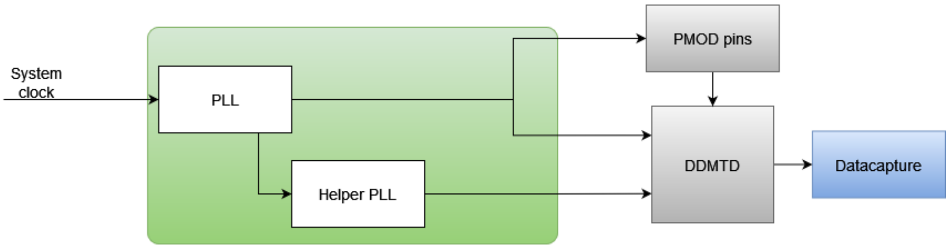


Figure 5.1: Top-level system model of the first experiment

5.2.3. Data-Acquisition Firmware

Retrieving signal data from the FPGA is non-trivial. An FPGA is not considered to be a Data Acquisition (DAQ) unit, rather it is used for a variety of signal processing applications. These signals are usually transferred, through a variety of connectors, such as coaxial cables, FPGA Mezzanine Card (FMC) connectors or optical fibers, to a dedicated DAQ for analysis. Accompanying this is non-standardized firmware that has to be written by the user in detail including clock and MGT configuration or Advanced eXtensible Interface (AXI) communication over the serial bus. This requires advanced knowledge of FPGA programming and dedicated hardware to act as a DAQ.

In order to circumvent this obstacle, research is conducted on alternatives to retrieve data in a simpler way, without requiring the complexity involved with a custom serial data stream. From this research, it was determined that MATLAB offers an add-on called `HDL Verifier`, which includes a firmware generator. This automatically generated firmware can be incorporated into user-written firmware to directly capture signal data from the FPGA core. This can transferred through a JTAG USB connection or Ethernet connection to a PC.

The method of using the JTAG connection does not require advanced knowledge about configuring complex communication steams. However, the Ethernet alternative requires configuration of the whole MGT interface on which it is located. For convenience, the JTAG connection is chosen.

MATLAB generated firmware is either VHDL or Verilog depending on the preferences of the user. It can be used in two ways depending on the firmware design. Some firmware designs, such as this particular piece of firmware, can be represented in a block diagram. An illustration of this is presented in figure 5.2. This code is then incorporated in the user code, and the desired signals to capture are connected to this firmware core. After this, a MATLAB instance can be initiated and a window of data can be captured. After capturing the signal data it is exported to the Logic Analyzer from MATLAB, as illustrated in figure 5.3, from which it can be exported to a MATLAB file.

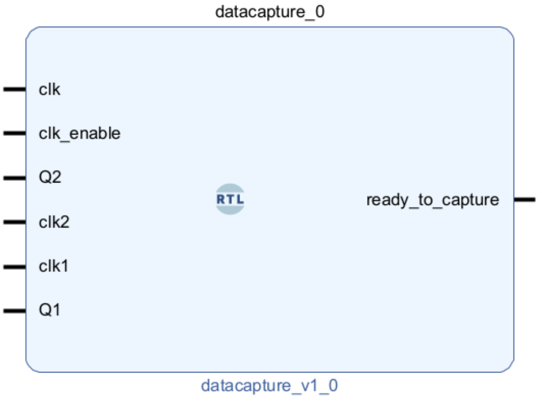
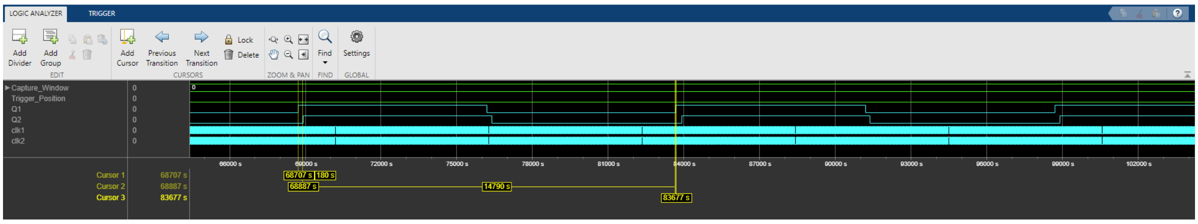


Figure 5.2: Datacapture block in Vivado ML 2021.2, the left side indicates all the inputs, the right side indicates the outputs.



▶ Capture_Window 0
Trigger_Position 0
Q1 0
Q2 0
clk1 0
clk2 0

Cursor 1 68707 s
Cursor 2 68887 s
Cursor 3 83677 s

Figure 5.3: Example of captured data using the HDL Verifier add-on from MATLAB

The data capture firmware for this experiment has six inputs, two of which are standard inputs next to the signals the user wants to capture and one output. These are defined in table 5.1.

Signal	Description
Q1	The first beating clock signal.
Q2	The second beating clock signal.
clk1	The transmitted signal.
clk2	The receiving signal.
clk	The sampling clock, provided by the user, used to sample the signal data at each rising edge of this particular clock.
clk_enable	A toggle to internally enable the data capture clock.
ready_to_capture	A signal that indicates whether the data capture firmware is ready to capture.

Table 5.1: Inputs and output for the MATLAB data capture IP implemented in the design

The most relevant signal being the clock signal, which determines the sampling rate of the data capture. There are limits, however, to the magnitude of the sampling frequency. It was fist hypothesized that the data is directly streamed through JTAG to a PC, which means that the limit of this method would ultimately be the component with the most limiting bandwidth of the USB connection. According to MATLAB, for Xilinx FPGAs, the JTAG ticking frequency is between 33 MHz and 66 MHz [1]. Under the aforementioned hypothesis, a maximum sampling frequency of 66 MHz would be achievable.

However, this did not appear to be the case as validation tests showed that the clocking frequency can go beyond this frequency, however, it does become increasingly unstable for frequencies beyond 200 MHz, hence from here on out the sampling frequency is set to 200 MHz. It was found that the type of connection did not matter in terms of sampling bandwidth as it is likely sampled by the FPGA itself and stored registers in Block Random Allocated Memory (BRAM), from which data is transferred to the PC at a frequency between 33 and 66 MHz.

The data capture IP operates comparable to an oscilloscope: a window of a set number of points measured around a set trigger position. Signals can be connected to the data capture and the capture window size and sampling frequency set. In this process, the data capture consumes a significant amount of BRAM depending on the amount of signals, the data width of these signals and the amount of samples per window. For large signals consisting of multiple bits, large window sizes may not be allowed.

When setting these parameters, a healthy balance between them is required so as to not exceed physical BRAM limits. Due to the limited number and width of signals, there is considerable flexibility in determining the window size. Hence, for this experiment, the maximum capture window of 1048576 samples is chosen.

Due to the instabilities that occur when sampling at high frequencies, use was made of the PMOD pins on the board. This allows external devices, such as an oscilloscope, to independently measure the signals when desired. The necessary signals to acquire a phase delay are routed in parallel to the PMOD pins of header J53 on the KCU105. `PMOD1_1_LS`, `PMOD1_5_LS`, `PMOD1_2_LS` and `PMOD1_6_LS` are connected to Q_1, Q_2, clk_1 and clk_2 respectively. The layout of this header is illustrated in figure 5.4.

[1] https://nl.mathworks.com/help/hdlverifier/xilinxfpgaboards/ug/hdlverifier.fpgadatareader-system-object.html, [accessed on 7-9-2023]

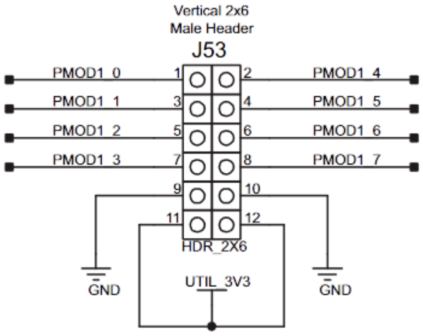


Figure 5.4: J53 header schematic of the KCU105 [23].

5.2.4. DDMTD Phase Detection Firmware

A DDMTD is in essence a digital phase frequency detector. It detects the phase of two square wave signals of the same frequency with respect to each other. The reason why this technique is preferred is that it allows for the measurement to be accurate below what the sampling clock permits. Lately this technique has been often employed by CERN in various networking applications to achieve phase stability in network solutions [24][8]. One way to achieve this phase stability is in part by measuring the phases of two clocking signals and using the resulting signals as a feedback to other components.

Primitively a DDMTD consists of three components, two DFFs and a PLL. The general schematic layout is illustrated in figure 2.7 from [20]. This picture also depicts a deglitcher and a time difference computational component. These are needed whenever all the signal processing needs to happen on the FPGA silicon, for example whenever the DDMTD is part of a bigger component such as TCLink [25]. For this experiment, this is excluded from the FPGA firmware, rather this is done in post-processing as DDMTD is used as the final means of measuring the phase. The helper PLLt is phase-locked to the frequency of the transmission signal (10 MHz), its output frequency is aimed to be close, but not equal to that of its incoming frequency. Preferably the helper PLL generates a signal that is within 1 Hz of the incoming signal, resulting in a high order of accuracy [8]. Within the category of phase locking mechanisms from the FPGA the user has two choices, a PLL and a Mixed-Mode Clock Manager (MMCM). The PLL works with an integer divider that has a value between 1 and 128. Resulting in high frequency differences between the incoming signal and the output of the PLL, normally being between 500 kHz and 1 MHz, resulting in a poorer accuracy.

A MMCM operates the same as a PLL, but works with floating values of 0.125 multiples instead of whole integers. Naturally, this allows the helper PLL frequency value to be closer to the incoming frequency, usually within a 100 kHz range of the incoming signal frequency. To get to a better result however, N-values in the range of 10,000 to 100,000, calculated using Equation 2.5, are required [8]. The frequency resulting from the helper PLL is 9.98 MHz which results in an N-value of 499.

The expected beating frequency of this system will be as follows:

$$f_{beat} = f_1 - f_{ddmtd} = 20 \text{ kHz} \tag{5.1}$$

Given this and a sampling frequency of 200 MHz the minimum discernible time difference resolution that is achievable by this system can be calculated. The minimum time difference between the two beating clocks that can be distinguished is equal to one clock cycle of the sampling clock, which is 5 ns. This results in the following minimum time difference that the system can measure and is calculated by substituting the aforementioned variables in Equation 2.7.

$$\Delta t(t) = 5 \cdot \frac{20 \cdot 10^3}{10 \cdot 10^6} = 0.01 \text{ ns} \tag{5.2}$$

The calculation from Equation 5.2 shows that this aforementioned resolution is 10 ps, which translates to a ranging resolution of about 3 cm. This could be improved by solely using an PLL mechanism that is capable of phase locking a signal with a frequency closer than currently is achievable with the on-board PLLs and MMCM.

Hence for projects such as White Rabbit [8], an external DDMTD PCB is made with a dedicated PLL that aims to closely lock on to the incoming clock signal. The method of employing an MMCM for DDMTD is not uncommon and is done by CERN [25], where they hit the same limitations.

5.2.5. System Clocks

The system is designed with three clocking mechanisms to get the appropriate signals. A block diagram of the clocking signals is illustrated in figure 5.5. In order to elaborate further of the system clocking, the description will follow all clocking mechanisms illustrated in the picture starting from left to right.

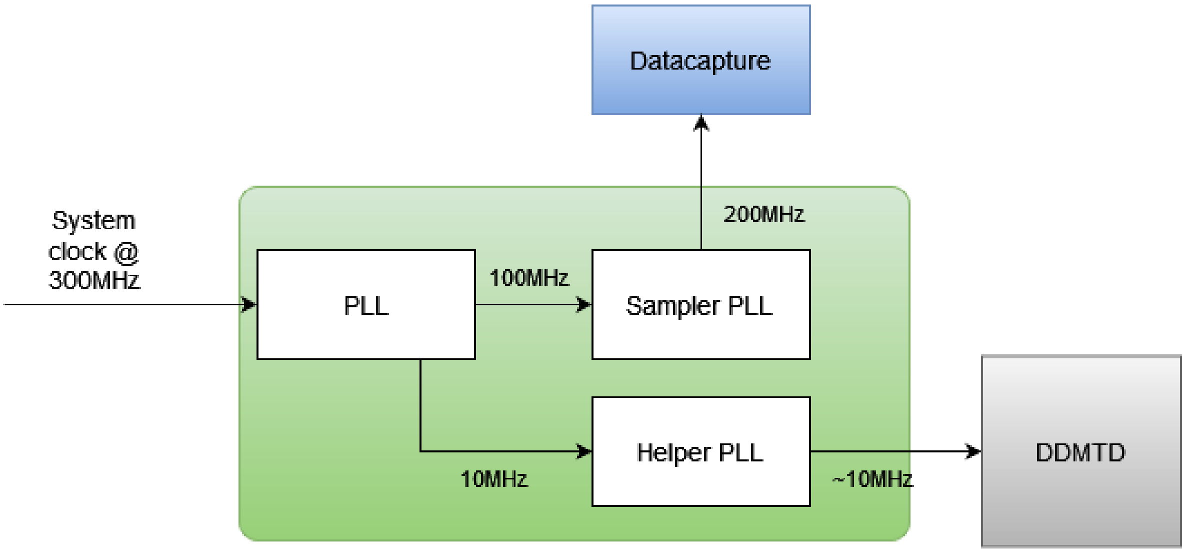


Figure 5.5: Block diagram of the clock signals in the system firmware for the first experiment

First the default system clock is provided as driving input to the firmware at a frequency of 300 MHz. This is provided to a PLL that reduces the frequency to the lowest achievable internal frequency given its input of 10 MHz and a clock signal with a frequency of 100 MHz.

The helper PLL is an MMCM, which takes the signal and phase locks it to the closest possible frequency of 9.98 MHz, providing this to the DDMTD. The sampler PLL is also an MMCM, up-scaling the signal to 200 MHz.

5.2.6. General Firmware Description

An overview of the entire firmware as it is programmed in Vivado ML 2021.2, is illustrated in figure 5.6. This block diagram is discussed from left to right to provide the reader with a clear overview of the signal flow.

Starting with the input signals on the left of the figure. First the `reset` signal comes from the on-board Central Processing Unit (CPU) reset. This is a reset signal that is initiated when power cycling the FPGA, or pressing the physical button. It is connected to the clocking mechanisms as these are the main driving parts of the firmware.

Secondly the default 300 MHz system clock is connected to the main clocking wizard, which uses it to derive the other necessary signals. Finally the `PMOD1_4_LS` is one of the PMOD pins at which the clock signal is routed back into. This is connected to the `PMOD_route_0` module, which responsibility it is to route this signal to the DDMTD.

`clk_wiz_0` is responsible for the generation of two clocking signals, `clk_out_1` and `clk_out_2`. `clk_out_1` is the 10 MHz signal and `clk_out_2` is a 100 MHz signal. The `locked` signal is an indicator whether or not the PLL is phase locked. `clk_out_1` is connected to the `helper_PLL` as an input to phase lock onto, `DFF_0` as part of the DDMTD, to the `PMOD_route_0` to be attached to one of the PMOD pins, to the `scope_probes_0` for external access and finally to `datacapture_0` to capture it digitally through MATLAB. `clk_out_2` is provided to the `Sampler_PLL`. Which phase locks a 200 MHz signal to be provided to the `datacapture_0` module. `locked` is routed to an AND gate. This AND gate combines all the `locked` signals from all the phase locking mechanisms.

The `helper_PLL` phase locks to the 10 MHz incoming signal and generates a signal that is slightly lower in frequency. This clocking wizard as well as the `Sampler_PLL` provide `locked` signals that are combined with an AND gate, which output is negated, technically making it a NAND gate. The output of this NAND gate is used as the reset of the DFFs, which will only be activated after all the clocking mechanisms have phase locked onto the incoming signal.

The DFFs are clocked by the `helper_PLL`. `DFF_0` has the transmitted 10 MHz signal as its input and `DFF_1` has the received 10 MHz signal as its input. This way the phase between these two can be compared in order to obtain time delay measurement. The outputs are routed to the datacapture module where they are exported to be read through MATLAB.

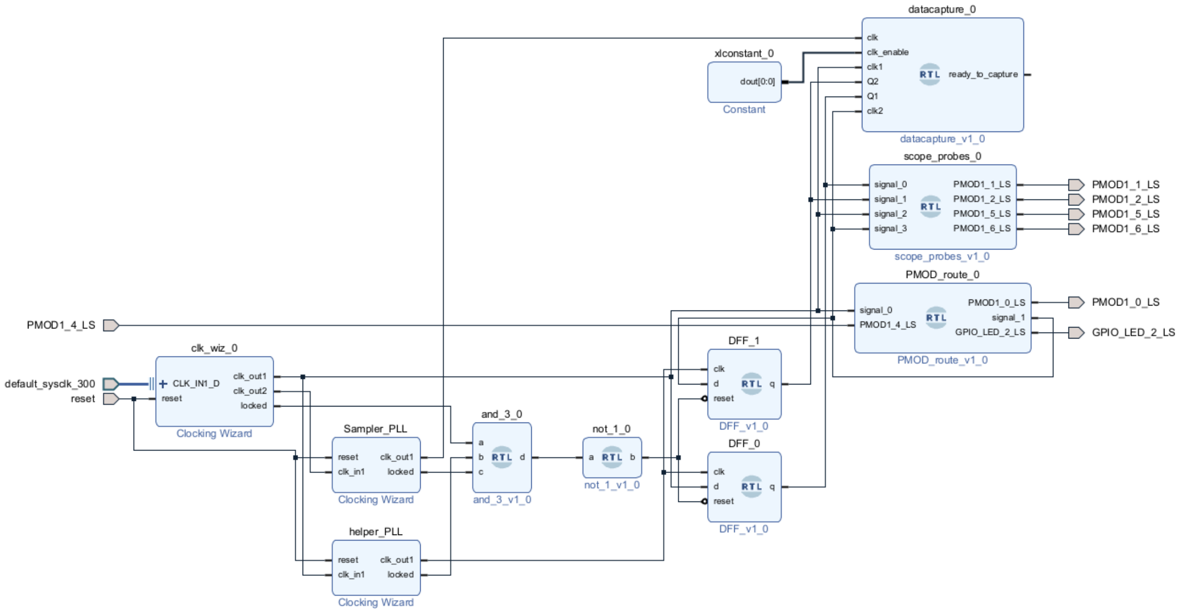


Figure 5.6: Block diagram of the firmware with the signal connections

5.3. Method

In order to map this internal delay, the type of signal transmitted, the method of measuring this delay and where to route this signal needs to be devised. Starting with the type of signal in subsection 5.3.1, the aim of it is to be as simple and straightforward as possible. This means a square wave signal with a low frequency that is allowed by the FPGA. Then the method of determining the phase difference is described in subsection 5.3.2

5.3.1. Signal type

An FPGA operates naturally on relatively high frequencies in the order of several hundreds of megahertz, often orders of magnitude higher than encountered within space communication, which is often in the range of several kilohertz. To generate these frequencies it uses crystal oscillators that are soldered on the board. These oscillators cannot go to arbitrarily low frequencies, thus the minimum achievable frequency on board is 10 MHz. As previously mentioned the signal from which the delay will be measured concerns a square wave with this frequency. The signal can be expressed mathematically by Equation 5.3, where f_1 is equal to 10 MHz.

$$x_1(t) = \mathrm{sgn}(\sin(2\pi f_1 t)) \tag{5.3}$$

Throughout this chapter, $x_1(t)$ will be referred to as the transmission signal.

5.3.2. Phase Detection

To effectively measure the delay of this signal, a phase detector must be programmed within the firmware. There exist several digital phase detectors that can be employed, each having advantages and disadvantages. The discussion of advantages and disadvantages is beyond the scope of this thesis. From the literature study conducted, the DDMTD is an exceptionally accurate method, capable of femtosecond level accuracy when configured appropriately [12] [8]. The inner workings of the device are presented in section 2.4

A DDMTD can be synthesized on the FPGA to measure this phase delay. The firmware design of this method is discussed in section 5.2. The system will have as input the transmission signal and the receiving signal $x_2(t)$, which is the transmission signal that has traveled through the internal electronics of both the FPGA and the PCB. The system's output consists of two oscillating clocks, which exhibit a significantly lower frequency compared to the input. This is illustrated in figure 2.8 taken from [10].

In figure 2.8 two important aspects are shown, $\Delta t(t)$ and $\Delta t_{beat}(t)$. The desired parameter from the experiment is $\Delta t(t)$, which is calculated though $\Delta t_{beat}(t)$ using Equation 2.7.

This in allows for a timing resolution that is orders of magnitude smaller than the potential sampling frequency of the system. For this system alone, the sampling clock allows for a resolution of 5 ns which is reduced to 0.01 ns, which is 500 times smaller.

5.3.3. Signal Routing

In an attempt to mimic the internal delay a signal would experience, first the physical location of the components must be considered. When communication is enabled, the signal will travel to the appropriate transceiver on the board. On the KCU105, the eventual laser signal will be sent through the SFP transceiver on the board. To mimic this, the aim is to get the signal as close to the SFP module as permissible.

5.4. Verification

In this section the verification of the firmware described in section 5.2 is presented. A block diagram of the firmware is illustrated in figure 5.6. First in subsection 5.4.1, system tests are performed with the use of a post-implementation functional analysis. Next, in subsection 5.4.2 the performance of the DDMTD is tested in the simulation.

5.4.1. System tests

The firmware from section 5.2 is simulated using Vivado ML Enterprise Edition VHDL simulator 2021.2 with the use of a testbench. The signals that are expected will be provided as input to the system, and the simulated output will be observed. The firmware block diagram used for verification is illustrated in figure 5.6.

A testbench is module writtin in VHDL or Verilog that virtually provides the inputs to the piece of firmware that has to be tested such that it can be simulated using a waveform simulator. For this firmware the differential clock input of the FPGA runs at a frequency of 300 MHz, hence the same values will be provided to the simulation. The CPU reset signal is forced to a constant "0" value, to enable all the clocking mechanisms. These are the only inputs that are required for the simulation to function.

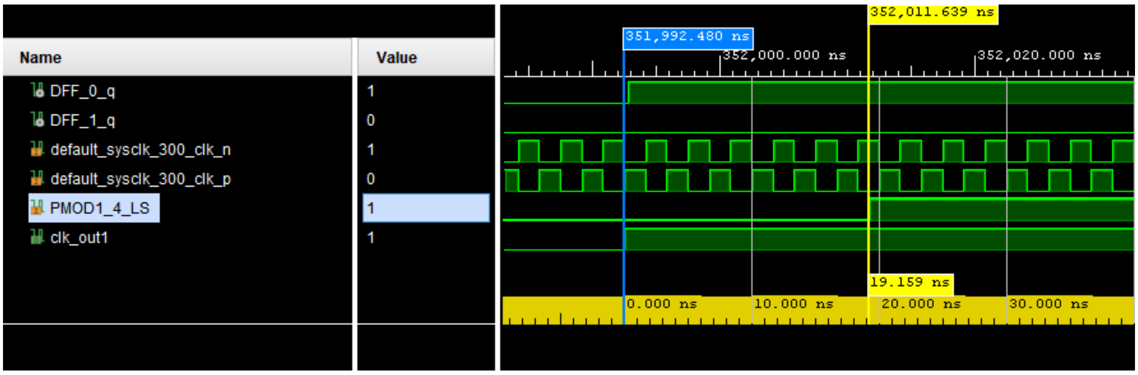
To start, it must be noted that instead of using the dedicated SFP pins to create an internal delay, the PMOD pins are used instead. As mentioned in section 5.2, Vivado did not allow to manually interface with MGT hardware.

Vivado allows for a post-implementation functional simulation after user code is written, synthesized and successfully implemented. This allows for a well-validated simulation to take place, specifically catered to the selected board within the software itself. The results of this simulation can be considered to be a fair representation of reality.

First, the simulation will run for 10 µs. It is expected that there is no phase delay as the PMOD pins are not digitally bridged yet. Next `PMOD1_4_LS` is forced to a clock with an arbitrary phase difference in comparison to `PMOD1_0_LS`. The simulation is run for another 10 µs.

After running for a total of 20 µs, no output is registered. The reason for this is that the PLL responsible for locking onto the f_1 frequency at a frequency of f_{ddmtd} takes a relatively long time. After roughly 64 µs the helper PLL appears to lock and the flip-flops are providing an output.

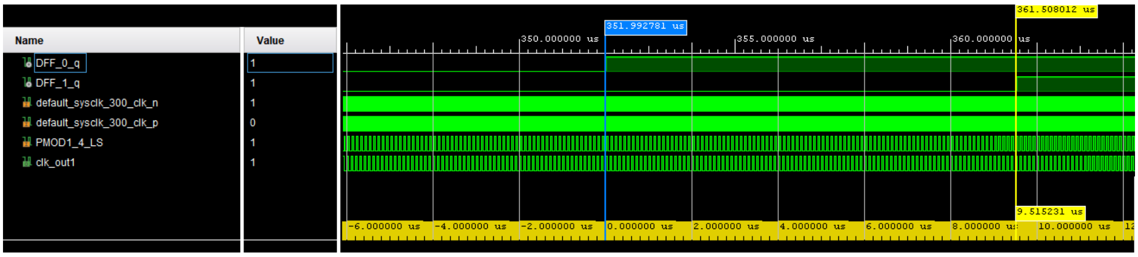
Observing the signals shows that the phase delay created is roughly 19.159 ns, as illustrated in figure 5.7. When using the other signals, as input of Equation 2.7, the outcome should be practically the same.



Name	Value
DFF_0_q	1
DFF_1_q	0
default_sysclk_300_clk_n	1
default_sysclk_300_clk_p	0
PMOD1_4_LS	1
clk_out1	1

Figure 5.7: Signal delay simulation for verification. The arbitrary phase delay between the two signals being 19.159 ns.

Observing the outputs from the DFFs, illustrated in figure 5.8, the $\Delta t_{beat}(t)$ at an arbitrary point in time is 9.515 231 µs.



Name	Value
DFF_0_q	1
DFF_1_q	1
default_sysclk_300_clk_n	1
default_sysclk_300_clk_p	0
PMOD1_4_LS	1
clk_out1	1

Figure 5.8: Time difference between the two beating clocks in a simulation

Substituting the obtained values in Equation 2.7, the delay can be computed resulting in the following:

$$\Delta t(t) = 9.515231 \cdot \frac{20 \cdot 10^3}{10 \cdot 10^6} = 19.03\ ns \tag{5.4}$$

The output has an absolute error of 128.54 ps translating to a relative error of about 0.7%. Furthermore, a slight deviation in the frequency f_1 is observed in the simulation. After calculating the frequency from the observed period of the waveform, instead of having a frequency of 10 MHz it has a frequency of 10.001 000 1 MHz. This is likely due to the nature of the MMCM responsible for generating this signal, a steady state error is present and expected. An MMCM can be compared to a closed-loop control system, where a steady state error is common.

The performance of the DDMTD firmware will be further investigated in both subsection 5.4.2 and subsection 5.5.2. The system has undergone the tests and they were successful.

5.4.2. Performance Tests

When observing the actual phase difference between in figure 5.7 and comparing that to the results from Equation 5.4, an absolute error exists even in simulation. Therefore this error is expected to persist during the firmware implementation on the FPGA. Consequently, it is crucial to generate results that can be compared with direct measurements obtained from the FPGA itself. In order to attempt to map the expected error through a simulation, the following strategy is employed.

First, two signals are programmed into the firmware, each having their own frequency f_1 and f_2 respectively, both equal to 10 MHz. Through the internal PLL, `clk_wiz_0`, a phase difference between the signals can be requested.

In order to map the performance, a range of phases are requested and subsequently compared. These phase-delays range from 90°down to 5°in 10°intervals, with extra steps at 45°and 5°. The simulation is run for 300 µs and the results are observed. These results can later be used for comparison during validation, as the same tests will be performed on the physical hardware.

Table 5.2: Results from the verification of the simulation using different pre-determined phase delays

Desired $\Delta\Phi$ (°)	Produced $\Delta\Phi$ (°)	True Δt (ns)	Measured Δt (ns)	$\Delta\varepsilon$ (ps)
90	90.00	25.000	24.850	150.299
80	80.25	22.292	22.244	47.176
70	69.75	19.375	19.238	136.522
60	60.00	16.667	16.633	33.399
50	50.25	13.958	13.828	130.677
45	45.00	12.500	12.425	75.150
40	39.75	11.042	11.022	19.620
30	30.00	8.333	8.216	116.900
20	20.25	5.625	5.611	13.775
10	9.75	2.708	2.605	103.123
5	5.25	1.458	1.403	55.527

From table 5.2, it can be observed that the error ranges from as high as 150.299 ps to as low as 13.775 ps. It is expected that, when the firmware is run on physical hardware, the error will increase due to internal jitter of the utilized PLLs and MMCM and other logic and electrical components on-board as well as other noise artifacts.

When the phase detection error remains consistent for a specific type of phase delay, this error can, in theory, be calibrated for in post-processing. Therefore, mapping these errors is crucial to improve system performance. A similar test will be performed on the physical hardware and it will be discussed in subsection 5.5.2.

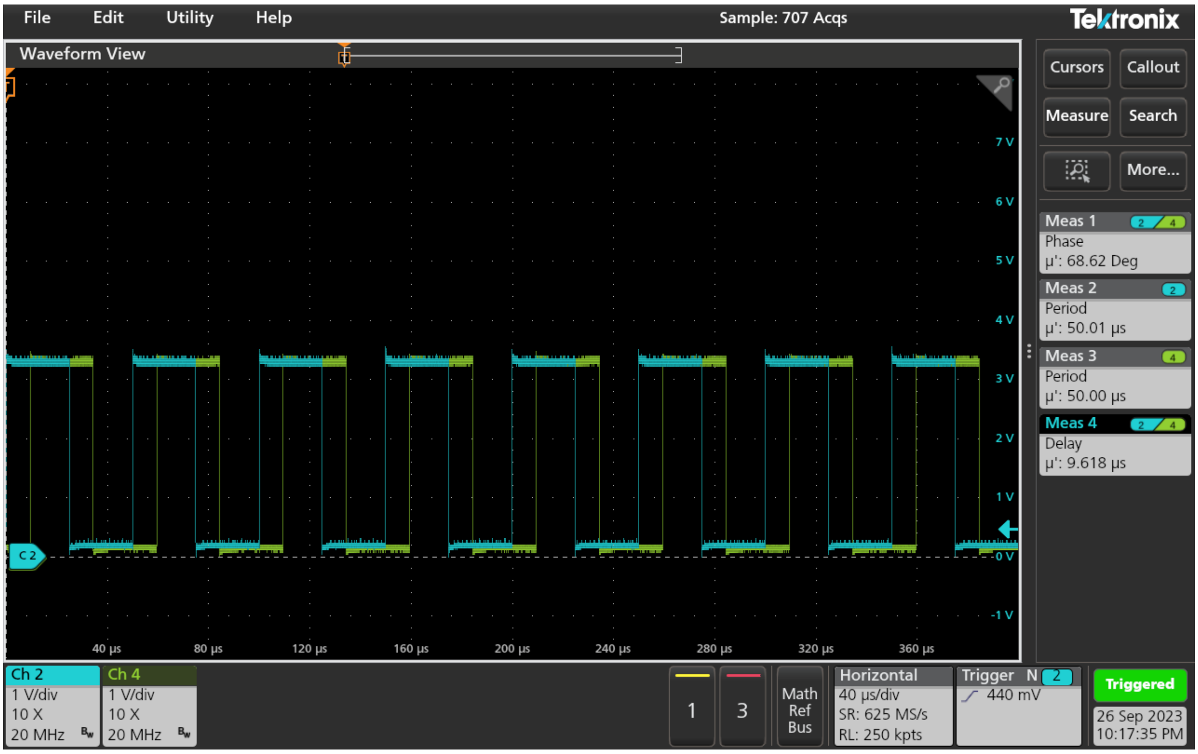
5.5. Validation

In this section, the firmware validation is presented. The performance of the data capture firmware is briefly compared to that of an oscilloscope in subsection 5.5.1. The performance of the DDMTD is presented in subsection 5.5.2. To validate that the internal delay is not larger than one clock cycle, a test is executed in subsection 5.5.3.

5.5.1. Data Acquisition Validation

The data capture IP provided by MATLAB can be considered to be well verified and validated. Nonetheless it is generally a good idea to consider double checking the performance of this firmware by performing a random test validation test and comparing the result to that captured by the data capture.

A known phase difference between the transmitting and receiving 10 MHz clock signals is set to 69.75°. The outputs of the DDMTD are connected to the PMOD pins as depicted in figure 5.10 and the outputs of which are measured by a Tektronix Series 2 oscilloscope at a sampling rate of 625 MS/s. The results are illustrated in figure 5.9.



Meas 1 Phase µ': 68.62 Deg

Meas 2 Period µ': 50.01 µs

Meas 3 Period µ': 50.00 µs

Meas 4 Delay µ': 9.618 µs

Ch 2: 1 V/div, 10 X, 20 MHz

Ch 4: 1 V/div, 10 X, 20 MHz

Horizontal: 40 µs/div, SR: 625 MS/s, RL: 250 kpts

Trigger N 2: 440 mV

26 Sep 2023 10:17:35 PM

Figure 5.9: DDMTD beat frequencies sampled by a Tektronix Series 2 oscilloscope

The Δt_{beat} is 9.618 µs on average yielding a Δt of 19.236 ns. Comparing this to the result from the data capture in table 5.3, which is 19.072 ns, the difference between these two values is 164 ps. Especially considering that the signals experience a different internal delay depending when being routed to the appropriate PMOD pins.

5.5.2. Performance Validation & Analysis

After the verification by means of a behavioral simulation done using Vivado and validating that the system works as intended when programmed onto the FPGA, the same firmware signals are sampled using the MATLAB generated IP firmware. Similar to the verification phase, a pre-determined phase delay is programmed into the signal by means of an MMCM. These phases also range from 90°down to 5°in 10°decrements, with special stops at 45°and 5°.

The firmware block is illustrated in figure 5.10. `clk_wiz_0` has three clock outputs derived from the default system clock. This is where the phase delay is created internally. `clk_out1` is a 10 MHz signal with zero phase delay with respect to the default clock. This signal is routed to the helper PLL and used as the f_1 signal in the design. `clk_out2` is a 100 MHz phase locked signal derived from the default system clock that is routed to the sampler PLL. The sampler PLL upscales its input frequency. Providing it with the 10 MHz signal would limit the maximum achievable sampling frequency that can be derived without issues. Similarly, if the sampling frequency was directly created from `clk_wiz_0`, the phase difference between `clk_out1` and `clk_out3` would become problematic as flexibility in these values would be reduced if another signal is present that is several orders of magnitude higher than the other frequencies. Therefore, this double PLL configuration is chosen.

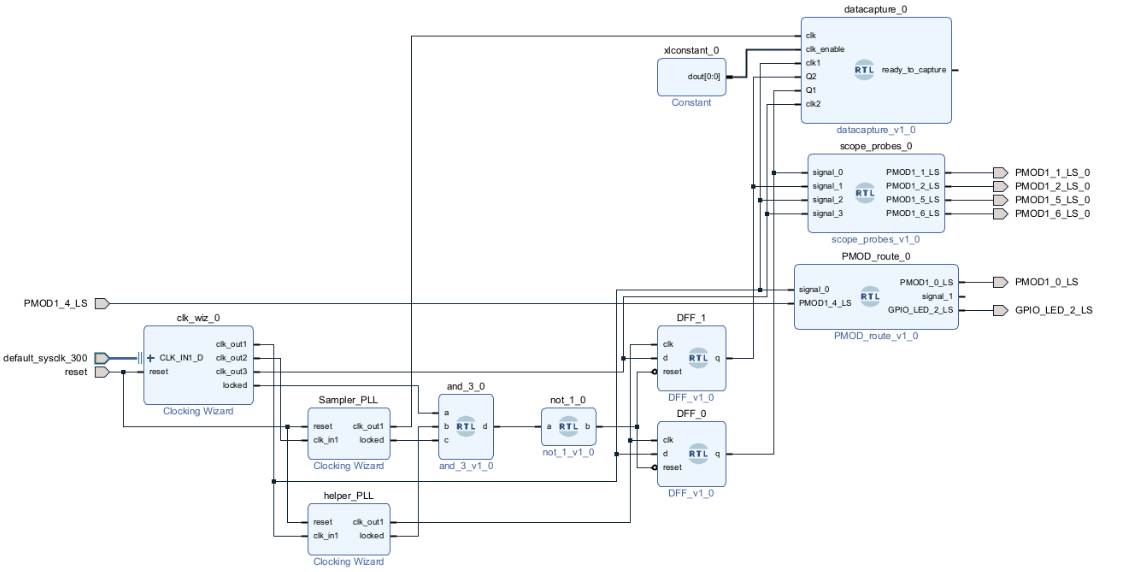


Figure 5.10: Firmware block diagram for validation testing. Instead of routing the signal through the PMOD pins, an artificial delay is generated by setting a phase difference in the PLL

The FPGA is connected to a PC using a micro-usb to USB-A cable connected to the JTAG port on the FPGA and the aforementioned strategy is employed to generate the results presented in table 5.3

As the MMCM may not always be able to precisely generate the desired phase output, compromises need to be made between the desired phase and the achievable phase. Signals Q_1, Q_2, f_1 and f_2 are sampled and their waveforms are saved for analysis.

The sampling frequency is set to 200 MHz. With this setting, a time delay from the signals can be computed, the results of which are presented in figure 5.11 until and including figure 5.21.

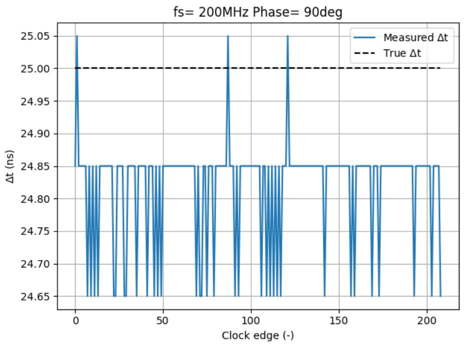


Figure 5.11: Validation results with set phase difference of 90°

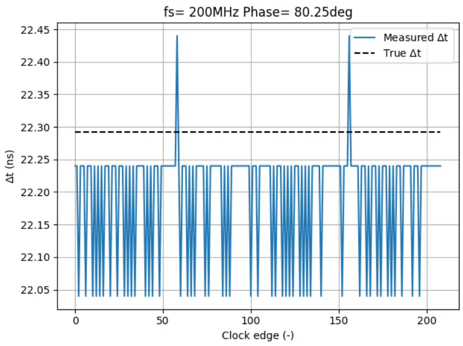


Figure 5.12: Validation results with set phase difference of 80°

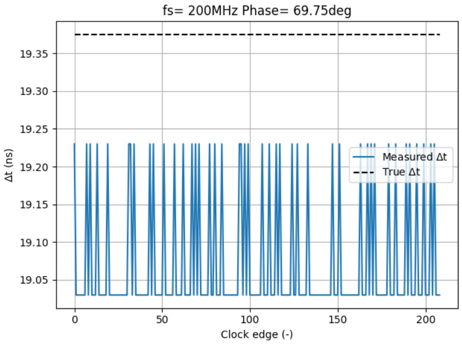


Figure 5.13: Validation results with set phase difference of 70°

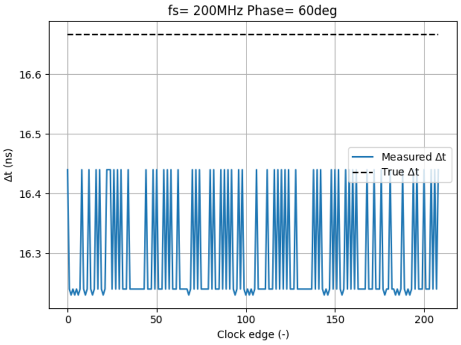


Figure 5.14: Validation results with set phase difference of 60°

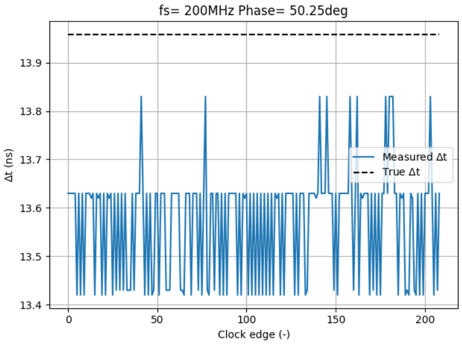


Figure 5.15: Validation results with set phase difference of 50°

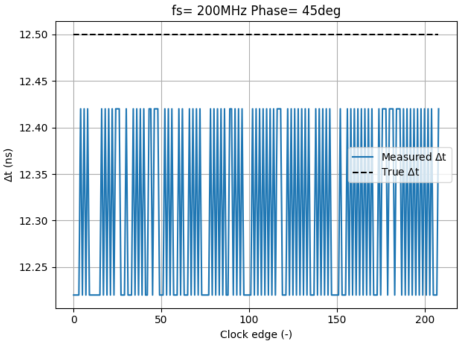


Figure 5.16: Validation results with set phase difference of 45°

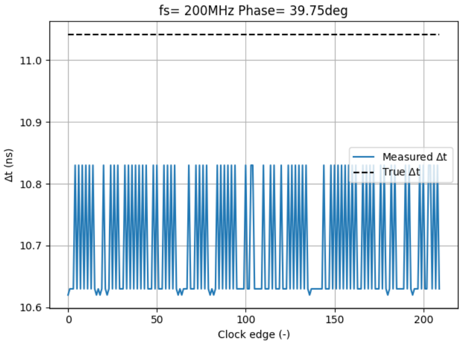


Figure 5.17: Validation results with set phase difference of 40°

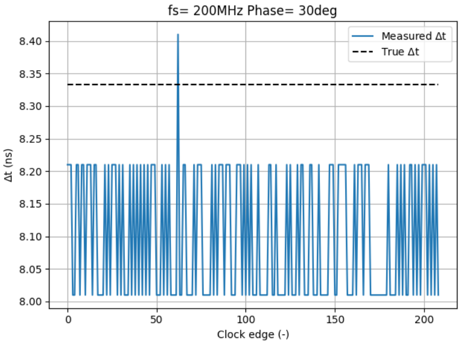


Figure 5.18: Validation results with set phase difference of 30°

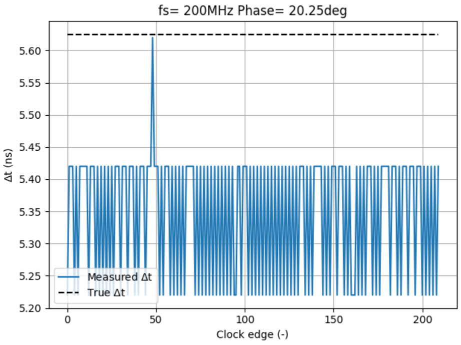


Figure 5.19: Validation results with set phase difference of 20°

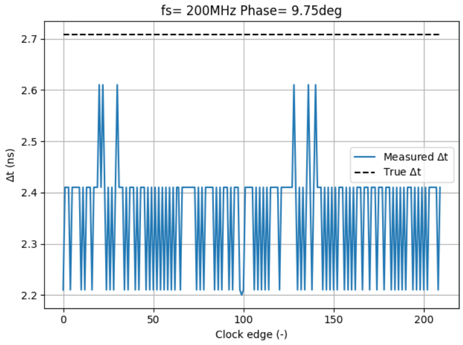


Figure 5.20: Validation results with set phase difference of 10°

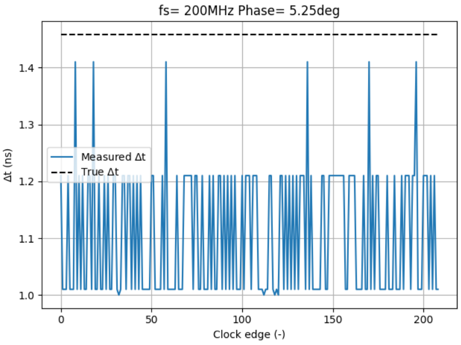


Figure 5.21: Validation results with set phase difference of 5°

The performance of the system is slightly lower than anticipated from the simulation in section 5.4. The phase readings are oscillatory, however in most cases a baseline is present such as in figure 5.11. In this case a clear baseline is present at 24.85 ns. The oscillatory behaviour is mostly exactly 20 ns, translating to 4 sampling periods. In cases such as figure 5.18, this baseline is not evident from the graph, it appears there are equal amounts of points present at 8.21 ns and 8.01 ns.

It does however display the capability of the system to measure phase differences below what the sampling clock permits, the final two time delays are not measurable given the constraint of 5 ns, using DDMTD they are. The results are analysed statistically. The mode, average, and standard deviation of the data are calculated. The results of this analysis are presented in table 5.3.

Table 5.3: Statistics from the validation experiment where the true time delay is compared to the average and the mode of the measured time delay

Δt_{true} (ns)	Δt_{mode} (ns)	$\Delta t_{\text{average}}$ (ns)	Δt_{std} (ns)	$\Delta\varepsilon$ w.r.t mode (ps)	$\Delta\varepsilon$ w.r.t average (ps)
25.000	24.85	24.820	0.079	150.000	179.665
22.292	22.24	22.194	0.089	51.667	97.600
19.375	19.03	19.072	0.082	345.000	302.895
16.667	16.24	16.296	0.091	426.667	371.116
13.958	13.63	13.573	0.113	328.333	385.463
12.500	12.22	12.306	0.099	280.000	193.876
11.042	10.63	10.698	0.095	411.667	343.619
8.333	8.01	8.100	0.101	323.333	233.381
5.625	5.42	5.346	0.099	205.000	279.286

Continued on the next page

Table 5.3 – continued from the previous page

$\Delta\mathbf{t}_{\text{true}}$ (ns)	$\Delta\mathbf{t}_{\text{mode}}$ (ns)	$\Delta\mathbf{t}_{\text{average}}$ (ns)	$\Delta\mathbf{t}_{\text{std}}$ (ns)	$\Delta\varepsilon$ w.r.t mode (ps)	$\Delta\varepsilon$ w.r.t average (ps)
2.708	2.41	2.352	0.103	298.333	356.476
1.458	1.01	1.106	0.111	448.333	351.874

It can be observed that the difference between the mode and the average of the time delay computation is minimal, in the order of several tens of picoseconds. The maximum difference between the mode and the average is 96 ps at 5°phase delay with a minimum difference of 30 ps at 90°. As expected, when comparing the error residual to the results found during firmware verification from table 5.2, the residuals found are relatively larger. Comparing the time delay measurements from the simulation to the the results from table 5.3 shows that the measurements do not differ significantly. This is illustrated in the graph in figure 5.22.

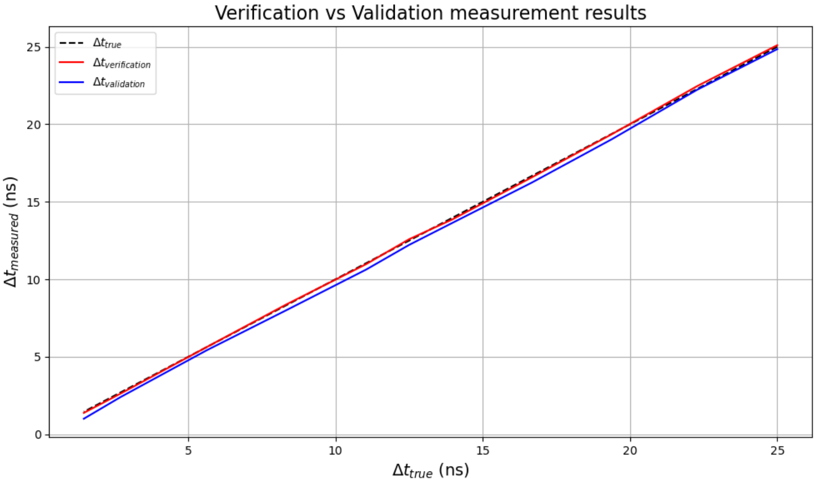


Figure 5.22: Result comparison of the verification results against the validation results.

It can be observed that the time measurements almost align, nonetheless these slight deviations must be noted. The measured time delay is in all cases slightly less than the expected time delay represented by the black line. This could be attributed to the accumulative jitter that exists within the clocking mechanisms of the FPGA as well as the DFFs. This can be seen as a more or less constant bias. It does not pose a tremendous issue with regards to the measurements as it can be calibrated out using least-squares.

First, assuming that the measurements adhere to a Gaussian distribution, a probability interval can be calculated using the standard deviation from the results. According to a probability interval of 99.5%, the z-value is 2.576. When computing this interval, the residual of the time delay measurement becomes evident, as the probability interval narrowly includes the true time delay as illustrated in figure 5.23, and a magnification of the error residual in figure 5.24.

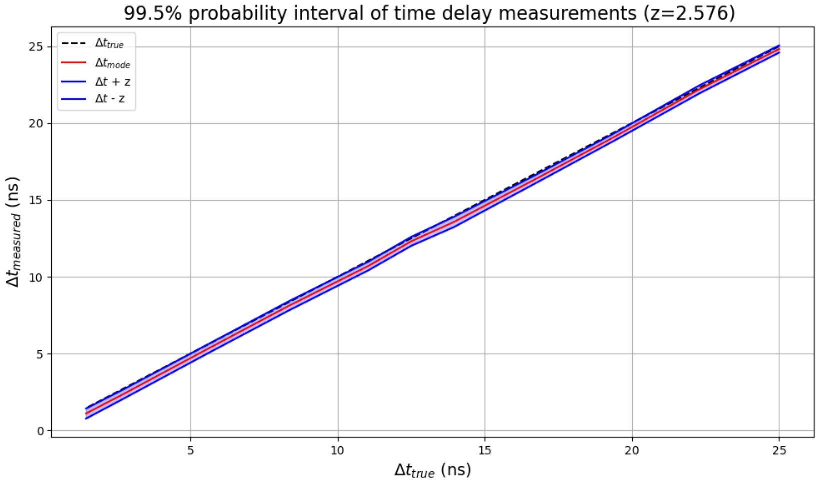


Figure 5.23: 99.5% interval of the time delay measurements during validation of the firmware

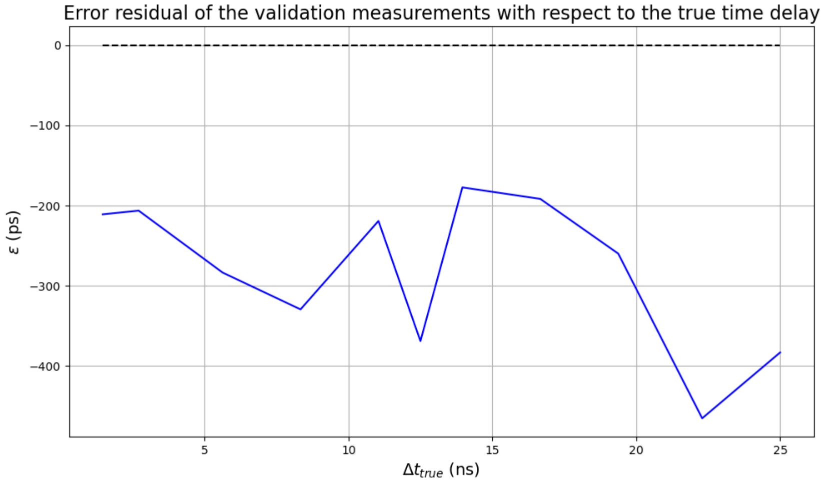


Figure 5.24: Magnification of the error residual from figure 5.23.

When adding the average error of the average time delay measurement to itself, the distribution closely coincides with the true measurement as illustrated in figure 5.25, and a magnification of the error residual in figure 5.26.. The average of the error with respect to the average of the time measurement is 281.4 ps.

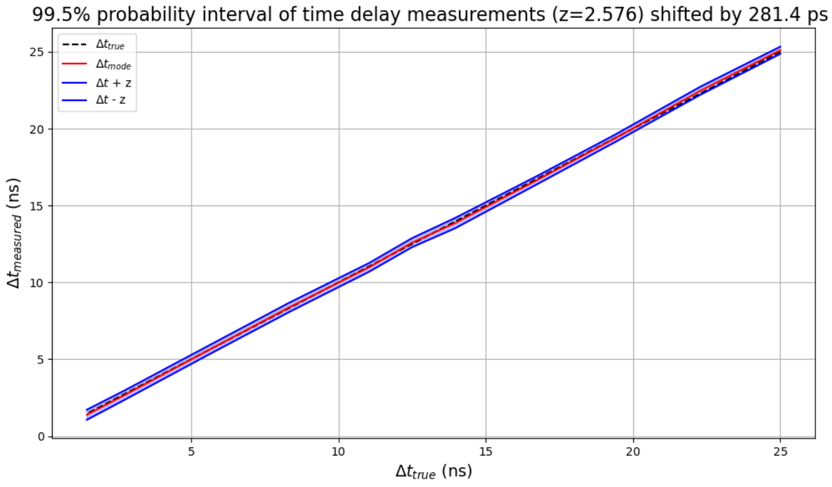


Figure 5.25: 99.5% interval of the time delay measurements during validation of the firmware shifted by the average of the error.

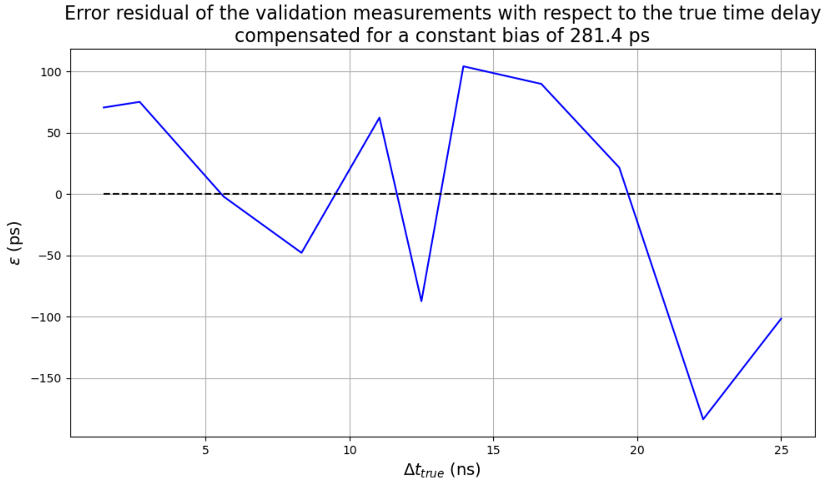


Figure 5.26: Magnification of the error residual from figure 5.25 after compensating for the bias.

After implementing this shift, the resulting errors are shown in Table table 5.4. On average, when calibrating the measurements obtained with the average error, there is a 73% reduction in error compared to the uncorrected time-delay measurements.

Imagining the same error behaviour would show itself when connecting this system to some laser terminal, the ranging anticipated ranging error would be at most 5.51 cm for this particular system.

Table 5.4: Error in time delay measurement when shifting by the average error compared to the standard errors

Δt_{true} **(ns)**	$\Delta t_{corrected}$ **(ns)**	$\Delta\varepsilon$ **(ps)**
25.000	25.102	101.721
22.292	22.475	183.787
19.375	19.353	21.508
16.667	16.577	89.730
13.958	13.854	104.076
12.500	12.588	87.511
11.042	10.979	62.233
8.333	8.381	48.005
5.625	5.627	2.101
2.708	2.633	75.090
1.458	1.388	70.488

From this can be concluded that the obtained time delay measurement can be corrected by Δt_{cal} = 281.4 ps to increase the accuracy of the measurements.

5.5.3. Resolving ambiguity

The transmitted signal is a periodic 10 MHz signal with a 50% duty cycle. This results in the following ranging ambiguity:

$$d_{RA} = \frac{c}{2 \cdot f} = 15m \tag{5.5}$$

Where c is the speed of light and f is 10 MHz. From the calculation above, the time constraint for the system is 50 ns. To ensure the system meets this constraint, it needs to be validated that the propagation does not exceed this limit.

To resolve this issue, the 10 MHz is to be replaced by a slow clock. The internal clocking mechanisms of the FPGA only allows for stable clock generations as low as 10 MHz, therefore the internal clocking IP cannot be used to generate these slow signals. To resolve this, a compromise on stability has to be made and a counter is implemented to create a slow clock: this signal is only used to confirm if the timing limit is not exceeded.

A 4-bit counter is implemented as illustrated in figure 5.27, resulting in a periodic signal with a frequency of 625 kHz and a duty cycle of 6.25%. This signal has an ambiguity of 800 ns, which is highly unlikely that the propagation delay will exceed.

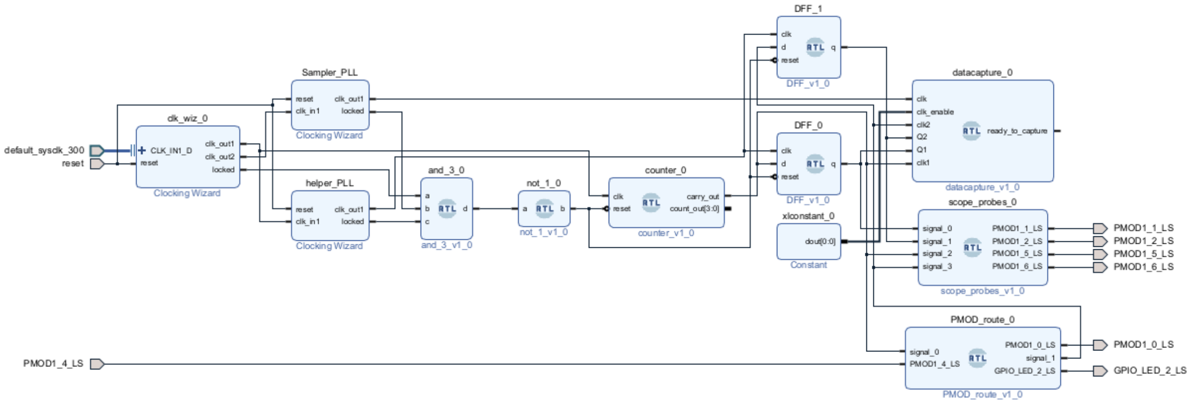


Figure 5.27: Block diagram of the firmware transmitting a single pulse

The difference between the block diagram in figure 5.27 and in figure 5.6 is the 4-bit counter, the rest of the firmware remained the same.

Running the firmware on the FPGA results in the following waves captured by MATLAB illustrated in figure 5.28. It can be observed that there is a slight delay between the two clock signals of 4 data points. The sampling frequency is 200 MHz, resulting in a time difference of 20 ns. It can be safely assumed that the internal delay does not exceed the aforementioned 50 ns.

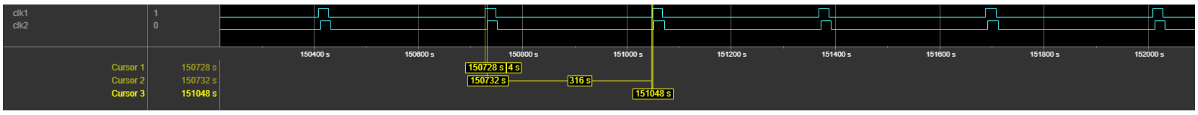


Figure 5.28: Time difference between two consecutive pulses

5.6. Results

The firmware depicted in figure 5.10 is programmed onto the KCU105 FPGA using Vivado ML Enterprise Edition 2021.2. The experiment is conducted by bridging two pins, the `PMOD1_0` and `PMOD1_4`, of the J53 header on the board. The gap will be bridged by a piece of copper wire. The dimension of the J53 header is illustrated in figure 5.29 and the copper bridge is illustrated in figure 5.30.

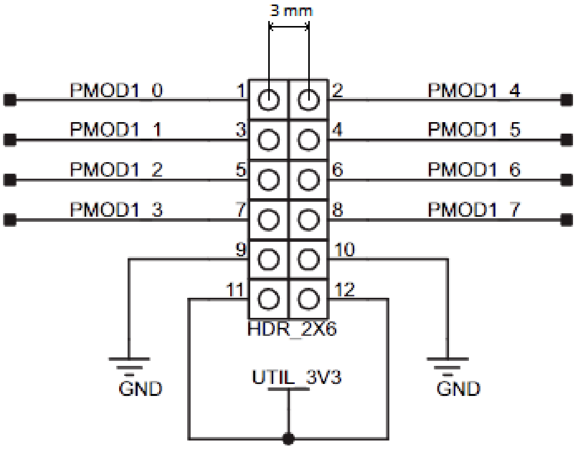


Figure 5.29: J53 Header pin dimension

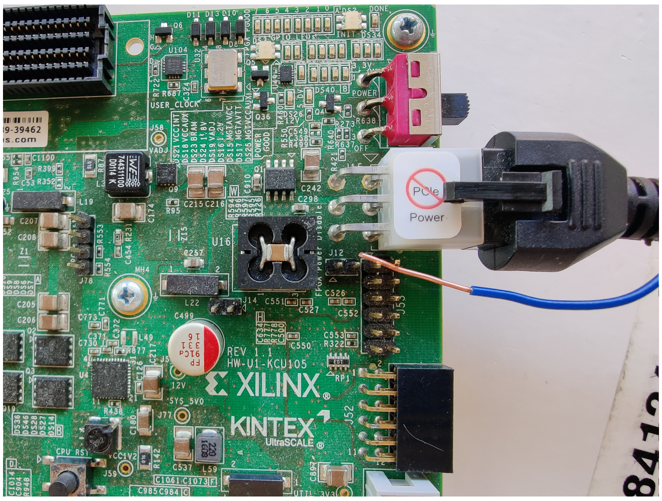


Figure 5.30: Bridging of two PMOD wires with a copper wire to create a delayed signal

During the process of programming the firmware onto the FPGA, the logic gates are assigned to specific locations on the chip. Normally, Vivado handles this task automatically and calculates the optimal placement for each component. However, it is important to recognize that the positioning of these logic components on the chip might have a slight impact on the firmware's performance, particularly in terms of accuracy when conducting physical time delay measurements using the copper bridge. For the purpose of demonstration, the location of the logic components is illustrated in figure 5.31.

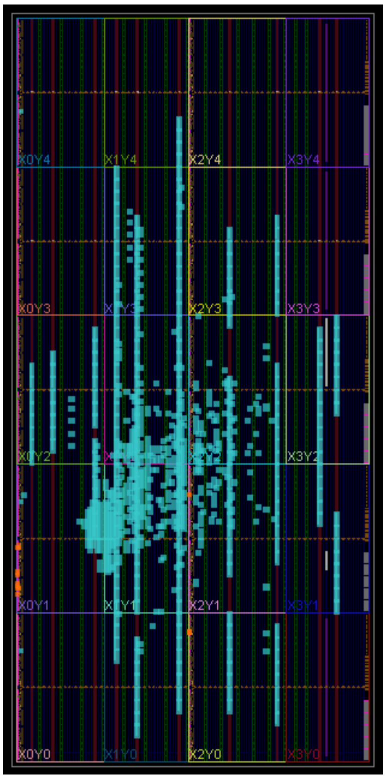


Figure 5.31: The physical location of logic components on the FPGA chip on board of the KCU105 for experiment 1

The 3 mm gap is bridged at the bottom of the pins, this will generate a slight additional time delay to the system that needs to be taken into account. An electrical signal travels at approximately $\frac{2}{3}$ the speed of light. Therefore the additional delay experienced by the signal is the following:

$$\Delta t_{copper} = \frac{3 \cdot L}{2 \cdot c} = 15\ ps \tag{5.6}$$

The value computed by Equation 5.6 is theoretically large enough to be recognized by the digital phase detector as described in section 5.3.

The test is conducted 10 consecutive times each test containing 1048576 data points at a resolution of 5 ns.

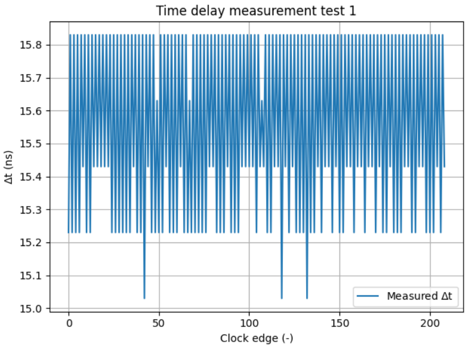


Figure 5.32: Time delay measurement of experiment 1, test 1.

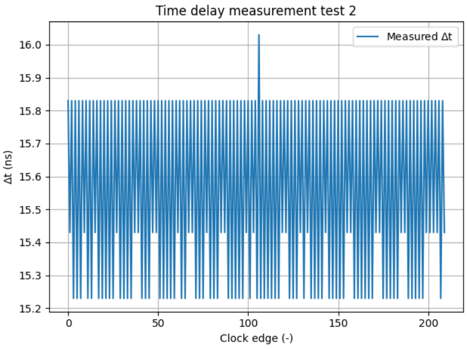


Figure 5.33: Time delay measurement of experiment 1, test 2.

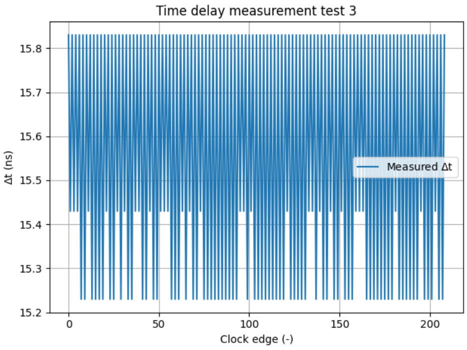


Figure 5.34: Time delay measurement of experiment 1, test 3.

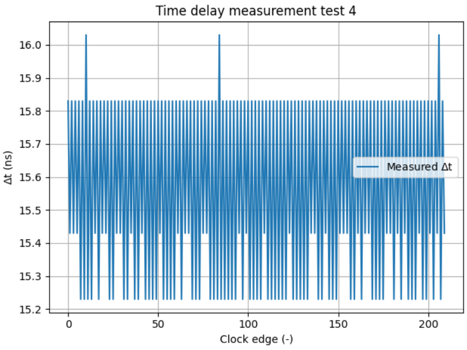


Figure 5.35: Time delay measurement of experiment 1, test 4.

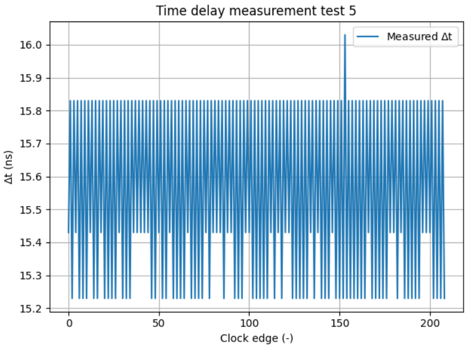


Figure 5.36: Time delay measurement of experiment 1, test 5.

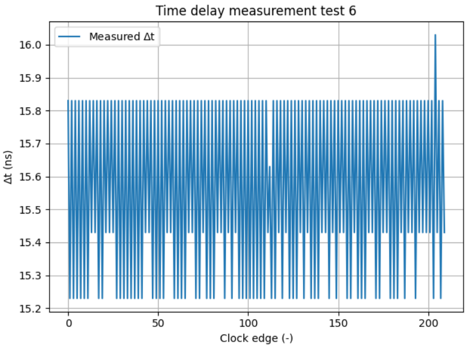


Figure 5.37: Time delay measurement of experiment 1, test 6.

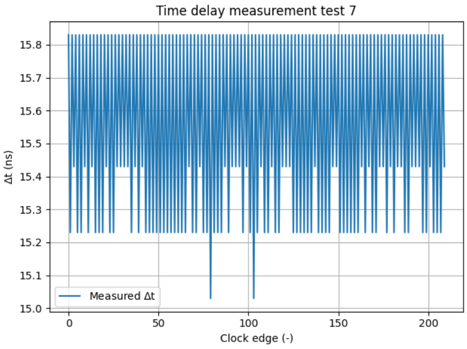


Figure 5.38: Time delay measurement of experiment 1, test 7.

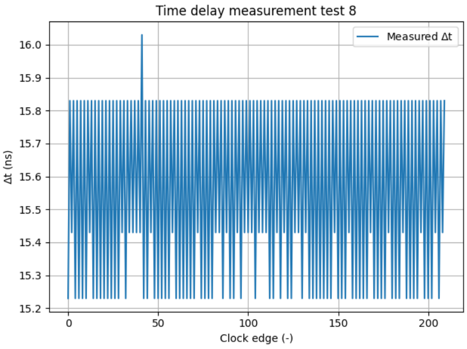


Figure 5.39: Time delay measurement of experiment 1, test 8.

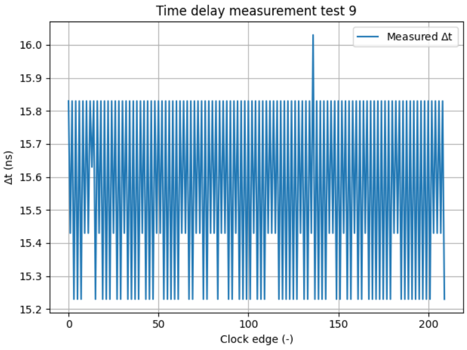


Figure 5.40: Time delay measurement of experiment 1, test 9.

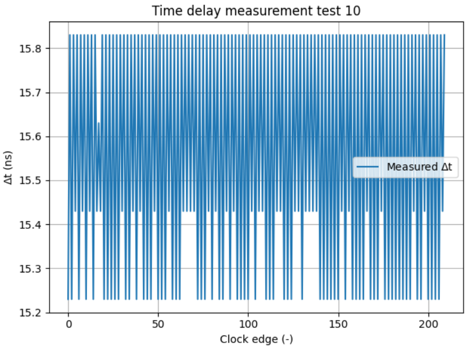


Figure 5.41: Time delay measurement of experiment 1, test 10.

The test results are illustrated in the form of graphs shown in figure 5.32 until and including figure 5.41. Similar to the validation of the internal firmware, the measurements are oscillatory in a similar fashion. However the standard deviation of these measurements are roughly three times as large as the values found in during the validation in section 5.5. Furthermore, the mode that is evident during the validation is not present during the physical tests. From these graphs, the mode, average and standard deviation are calculated. Next, the average is corrected by the expected delay of the copper bridge and the correction factor obtained from section 5.5. The final time delay measurement can therefore be calculated by using Equation 5.7.

$$\Delta t_{corrected} = \Delta t_{measured} - \Delta t_{copper} + \Delta t_{cal} \tag{5.7}$$

The statistical results of the measurements are presented in table 5.5.

Table 5.5: Statistics of the measurement results of experiment 1

Test	Δt_{mode} **(ns)**	$\Delta t_{average}$ **(ns)**	Δt_{std} **(ns)**	$\Delta t_{corrected}$ **(ns)**
1	15.83	15.467	0.273	15.828
2	15.83	15.561	0.279	15.828
3	15.83	15.561	0.278	15.827
4	15.83	15.568	0.277	15.834
5	15.83	15.563	0.276	15.829
6	15.83	15.568	0.271	15.834

Continued on next page

Table 5.5 – continued from previous page

Test	Δt_{mode} (ns)	$\Delta t_{average}$ (ns)	Δt_{std} (ns)	$\Delta t_{corrected}$ (ns)
7	15.83	15.563	0.276	15.830
8	15.83	15.562	0.278	15.829
9	15.83	15.565	0.276	15.832
10	15.83	15.563	0.273	15.830

From the test results from table 5.5 show that the internal time delay on average is equal to 15.83 ns with a standard deviation of 275.7 ps. In terms of range measurements, this translates to an average of 4.75 m of internal delay, with a standard deviation of 8.27 cm.

5.7. Conclusion

Using a Xilinx KCU105 FPGA, an experiment was carried out to quantify the internal delay of a signal as a precursor to the experiment that follows in chapter 6. The primary objective was to answer the second and third sub-question listed in section 1.2.

The system design is presented in section 5.2 starting with a top-level system model of the firmware. In order to capture data from the FPGA, MATLAB IP was used. This allows for easy to use and robust data capture of user signals over a JTAG connection to a PC. The firmware operates similarly to an oscilloscope, by capturing a window of a set amount of points based on a trigger position at a sampling frequency. The maximum obtainable sampling rate is 200 MHz, while higher sampling rates will cause glitches in the captured data making it unusable. The window size that is selected contains 1048576 samples, limited by the available memory on the FPGA.

The DDMTD implemented to measure the phase difference between the transmitted and received signal has a beating frequency of 20 kHz and a resolution of 0.01 ns which is 500 times smaller than what would be observable by the data capture.

The method followed for the experiment is presented in section 5.3. First, the signal type chosen to transmit through the FPGA is a simple square wave with a frequency of 10 MHz. This frequency is chosen because this is the lowest convenient signal that can be achieved through internal clocking mechanisms. To quantify the delay, the phase between the transmitted signal and the received signal must be measured. One digital phase frequency detector is the DDMTD. It is able to measure the phase between two square signals below the accuracy of the obtainable by the system clock.

Because the Vivado synthesizer does not allow the user to manually interfere with the MGT hardware, the signal is transmitted to a PMOD pin instead in an attempt to measure the internal delay. This pin will connected to another pin that is used as the hypothetical receiver. The path the signal travels will introduce a delay, which can then be measured and used as representation of the internal delay experienced by a signal.

The firmware is verified by performing a post-implementation functional simulation in Vivado, which simulates the logical behavior to near reality, presented in section 5.4. For verification, the capabilities of the DDMTD is simulated on two different signals of which the phase difference is set by a PLL, showing it is capable of measuring the set time delay within 151 ps of the target delay.

This behaviour is validated in section 5.5. Here the phase of the two signals is set as well, however this time it is programmed onto the FPGA and the output of the DDMTD is sampled through the data capture and analysed. The phase was correctly identified within 386 ps. This validation showed that on average it has a bias of negative 281.4 ps. Compensating for this bias results in a 73% reduction in errors with the maximum error being 184 ps for a phase difference of 80°, resulting in about 5.5 cm.

Finally the results are presented in section 5.6. An internal delay of 15.83 ns was found with as standard deviation of 276 ps. This value however, only accounts for the internal delay of the signal through the FPGA, to the dedicated route on the PCB and back. It does not however account for the internal delay that is expected when using the MGT, the internal delay will be greater to account for the additional hardware that is being used to transmit the signal.

Only a partial answer can be provided for the second sub-question:

> What methods can be utilized to generate a stable and consistent signal on an FPGA?

Creating a stable and consistent signal can be done with the use of Xilinx Clocking Wizard IP. These pieces of firmware are versatile, well validated and easy to use. However, in order to send a signal through a transceiver, more than just a clocking manager required to create a stable signal.

For the answer to the third sub-question:

> How can the internal propagation delay of the FPGA be quantified?

The answer is more complex than the results obtained in section 5.6. The result of 15.83 ns can be considered to be a correct result for the signal traversing through the PCB to the PMOD pins and back, it does not however account for the more complex structure of the MGT. Its answer can therefore not be considered an appropriate answer for this sub-question and a more detailed analysis is required to find the correct answer.

5.8. Discussion

Even though the conclusion of this experiment is that the results from the tests are not usable for the experiment in chapter 6, the objectives are met partially and most of the goals set in section 5.1 are met. It provided a solid foundation for the further development of a more complex system.

Ideally, with the system from this chapter, one should manually wire a laser transmitter to several PMOD pins for direct control over that module. This could be feasible, however the research, hardware iterations and software iterations required are not possible in the given time constraints imposed on this thesis.

The basis for a more complex system is set with this experiment, namely that proper clocking constraints can be managed, familiarity is gained with programming the FPGA as well as performing post-implementation function simulations using Vivado and most importantly data can be extracted from the FPGA. Which are crucial for the development of any system that follows.

Furthermore, it also demonstrated the basic principles and inner working of a DDMTD implemented digitally on an FPGA. The digital implementation is not new and has been done before in TCLink from CERN [25]. The strategy of implementation, is more advanced, however the basic driving principles are similar. Such as the fact that a MMCM is used to drive the helper PLL of the DDMTD and the employment of two DFFs.

It also differs in the way the DDMTD is implemented. For TCLink it used as a means to an end, in this work it is used as one of the final elements. This difference in implementation does however mean an introduction to an added layer of complexity. For CERN, the DDMTD is used in the process of stabilizing the delay between two points, they are indifferent to the magnitude of this delay and do not actively attempt to quantify this. In this thesis, the delay must be quantifiable, meaning that retrieving this data and the accuracy of this retrieval is important and currently limited to a resolution of 5 ns.

5.9. Recommendations

In this section the recommendations for future work are listed.

- **Quantify the internal delay with the use of the MGT**
  As mentioned before, the result obtained through this method is not representative of a signal that is transmitted over an MGT. For this to be measured the MGT needs to be actively implemented in the design in order to attempt to measure its internal delay. From the user guide [26] of the GTH Ultrascale Transceivers, there exists a method to provide a virtual signal to enable a loopback mode. This is one potential avenue that can be explored in chapter 6.

6 Laser Signal Loopback over Optic Fiber

The experiment outlined in chapter 4 represented a laser ranging experiment, without a dedicated data stream. From this it became evident that the limiting factors on the range measurement were the sampling frequency of the oscilloscope and inaccurate calibration for ranging. The experiment presented in chapter 5 proved that an FPGA is able to quadruple the measuring speed with sufficient measurement accuracy and an attempt at mapping the internal delay for calibration was made. This, however did not prove to be representative for the communication system. Hence this chapter aims to improve these shortcomings.

In this chapter, the objective and goals are outlined in section 6.1. The system setup used for the experiment is presented in section 6.2. Next the system design will be elaborated upon in section 6.3. The methods in place for achieving these goals with the provided system setup are described in section 6.4. Verification and validation of the system design are then presented in the section 6.5 and section 6.6 respectively. Finally, the results of the experiment are presented in section 6.7. These results are then substituted in a orbital simulation to grasp the in-orbit performance and compare the systems in section 6.8 followed by the conclusion in section 6.9 and recommendations for future work in section 6.11.

6.1. Objectives and Goals

From the previous chapters, several recommendations have been made. This chapter aims to fulfil several of them. Firstly, one objective is to improve the measurement uncertainty from chapter 4 by increasing the sampling frequency by 400% as proven feasible in chapter 5.

Secondly, the objective is to conduct ranging based on the data structure of an optical signal rather than phase measurements, as outlined in chapter 5. The FPGA will use a standardized MGT which will take a universal SFP connector to transmit and receive an optical signal.

Thirdly, the parameters that influence the range measurement shall be parameterized such as internal latency and data-rate.

Furthermore it aims to answer the remaining sub-questions:

1. What methods can be utilized to generate a stable and consistent signal on an FPGA?
2. How can the internal propagation delay of the FPGA be quantified?
3. How does the line rate of the communication influence the range measurements of the FPGA?

The goals of this chapter are to:

- Create a signal transmissible over the MGT of the FPGA.

- Re-calibrate to compensate for MGT internal delays.
- Obtain range measurements through optical telemetry ranging.
- Create different firmware versions operating at different line rates.
- Identify components introducing uncertainty in the range measurements.
- Demonstrate the potential in-orbit system performance.

6.2. System Setup

The hardware setup is illustrated in figure 6.1. It is a straightforward setup where the JTAG and Universal Asynchronous Receiver/Transmitter (UART) bridges are connected to a PC, and the SFP module on the board is looped back with a optic fiber cable, the length of which is to be measured through the ToF measurements

The system is built using a KCU105 FPGA board from Xilinx. It uses the standard commercial MGT transceiver in combination with a SFP module. These components are Commercial Off The Shelf (COTS) and readily accessible. Instead of free-space, single mode optic fibers are attached to the SFP module. The module is operating at a wavelength of 1550 nm, compatible with the 9/125 (OS2) G.657.A1 single mode optic fiber.

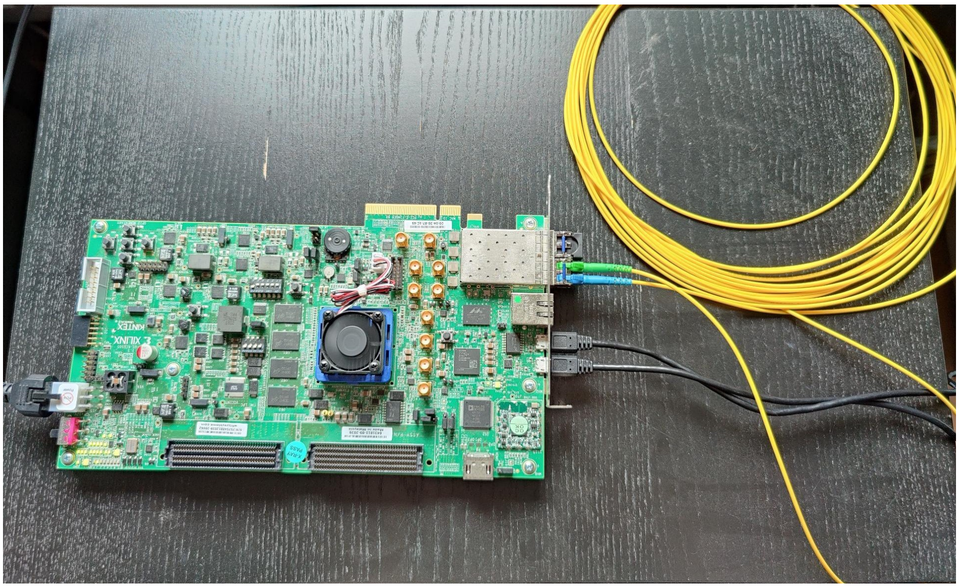


Figure 6.1: Setup of the KCU105 FPGA board to run the tests, a single mode optic fiber is looped back into one of its receivers to measure the length of the cable.

SFPs are a universal standard and come with laser diodes of different wavelengths and with different data-transfer speeds. The SFPs that are used are ASF15-24-80-D modules operating at a wavelength of 1550 nm with a maximum range of 80 km over single-mode optic fiber. The maximum data rate supported by this model is 1.25 Gbps. A photo of the models is depicted in figure 6.2.

Figure 6.2: ASF-15-24-80-D 1550-nm SFP modules used for in the FPGA

Each optic fiber cable has an approximate length of 10 m. This is hard to confirm as the optic fiber cables naturally curl, and stretching them out still leaves some curves. Nonetheless they were measured using a tape measure and the lengths were confirmed. The cables can be connected to one and other using connectors to create an array of lengths, ranging from 10 m to 50 m.

6.3. System Design

In this section the system design is presented. Starting with a system model in subsection 6.3.1, where a top level system model is provided. Followed by the system clocks and clock domain crossings in subsection 6.3.2. The configuration of the GTH and the CDC is discussed in subsection 6.3.3. Followed by the signal generation and line encoding in subsection 6.3.4 and subsection 6.3.5 respectively.

6.3.1. System Model

Figure 6.3 illustrates a top-level system model diagram. From this perspective, the system can be subdivided into three general parts. The first part is the signal generation, depicted in blue in the block diagram. Here a 16-bit counter is instantiated to provide the RCIDs for the data stream. This value is then transferred to the second part, the GTH. In the block diagram, it is separated for illustrative purposes, within the FPGA firmware both the transmitter and receiver channels are merged. A GTH is a type of MGT as there exist different types equipped for various data rates. The main reason for separating it , is that they share the same physical location on the FPGA and share some clocking resources. It must be noted that as soon as data is inserted into the GTH transceiver, some intermediate steps it undertakes are not observable through the data capture firmware, or the Internal Logic Analyzer. The final part of the system are the SFP modules. These modules translate the electrical signal to the optical domain and back.

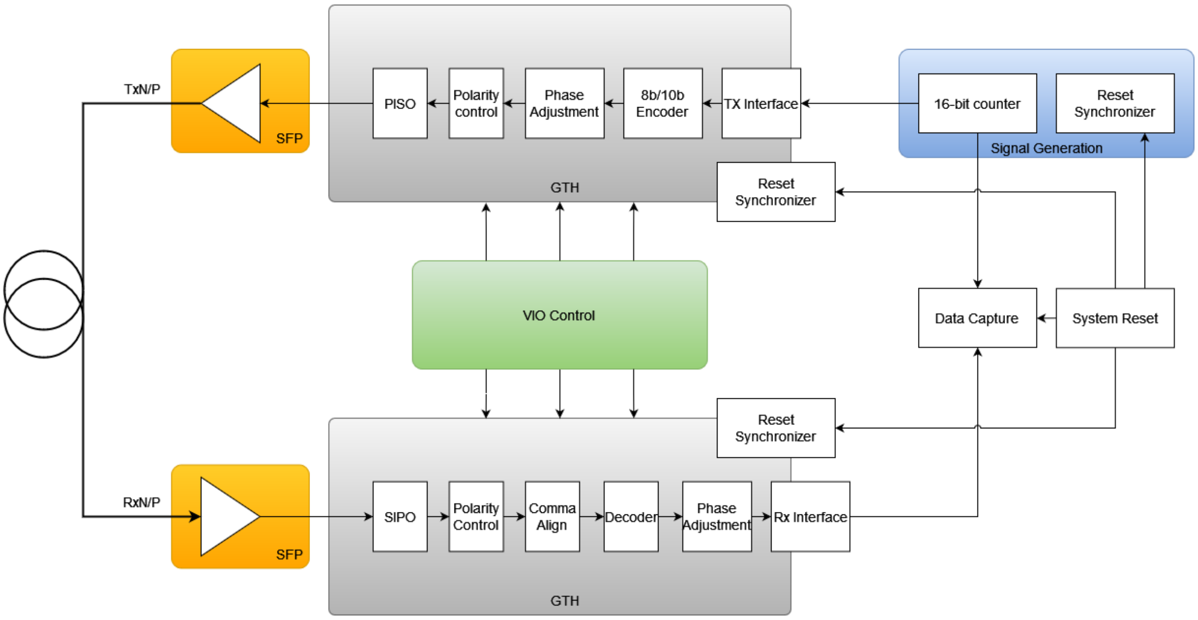


Figure 6.3: Top level system model block diagram of the system built for the second experiment

The GTH transceiver is configured using Xilinx IP and is further elaborated upon in subsection 6.3.3. The data is inserted into the GTH by the Tx interface, this automatically communicates with all the necessary and configured components of the transceiver. The data has a width of 16-bits representing the RCID value, the reason for selecting this values is presented in subsection 6.3.4. The Tx interface relays the parallel data to the 8b/10b encoder (subsection 6.3.5) where the data is encoded according to the selected line encoding scheme.

Then the phase adjustment and the polarity control take place, this is mainly done to the underlying clock of the data subsection 6.3.2 (this is an automated process performed by the IP). After these changes the still parallel data is insterted into the Parallel Input Serial Output (PISO) register, where it is translated to serial data. The data is On-Off Keying (OOK) modulated and provided to the SFP driver, converting the signal from the electrical domain to the optical domain.

Once the data is received on the receiver side of the SFP, the data is de-serialized using a Serial Input Parallel Output (SIPO) register. Here the polarity of the signal is checked and controlled in the GTH firmware. As mentioned before the signal is 8b/10b encoded, which in the process embeds commas (table 6.3) at the start of transmission to align the data accordingly. This process is described in more detail in subsection 6.3.5.

After alignment, data is decoded, phase adjusted and transferred to the Rx interface where the data is made accessible again to the user and the 16-bits parallel data is routed to the data capture firmware which acquires the data as described in subsection 5.2.3. Here, it is sampled at 200 MHz with a window size of 262144 samples and transferred to a PC over a JTAG connection.

6.3.2. System Clocking

In order to use the Ultrascale GT IP for gigabit networking, some clocks have to be configured manually and some are automatically derived by the firmware itself. The clocking mechanisms that the user has to configure and can use are defined as *User Clocks*. The ones that are automatically derived are called *GTH Clocks* as they reside within the GTH firmware and cannot directly be accessed by the user. Special attention will be dedicated to these mechanisms in subsection 6.3.3 as it is important to understand the inner workings.

The user clocks diagram is illustrated in figure 6.4. Here all the arrows represent a clocking signal that is accessible to the user and can be integrated in custom written firmware outside of the IP.

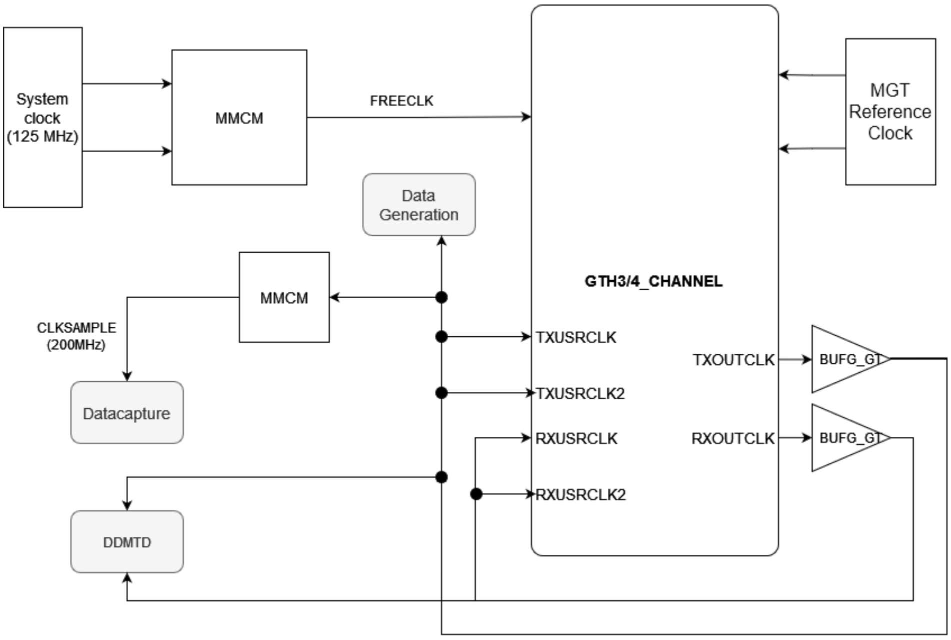


Figure 6.4: User Clocks Block Diagram of the System

The IP only needs two inputs in terms of clocking, a free-running Dynamic Reconfiguration Port (DRP) and a MGT reference clock. These need to be configured by the user to fit the requirements of the GTH channel. These are conveniently listed during the configuration of the IP. The free-running DRP clock or FREECLK is dependent on the line rate at which the system operates and the encoding. For line rates below 2 Gbps and excluding 64b/66b encoding, the FREECLK frequency can be determined by Equation 6.1. For the determination of high line rate applications, the guidelines from [26] shall be followed.

$$f_{DRP} = \frac{R_{line}}{S} \tag{6.1}$$

Where R_{line} is defined as the line rate in bits per second, and S is defined as the symbol length in bits. In order to derive the proper DRP frequency, it needs to be scaled down from a system source. The FPGA provides only two system clock sources, one of which is has a frequency of 125 MHz. Naturally this may need to be scaled up, or down depending on the requirements imposed on the line rate.

In order to scale this frequency a MMCM is used. This is a IP module that will derive and phase lock a desired clock signal from its input, which in this case is one of the system clocks. The required frequencies will be listed in the sections of the various tests as DRP frequency.

The second input is the MGT reference clock. This is used to create a stable reference for other signals within the GTH and is common in networking practices such as Ethernet for example. The magnitude of this frequency is not necessarily important, it is however common practice to set it at least higher than the DRP frequency[26].

The reference clock must have its own dedicated source, to further explain this, first it must be understood how the physical hardware in the FPGA silicon is divided. It is divided into quadrants or quads, which each have their own X and Y location on the hardware as illustrated in figure 5.31. Naturally some signal types, especially those with a high frequency, cannot traverse the entirety of the FPGA silicon. Hence a physical limiation is imposed that a reference source can only be taken from a quad north or south of the master quad or itself. The quad that inhibits all the necessary connections for the SFP hardware is quad X0Y2, meaning that reference can only be taken from X0Y1, X0Y2 or X0Y3. Looking at the user guide of the KCU105 [23], the clocks can be sourced from the SubMiniature version A (SMA) GTH inputs, the FMC Low-Pin Count (LPC) connector input, the Peripheral Component Interconnect Express (PCIE) clock or MGT Si570 clock or the Si5328 jitter attenuator clock.

The first three options listed are dedicated to external inputs to the FPGA, leaving the two internal options, the Si570 and the Si5328 as the remaining options. Even though the Si570 is advertised as being an MGT clock, for communication applications, the Si5328 jitter attenuator is often used due to its excellent phase noise performance required by most systems[27]. Hence the jitter attenuator is also used for this design.

This clock is running at 0 Hz by default on the FPGA. The frequency of the jitter attenuator can be set through an UART connection via USB. Xilinx recommends the use of Tera Term, a terminal emulator designed to communicate with a wide variety of devices with configurable settings. First the appropriate Silicon Labs drivers must be installed. These divers are the Silicon Labs Dual CP2105 USB to UART bridge drivers. After this the serial connection in Tera Term can be configured.

Connecting the KCU105 to a PC, the PC recognizes the board even when the power is off, if the drivers are configured correctly. For the configuration of Tera Term, the Enhanced COM Port (COM5) is selected on the opening screen. The default Baud rate of Tera Term is 9600 bits/s, this needs to be changed to 115 200 bit/s. When powering on the KCU105, a menu shall be presented from which clock configuration is straight forward.

As illustrated in figure 6.4, the GTH channel also generates two clocks to be used by the user in the firmware design. They are routed through a buffer making them accessible in the user code. These clocks are directly linked to the transmitted and received data, meaning that all aspects related to the transmitted data is preferably clocked using either `TXUSRCLK` or `TXUSRCLK2` and for the receiver either `RXUSRCLK` or `RXUSRCLK2`. Hence data generation is driven by `TXUSRCLK`.

The data capture firmware is clocked through a MMCM phase locked to the `TXUSRCLK`. Phase locking the sample clock to this signal causes the Tx User Data to be aligned with the data capture itself. From tests it was found that, in case the sample clock was derived from the system clock, both Rx and Tx User Data contained random spikes attributable to the out of sync sampling clock versus the data clocks. Doing so however does not get rid of these attributes on the receiver data. On the receiver data there will still be artifacts attributable due to the transmitter and receiver clocks not being synchronized, that can be filtered out with post-processing (subsection 6.4.4).

6.3.3. GTH and Clock Domain Crossing

The clocks outside of the GTH channel have been described in subsection 6.3.2. In figure 6.4, this has been illustrated as a closed box, the inner workings of which will now be elaborated upon.

These internal clocking mechanisms are automatically derived by the IP through the configuration performed by the user. Configuring these transceivers on an FPGA however is a non-trivial task as it requires dedicated firmware to run the MGT transceivers as desired, including setting up all involved clocks, line-encoding, buffer settings and reset synchronization.

These GTHs are complex systems with dedicated firmware, which is generated as part of the IP from Xilinx. This allows the user to only provide external toggles to the system in order to configure these specific settings through a series of Multiplexers (MUXs).

To start, the GTH can be subdivided into transmitter and receiver channels. A walk-through of both is described, elaborating on the choices made and how the GTH works internally to further understand its behaviour. First the transmitter part is illustrated in figure 6.5.

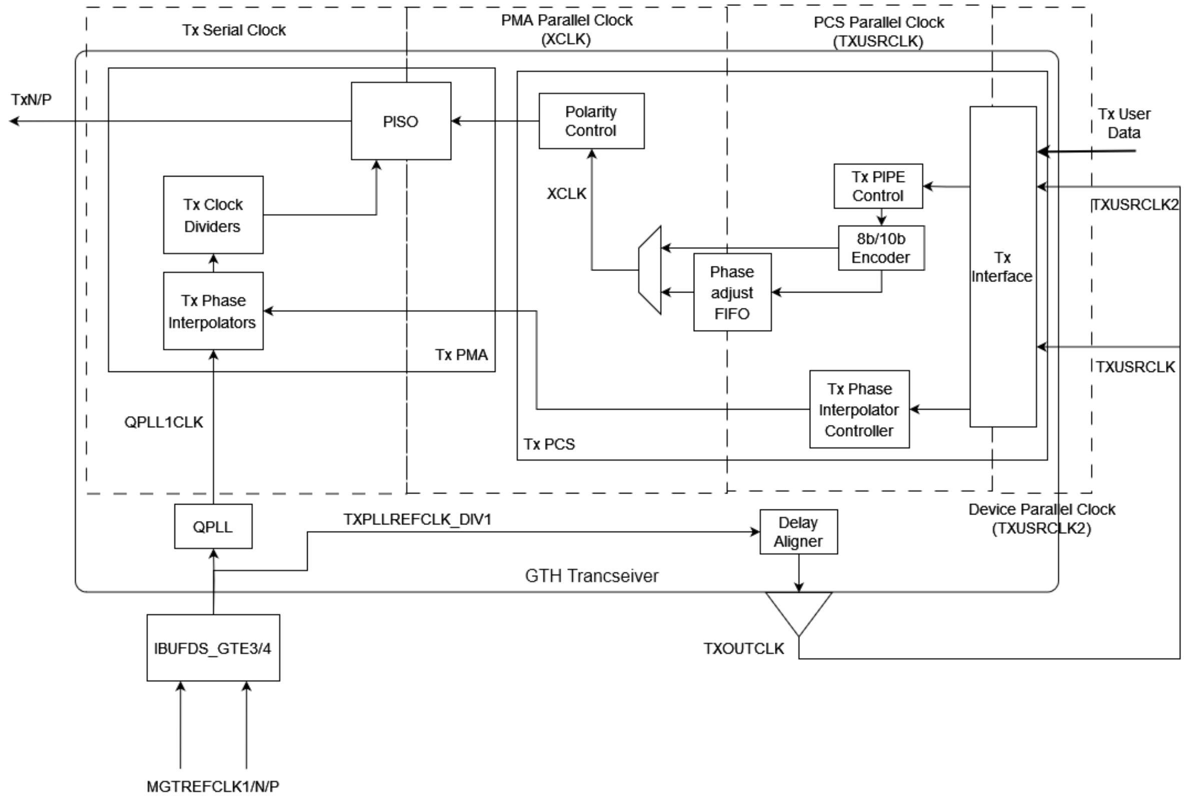


Figure 6.5: Clock domains and CDC of the transmitter GTH

From the transmitted data generated externally as illustrated in figure 6.3, a clock (TXOUTCLK) is derived from the reference, that is aligned to the start of each symbol at a frequency of f_{DRP}. This signal can be found at the bottom of figure 6.5. This is simultaneously provided as input to the IP as per recommendation[26].

It is recommended to derive the TXOUTCLK from the TXPLLREFCLK, and it is required to do so in case the buffer is bypassed [26]. The TXOUTCLK is routed through a buffer outside of the GTH firmware to be used as input for both TXUSRCLK and TXUSRCLK2.

Here, on the right side of figure 6.5, it enters the PCS layer clock domain through the Tx interface. The PCS layer is standard firmware that is part of the GTH. The Tx interface allows for easy communication from the outside of the transceiver with respect to the other layers, while facilitating automatic CDC.

In the PCS layer the TXUSRCLK signal is used to drive the PIPE control, the encoder, the phase interpolator controller and potentially the phase adjustment First In First Out (FIFO). Here the first design choices in terms of communication protocol have to be made, starting with the encoding type. The encoding protocol is further elaborated upon in subsection 6.3.5. To ease the complexity of the communication, 8b/10b encoding was chosen because it allowed for automatic alignment of the data without user intervention, ultimately saving time.

The next configurable aspect is whether or not to bypass the buffer, the phase adjustment FIFO, on the transmitter. The buffer is there by default, responsible for the phase alignment between the PMA Parallel Clock (XCLK) and TXUSRCLK rate in order for successful transmission. Alternatively there exists also a phase alignment circuit that can be utilised. Bypassing the buffer has different consequences for the transmitter. The trade-off criteria for bypassing the buffer are given in the user guide [26] and presented in table 6.1.

Table 6.1: Trade-off criteria for bypassing the Tx buffer, taken from [26]

	Tx Buffer	**Tx phase alignment**
Ease of Use	The Tx buffer is robust and easy to operate	Requires extra logic and additional constraints on the clock sources.
Latency	Higher latency	Uses fewer registers to achieve lower and deterministic latency.
TXUSRLK jitter sensitivity	No sensitivity to TXUSRCLK jitter	Sensitive to TXUSRCLK jitter

In this table, the FIFO or buffer, is compared to a phase alignment circuit. The first trade-off criterion is the ease of use. The buffer is easy to use while the phase alignment circuit requires extra logic. Xilinx however, automatically provides a template for this exact logic. In terms of latency, the buffer has a higher latency, while the circuit has a lower latency that is more deterministic according to [26]. Finally the jitter sensitivity is considered. Preferably the transmitter shall not be sensitive to the jitter in the `TXUSRCLK`. Taking these trade-off criteria into consideration it was opted to bypass the buffer to create a more deterministic system.

After these parts of the GTH the clock crosses into the PMA layer, driving the polarity control and part of the PISO. Here it gets combined with the externally provided MGT reference clock, coming from the Quad Phase-Locked Loop (QPLL). As explained the FPGA is broken up into quads, four of which are MGT quads. Each one of these has two QPLLs, called QPLL0 and QPLL1 respectively. The main advantage of these PLL variants is that they operate on exceptionally high frequencies, 16 and 8 GHz respectively [28]. QPLLs are often used for high line rate applications with better overall jitter performance [28]. Hence QPLL1 is selected to drive the the transmitter phase interpolators and the clock divider, which are necessary for the transmission and controlling the SFP.

At this point in the design, all the clocks and design decisions for the transmitter have been made. This has resulted in a OOK modulated optical signal that is receivable at another destination, which for this system will be the receiver.

The internal structure for the receiver part of the GTH is similar to that of the transmitter. It also consists of the PCS and PMA layer that will translate the clock signals from the SFP towards a Rx interface. The structure including all the CDCs are illustrated in figure 6.6.

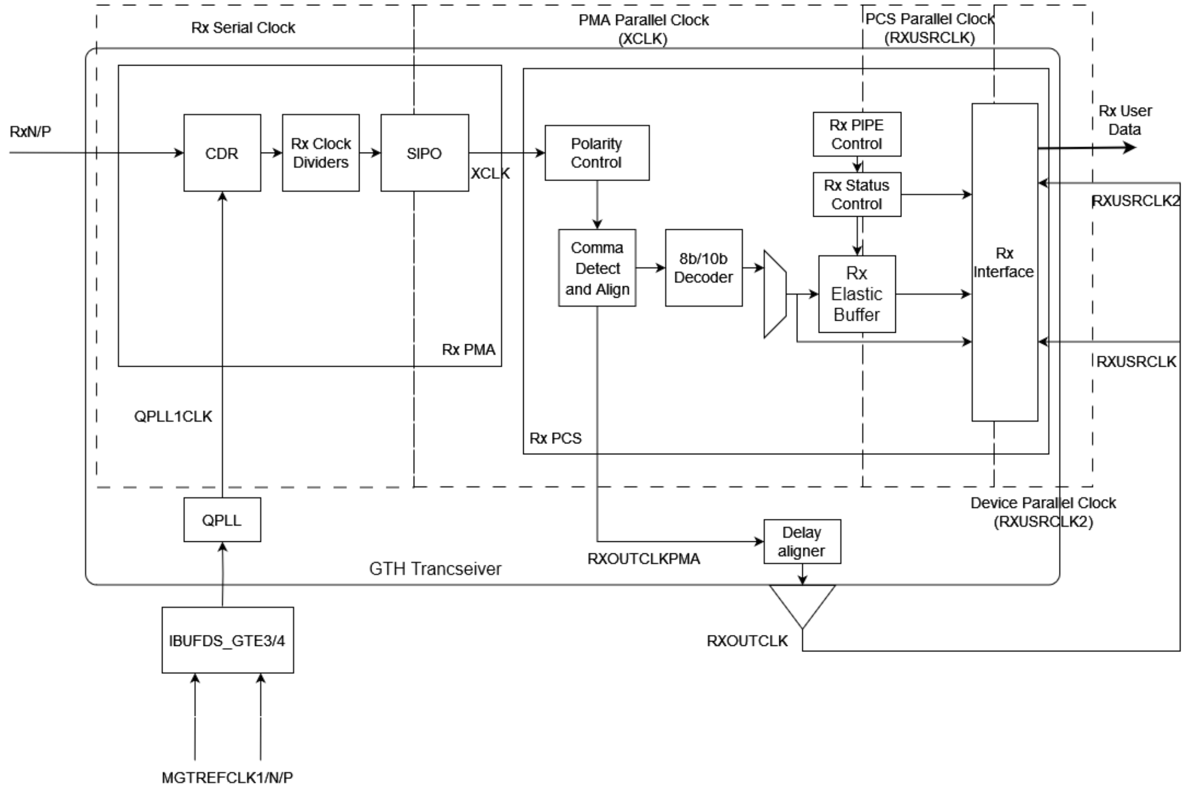


Figure 6.6: Clock domains and CDC of the receiver GTH

Starting this time at the point where the data is received, on the left hand side of figure 6.6. Here Clock Data Recovery (CDR) firmware recovers a clock that is embedded in the received data, which uses the same reference clock source from QPLL1 as the transmitter. This clock is divided into to the appropriate magnitude and drives the SIPO, where it crosses into the PMA clock domain. The recovered clock, now the `XCLK` drives the polarity control, the comma detect and align circuit and the decoder.

A comma, is a special bit sequence used to align symbols in a continuous data stream, which will be further elaborated upon in subsection 6.3.5. The presence of the decoder is a direct consequence of having 8b/10b line encoding selected in the transmitter. The `XCLK` driving these parts, is directly exported outside of the GTH as `RXOUTCLK`, driving the Rx interface.

From the decoder it enters a MUX, which is placed here to illustrate the possibilities of having different decoders for different line encoding options, or no line encoding at all. After the MUX the option to bypass the receiver buffer presents itself. Again a trade-off can be made based on the criteria from [26], presented in table 6.2. The option between the elastic buffer and a phase alignment circuit exists.

Table 6.2: Trade-off criteria for bypassing the Rx buffer, taken from [26]

	Rx Buffer	Rx phase alignment
Ease of Use	The Rx buffer is robust and easy to operate	Requires extra logic and additional constraints on the clock sources.
Clocking Options	Can use Rx recovered clock or local clock with clock correction	Must use the Rx recovered clock
Initialization	Works immediately	Must wait for all clocks to stabilize before performing the Rx phase and delay alignment procedure
Latency	Buffer latency depends on features used	Lower deterministic latency

For the first criterion a similar trade-off can be made as with the transmitter. As Xilinx provides a basic template that is suitable for this application, the ease of use can be negated. The clocking options are irrelevant in the trade-off as it is a matter of simple selection. The initialization is according to the simulations several hundred nanoseconds in which nothing happens, so it does not influence the ranging performance. Finally, in terms of overall latency, bypassing the buffer results in a lower deterministic latency, which is preferred, hence the buffer is bypassed.

After this bypass, the clocks cross into the PCS layer, operating at RXUSRCLK. From the Rx interface the received data can be extracted accordingly to be processed further into the user code.

6.3.4. Signal data generation

The firmware is flexible to what type of signal data is inserted. By default a Pseudo Random Bit Sequence (PRBS) is selected that has the desired length of the selected data width of the signal. The Ultrascale GT wizard allows for data widths of 16, 32 and 64 bits. Within the system firmware there is a small part that is dedicated to verifying the validity of the received PRBS integer against the transmitted PRBS integer. This does however not allow for easy verifiable results as the measured signal is seemingly random. An internal error checking mechanism can be employed, but the random nature of the PRBS itself, makes post-processing, verification, and validation significantly more complex.

Employing a PRBS makes visibly verifying the validity of the signal impossible and, as presented in section 6.6, the data can include errors due to various artifacts. Therefore, it was decided to transmit a counter instead, of which the behaviour is predictable and errors can be observed with relative ease.

The data acquisition firmware from chapter 5 imposes limitations on the data width of the signal itself. The data is temporarily stored in memory, which is limited in size. Larger data widths therefore results in a smaller window in time that can be observed and vice versa.Therefore a compromise is made between the data width of the counter and the capture window size.

The system allows for two signals, transmitting and receiving, to be captured when both have a data width of 16-bits, with a capture window width of 262144 samples. This allows for a 16-bit counter to be implemented as the data to be transmitted and received. The maximum unsigned 16-bit integer this counter is able to count to will be:

$$N_{max} = 2^{16} - 1 = 65535 \tag{6.2}$$

As depicted in subsection 6.3.2, the counter is running on TXUSRCLK2, derived from the 25 MHz free running clock provided to the MGT. Naturally, this clock may drift slightly from the free running clock but the frequency at which it operates remains practically constant. But this does introduce jitter into the system which will have to be tolerated. The rollover time of the counter is thus:

$$t_{rollover} = \frac{1}{25 \cdot 10^6} \cdot 65535 = 2621.4 \mu s \tag{6.3}$$

This in turn would translate to an ambiguity range of 786.42 km, imposing an upper limit on the current system that the total delay of the signal is no more than 2621.4 µs. This delay is not achievable with the experiment, therefore a 16-bit counter is considered to be sufficient.

6.3.5. 8b/10b line encoding and word alignment

8b/10b encoding is a line code that transforms an 8-bit word to 10-bit symbols, achieving DC balance and bounded disparity, simultaneously allowing reasonable clock recovery [29]. It achieves DC balance by keeping track of the Running Disparity (RD) of the serial transmission and adapting the line code to balance the RD around zero. Because of this, it provides the data with frequent transitions between one and zero bits, easing clock data recovery. Depending on the RD, a block of data is translated to a pre-determined sequence according to [30].

The way the GTH transceiver operates is by transmitting words with a pre-determined word length. It is important to denote the difference between symbols and words for the following sections. With 8b/10b line encoding, 8 bits of data is translated to a symbol of 10 bits. A word that is transmitted through the GTH transceiver consists of N symbols.

With the devised signal from subsection 6.3.4, the data to be transmitted is 16 bits wide, this is translated into two symbols with a width of 10 bits each. In a continuous serial communication link, the receiver needs to find the symbol boundaries such that the bits of the words align with the transceiver.

Word alignment is achieved using plus commas and minus commas. These commas are set inside of the GTH transceiver architecture. In the current design the comma values are left on the default value from the GTH IP. The values are presented in table 6.3. When these commas are detected on the serial link, the word boundary slips accordingly until the commas are aligned. The system then locks the alignment for the remainder transmission such that all the symbols are received properly.

Table 6.3: Plus and Minus commas used for word alignment in the GTH transceiver

Plus Comma	Minus Comma
1001111100	0110000011

The comma alignment process is necessary, and for 8b/10b encoding this is automated by the Xilinx IP. It does, however, introduce some extra latency to the system. Manual alignment can be implemented using the `RXSLIDE` feature and it can be done to the same extent, requiring custom firmware. Institutions such as CERN often opt to write their own alignment controller [24], [31]. However, for the sake of convenience it is opted to use the automatic alignment to save time, while having to tolerate the jitter penalty.

6.4. Method

As mentioned in the objective in section 6.1, a signal must be transmitted through one of the optical transceivers on the FPGA board. The KCU105 has a SFP+ cage that is able to hold two SFP modules, allowing for a total of two optical transmitters and receivers respectively, creating two channels in total. This cage can be found on the board as illustrated in figure 1.1. SFP is a currently a universal standard for networking and telecommunications equipment. Using these modules in theory makes the system implementable in a wide variety of free-space test setups, where a dedicated SFP module can be tailored to free-space communications.

For optical communication systems implemented in satellites, the ToF is recovered from the signal structure itself. Hence signal generation and recovering ToF measurements lies at the core of this experiment, thus this section aims to outline the methods used to retrieve ranging measurements from the communication stream. Firstly, the signal generation method is outlined in subsection 6.4.1. Secondly, the ToF calculation method from the communication stream is presented in subsection 6.4.2. When implementing optical communications, in a space system, the range is recovered from. Next the calibration of the system is described in subsection 6.4.3, followed by the post-processing methods in subsection 6.4.4.

6.4.1. Signal Generation

Similar to literature [6], [16], [32], the proposed communication convention includes a RCID. A detailed description of the proposed convention for optical telemetry is described in the literature study or the technical report proposing the convention [12], [32].

The proposed data steam consists of RCW. It consists of SMCW that are delimited by a synchronization marker, followed by data codewords. The first S amount of symbols after the first synchronization marker of a data packet are dedicated to the RCID. This is illustrated in figure 6.8.

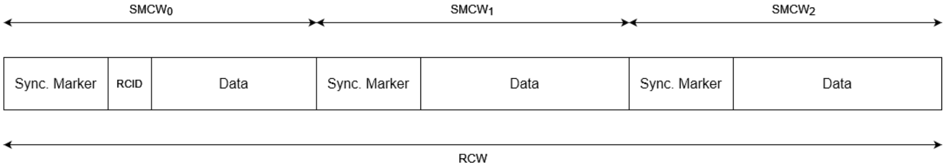


Figure 6.7: Data stream proposed for optical telemetry ranging by [32]

These RCIDs are unique in the sense that, for a given RTLT, the same RCID is not transmitted multiple times before the response corresponding to that RCID is received. Meaning that the rollover period of the RCID sequence needs to be larger than the anticipated RTLT in order to avoid ambiguity.

$$t_{rollover} \geq t_{RTLT} \tag{6.4}$$

Moreover the RTLT is also limited by the width of the internal data capture firmware, which is in turn limited by the rollover time. The data capture firmware for this experiment has an internal width of 262144 data points at 200 MHz. In order for the system to be able to capture a response to a transmitted RCID, the RTLT must be smaller than the total width in terms of time of the capture window.

$$t_{RTLT} \leq t_{dc} \tag{6.5}$$

For the purpose of this thesis, only the RCID symbols will be transmitted, as these are only relevant to the ranging measurements, without insertion of any data block. The system that dictates the value of the RCID is chosen to be a standard counter. The value of the RCID increases every rising edge of the data clock. The data stream is illustrated in figure 6.8.

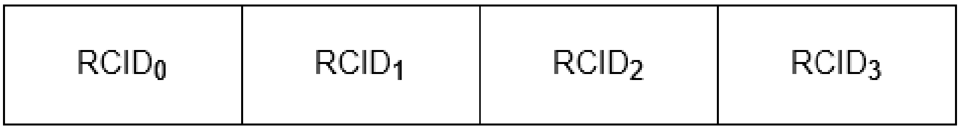


Figure 6.8: Data stream implemented in the system

Simplifying the signal to only transmitting the RCID does however have the downside that the amount of range measurements that can be conducted per second is substantially higher than what can be conducted while embedding the RCID into the normal data stream. This is due to the fact that the number of ranging symbols per second that are transmitted is reduced and padded by other data such as scientific data. This has the consequence that, for a real system the measurement rate is reduced causing a higher uncertainty for the same integration time.

6.4.2. Time-of-Flight Computation

The signal structure from subsection 6.4.1 is in turn used to compute a range measurement with for the system. This can be done using the attributes listed in table 6.4.

Table 6.4: Variables involved in retrieving the range measurement from the data stream.

Transceiver Variables	
$RCID_u(t_T)$	The RCID of an uplink data packet at the time of transmission
$\psi_u(t_R)$	The phase of the uplink signal at the time of reception
$t_T - t_0$	The time of transmission relative to the start of a measurement
Receiver Variables	
$RCID_d(t_R)$	The RCID of an uplink data packet at the time of reception
$\psi_d(t_R)$	The phase of the downlink signal at the time of reception
$t_R - t_0$	The time of reception relative to the start of a measurement

These variables and methods are derived from the original concepts from [6] and [16]. These two works consider the concept of optical telemetry ranging, where identifiers are embedded into the data stream in addition to some of the variable listed in table 6.4. These works consider a master-slave configuration of two terminals A and B, which can be considered to be a ground station - Spacecraft (SC) or SC-SC system. It can be represented by a simple block diagram as depicted in figure 6.9.

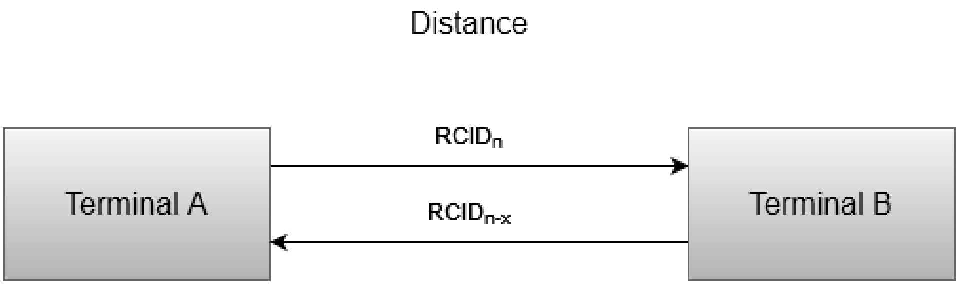


Figure 6.9: High level block diagram of the system considered for optical telemetry ranging.

For this system the following assumptions hold for this experiment:

- The data clocks are perfectly synchronized and remain that way for a long enough duration to perform ranging.
- The transmission of a RCW at terminal B starts at the instant a RCW is received at terminal B.

In this figure an RCID with number N is transmitted. This is processed then processed at terminal B, which immediately sends a response to an RCID it received x instances ago. For the system discussed in this chapter, as it is a loopback system, terminal B from figure 6.9 can be considered to be a terminal with no internal delay, or in other terms a simple mirror. Hence the response of terminal B to RCID N is the same identifier.

The data stream is then better illustrated using the original figures from the technical reports [6], [16], depicted in figure 6.10 and figure 6.11. Both images have quite the overlap, but use different terms as figure 6.10 depicts the original concept, and figure 6.11 presents a particular use case of the concept.

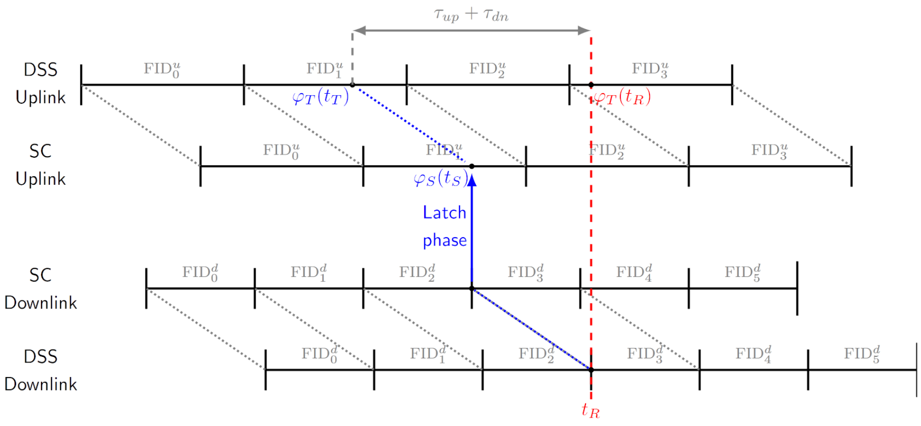


Figure 6.10: Optical telemetry ranging concept from [6].

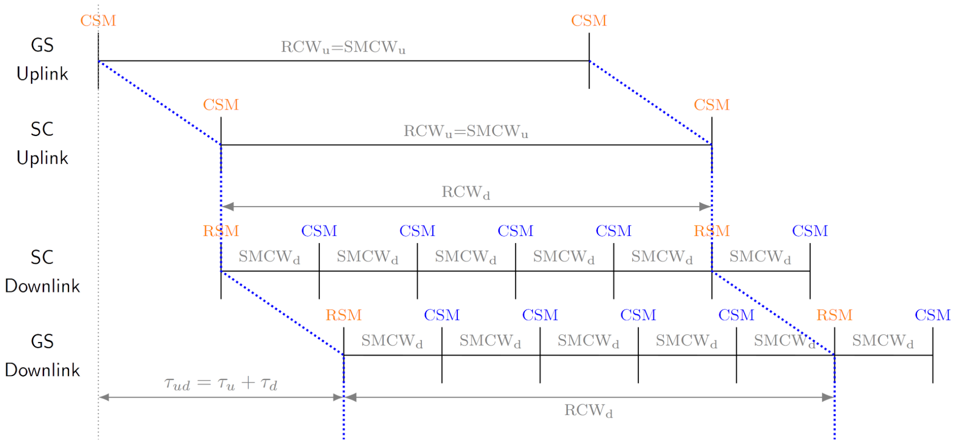


Figure 6.11: Original synchronous optical telemetry ranging concept from [16].

The Frame Identifier (FID) in figure 6.10 is the same as a RCID in figure 6.11 as it shares the same purpose. These figures do however show two different data structures on one link, as may be customary when using multiple terminals. For the system discussed in this chapter however, the up- and downlink have the same structure. Hence the situation can be condensed to the illustration in figure 6.12. t_0 is taken as the time a measurement is initiated. In this case this will be the start of a capture window from the data capture firmware.

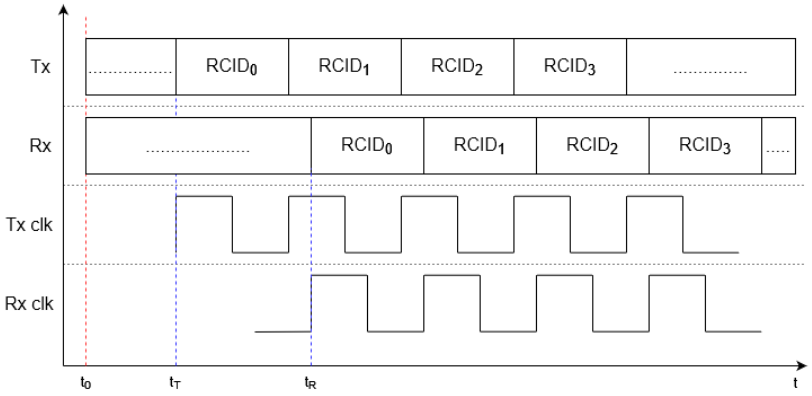


Figure 6.12: Signal transmitter and receiver responses with accompanying data clocks.

The way the data capture is structured, signal measurement starts at an arbitrary point in time t_0. From there, a window is captured with 262144 samples at 200 MHz. The RCIDs of both the transmitter and receiver are captured simultaneously as well as their associated clocks called `TXUSRCLK2` and `RXUSRCLK2`, these clocks directly correspond to the respective data stream [26].

The data stream is captured and exported to Python for further processing. This procedure involves the extraction of range information from the measurement data by looking at the RCID values for the up- and downlink, along with the two clocks. The algorithm initiates by indexing each unique RCID value in the transmission data and due to the lower frequency of both the transmission data and the sampling clock, a single RCID may be sampled multiple times.After indexing the RCID values in the transmission data, the algorithm proceeds to identify the first subsequent occurrence of that exact RCID in the receiver data, where it is similarly indexed. The subsequent step involves subtracting these indexes, allowing the computation of the difference between $RCID_d$ at t_R and $RCID_u$ at t_T using Equation 6.6.

$$\Delta s_i = \{\mathrm{RCID}_d(t_R)\}_i - \{\mathrm{RCID}_u(t_T)\}_i \tag{6.6}$$

Here Δs_i represents the difference in samples between two RCIDs on the receiver and transmitter at point i. This value can be converted to the total delay time using Equation 6.7.

$$\Delta t_{total\ i} = \Delta s_i \cdot T_s \tag{6.7}$$

However, this method determines the total delay at a given instance i of the setup. This total time can be deconstructed into two parts, the ToF of the signal as well as a term that accounts for all the other delays experienced inside of the FPGA (Equation 6.8).

$$\Delta t = \Delta t_{total} - \Delta t_{internal} \tag{6.8}$$

At this stage, the internal delay should have been thoroughly calculated in chapter 5. However this is not the case as the MGT is implemented, introducing a significant internal delay. To measure this internal delay calibration is again required, the method of which will be presented in subsection 6.4.3.

Another important aspect to take into account when determining the ToF using Equation 6.8 is the refractive index of the medium through which the photons travel. This influences the propagation speed significantly, being only a fraction of the speed of light in vacuum. As mentioned in section 6.2, the medium is a G.657.A1 optic fiber with a silica core: this has an effective group index of refraction at 1550 µm of 1.468[1].

The range will need to be corrected in terms of the refraction in the medium. This can be done by employing Equation 4.9. Therefore the final range that is computed for a given measurement can be described by Equation 6.9.

$$\Delta r_i = \frac{\Delta t \cdot c}{\bar{n}} \tag{6.9}$$

Where Δr_i is the range measured at a given instance i.

6.4.3. Calibration

In order to get meaningful measurements, the final system will have to be calibrated appropriately. Initially, the methods in chapter 5 were devised to calibrate the system for this experiment. The problem with this method is that the signal in that experiment, does not travel through the MGT, only through the board to the GPIO pins. Hence the delay found in chapter 5 cannot be considered to be representative for this experiment and the calibration has to be re-evaluated.

In an attempt to quantify this delay, the internal MGT loopback feature can be employed. This is a feature for which the user can provide input through the Virtual Input Output (VIO) core in the FPGA firmware. Using this feature in theory can show the results of a transmission when passing through the GT firmware. This loopback path is illustrated in figure 6.13.

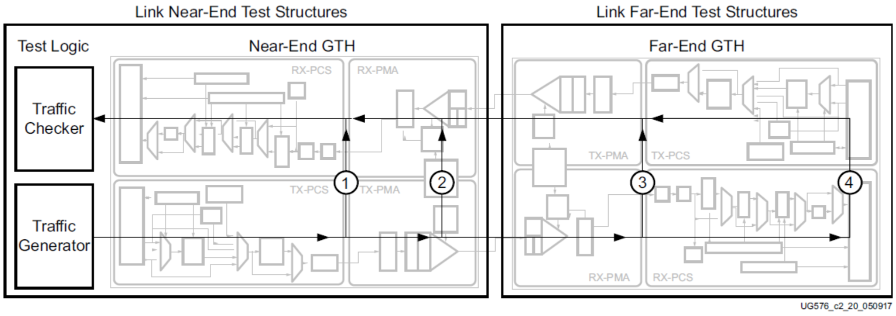


Figure 6.13: Schematic of the internal loopback feature provided in the GT firmware, taken from Xilinx User Guide [26]

[1] https://www.optokon.com/ms/product/260-single-mode-fiber-g-657-a1, [accessed 21-9-2023]

From figure 6.13 can be seen that there are 4 loopback modes, near-end PCS and PMA loopback and far-end PMA and PCS. Providing the appropriate VIO input to the system routes traffic through that region of the GTH. Based on these results a calibration about the internal delay can be made.

However, in order for the loopback to work as intended, additional settings are required next to only providing this VIO input. The process of providing these additional settings introduces various errors and bugs in the system, in addition to other technical difficulties, which causes the internal loopback to malfunction, providing unusable results.

Solving this issue would be time consuming and given the timing constraints imposed on the thesis, achieving this will not be possible in the given time frame. Given the importance of this internal delay on the accuracy of the timing measurements, alternative methods were selected in the form of physical loopback adapters: an optical fiber loopback adapter and a bridged SFP loopback adapter, illustrated in figure 6.14 and figure 6.15 respectively.

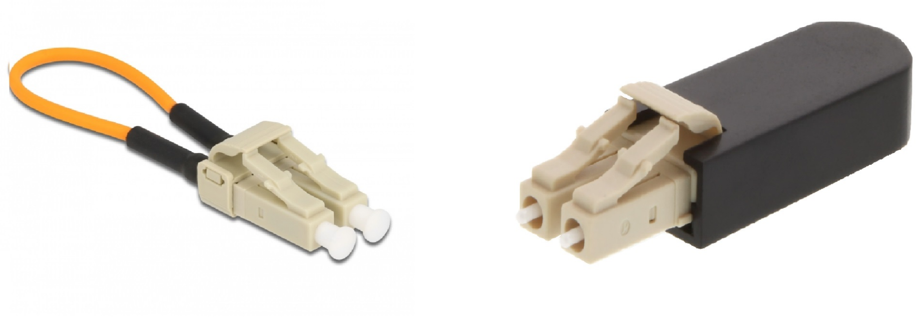


Figure 6.14: Optic fiber external loopback adapter. On the left is the same adapter without cover, on the right the cover is put back in place

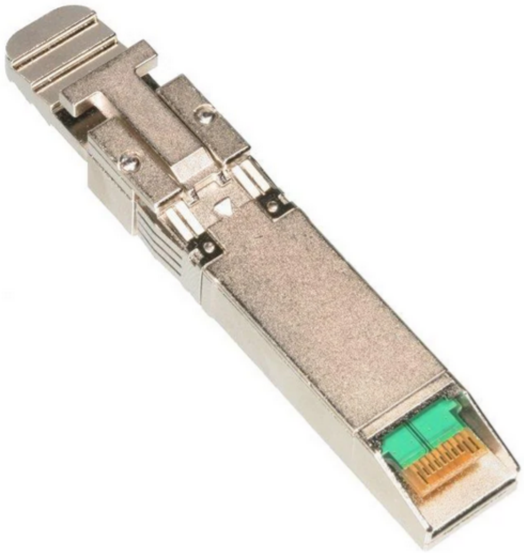


Figure 6.15: SFP loopback adapter. The internal terminals are electronically bridged.

These adapters can be inserted into the SFP slot and create a near instant loopback, from which the results can be used. The difference between the two devices is that the optic fiber loopback adapter from figure 6.14 is inserted into the SFP module by which the measurements are made, while the other device replaces this SFP module entirely.

Theoretically, the adapter from figure 6.14 will account for the added latency from the laser and photo-detectors in the SFP module itself as well as a relatively small path through the optic fiber cable. While the adapter from figure 6.15 excludes these delays and therefore provides a better representation of the real internal delay. The disadvantage however is that the latency on the laser and photo-detector remains unknown.

The difference between the two values are expected to be negligible as the delays added by the laser, photo-detector and small optic fiber cable are not expected to be noticeably large, meaning that they exceed the resolution of the system. The resolution of the system can be calculated using Equation 4.9. Where c is the speed of light in vacuum, $\bar{n}$ is the refractive index of the medium in which the photons are travelling and τ is, in order to calculate the resolution, the minimum distinguishable unit of time, which is 5 ns for this system. Combining this with the refractive index of the optic fiber of 1.468 yields a resolution of 1.02 m. The length of the calibration devices do not exceed this resolution, hence the assumption that the delay is unnoticeable is theoretically confirmed. However, calibration will be performed using both of these devices to validate this hypothesis.

The same signal as devised in subsection 6.4.1 will be transmitted as with the regular tests. The same method to recover the ToF will be employed, however Equation 6.10 will now be assumed to hold true.

$$\Delta t_{total} = \Delta t_{internal} \tag{6.10}$$

With the results from calibration, the measurements from the final tests can be corrected accordingly by subtracting the obtained results by the calibration time according to Equation 6.8.

It is expected that the internal delay depends on the line rate of the system. Before parallel data is transferred from the input to output, a minimal amount of absolute clock cycles will elapse. Hence, for slower line rates the internal delay will be larger than for faster line rates.

6.4.4. Post-Processing

The data obtained from the measurements will need post-processing. This is due to the fact that the clocks responsible for measuring the data are not synchronized to the clocks responsible for transmitting and receiving, causing evident glitches in the data. An example of such events is illustrated in figure 6.16.

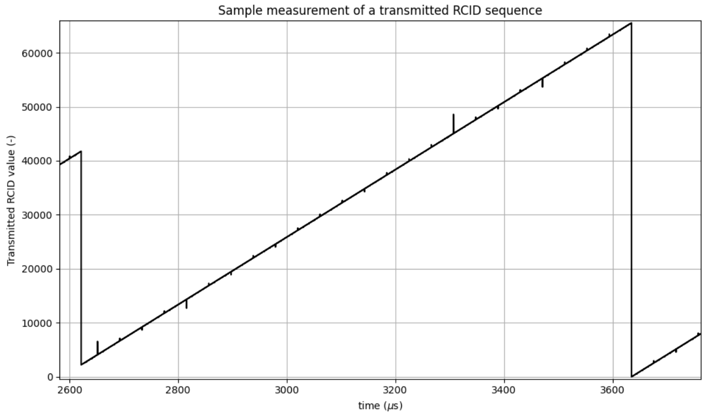


Figure 6.16: Spikes in the transmitted RCID data due to out-of-sync transmitter and sampler clocks

To confirm these spikes are due to synchronization issues, the receiver side is also observed. In case the data is faulty, it shall naturally propagate to the receiver as well. But as observed in figure 6.17, the data glitches did not propagate to the receiver: in fact the received values of this particular captured window are mostly glitchless.

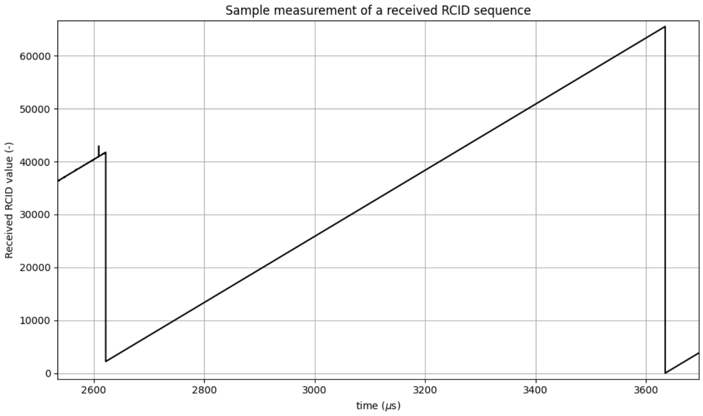
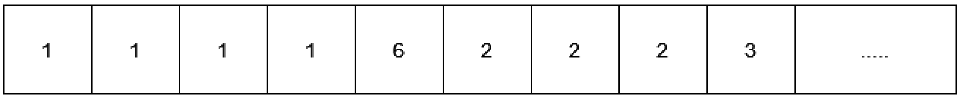


Figure 6.17: no-spikes in the received RCID data, clocks are still out-of-sync.

It may be concluded that these are in fact errors in the observations rather than actual errors in the data. These glitches, referred to as spikes, can be statistically ignored. They can be filtered in the FPGA hardware itself, however due to time constraints, post-processing was opted for.

Each RCID value is expected sampled $f_s/f_{USRCLK2}$ times. Each unique value in the sequence can be listed and it can be counted how many times these values occur and where, moreover it is known that the RCID is set to increase by one every $f_s/f_{USRCLK2}$ clock cycles. Knowing this, the spikes can be identified using Equation 4.4. In case the value for $y[i+1]$ is not equal to 1, it may be considered to be a spike. In order to correct the spike to its original value, it must be determined to which RCID value it belongs.

In order to do this the values before and after the spike need to be observed. As mentioned before, in general one RCID should occur $f_s/f_{USRCLK2}$ clock cycles, hence it is expected to find $f_s/f_{USRCLK2}$ consecutive RCIDs before and after the outlier. An illustration of such a situation is presented in figure 6.18.



1	1	1	1	6	2	2	2	3	.....

Figure 6.18: Illustration of a situation where an outlier is embedded in the RCID data stream.

As can be seen in this figure, the amount of RCIDs that are expected consecutively is 4. There is definitely an outlier between 1 and 2 as the increment is more than 1, and by looking at the values before and after the position of 6, it can be determined that the 6 should have been a 2 as there are only 3 twos where 4 are anticipated. Similarly if it would have been the other way around if the number of ones were less than the anticipated number of repetitions.

To mitigate the issue however, as explained in subsection 6.3.2, the transmitter clock and sampler clock are synchronized to avoid the majority of the outliers, excluding errors that may occur due to the naturally present Bit Error Rate (BER). With this design, there is only one channel that has a large amount of spikes in its data due to synchronization issues.

6.5. Verification

In this section, the verification of the system is presented: it is done with a post-implementation functional simulation using the simulator included in Vivado ML Edition 2021.2. A testbench is written to simulate nominal operations of the system, and the results of the overall system test are presented in subsection 6.5.1. The expected performance of the firmware is then analysed in subsection 6.5.2.

6.5.1. System Tests

A VHDL testbench is written to simulate the behaviour of the firmware. After implementation, a post-implementation function simulation can be conducted. In total, three different tests will be conducted at three different line rates with the system properties facilitating the line rates.

The most important aspect of the firmware are the clock configurations. Most of the firmware is Xilinx IP, which only processes user inputs, which are the user data, clocks, and reset signals. To verify the correct functioning of the system therefore, the free-running clock, `USRCLK`, `USRCLK2`, sample clock, user data and line rate will be verified.

The verification of the clocks will be done with respect to frequency and alignment to the user data in the Vivado simulator. To verify the user data, the progression of the packets themselves will be investigated to verify the increments.

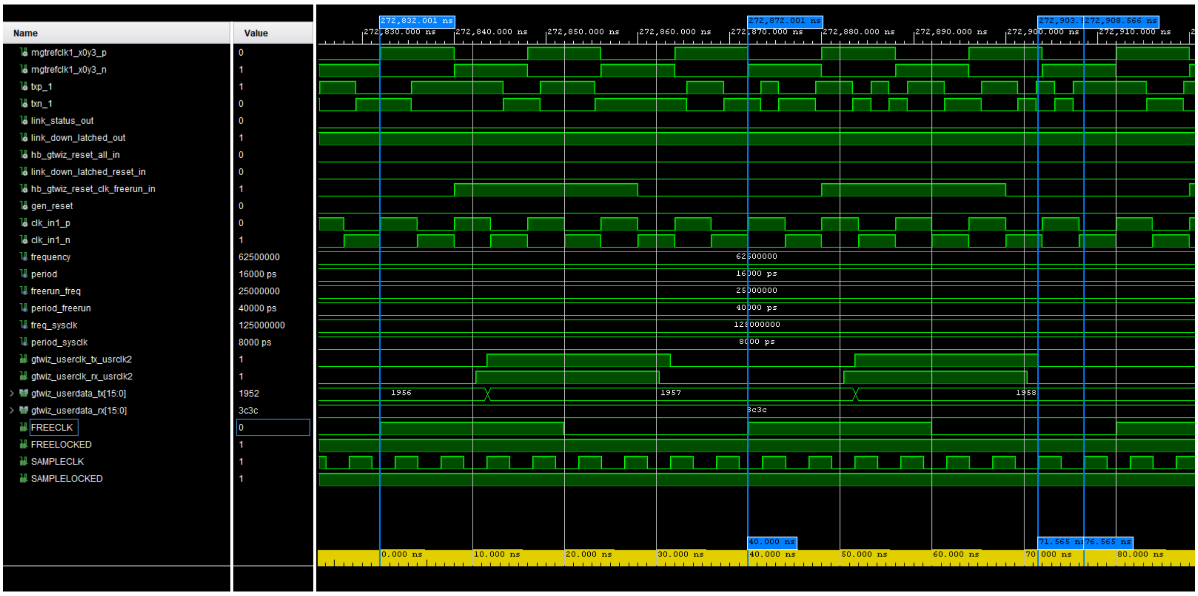
6.5.1.1. Simulation Test 1

A summary of the system parameters of this test is presented in table 6.5.

Table 6.5: System properties Test 1

Property	Value	Unit
Sampling frequency	200	MHz
MGT reference frequency	62.5	MHz
DRP frequency	25	MHz
Line rate	0.5	Gbps
Symbol rate	25	MHz

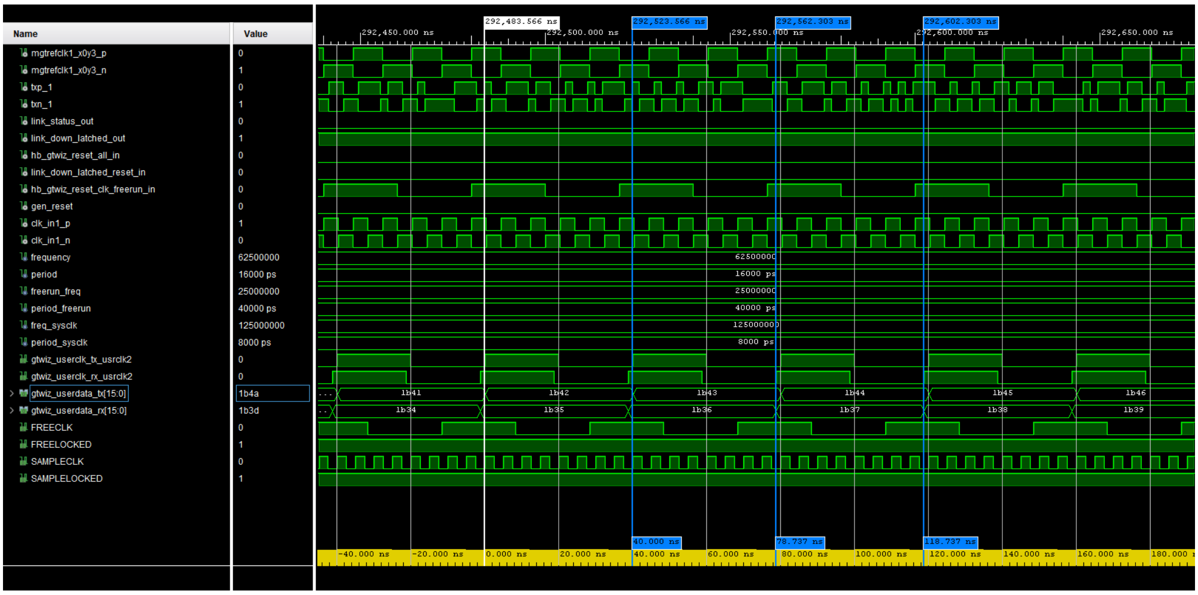
The simulation is run for 500 µs and the signals in the design are observed and illustrated in figure 6.19 until and including figure 6.21. On the left side of these figures the signal names are listed and their corresponding bit value with respect to time is presented by a green waveform. Looking at the free-running clock and the sample clock, `FREECLK` and `SAMPLECLK` respectively, in figure 6.19, it can be observed that from the timestamps that these are derived properly from the input system clock. The period of the sample clock is 5 ns and the period of the free-running clock is 40 ns.



Name	Value
mgtrefclk1_x0y3_p	0
mgtrefclk1_x0y3_n	1
txp_1	1
txn_1	0
link_status_out	0
link_down_latched_out	1
hb_gtwiz_reset_all_in	0
link_down_latched_reset_in	0
hb_gtwiz_reset_clk_freerun_in	1
gen_reset	0
clk_in1_p	0
clk_in1_n	1
frequency	62500000
period	16000 ps
freerun_freq	25000000
period_freerun	40000 ps
freq_sysclk	125000000
period_sysclk	8000 ps
gtwiz_userclk_tx_usrclk2	1
gtwiz_userclk_rx_usrclk2	1
gtwiz_userdata_tx[15:0]	1952
gtwiz_userdata_rx[15:0]	3c3c
FREECLK	0
FREELOCKED	1
SAMPLECLK	1
SAMPLELOCKED	1

Figure 6.19: Sample and free-running clock verification using post-implementation functional analysis for a line rate of 500 Mbps

Next, the user clock frequencies, `gtwiz_userclk_tx_usrclk2` and `gtwiz_userclk_rx_usrclk2`, are verified: they are determined by setting the appropriate clocking constraints. For a correct functioning system, both TXUSRCLK2 and RXUSRCLK2 shall have the same frequency, and the rising edge of these clocks must coincide with the start of a data packet, or else the system is misconfigured.

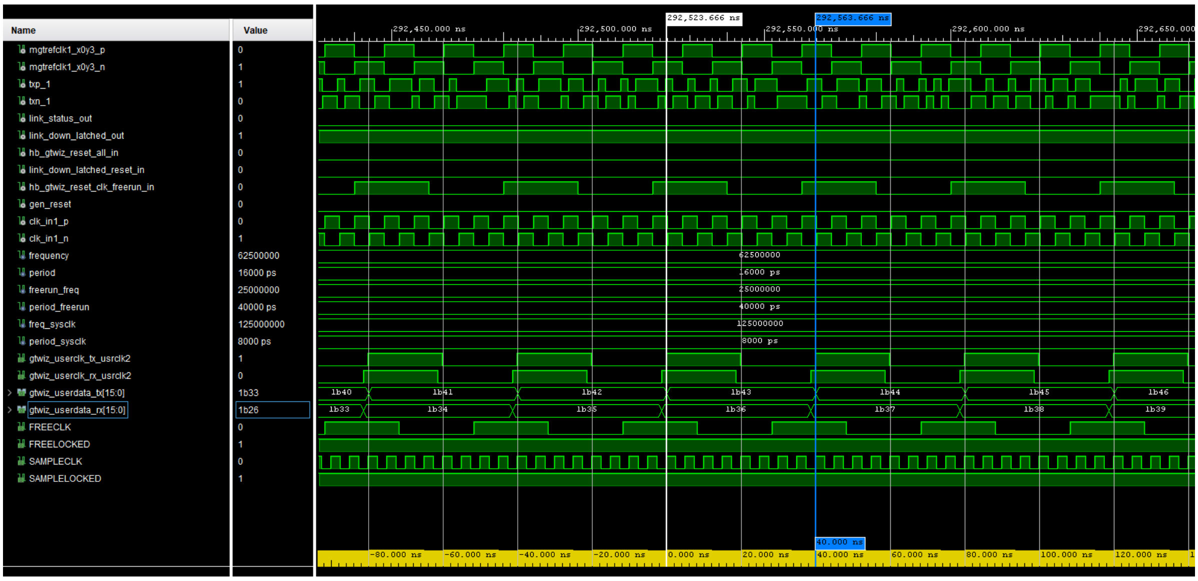


Name	Value
mgtrefclk1_x0y3_p	0
mgtrefclk1_x0y3_n	1
txp_1	0
txn_1	1
link_status_out	0
link_down_latched_out	1
hb_gtwiz_reset_all_in	0
link_down_latched_reset_in	0
hb_gtwiz_reset_clk_freerun_in	0
gen_reset	0
clk_in1_p	1
clk_in1_n	0
frequency	62500000
period	16000 ps
freerun_freq	25000000
period_freerun	40000 ps
freq_sysclk	125000000
period_sysclk	8000 ps
gtwiz_userclk_tx_usrclk2	0
gtwiz_userclk_rx_usrclk2	0
gtwiz_userdata_tx[15:0]	1b4a
gtwiz_userdata_rx[15:0]	1b3d
FREECLK	0
FREELOCKED	1
SAMPLECLK	0
SAMPLELOCKED	1

Figure 6.20: User clock verification using post-implementation functional analysis for a line rate of 500 Mbps

Looking at the waveforms in figure 6.20, the user clocks have a period of 40 ns, resulting in a frequency of exactly 25 MHz as designed. Furthermore, it can be noted that both clocks are closely coinciding with the rising edge of their respective data packet. This proves that the constraints imposed on the user clocks are set correctly and the GTH transceiver is configured to correctly scale down the provided reference clock. Upon closer inspection on the location of the rising edge however, the clock is 100 ps offset from the user data. This offset appears to be constant over the simulation period. This does not appear to influence the functioning of the system and is therefore considered to work as intended. It does not have an influence on the eventual range measurement as the time difference between the transmitter and receiver clock is the same as the time difference between packet.

To conclude, the line rate is verified. As explained in subsection 6.3.3, the internal data rate is proportional to the ratio of the line encoding multiplied by the unencoded word length multiplied by the DRP frequency. figure 6.21 shows the waveform generated by the simulation. A word with a length of 16-bits, lasts exactly 40 ns, translating to a word rate of 400 Mbps. Multiplying this by the encoding ratio of 10/8 yields exactly a line rate of 500 Mbps.



Name	Value
mgtrefclk1_x0y3_p	0
mgtrefclk1_x0y3_n	1
txp_1	1
txn_1	0
link_status_out	0
link_down_latched_out	1
hb_gtwiz_reset_all_in	0
link_down_latched_reset_in	0
hb_gtwiz_reset_clk_freerun_in	0
gen_reset	0
clk_in1_p	0
clk_in1_n	1
frequency	62500000
period	16000 ps
freerun_freq	25000000
period_freerun	40000 ps
freq_sysclk	125000000
period_sysclk	8000 ps
gtwiz_userclk_tx_usrclk2	1
gtwiz_userclk_rx_usrclk2	0
gtwiz_userdata_tx[15:0]	1b33
gtwiz_userdata_rx[15:0]	1b26
FREECLK	0
FREELOCKED	1
SAMPLECLK	0
SAMPLELOCKED	1

Figure 6.21: Line rate verification using post-implementation functional analysis for a line rate of 500 Mbps

From the above figure, it can also be found that the user data on both channels is coherent, indicative of correct functioning of the GTH transceiver.

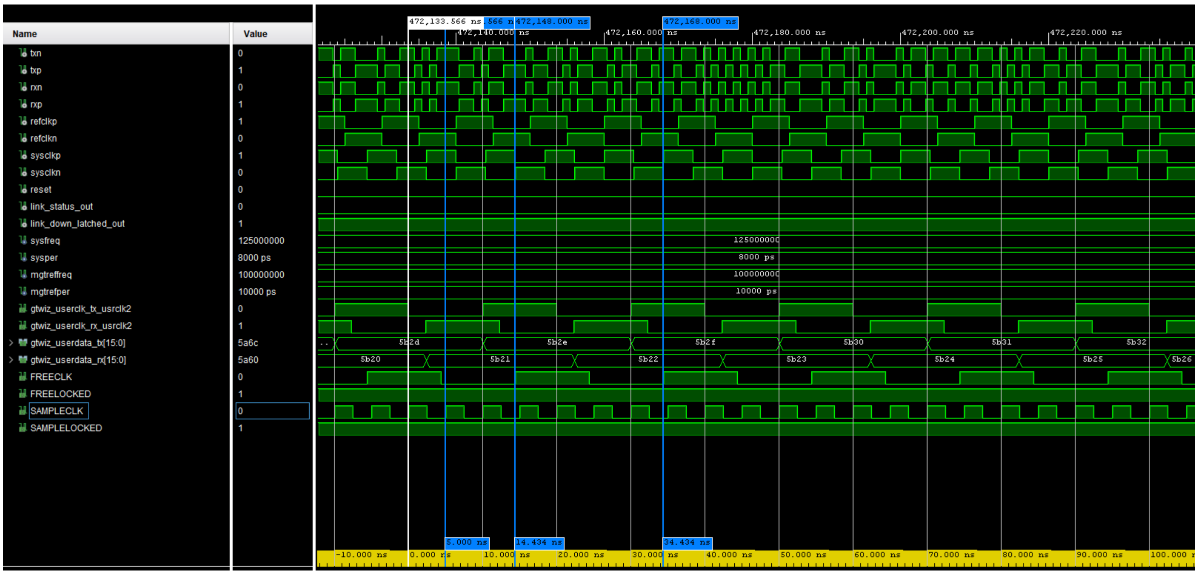
6.5.1.2. Simulation Test 2

The system properties of system 2 are presented in table 6.6.

Table 6.6: System properties Test 2

Property	Value	Unit
Sampling frequency	200	MHz
MGT reference frequency	100	MHz
DRP frequency	50	MHz
Line rate	1	Gbps
Symbol rate	50	MHz

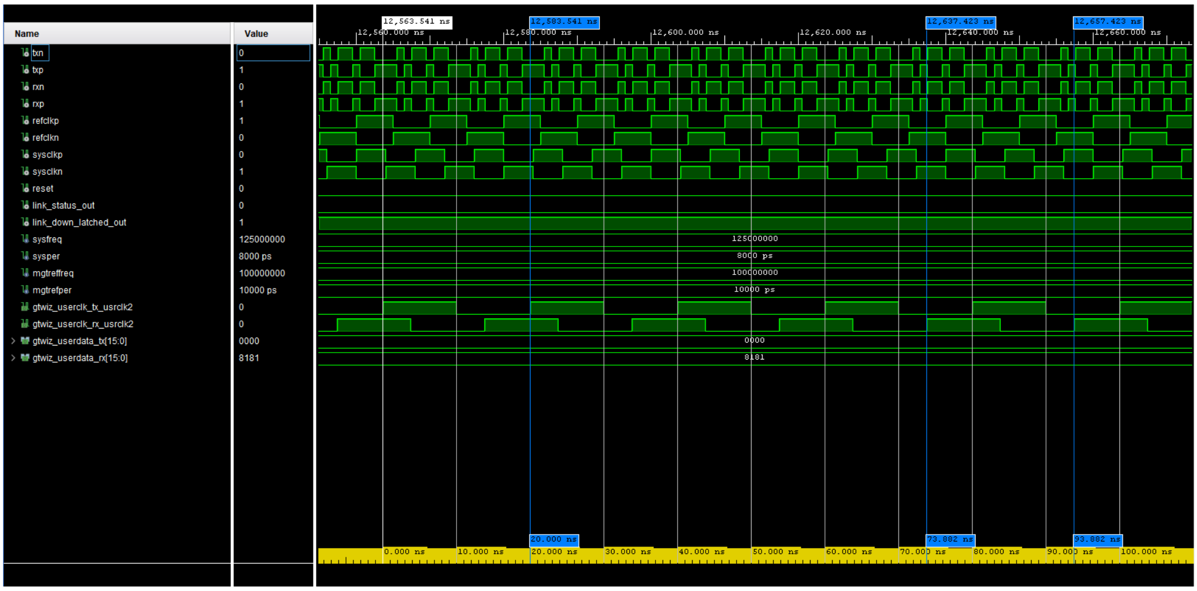
Following the same procedure as before, the simulation is run for 500 µs and the results are carefully observed. First the free-running and sampling clock are verified. The results of which are illustrated in figure 6.22. The constraints of these clocks hold and they are expected to function as intended.



Name	Value
txn	0
txp	1
rxn	0
rxp	1
refclkp	1
refclkn	0
sysclkp	1
sysclkn	0
reset	0
link_status_out	0
link_down_latched_out	1
sysfreq	125000000
sysper	8000 ps
mgtreffreq	100000000
mgtrefper	10000 ps
gtwiz_userclk_tx_usrclk2	0
gtwiz_userclk_rx_usrclk2	1
gtwiz_userdata_tx[15:0]	5a6c
gtwiz_userdata_rx[15:0]	5a60
FREECLK	0
FREELOCKED	1
SAMPLECLK	0
SAMPLELOCKED	1

Figure 6.22: Sample and free-running clock verification using post-implementation functional analysis for a line rate of 1 Gbps

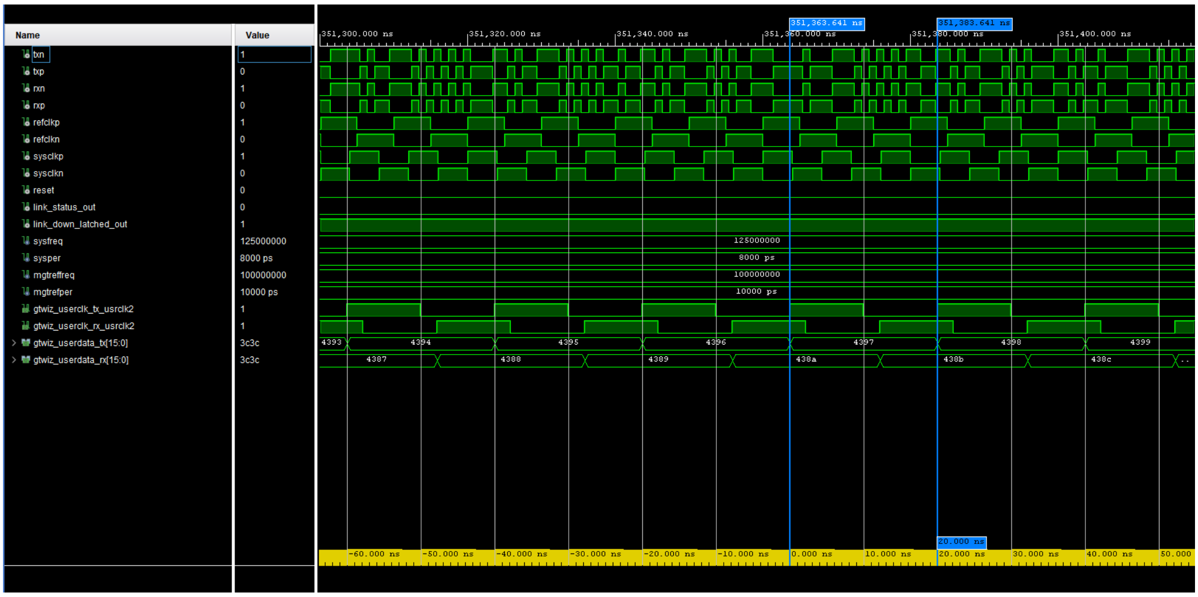
The user clocks, illustrated in figure 6.23, function as intended with a frequency of 50 MHz.



Name	Value
txn	0
txp	1
rxn	0
rxp	1
refclkp	1
refclkn	0
sysclkp	0
sysclkn	1
reset	0
link_status_out	0
link_down_latched_out	1
sysfreq	125000000
sysper	8000 ps
mgtreffreq	100000000
mgtrefper	10000 ps
gtwiz_userclk_tx_usrclk2	0
gtwiz_userclk_rx_usrclk2	0
gtwiz_userdata_tx[15:0]	0000
gtwiz_userdata_rx[15:0]	8181

Figure 6.23: User clock verification using post-implementation functional analysis for a line rate of 1 Gbps

For the line rate, illustrated in figure 6.24, the word rate is calculated to be 800 Mbps, translating to a line rate of 1 Gbps. Furthermore, the user data on both channels is found to be coherent with increments of 1, indicative of a correct configuration of the GTH transceiver firmware block and counter.



Name	Value
txn	1
txp	0
rxn	1
rxp	0
refclkp	1
refclkn	0
sysclkp	1
sysclkn	0
reset	0
link_status_out	0
link_down_latched_out	1
sysfreq	125000000
sysper	8000 ps
mgtreffreq	100000000
mgtrefper	10000 ps
gtwiz_userclk_tx_usrclk2	1
gtwiz_userclk_rx_usrclk2	1
gtwiz_userdata_tx[15:0]	3c3c
gtwiz_userdata_rx[15:0]	3c3c

Figure 6.24: Line rate verification using post-implementation functional analysis for a line rate of 1 Gbps

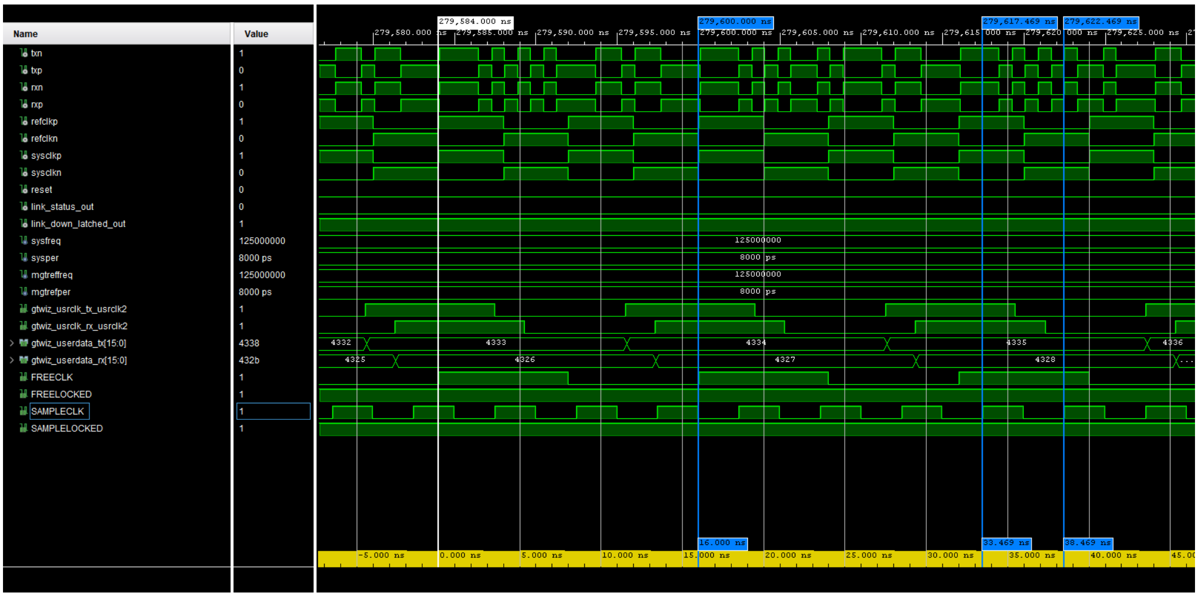
6.5.1.3. Simulation Test 3

The system properties of system 2 are presented in table 6.7.

Table 6.7: System properties Test 3

Property	Value	Unit
Sampling frequency	200	MHz
MGT reference frequency	125	MHz
DRP frequency	62.5	MHz
Line rate	1.250	Gbps
Symbol rate	62.5	MHz

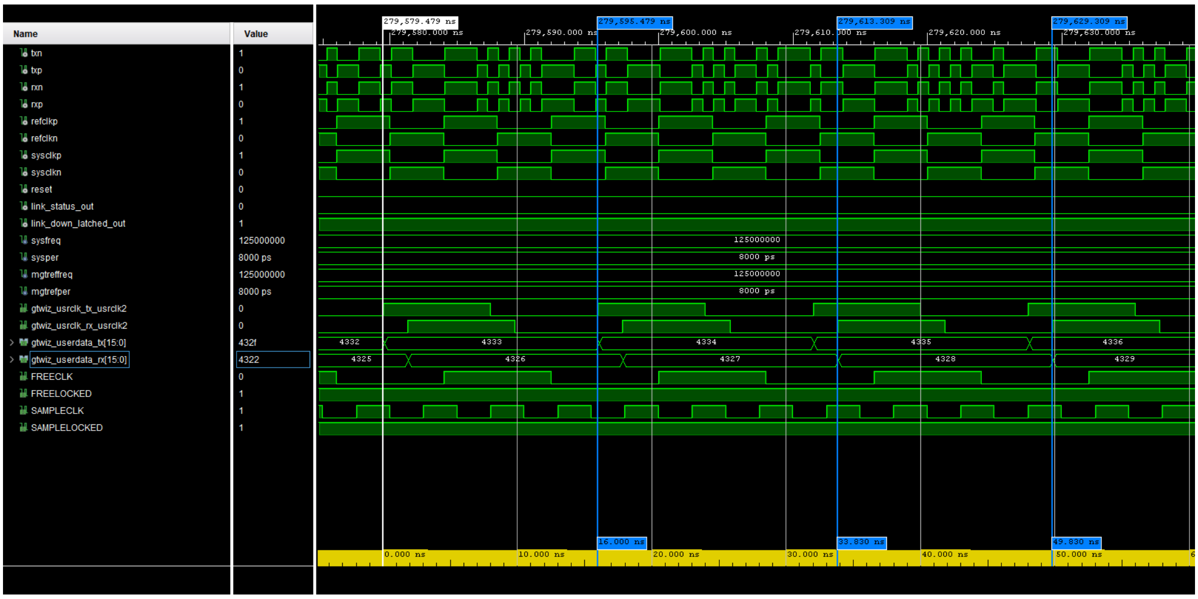
Following the procedure from the previous tests. The results from the free-running and sample clock are illustrated in figure 6.25. The test results in normal functioning clocks with their desired frequencies.



Name	Value
txn	1
txp	0
rxn	1
rxp	0
refclkp	1
refclkn	0
sysclkp	1
sysclkn	0
reset	0
link_status_out	0
link_down_latched_out	1
sysfreq	125000000
sysper	8000 ps
mgtreffreq	125000000
mgtrefper	8000 ps
gtwiz_usrclk_tx_usrclk2	1
gtwiz_usrclk_rx_usrclk2	1
gtwiz_userdata_tx[15:0]	4338
gtwiz_userdata_rx[15:0]	432b
FREECLK	1
FREELOCKED	1
SAMPLECLK	1
SAMPLELOCKED	1

Figure 6.25: Sample and free-running clock verification using post-implementation functional analysis for a line rate of 1.25 Gbps

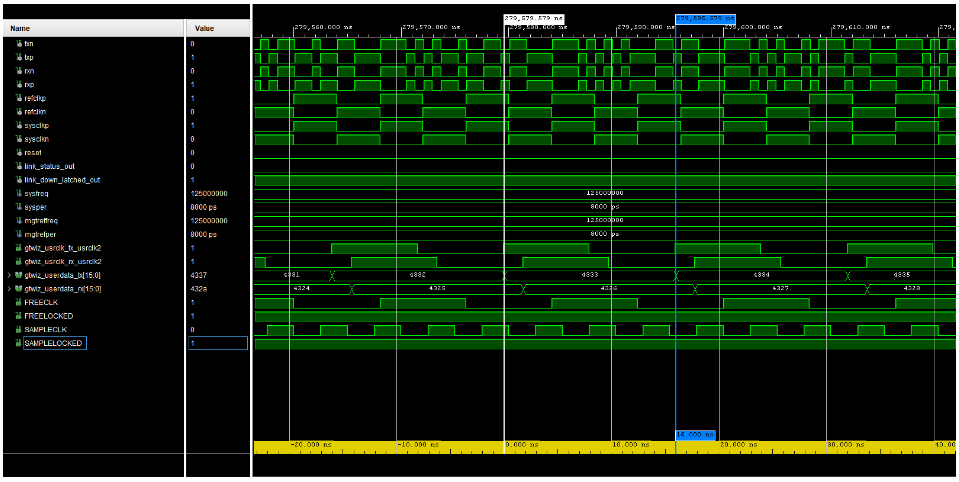
The user clocks for system 3 are simulated to run at their supposed frequencies. The clocks closely coincide with the user data, lagging 100 ps behind the corresponding data packet, similar to the other simulation results.



Name	Value
txn	1
txp	0
rxn	1
rxp	0
refclkp	1
refclkn	0
sysclkp	1
sysclkn	0
reset	0
link_status_out	0
link_down_latched_out	1
sysfreq	125000000
sysper	8000 ps
mgtreffreq	125000000
mgtrefper	8000 ps
gtwiz_usrclk_tx_usrclk2	0
gtwiz_usrclk_rx_usrclk2	0
gtwiz_userdata_tx[15:0]	432f
gtwiz_userdata_rx[15:0]	4322
FREECLK	0
FREELOCKED	1
SAMPLECLK	1
SAMPLELOCKED	1

Figure 6.26: User clock verification using post-implementation functional analysis for a line rate of 1 Gbps

Verifying the line rate with the waveforms in figure 6.27, yields a word rate of 1 Gbps, translating to a line rate of 1.25 Gbps. Furthermore no abnormal increments on the user data channels are observed.



Name	Value
txn	0
txp	1
rxn	0
rxp	1
refclkp	1
refclkn	0
sysclkp	1
sysclkn	0
reset	0
link_status_out	0
link_down_latched_out	1
sysfreq	125000000
sysper	8000 ps
mgtreffreq	125000000
mgtrefper	8000 ps
gtwiz_usrclk_tx_usrclk2	1
gtwiz_usrclk_rx_usrclk2	1
gtwiz_userdata_tx[15:0]	4337
gtwiz_userdata_rx[15:0]	432a
FREECLK	1
FREELOCKED	1
SAMPLECLK	0
SAMPLELOCKED	1

Figure 6.27: Line rate verification using post-implementation functional analysis for a line rate of 1 Gbps

From the tests it can be confirmed that all 3 systems function as designed at their respective line rates.

6.5.2. Calibration Tests

The post-implementation functional simulation is considered to be extremely powerful because the results are validated to be exceptionally close to reality. Hence the simulation can provide a close estimate of the internal system delays. These values can be used to compare to the system calibration values obtained in section 6.6.

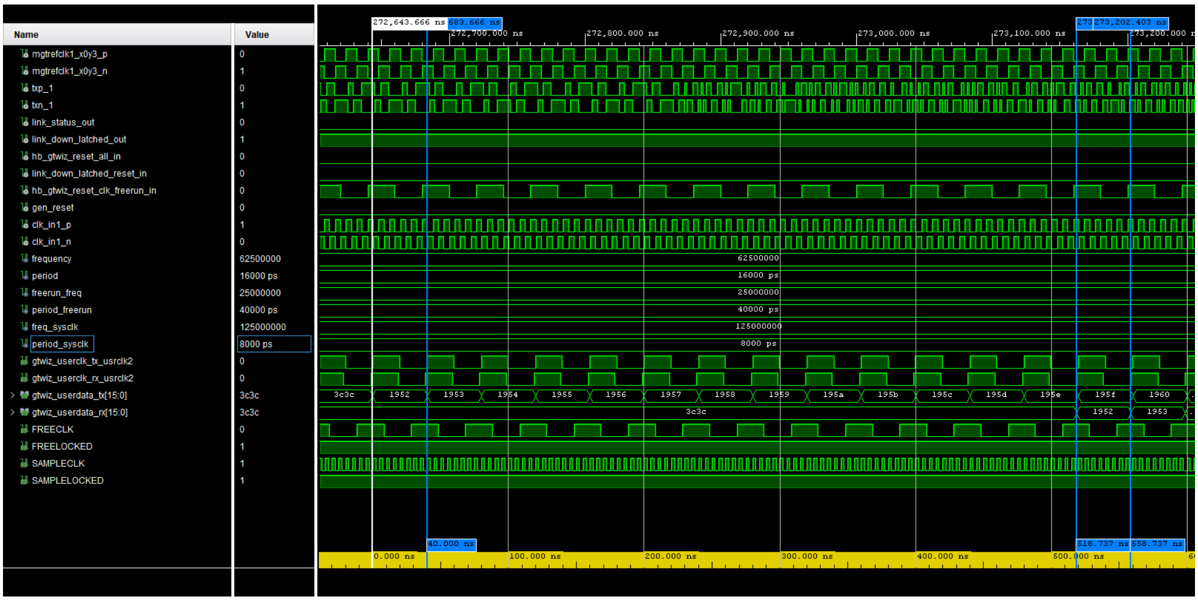
To measure the internal delay from simulations, the start of a transmitted data packet needs to be selected and a data packet with the same value needs to be found on the receiver data. This is done when the GTH is at the end of its boot-up sequence, the first data packet transmitted is the first data packet received. As explained in subsection 6.4.3, a set number of clock cycles will elapse before the internal logic is fully operational. Hence for different DRP frequencies, different values for the internal delay are expected.

6.5.2.1. Calibration Tests System 1

The results from the simulation of system 1 are illustrated in figure 6.28. The simulated internal delay is found to be 518.737 ns. Given a clock DRP frequency of 25 MHz, the absolute clock cycles required to bridge the internal FPGA logic is calculated according to Equation 6.11

$$N_{cycles} = \Delta t_{internal} \cdot f_{DRP} \tag{6.11}$$

For this simulation, the number of clock cycles that have elapsed is 12.97, rounding it to the nearest integer results in 13 clock cycles.

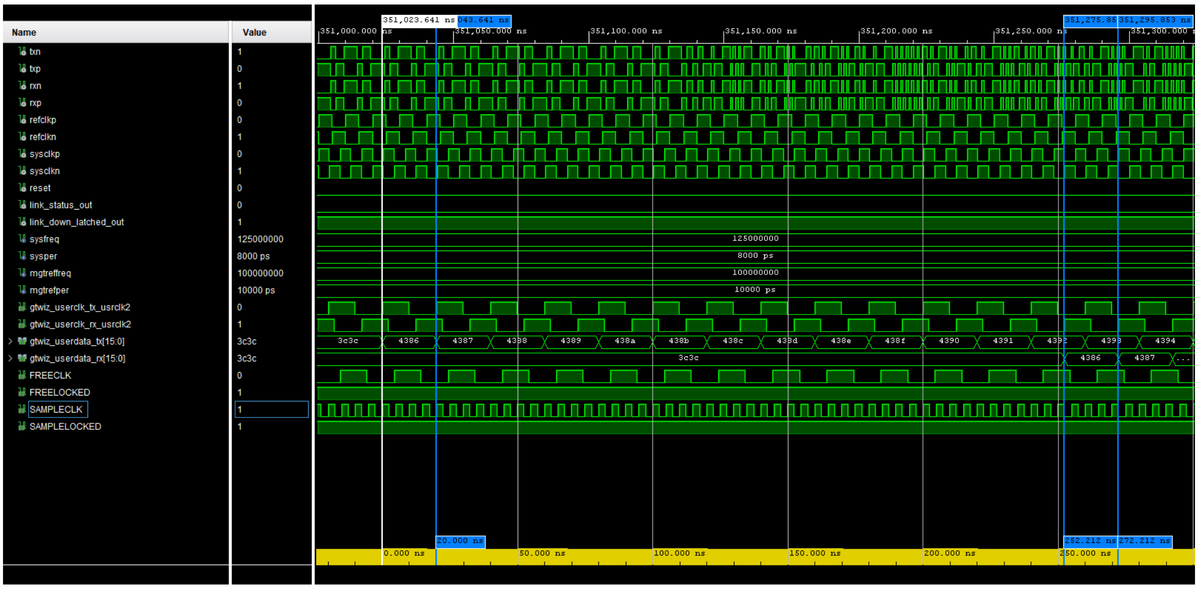


Name	Value
mgtrefclk1_x0y3_p	0
mgtrefclk1_x0y3_n	1
txp_1	0
txn_1	1
link_status_out	0
link_down_latched_out	1
hb_gtwiz_reset_all_in	0
link_down_latched_reset_in	0
hb_gtwiz_reset_clk_freerun_in	0
gen_reset	0
clk_in1_p	1
clk_in1_n	0
frequency	62500000
period	16000 ps
freerun_freq	25000000
period_freerun	40000 ps
freq_sysclk	125000000
period_sysclk	8000 ps
gtwiz_userclk_tx_usrclk2	0
gtwiz_userclk_rx_usrclk2	0
gtwiz_userdata_tx[15:0]	3c3c
gtwiz_userdata_rx[15:0]	3c3c
FREECLK	0
FREELOCKED	1
SAMPLECLK	1
SAMPLELOCKED	1

Figure 6.28: Simulation of the internal delay expected from a post-implementation functional simulation for a line rate of 500 Mbps

6.5.2.2. Calibration Tests System 2

The outcomes of the simulation for system 2 are depicted in figure 6.29. The simulated internal delay is determined to be 252.212 ns. Taking into account a clock DRP frequency of 50 MHz, we compute the exact clock cycles needed to traverse the internal FPGA logic using Equation 6.11, resulting in 12.61 clock cycles, which rounds up to 13 clock cycles when expressed as an integer value.

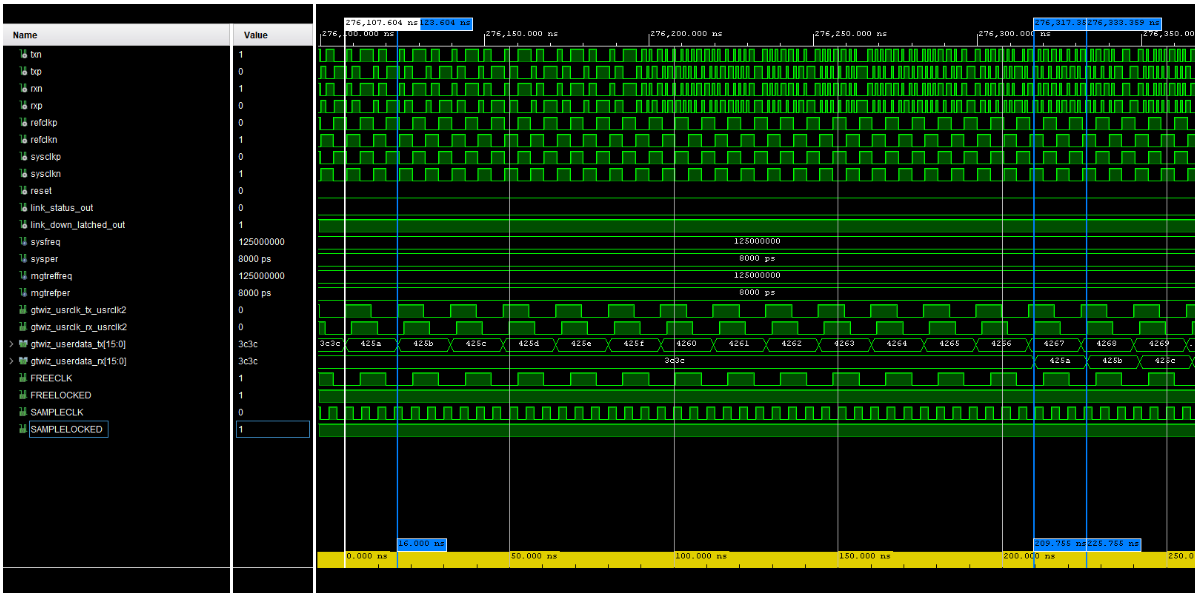


Name	Value
txn	1
txp	0
rxn	1
rxp	0
refclkp	0
refclkn	1
sysclkp	0
sysclkn	1
reset	0
link_status_out	0
link_down_latched_out	1
sysfreq	125000000
sysper	8000 ps
mgtreffreq	100000000
mgtrefper	10000 ps
gtwiz_userclk_tx_usrclk2	0
gtwiz_userclk_rx_usrclk2	1
gtwiz_userdata_tx[15:0]	3c3c
gtwiz_userdata_rx[15:0]	3c3c
FREECLK	0
FREELOCKED	1
SAMPLECLK	1
SAMPLELOCKED	1

Figure 6.29: Simulation of the internal delay expected from a post-implementation functional simulation for a line rate of 1 Gbps

6.5.2.3. Calibration Tests System 3

The results from the simulation of system 2 are illustrated in figure 6.30. The simulated internal delay is found to be 209.755 ns. Given a clock DRP frequency of 62.5 MHz, the absolute clock cycles required to bridge the internal FPGA logic is calculated according to Equation 6.11, resulting in 13.11 clock cycles, rounding off to the nearest integer value yields 13 clock cycles.



Name	Value
txn	1
txp	0
rxn	1
rxp	0
refclkp	0
refclkn	1
sysclkp	0
sysclkn	1
reset	0
link_status_out	0
link_down_latched_out	1
sysfreq	125000000
sysper	8000 ps
mgtreffreq	125000000
mgtrefper	8000 ps
gtwiz_usrclk_tx_usrclk2	0
gtwiz_usrclk_rx_usrclk2	0
gtwiz_userdata_tx[15:0]	3c3c
gtwiz_userdata_rx[15:0]	3c3c
FREECLK	1
FREELOCKED	1
SAMPLECLK	0
SAMPLELOCKED	1

Figure 6.30: Simulation of the internal delay expected from a post-implementation functional simulation for a line rate of 1.25 Gbps

The anticipated number of clock cycles necessary to traverse the internal logic is approximately 13 clock cycles. The simulation supports the hypothesis that the internal delay depends on the DRP frequency. This suggests that, for each chosen line rate, re-calibration is necessary, aligning with the expected internal delay as described in Equation 6.12.

$$\Delta t_{internal} \approx \frac{13}{f_{DRP}} \tag{6.12}$$

To summarize the internal delay found through verification is presented in table 6.8.

Table 6.8: Internal delay corresponding to the respective line rate and symbol rate of the system found through post-implementation functional simulation.

$\Delta t_{\text{internal}}$(ns)	Line rate (Gbps)	Symbol rate (MHz)
518.737	0.5	25
252.212	1	50
209.755	1.25	62.5

6.6. Validation

In this section the system validation is presented. First the general system validation is presented in subsection 6.6.1, aiming to validate general system functioning. Part of the validation is to perform system calibration in order to prepare for the results in section 6.7 and compare these results to the ones found in section 6.5. The validation on the internal delay calibration is presented in subsection 6.6.2.

6.6.1. System Validation

After the post-implementation functional simulation, it is expected that the system will function as intended. Nonetheless, unwanted behavior may occur after FPGA programming. To validate the system, it is run with the components included with the FPGA. The firmware is programmed and a time period is observed.

Through these observations it was found that the system operates as intended. However there are some exceptions. In some cases the received channel first 8 Most Significant Bit (MSB) become the last 8

Least Significant Bit (LSB). Whenever this occurs the RCID can be reconstructed in post-processing by replacing the first 8-bits with the last 8-bits and vice-versa.

This happens due to a system bug in which the transmitter ready signal is low for a short period of time, causing the GTH to shut down and restart. In this process the bits are switched around as seen in figure 6.31. As of writing the report the exact reason why this switching occurs is unknown. For a normal data stream, depending on the system at the other end of the link, the ready signal should remain high unless devices such as a FIFO are overfull and respond with a low ready signal. In this system there is no storage or further data processing, it is simply changing wire values.

There may be some misconfiguration within the Ultrascale GT Wizard that causes this ready signal to be periodically low, for example that the PISO is handling too little data as the rate at which the data enters is less than the rate at which it is transmitted, causing periodic resets. This is however speculative and not confirmed.

Due to the time constraints imposed on the thesis and the possibility of fully recovering these packets in post-processing, it was deemed to function as intended. In the future, however, it is recommended to further investigate this behavior.

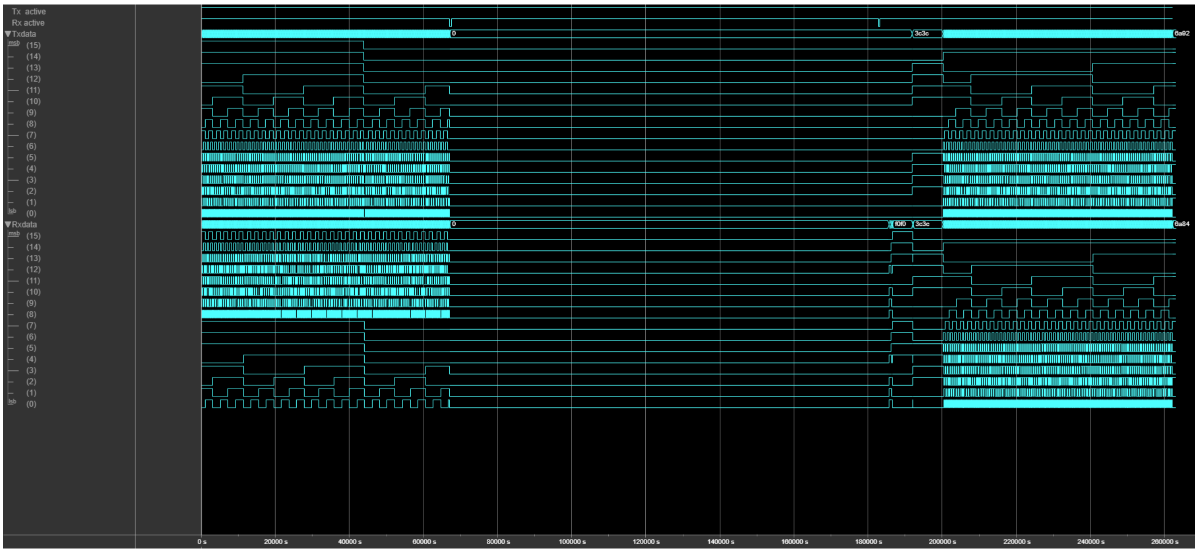


Figure 6.31: Clock phase drift captured by the data capture. The receiver is reset causing for an inversion in the RCID counter. This may happen both ways.

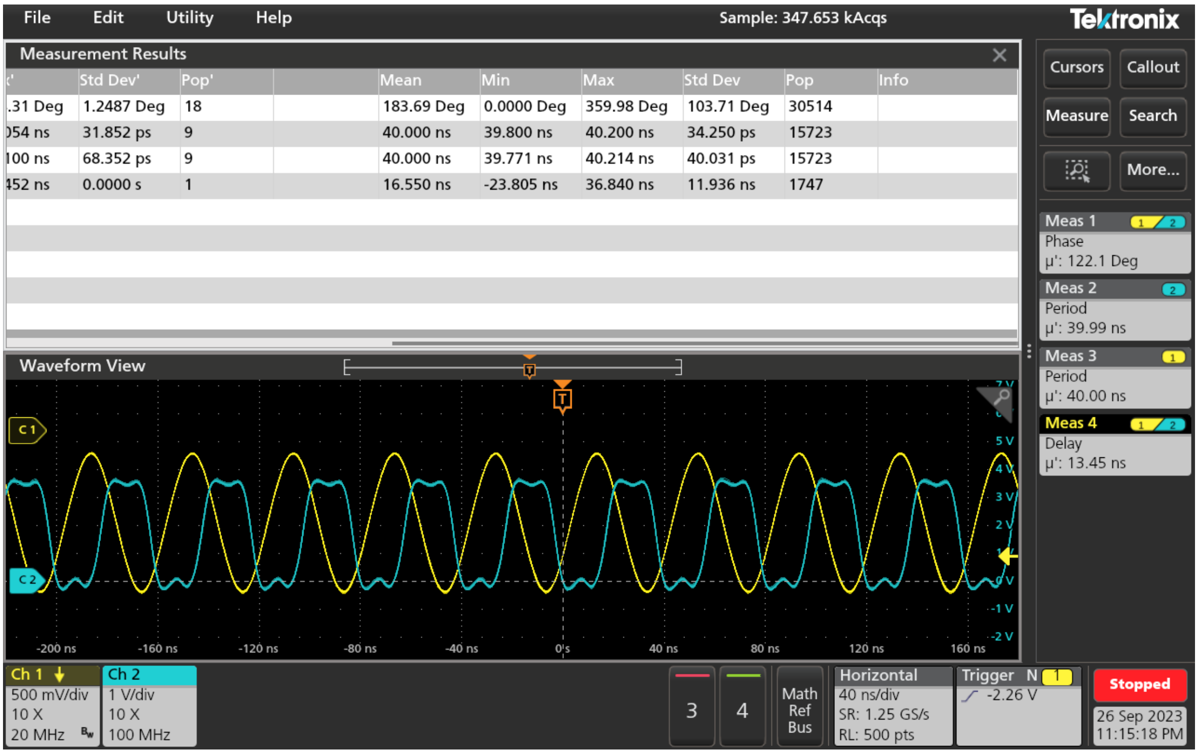
Whenever the system is started, it locks the user clocks to a certain phase with respect to each other through the internal comma and phase alignment firmware. As far as it can be observed through the data capture firmware, this phase is constant for the duration of a transmission. This phase difference is naturally discretized by the sampling frequency of the data capture firmware. From this, it can be concluded that for a given DRP frequency f_{DRP} and a sampling frequency f_s, the possible observable phase differences can be described by the relation described in Equation 6.13.

$$N_{phase} = \frac{f_s}{f_{DRP}} \tag{6.13}$$

Hence every reset, the phase may be locked to f_s/f_{DRP} positions. Because the system periodically resets, the shift in phase causes a jitter in internal delay injecting noise into the system that can be characterized by these shifts in phase.

The frequency of these resets are impossible to measure as the way the system is currently configured does not allow for a continuous stream of data. Only a limited window size can be captured with a period of time that cannot be observed in between windows. From the data capture of the results however, no reset sequences are captured. These sequences can only be captured when setting a trigger on the recognisable properties such as a low ready signal.

This can be confirmed by pulling both TX and RXUSRCLK2 out to a PMOD pin. Both clocks are single-ended clocks and cannot be output to a SMA connector pin. A recording is started using a Tektronix Series 2 Oscilloscope and the results are presented in figure 6.32.



Measurement Results

'	Std Dev'	Pop'		Mean	Min	Max	Std Dev	Pop	Info
.31 Deg	1.2487 Deg	18		183.69 Deg	0.0000 Deg	359.98 Deg	103.71 Deg	30514	
)54 ns	31.852 ps	9		40.000 ns	39.800 ns	40.200 ns	34.250 ps	15723	
100 ns	68.352 ps	9		40.000 ns	39.771 ns	40.214 ns	40.031 ps	15723	
452 ns	0.0000 s	1		16.550 ns	-23.805 ns	36.840 ns	11.936 ns	1747	

Meas 1
Phase
µ': 122.1 Deg

Meas 2
Period
µ': 39.99 ns

Meas 3
Period
µ': 40.00 ns

Meas 4
Delay
µ': 13.45 ns

Ch 1
500 mV/div
10 X
20 MHz

Ch 2
1 V/div
10 X
100 MHz

Horizontal
40 ns/div
SR: 1.25 GS/s
RL: 500 pts

Trigger N
-2.26 V

Stopped

26 Sep 2023
11:15:18 PM

Figure 6.32: Capture of oscilloscope phase measurment between TXUSRCLK2 and RXUSRCLK2

The top row of the table in figure 6.32 represents the phase difference between both clocks measured for 30514 acquisitions. From the values it can be seen that the minimum and maximum phase difference between the clocks is 0 and 360 degrees respectively with a mean at almost exactly the center at 180 degrees and a standard deviation of 104 degrees, meaning that the phase of these clocks cycle continuously causing the clock phase drift noise.

6.6.2. System Calibration

The system model described in subsection 6.3.1, there are multiple aspects of the system that may contribute to the latency of the overall system. The complexity of the GTH transceiver firmware and hardware however, only allows for limited analysis on the contribution of these components. The parts within the grey blocks illustrated in figure 6.3 can currently only be mapped to a certain extent.

Nonetheless, this validation step aims to map parts of this latency with the use of a set of hardware devices as described in subsection 6.4.3. Two calibration tests are done with the bridged SFP adapter and the optic fiber loopback adapter to, first of all, validate the internal latency found through verification in section 6.5 and secondly identify the observable noise on the signal itself.

When inserting the bridged SFP, the system model boils down to the block diagram illustrated in figure 6.33. The SFP drivers are in essence removed from the link and only the noise and latency introduced from signal generation through the GTH channel and back is measured. Noise and latency components related to the embedded laser transmitter and receiver of an SFP module are effectively ignored.

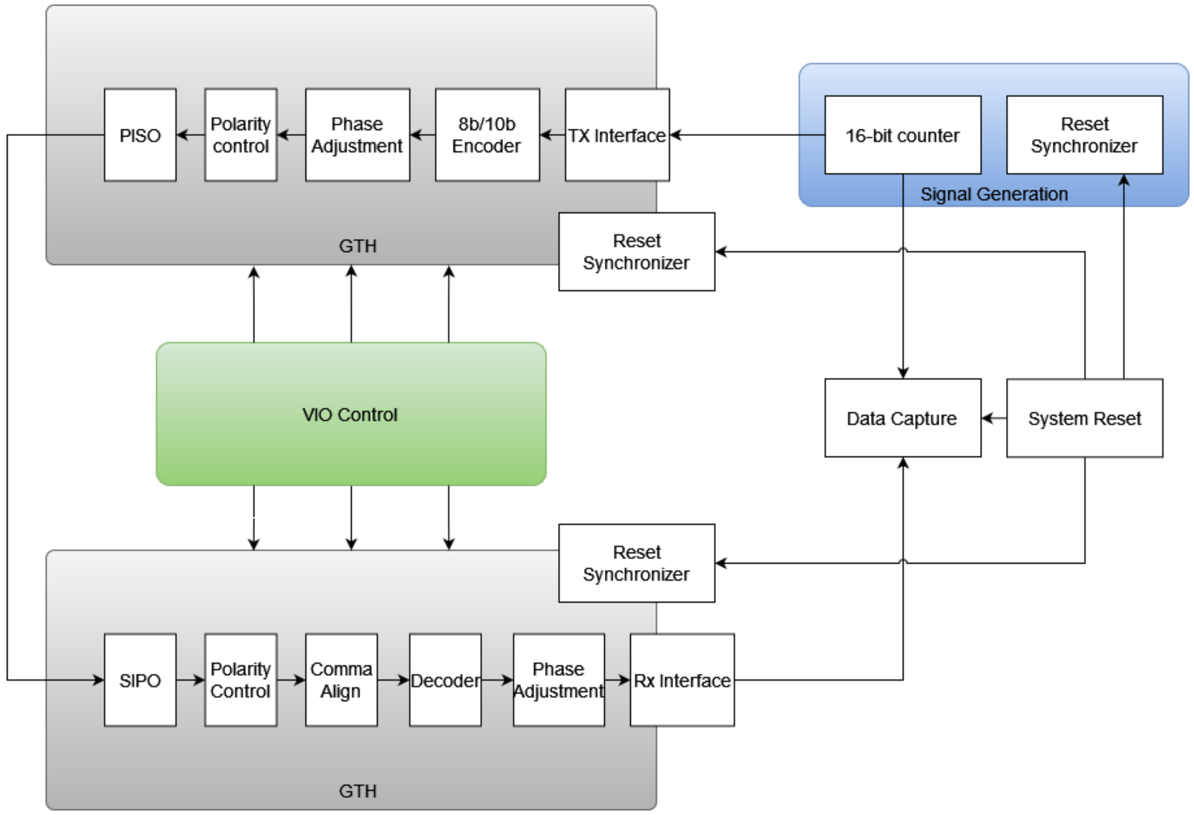


Figure 6.33: Block diagram of the system model with a bridged SFP adapter

Noise and latency are theoretically added when the adapter is replaced with a regular module and the fiber optic loopback adapter. The same system model as illustrated in figure 6.3 is true, however the path travelled by the photons is at centimeter level, which is much lower than the 1.02 m resolution of the system. It is therefore assumed that the transmitted signal can be considered to be instantaneously received at the receiver.

6.6.2.1. Test 1: 500 Mbps

Starting with a line rate of 500 Mbps, 500 windows of data, totalling 131 million samples are acquired and the time delay is computed according to the algorithm described in section 6.4. The results for the bridged adapter and loopback adapter are shown in the graphs in figure 6.34 and figure 6.35 respectively.

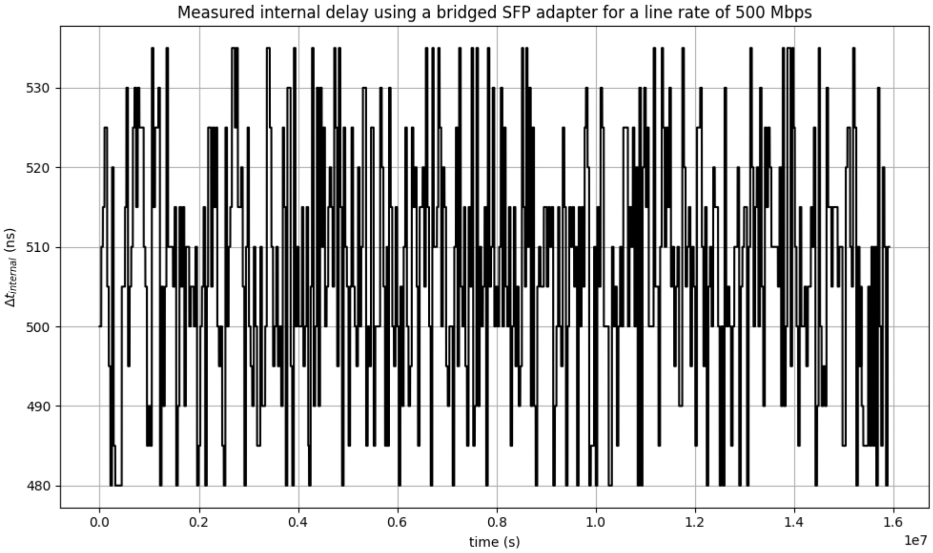


Figure 6.34: Internal delay measurement using a bridged SFP adapter with a line rate of 500 Mbps

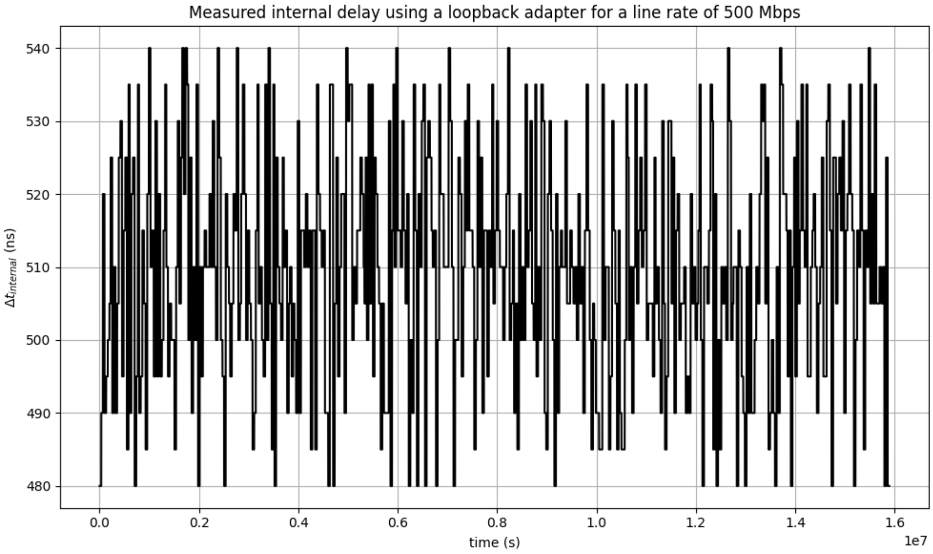


Figure 6.35: Internal delay measurement using a optic fiber loopback adapter with a line rate of 500 Mbps

The phase noise due to the phase shift of the `TXUSRCLK2` and `RXUSRCLK2` is evident, as a more or less flat line in these figures is the ideal measurement. Looking at the minimum and maximum of figure 6.34, the minimum if 480 ns and the maximum is 535 ns, thus the internal delay has an amplitude of 55 ns. Doing the same for figure 6.35, it can be observed that the minimum, maximum and resulting amplitude are 480 ns, 540 ns and 60 ns respectively.

The internal delay measurement can be represented by two Gaussian distributions given in Equation 6.14 and Equation 6.15 respectively. μ is given in nanoseconds and σ^2 is given in nanosecond squared. The difference in the two distributions is minimal. The time delay measurements closely coincide, the lower bound of both the bridged SFP and the loopback measurements are identical, however for the loopback the upper bound appears to sometimes exceed by 1 sampling clock cycle. This can be attributed to the change added delay of the transmitter and receiver driver within an SFP module, though it only occurs when the phase drift of the two user clocks is driven to their extreme. Therefore it is likely a relatively small value compared to the phase drift noise, only shifting the average internal delay up slightly as well as introduces roughly 300 ps more standard deviation.

$$N_{SFP}(\mu, \sigma^2) \sim N_{SFP}(506.998, 15.435^2) \tag{6.14}$$

$$N_{loopback}(\mu, \sigma^2) \sim N_{loopback}(509.316, 15.782^2) \tag{6.15}$$

Both of these values are close to the value of 518.737 ns found during verification, and thus the simulation can be considered valid. Furthermore for the tests conducted with this line rate, the phase drift noise component of the loopback adapter is taken, as this is a more representative value for calibration when doing tests with the SFP and optic fiber. It may be concluded that the noise from the SFP is overwhelmed but is definitely embedded in the internal delay measurements.

6.6.2.2. Test 2: 1Gbps

Similar to the previous calibration test, 500 windows of data are captured of a test with a line rate of 1 Gbps. It is expected that the uncertainty is lower as the DRP frequency is closer to the sampling frequency, reducing the number of discrete phase drift possibilities.

The results of these tests are presented in figure 6.36 and figure 6.37 respectively.

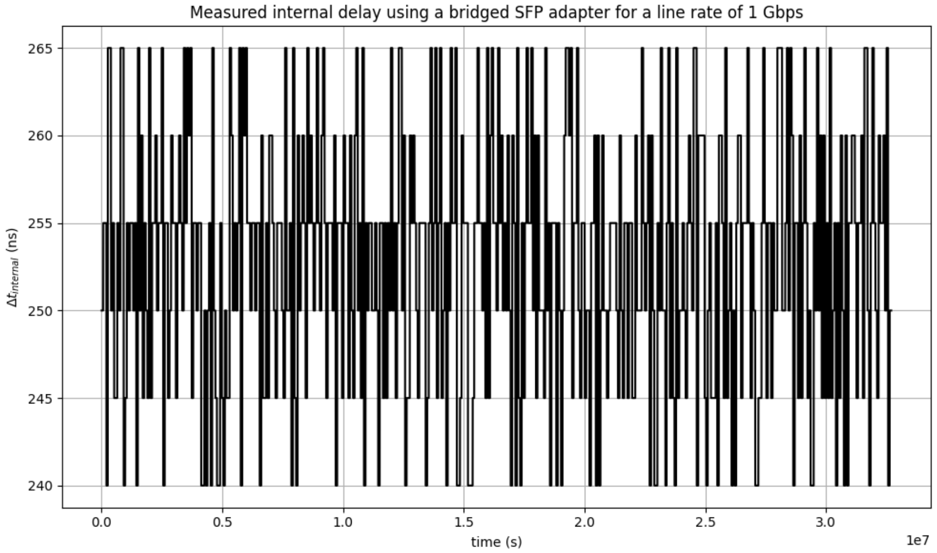


Figure 6.36: Internal delay measurement using a bridged SFP adapter with a line rate of 1 Gbps

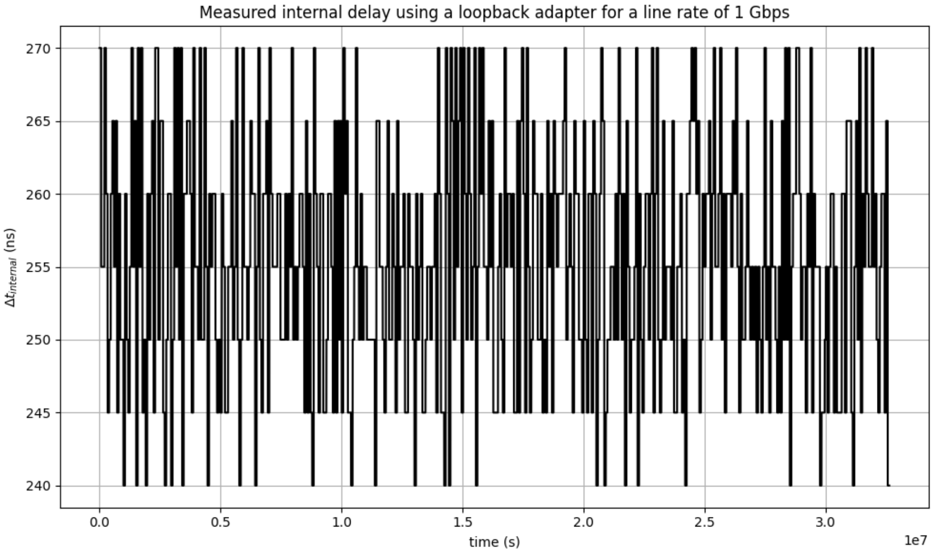


Figure 6.37: Internal delay measurement using a optic fiber loopback adapter with a line rate of 1 Gbps

The minimum, maximum and amplitude of figure 6.36 are 240 ns, 265 ns and 25 ns. Similarly, from figure 6.37 these properties are 240 ns, 270 ns and 30 ns respectively. As anticipated, bot the internal latency and standard deviation in the measurement is roughly half that of the measurement with 500 Mbps. This is due to the fewer discrete possibilities the clock drift is able to be measured. These results are in line with the results from the previous test, with the loopback adapter the upper bound is one clock cycle larger than for the bridged SFP while the lower bound remains constant. The Gaussian distributions describing the measurements are presented in Equation 6.16 and Equation 6.17 respectively.

$$N_{SFP}(\mu, \sigma^2) \sim N_{SFP}(252.719, 7.421^2) \tag{6.16}$$

$$N_{loopback}(\mu, \sigma^2) \sim N_{loopback}(255.418, 8.433^2) \tag{6.17}$$

The validation values are close to the value found during verification of 252.212 ns. This behaviour is expected as the phase drift discretization possibilities decrease with an increase of DRP frequency.

Moreover in line with the previous tests, the loopback adapter adds a little noise to the system causing the reading to be one clock cycle higher than for the bridged SFP adapter in some high latency cases. Nonetheless, this value is most representative to use to quantify the internal delay.

6.6.2.3. Test 2: 1.25 Gbps

Repeating the test with the same parameters as before, 500 windows of data are captured of a test with a line rate of 1.25 Gbps. Again, as the DRP frequency is increased, it is expected that the uncertainty is again lower than measured in the before tests.

figure 6.38 shows the internal delay measurements of using only a bridged SFP adapter and figure 6.39 illustrates the internal delay measurement using a loopback adapter.

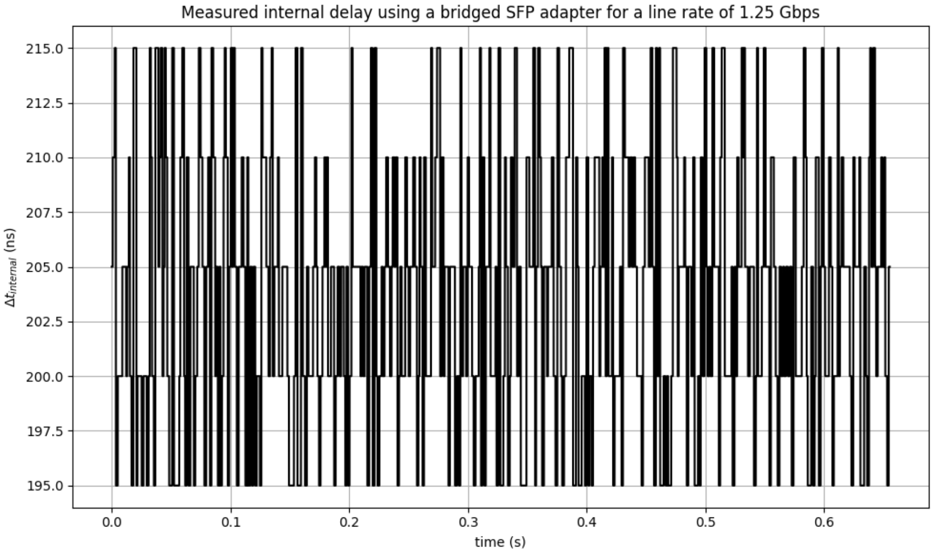


Figure 6.38: Internal delay measurement using a bridged SFP adapter with a line rate of 1.25 Gbps

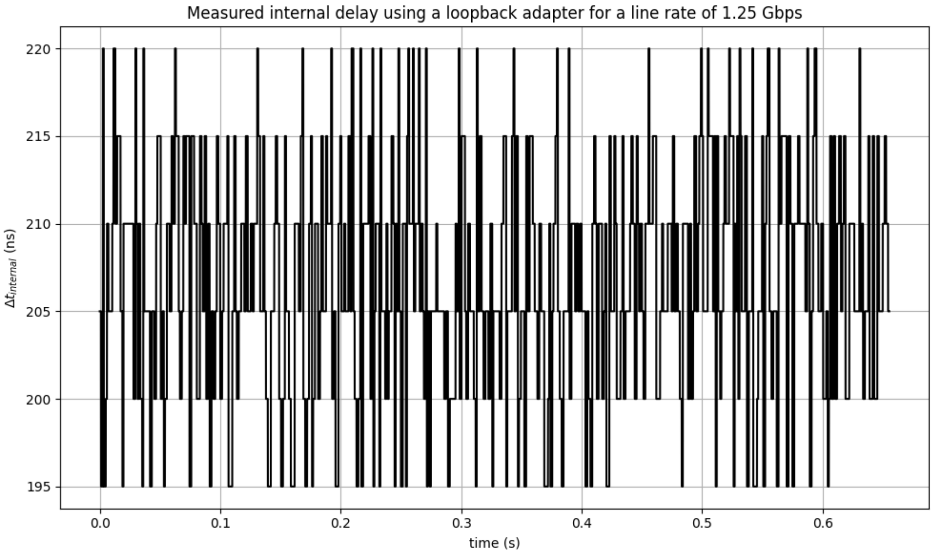


Figure 6.39: Internal delay measurement using a optic fiber loopback adapter with a line rate of 1.25 Gbps

From figure 6.38 the minimum, maximum and amplitude are 195 ns, 215 ns and 20 ns respectively. The minimum, maximum and amplitude from figure 6.39 are 195 ns, 220 ns and 25 ns respectively.

The delays can be expressed with a Gaussian distribution as presented in Equation 6.18 and Equation 6.19 for the bridged SFP adapter and loopback adapter respectively. Even though the amplitude of the amplitude is equal to the figures from the 1 Gbps, however the standard deviation has decreased as anticipated with an increase in line rate.

$$N_{SFP}(\mu, \sigma^2) \sim N_{SFP}(203.62, 6.277^2) \tag{6.18}$$

$$N_{loopback}(\mu, \sigma^2) \sim N_{loopback}(206.525, 6.678^2) \tag{6.19}$$

Comparing this to the simulation, they are relatively close to one and other, validating the results obtained in the simulation. The results are in line with what was obtained during the previous tests and similarly, the measurement value from the loopback adapter will be used for internal delay calibration.

6.7. Results

For each line rate a test is run. With this test and the obtained calibration values from the validation ranging can be performed with the system. Each test is run with 5 different single mode optic fiber cable lengths, ranging from 10 m to 50 m with 10 meter increments. The cable is composed from pieces with a length of 10 m each, coupled by a coupling piece.

From the validation measurements in section 6.6, it is expected that the range measurement will exhibit at least the same level of uncertainty due to the clock drifting issue. Furthermore, the increase in the length of the medium is guaranteed to add noise, potentially compounding to the standard deviation of the measurement.

6.7.1. Test 1: 500 Mbps

The FPGA is programmed to transmit the RCID sequence with a line rate of 500 Mbps. A summary of the relevant system properties is summarized in table 6.5. For each cable length a total of 131 million samples are taken. Based on the methodology explained in section 6.4 the range is computed. The results from these tests are presented in graphs and displayed in figure 6.40 until and including figure 6.44.

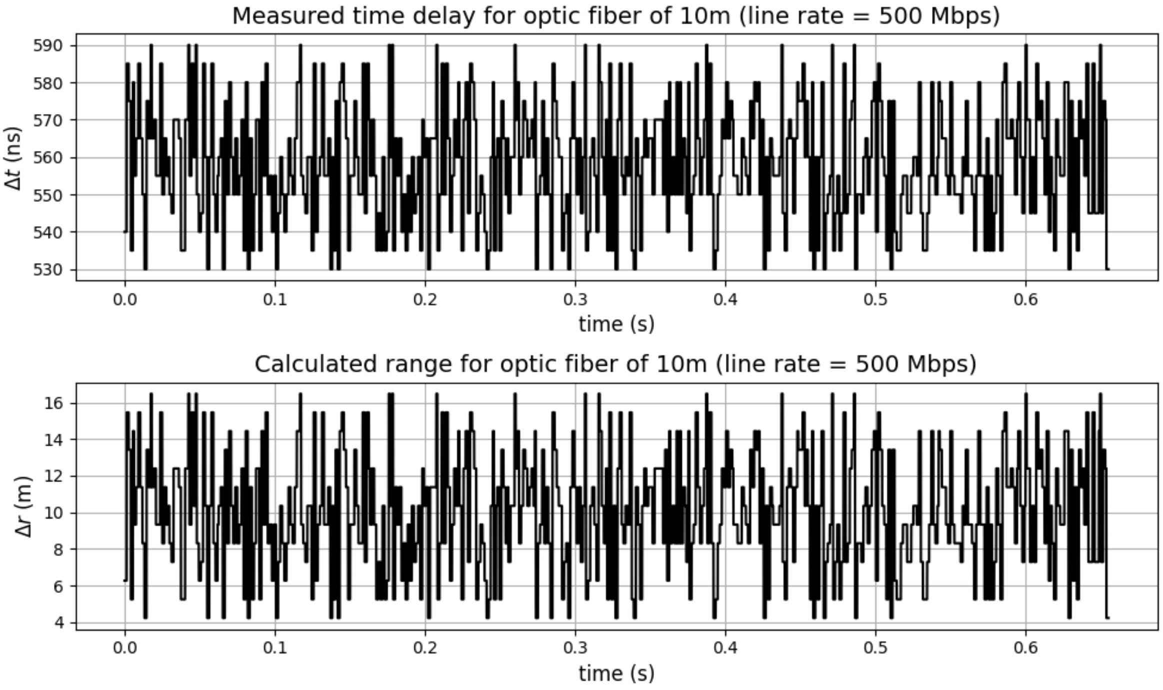


Figure 6.40: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 10 m at a line rate of 500 Mbps

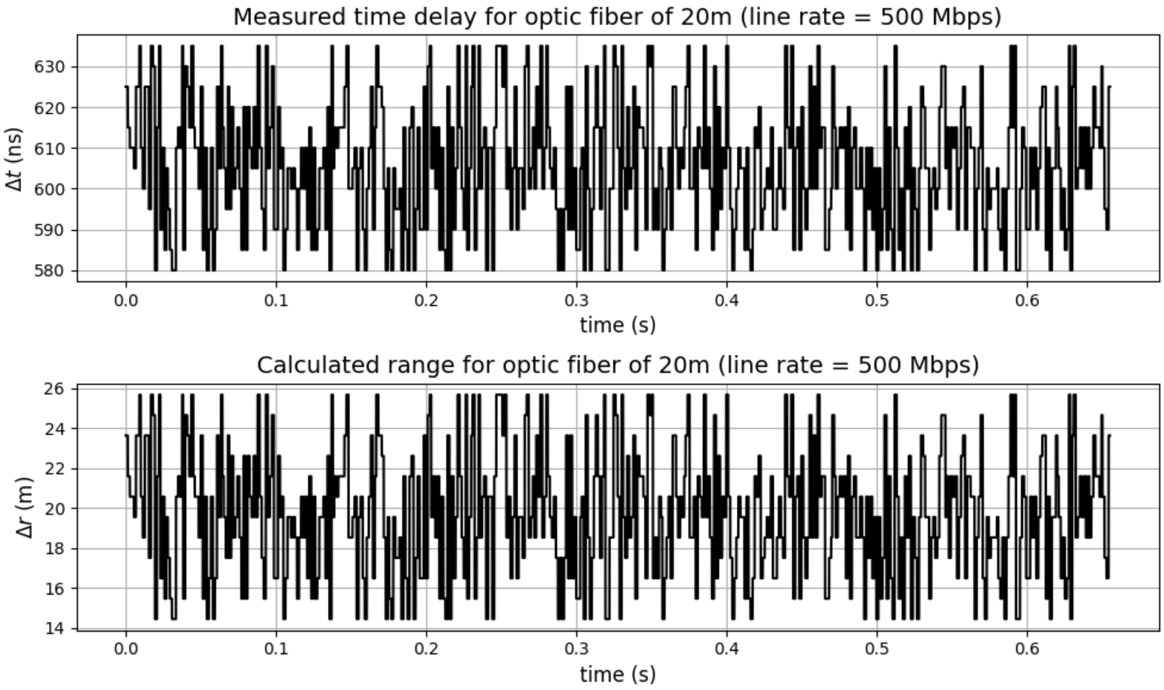


Figure 6.41: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 20 m at a line rate of 500 Mbps

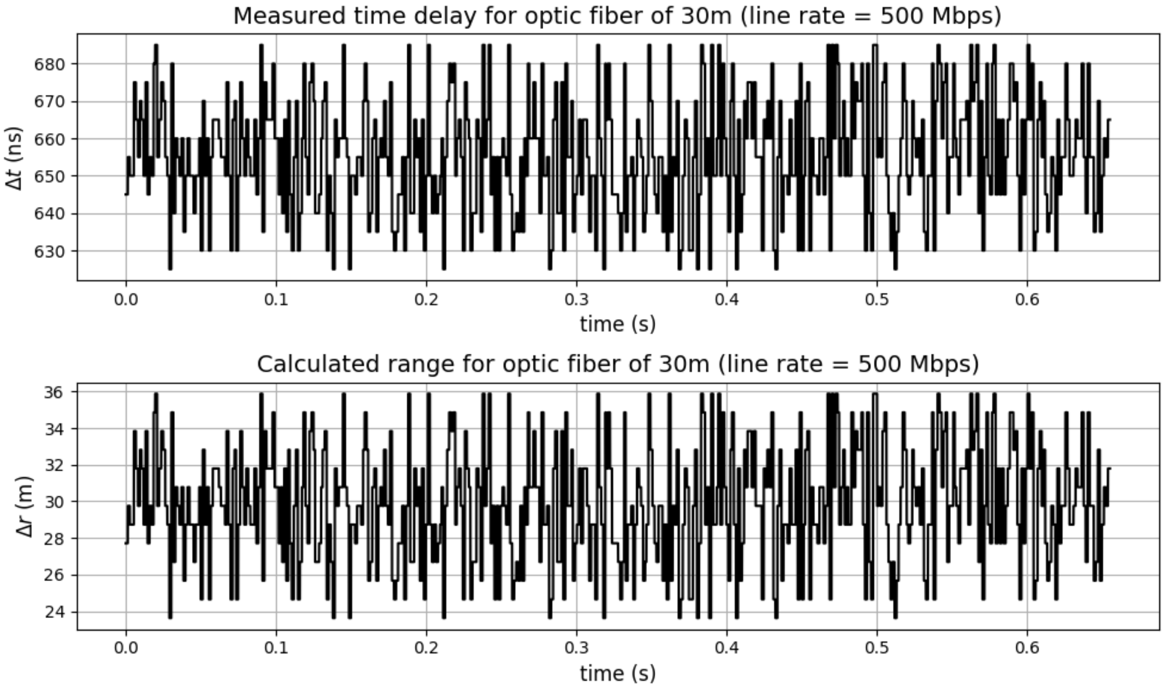


Figure 6.42: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 30 m at a line rate of 500 Mbps

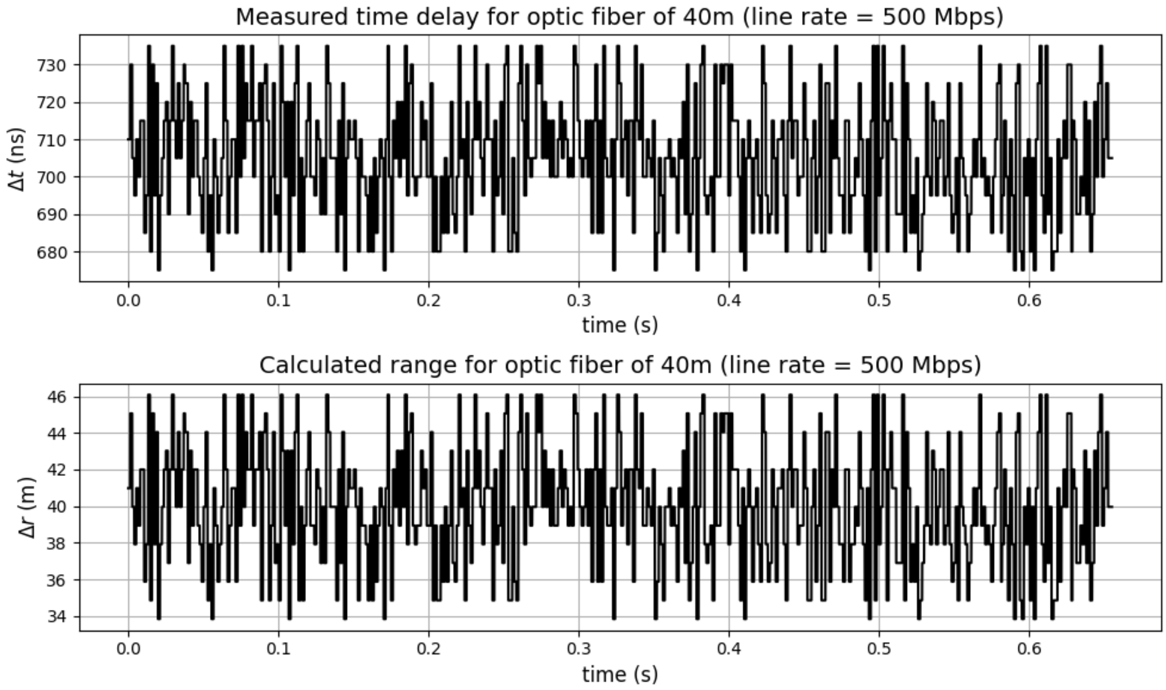


Figure 6.43: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 40 m at a line rate of 500 Mbps

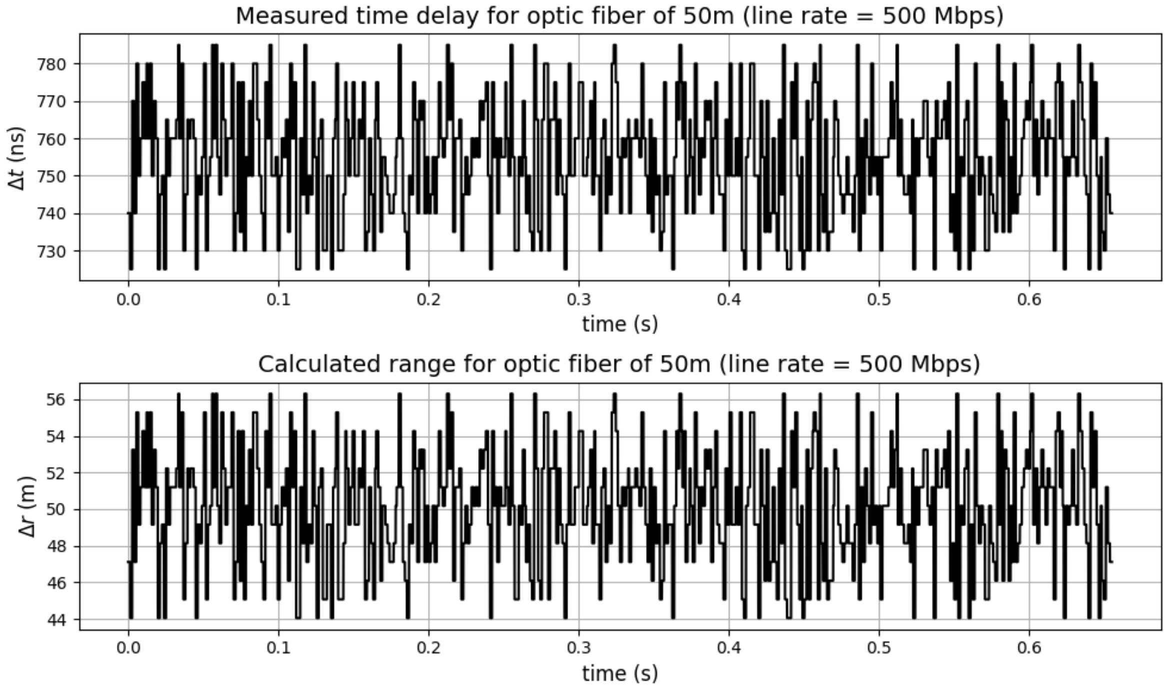


Figure 6.44: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 50 m at a line rate of 500 Mbps

From the figures the same noise level attributable to the clock phase drift is present for each measurements, with the same magnitude of 60 ns as before which was expected. To calibrate for the internal delay the average value from the calibration measurements with the corresponding line rate was taken (509.316 ns).

Using Equation 6.8, the Δt can be computed and transformed to a range measurement with Equation 6.9, correcting for the refractive index of the fiber according to section 6.4.

Essentially, the measurement at a certain distance is comparable to the calibration measurement offset with additional time delay. As the underlying noise behaviour is nearly identical in addition to a small component added by the uncertainty of the optic fiber cable itself, which is largely overwhelmed by the clock drift.

From the figures figure 6.40 until and including figure 6.44 can be seen that the measurement is centered around the target range, making the averages an exceptional representation of the measured range. Based on the data, the Gaussian distribution properties are calculated and summarized in table 6.9. Naturally, the standard deviation expressed in meters depends on the refractive index of the medium, in this case being optic fiber with $\bar{n} = 1.468$.

$\mathbf{r_{target}}(\mathbf{m})$	$\mu(\mathbf{ns})$	$\sigma(\mathbf{ns})$	$\mu(\mathbf{m})$	$\sigma_{\mathbf{fiber}}(\mathbf{m})$	$\varepsilon(\%)$
10	557.919	15.804	9.927	3.228	0.7433
20	605.547	15.666	19.652	3.199	1.739
30	655.468	15.884	29.847	3.243	0.511
40	705.03	16.311	39.968	3.331	0.079
50	754.463	16.003	50.064	3.268	0.127
Average		15.841		3.235	0.64

Table 6.9: Results range measurement for a line rate of 500 Mbps

Comparing the standard deviations to each other as well as to the standard deviations found in section 6.6, the values are relatively constant, further enforcing that the measurement is constructed of a constant deterministic part that represents the average of a Gaussian distribution, and a stochastic parts representing the jitter causing the standard deviation.

Moreover, there appears to be no clear correlation between the relative error and the measured distance, as it does not increase or decrease with respect to the increase in distance. The standard deviation however is relatively constant.

6.7.2. Test 2: 1 Gbps

The same test is repeated with five different cable lengths at a line rate of 1 Gbps. Based on the methodology explained in section 6.4 the range is computed. The results from these tests are presented in graphs and displayed in figure 6.45 until and including figure 6.49.

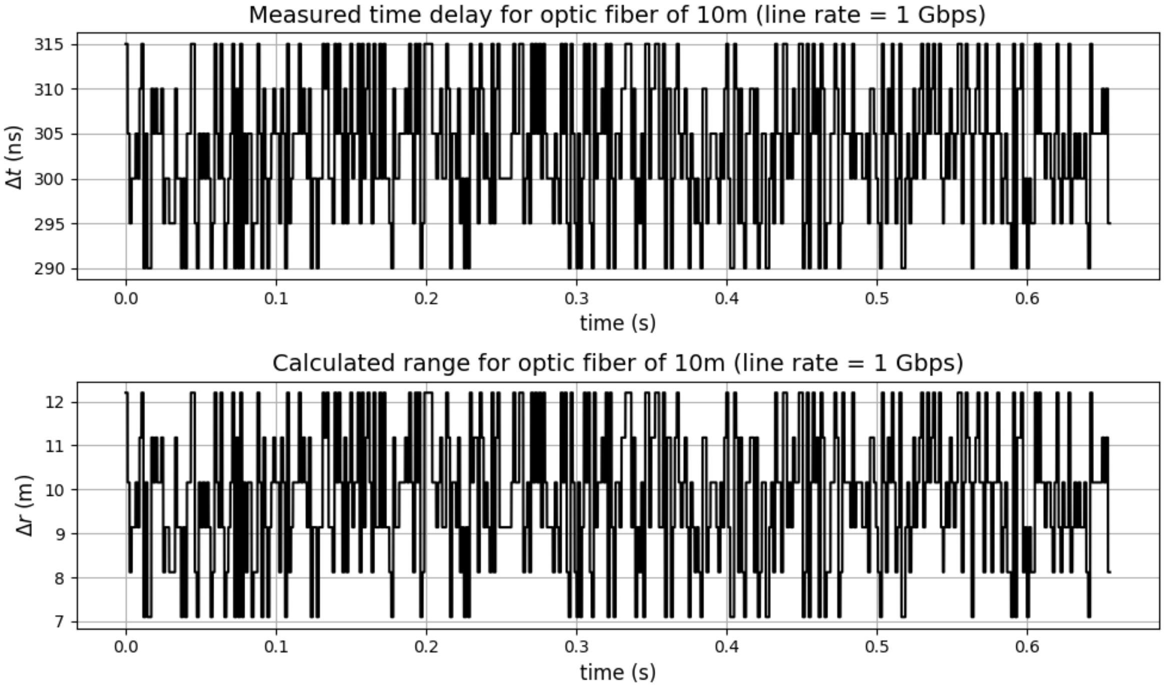


Figure 6.45: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 10 m at a line rate of 1 Gbps

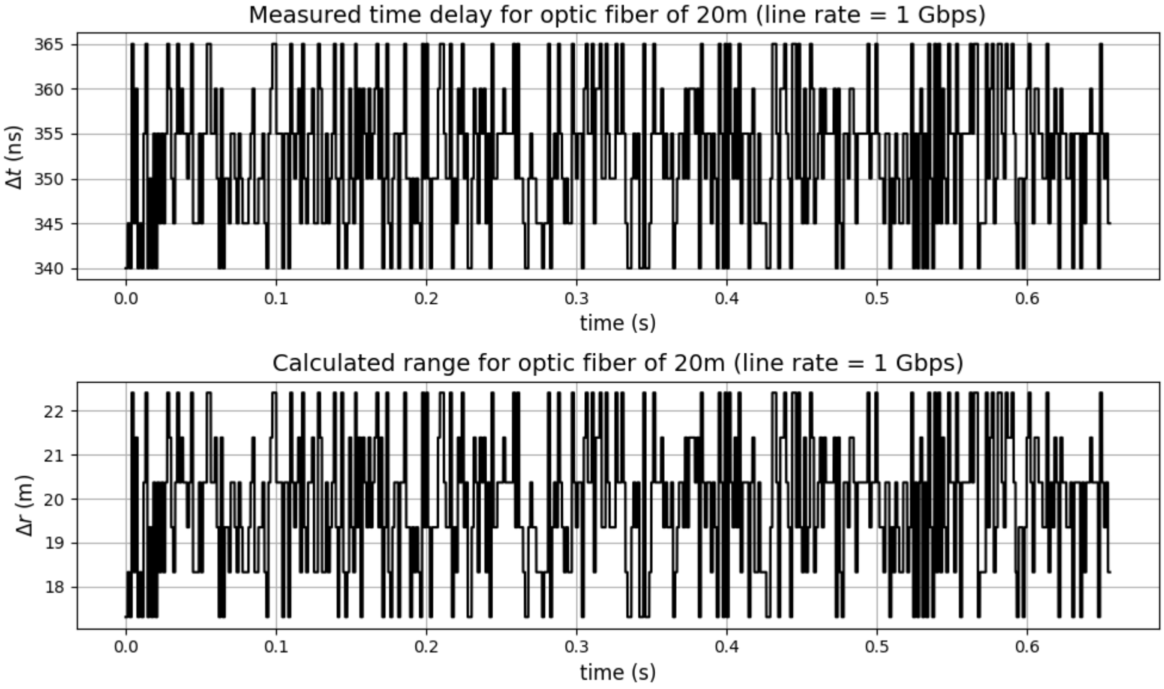


Figure 6.46: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 20 m at a line rate of 1 Gbps

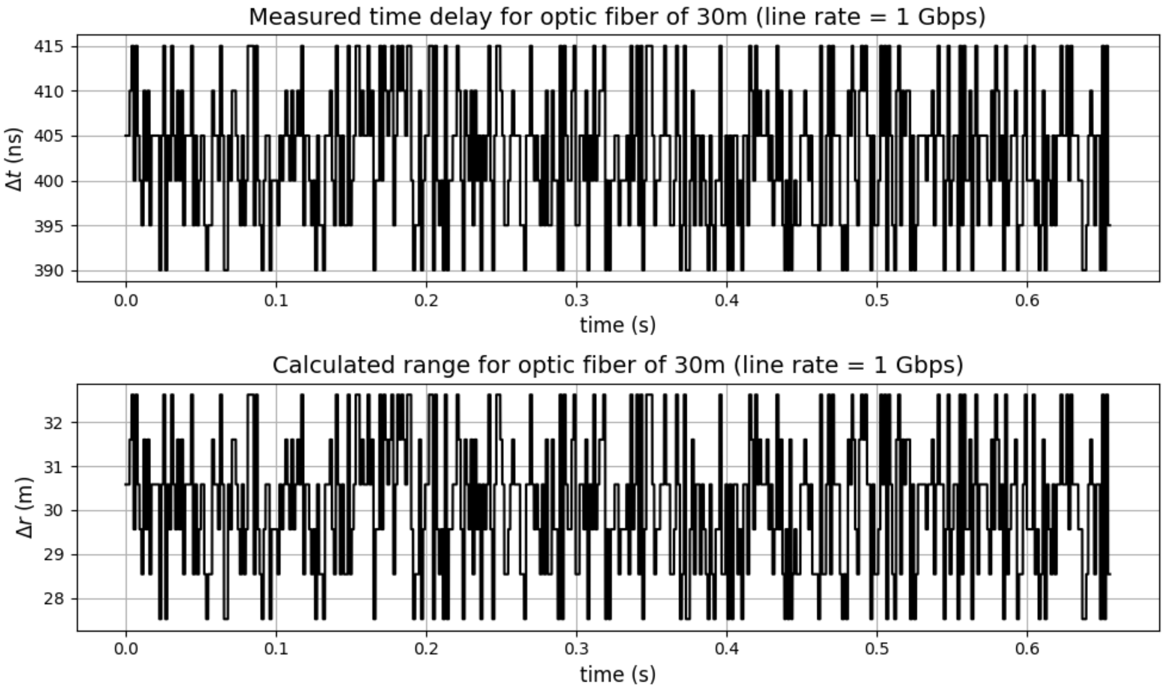


Figure 6.47: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 30 m at a line rate of 1 Gbps

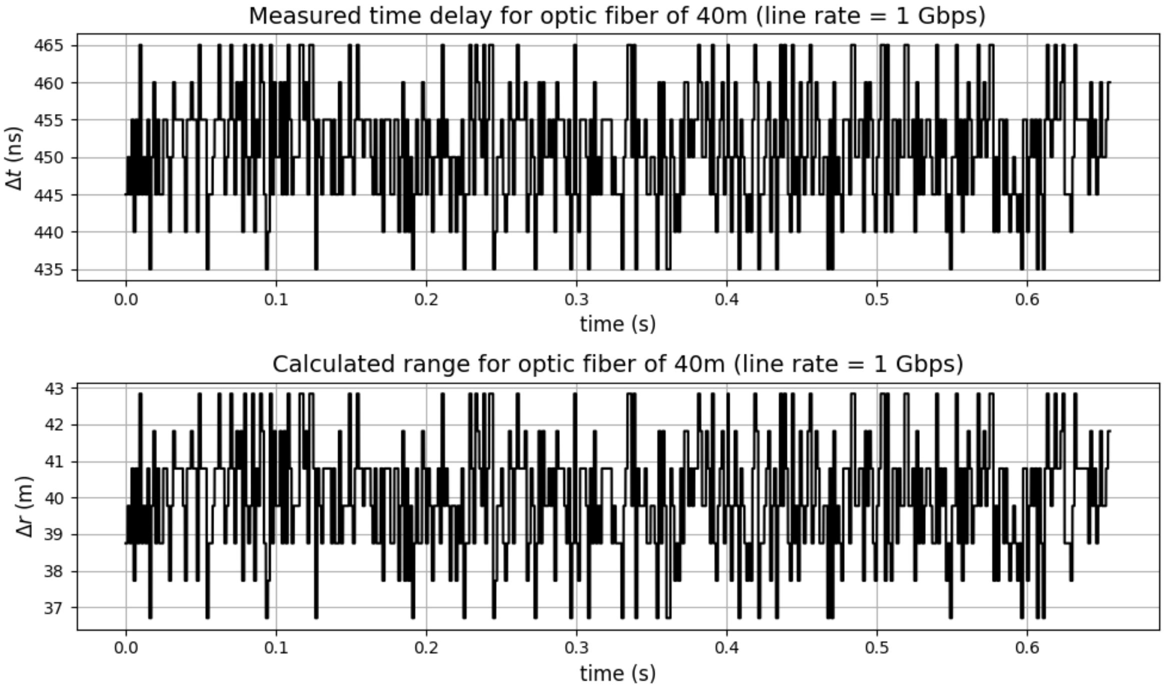


Figure 6.48: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 40 m at a line rate of 1 Gbps

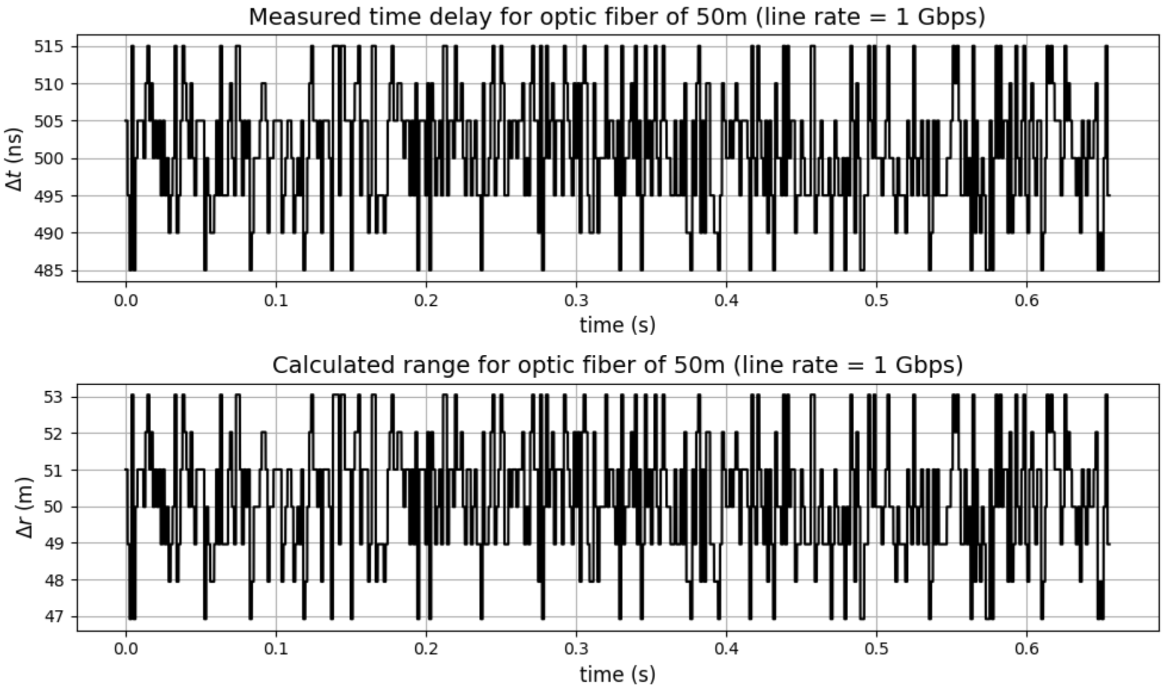


Figure 6.49: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 50 m at a line rate of 1 Gbps

With the previous tests the magnitude of the oscillation coincided with the magnitude of the loopback adapter. For this line rate, this behavior is only observed for cable lengths of 40 m and larger. This has minimal influence on the noise of the data, which is presented in table 6.10.

$\mathbf{r_{target}(m)}$	μ(**ns**)	σ(**ns**)	μ(**m**)	$\sigma_{\mathbf{fiber}}$(**m**)	ε(%)
10	303.508	7.791	8.758	1.591	12.418
20	352.687	7.775	18.802	1.588	5.992
30	402.440	7.759	28.962	1.585	3.460
40	451.158	7.807	38.911	1.594	2.722
50	501.124	8.116	50.178	1.657	0.355
Average		7.924		1.618	4.989

Table 6.10: Results range measurement for a line rate of 1 Gbps

The standard deviation has halved with respect to the previous test as expected due to the discretization of the clock phase drift. The closer the DRP frequency approaches the sampling frequency, the lower the standard deviation becomes.

Furthermore, the standard deviation remains relatively equal for the different cable lengths, the relative error however decreases with respect to the distance, contrary to the previous test where there was no clear correlation.

6.7.3. Test 3: 1.25 Gbps

The same test is repeated with five different cable lengths at a line rate of 1.25 Gbps. Based on the methodology explained in section 6.4 the range is computed. The results from these tests are presented in graphs and displayed in figure 6.50 until and including figure 6.54.

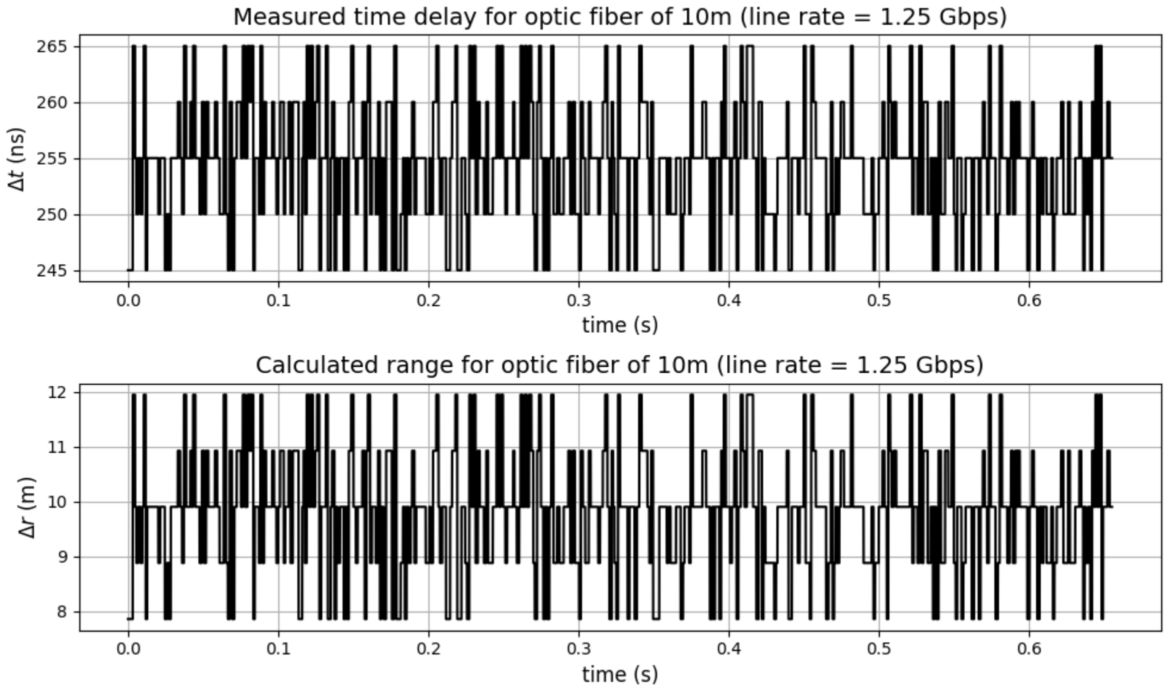


Figure 6.50: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 10 m at a line rate of 1.25 Gbps

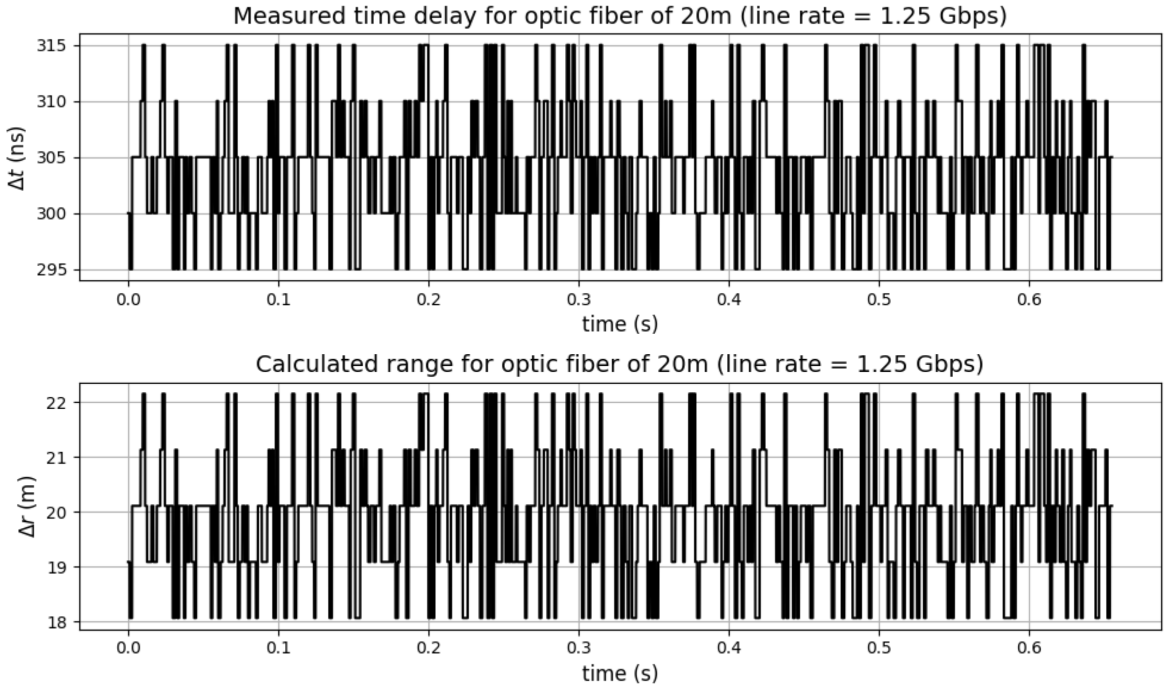


Figure 6.51: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 20 m at a line rate of 1.25 Gbps

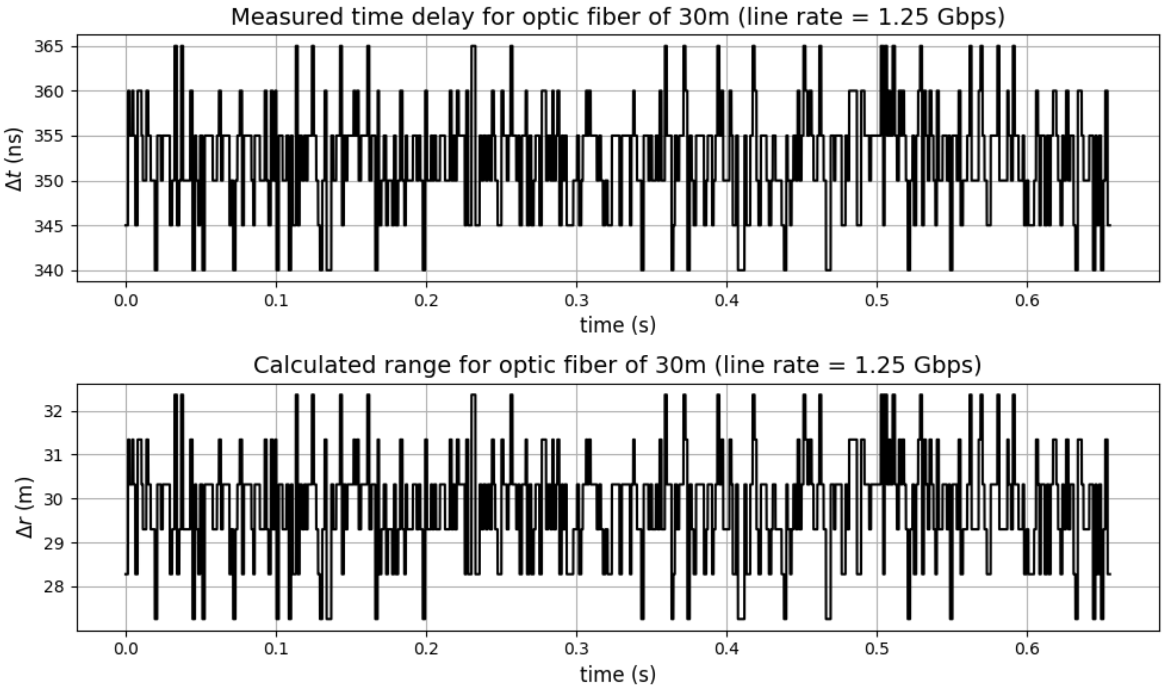


Figure 6.52: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 30 m at a line rate of 1.25 Gbps

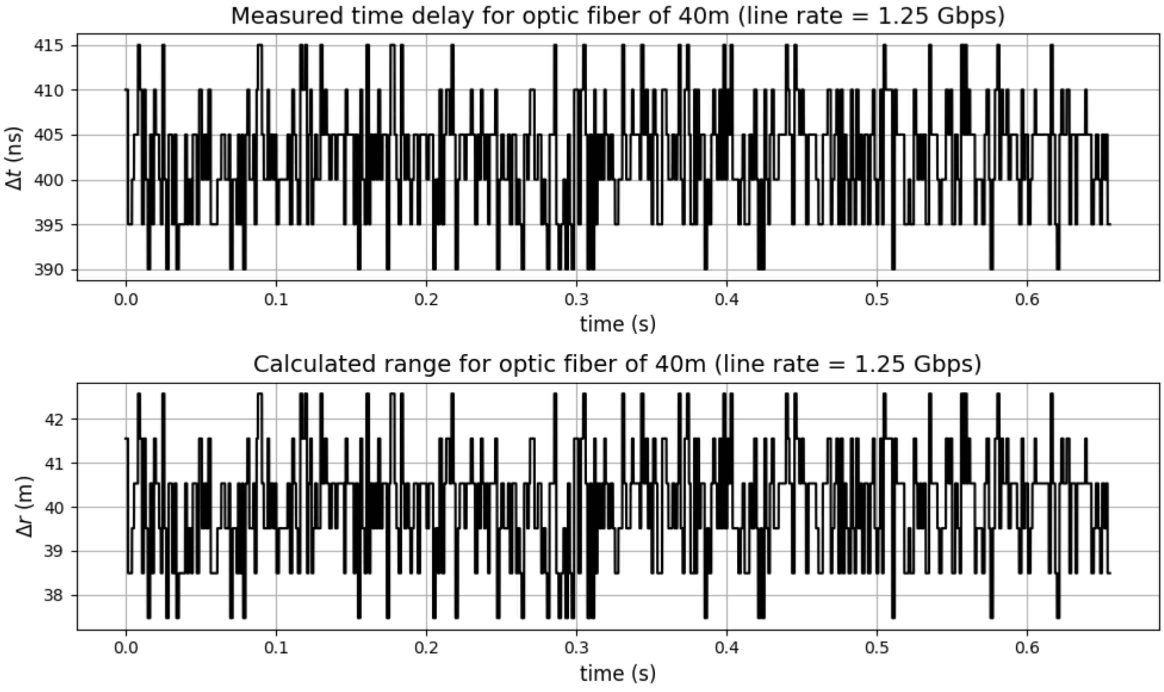


Figure 6.53: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 40 m at a line rate of 1.25 Gbps

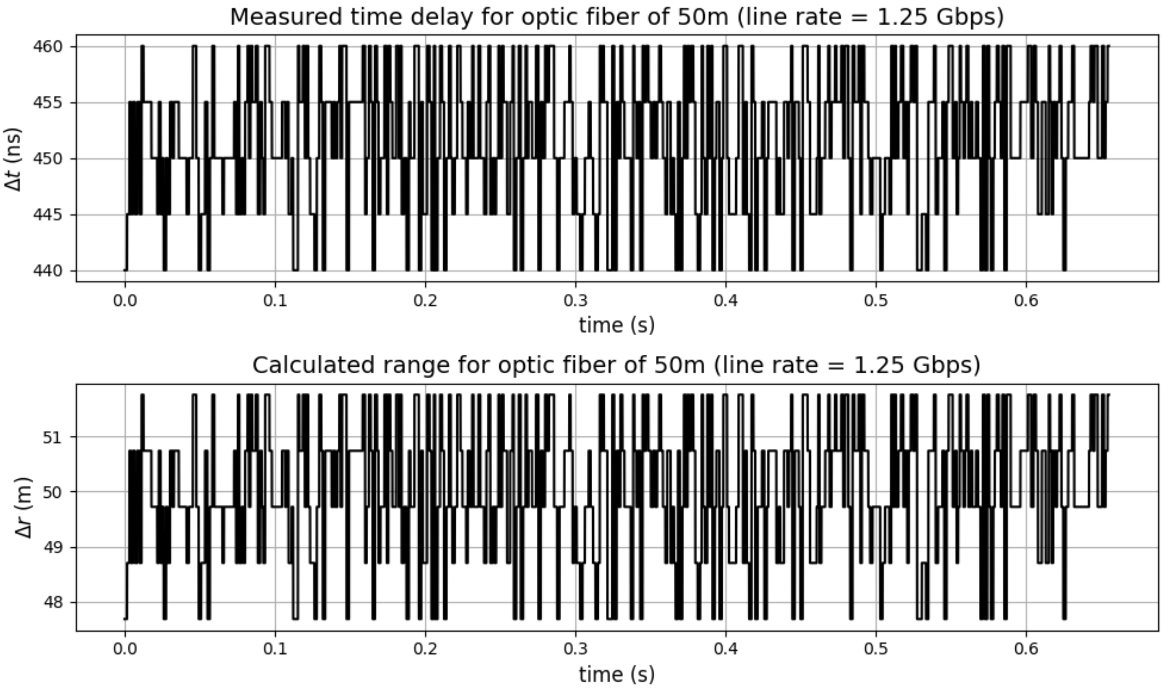


Figure 6.54: Plot of the results for a range measurement conducted on an optic fiber cable with a length of 50 m at a line rate of 1.25 Gbps

The results of these graphs are summarized in table 6.11. The standard deviation has decreased and remains constant for the different cable lengths. The relative error however steadily decreases in line with the results from the previous test.

$\mathbf{r_{target}(m)}$	μ(**ns**)	σ(**ns**)	μ(**m**)	$\sigma_{\mathbf{fiber}}$(**m**)	ε(%)
10	254.533	5.611	9.804	1.146	1.958
20	303.926	5.893	19.891	1.203	0.544
30	352.337	5.957	29.778	1.217	0.742
40	402.512	6.312	40.024	1.289	0.060
50	451.364	6.245	50.001	1.275	0.001
Average		6.004		1.226	0.661

Table 6.11: Results range measurement for a line rate of 1.25 Gbps

6.8. In-Orbit System Performance

In this section the system from this chapter will be simulated in-orbit using an orbital simulation made using Tudatpy[2]. With the known performance metrics of the system, a simulation can be made to demonstrate the potential performance of such as system in orbit. The simulation used to simulate the in-orbit performance, the Lunar Pathfinder and LUMIO orbit simulator from [7] is used. To summarize the general settings of the simulation the spherical harmonic gravity of the Earth and the Moon is modelled with a maximum order of 2 and 20, respectively, for the sake of simplicity and time efficacy.

It is not the objective of this thesis to create an accurate orbital simulation, but rather to serve as a comparison between the different systems and their respective line rates. Hence the gravitational model is simplified, and an existing orbital simulation is used, instead of trying to improve said model.

[2] `https://github.com/tudat-team/tudatpy`

6.8.1. Observation Setup

Depending on the line rate, the current system can make between 25 and 62.5 million range measurements per second. In a usable system, the RCIDs are embedded in the data stream, reducing the number of range measurements that can be made per second. In any case, the data rate is high enough that for large packet lengths, the requirement of measurements per second is still likely to be met or even exceeded. For an arbitrary packet length N_{packet} the range measurements per second R can be expressed using Equation 6.20.

$$R = \frac{N_{packet}}{N_{RCID}} \cdot f_{DRP} \tag{6.20}$$

Where N_{RCID} is the length of the RCID and f_{DRP} is one of the DRP frequencies corresponding to a chosen line rate from table 6.5, table 6.6 or table 6.7. For a large packet size the measurement rate is still in the order of several hundred kilohertz.

For orbital determination the amount of measurements per second matter as they statistically decrease the positional uncertainty. This is not the only factor however, it is also dependent on the position in orbit that is being measured. Preferably this is done intermittently spaced out over one or more orbits for the positional estimation to converge to a precise and accurate measurement.

This is however not the way the simulation is constructed. First of all, the orbit is simulated using the astrodynamics functions included in Tudatpy, with a time discretization step of Δt_{step} in seconds. Next the SCs are simulated to fly these orbits with a communication terminal that has a "noise" setting embedded into the code. The "noise" setting is called this by default, but for this simulation it is actually the uncertainty of the range measurement found in section 6.7. It is important to understand that this setting relates to the standard deviation in the range of the observations made by the simulation.

The way the simulation simulates observations made by a ground station or other terminal is at a particular integer multiple of Δt_{step}, referred to as K. Thus the measurement step can be defined as follows:

$$\Delta t_{measurement} = K \cdot \Delta t_{step} \tag{6.21}$$

To aid in visualizing this measurement process of the simulation, figure 6.55 aims to illustrate it. Every blue dot represents a simulation step, at every blue dot properties such position and velocity in 3D-space are updated according to the astrodynamics. At every red dot these steps are measured with a predefined noise. In this particular instance $K = 3$ as every third simulation step, an observation is made. This process is repeated for the entirety of the simulation, irrespective of the position in orbit of the SC.

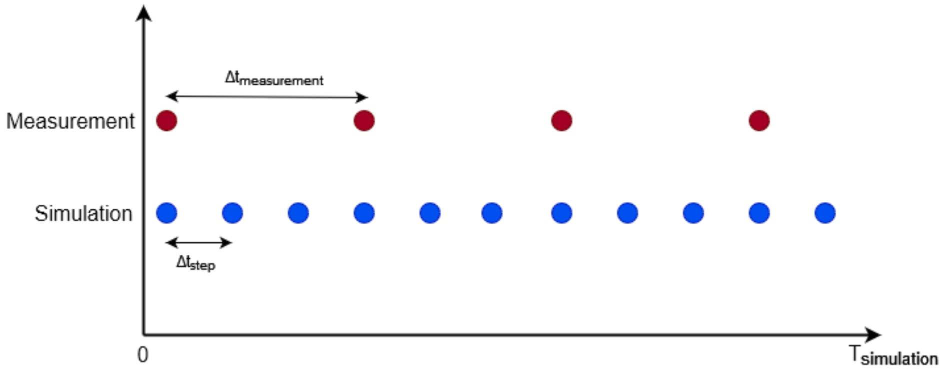


Figure 6.55: Illustration of the measurement method used in the Tudatpy simulation of LUMIO and LPF with $K = 3$.

This method of measuring has its limits with respect to the high measurement rate of the system. First of all is one that can never be avoided is that the time discretization step required to simulate this measurement rate would be in the order of nanoseconds, which is unrealistic and not feasible. Secondly, the theoretical limit is to conduct a measurement at every time step. This is however not allowed within the simulation as it results in errors and is generally a bad practice to do in orbital determination.

These reasons limit the simulated measurement rate to be in the in fractions of 1 Hz, which is not representative of the designed system and does not showcase the potential fast divergence. To circumvent this issue the uncertainty can be scaled with respect to the amount of measurements conducted at a specific interval according to Equation 6.22. Here σ is the total uncertainty based on the uncertainty of a single measurement σ_n after N repetitions. Hypothetically if the measurement time step is 5 s and the system runs at a 1 million measurements per second, N is then equal to 5 million, resulting in a small total uncertainty.

$$\sigma = \frac{\sigma_n}{\sqrt{N}} \tag{6.22}$$

The bypass is limited however by discretization errors in Python. As the measurement rate increases, the uncertainty becomes increasingly low, meaning that the positional error is prone to floating point errors which causes the simulation to raise errors and abort its process.

6.8.2. Simulation Results

As mentioned in the previous subsection, theoretically there is enough bandwidth to make several hundred thousand measurements per second. Realistically this is not ideal as one would be over sampling the noise, decreasing the orbital determination accuracy, due to the fact that the movement of the SC would not exceed the resolution limit of the system within this interval.

Knowing this and that the relative velocity of both LUMIO and LPF is roughly 300 m/s and that the resolution assuming free space would be 1.5 m, a measurement rate higher than 200 Hz would be over-fitting the noise. Hence a lower measurement rate of 10 Hz is chosen to demonstrate the in-orbit performance. The measurement rate is chosen this low, as trial and error showed that for measurement rates in the order of 100 Hz caused the simulation to diverge.

The simulation time is set to 12 days. The integration time step is set to 5 s and the measurement is set to 5 min. These parameters scale the uncertainties from section 6.7 to the values listed in table 6.12, calculated according to Equation 6.22.

Line rate (Gbps)	$\sigma_\mathbf{n}(\mathbf{m})$	$\sigma(\mathbf{m})$
0.5	3.235	0.06
1	1.618	0.03
1.25	1.226	0.022

Table 6.12: Uncertainties used in the simulation to account for a higher achievable measurement rate.

The results of this simulation are illustrated in figure 6.56 until and including figure 6.61. Starting with figure 6.56 the simulation starts to converge drastically around 6 days, after which it stabilizes until around 11 days, here there are some perturbations in the estimation. These perturbations appear to be consistent across all simulations. After these perturbations the system appears to stabilize until 12 days. This behaviour is similar in both the position and velocity estimation. Similarly in figure 6.57, the same perturbations can be observed. The rapid convergence for this simulation occurs in between 4 to 6 days, but is oscillatory in nature, which is expected of LPF. The final results of these graphs are presented in table 6.13. x, y and z are the positional errors in meters, v_x, v_y and v_z are the velocity errors. The 99.5% confidence interval is denoted by the 3σ in the table.

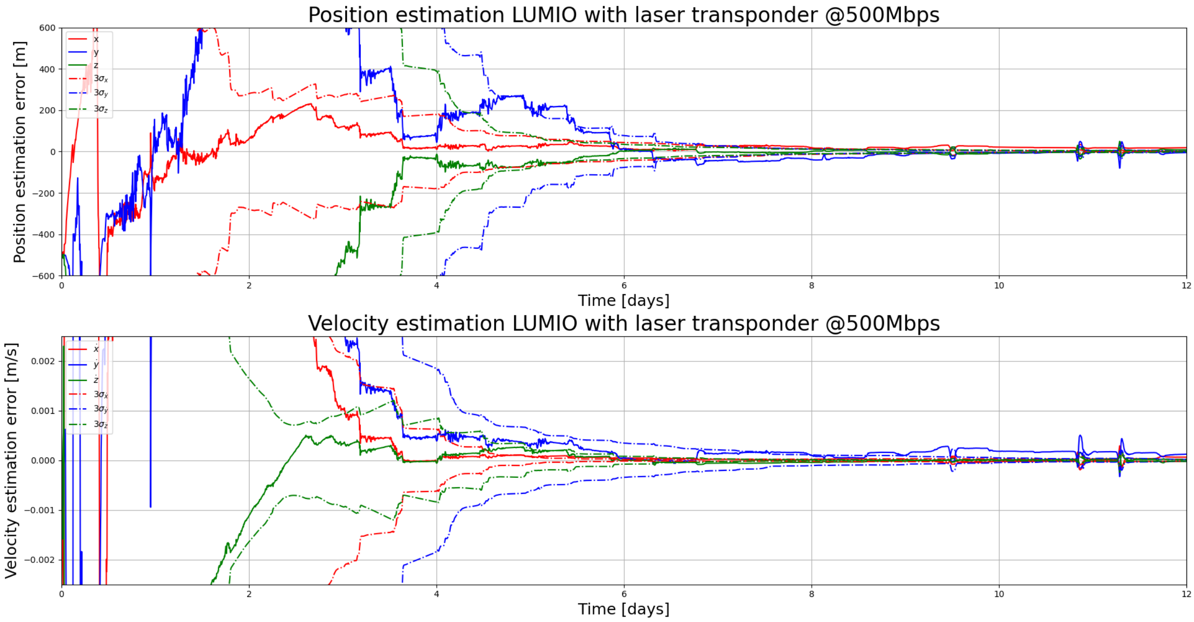


Figure 6.56: 3D Position and velocity estimation error of LUMIO with a laser transponder operating at 500 Mbps

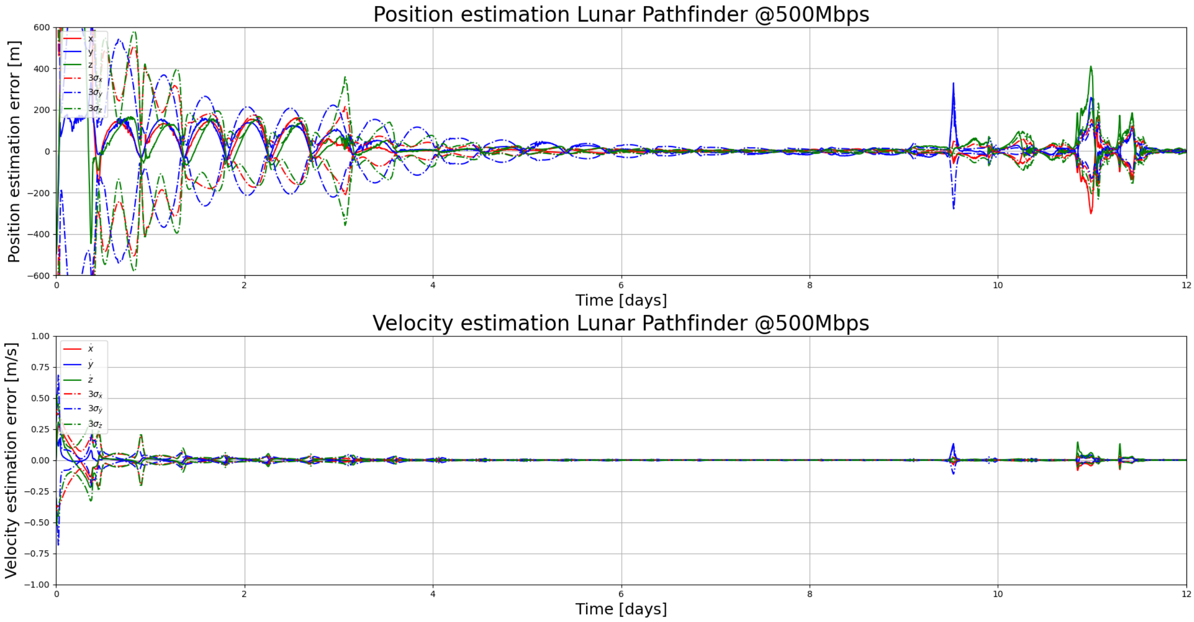


Figure 6.57: 3D Position and velocity estimation error of Lunar Pathfinder with a laser transponder operating at 500 Mbps

Table 6.13: Final results of a simulation of both LUMIO and LPF with a line rate of 500 Mbps after 12 days.

SC	**x (m)**	**y (m)**	**z (m)**	**v_x (m/s)**	**v_y (m/s)**	**v_z (m/s)**	**$3\sigma_x$ (m)**	**$3\sigma_y$ (m)**	**$3\sigma_z$ (m)**	**$3\sigma_{vx}$ (m/s)**	**$3\sigma_{vy}$ (m/s)**	**$3\sigma_{vz}$ (m/s)**
LUMIO	18.37	-5.30	8.14	$6.26 \cdot 10^{-5}$	$1.21 \cdot 10^{-4}$	$-2.60 \cdot 10^{-5}$	0.83	1.81	2.18	$2.00 \cdot 10^{-5}$	$1.56 \cdot 10^{-5}$	$1.27 \cdot 10^{-5}$
LPF	1.39	-1.88	1.46	$-2.62 \cdot 10^{-4}$	$2.58 \cdot 10^{-4}$	$-3.33 \cdot 10^{-4}$	4.27	2.66	5.90	$4.54 \cdot 10^{-4}$	$1.56 \cdot 10^{-4}$	$6.88 \cdot 10^{-4}$

Keeping all simulation parameters the same and increasing the line rate to 1 Gbps results in the graphs in figure 6.58 and figure 6.59. The simulations show to converge around the same period, between 4 and 6 days. The same perturbations occur in the estimations as with the previous simulation. The values at 12 days are presented in table 6.14.

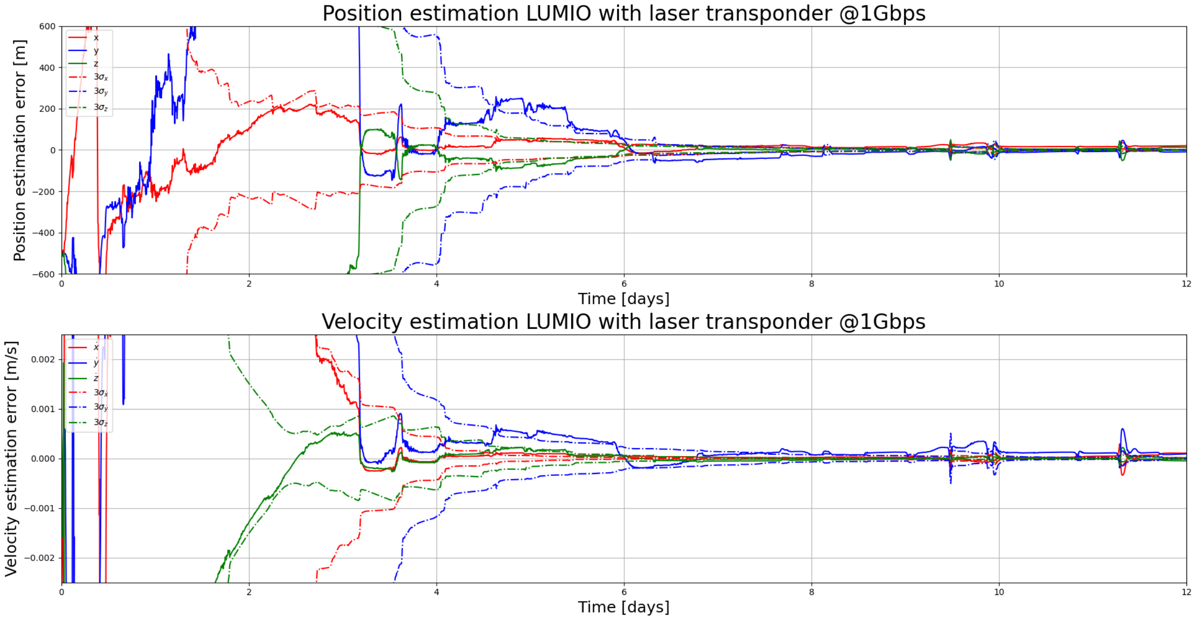


Figure 6.58: 3D Position and velocity estimation error of LUMIO with a laser transponder operating at 1 Gbps

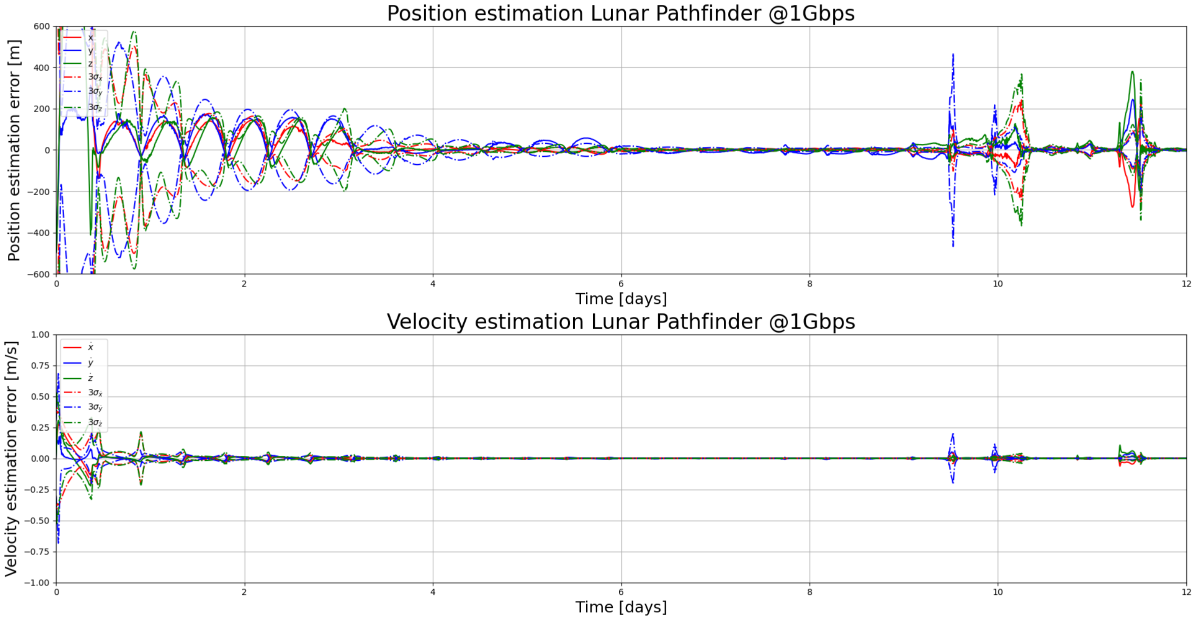


Figure 6.59: 3D Position and velocity estimation error of Lunar Pathfinder with a laser transponder operating at 1 Gbps

Table 6.14: Final results of a simulation of both LUMIO and LPF with a line rate of 1 Gbps after 12 days.

SC	x (m)	y (m)	z (m)	v_x (m/s)	v_y (m/s)	v_z (m/s)	$3\sigma_x$ (m)	$3\sigma_y$ (m)	$3\sigma_z$ (m)	$3\sigma_{vx}$ (m/s)	$3\sigma_{vy}$ (m/s)	$3\sigma_{vz}$ (m/s)
LUMIO	20.28	-9.23	12.56	$1.04 \cdot 10^{-4}$	$9.18 \cdot 10^{-5}$	$-4.88 \cdot 10^{-5}$	0.57	1.37	1.59	$1.33 \cdot 10^{-5}$	$1.11 \cdot 10^{-5}$	$9.40 \cdot 10^{-6}$
LPF	3.42	-5.44	2.00	$-1.78 \cdot 10^{-4}$	$1.56 \cdot 10^{-4}$	$-3.35 \cdot 10^{-4}$	2.06	1.69	2.87	$2.12 \cdot 10^{-4}$	$7.93 \cdot 10^{-5}$	$3.14 \cdot 10^{-4}$

Taking the similar approach as with the previous cases, the line rate is increased to 1.25 Gbps. This results in the graphs displayed in figure 6.60 and figure 6.61, with the corresponding final results after 12 days presented in table 6.15. The simulation abides by the same pattern as the cases discussed previously.

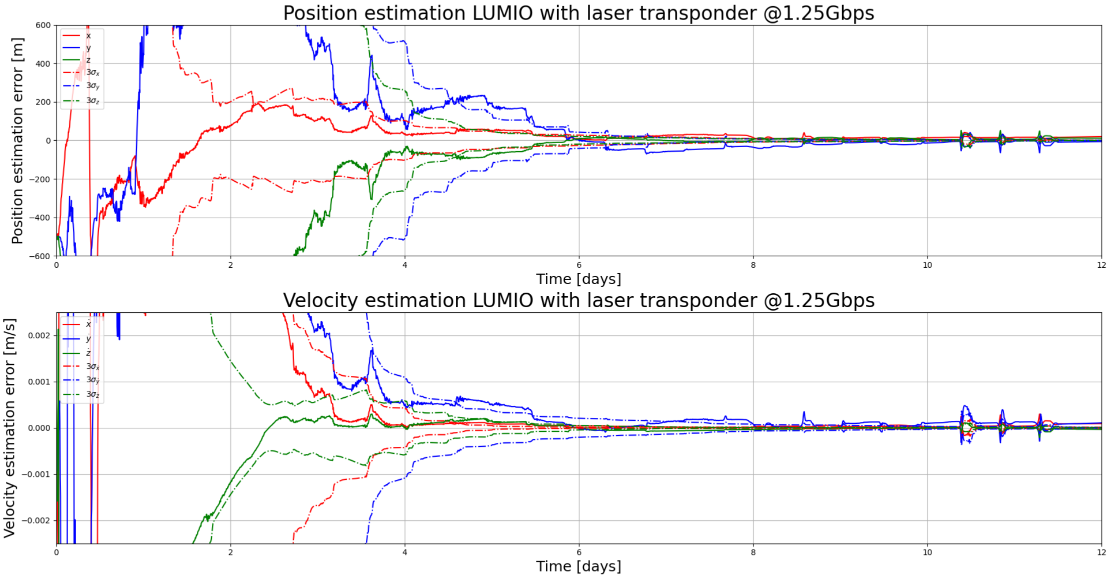


Figure 6.60: 3D Position and velocity estimation error of LUMIO with a laser transponder operating at 1.25 Gbps

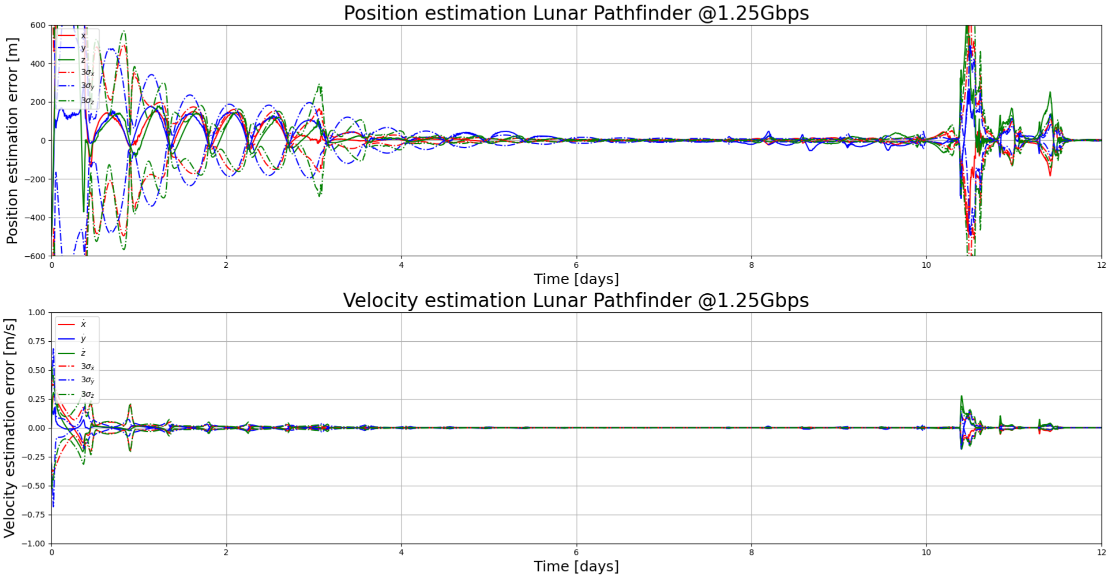


Figure 6.61: 3D Position and velocity estimation error of Lunar Pathfinder with a laser transponder operating at 1.25 Gbps

Table 6.15: Final results of a simulation of both LUMIO and LPF with a line rate of 1.25 Gbps after 12 days.

SC	x (m)	y (m)	z (m)	v_x (m/s)	v_y (m/s)	v_z (m/s)	$3\sigma_x$ (m)	$3\sigma_y$ (m)	$3\sigma_z$ (m)	$3\sigma_{vx}$ (m/s)	$3\sigma_{vy}$ (m/s)	$3\sigma_{vz}$ (m/s)
LUMIO	19.23	-6.59	11.15	$8.93 \cdot 10^{-5}$	$1.05 \cdot 10^{-4}$	$-3.61 \cdot 10^{-5}$	0.53	0.81	1.38	$1.20 \cdot 10^{-5}$	$8.70 \cdot 10^{-6}$	$7.12 \cdot 10^{-6}$
LPF	2.38	-2.67	-0.31	$-9.61 \cdot 10^{-5}$	$1.04 \cdot 10^{-4}$	$-3.96 \cdot 10^{-4}$	1.70	1.43	1.63	$1.21 \cdot 10^{-4}$	$4.96 \cdot 10^{-5}$	$2.00 \cdot 10^{-4}$

The estimation accuracy of these simulations are summarized in figure 6.62 and figure 6.63. As expected the higher the line rate, the faster the estimation accuracy improves. The final accuracies are listed in table 6.16. Overall the systems are relatively accurate for what they can currently achieve. The simulation however is not very sophisticated in terms of orbital determination, nonetheless it does provide a relevant comparison.

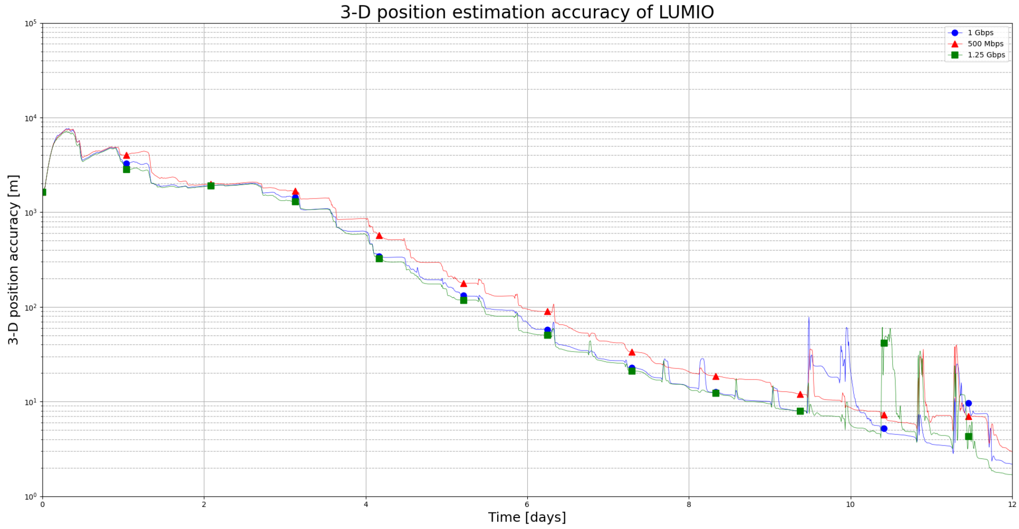


Figure 6.62: 3-D position estimation accuracy for LUMIO when equipped with a laser transponder operating at different line rates

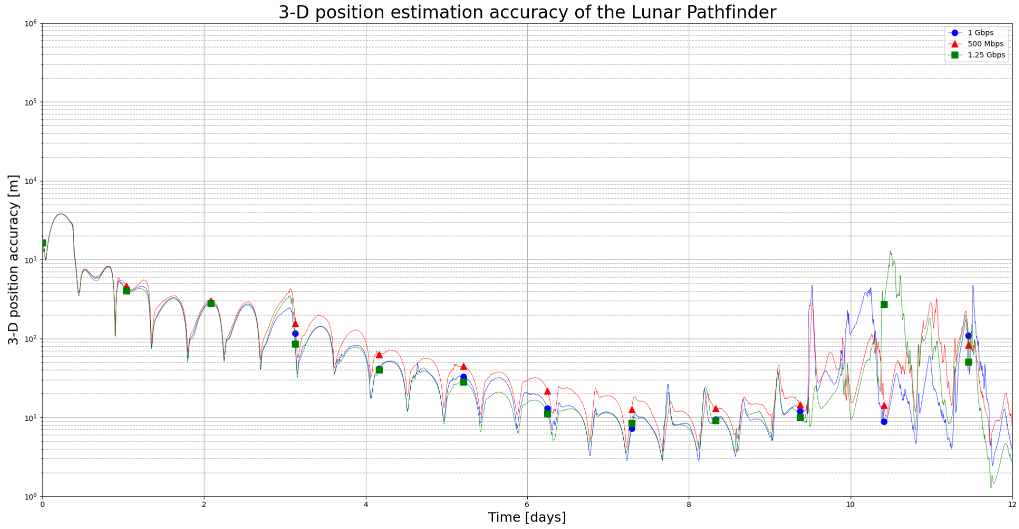


Figure 6.63: 3-D position estimation accuracy for the Lunar Pathfinder when equipped with a laser transponder operating at different line rates

SC	Accuracy 500 Mbps (m)	Accuracy 1 Gbps (m)	Accuracy 1.25 Gbps (m)
LUMIO	2.952	2.171	1.685
LPF	7.752	3.917	2.753

Table 6.16: 3D positional accuracy final values after 12 days with respect to various line rates.

6.9. Conclusion

Using a Xilinx KCU105 FPGA and the on-board SFP MGT transceivers, an experiment was conducted to demonstrate the achievable ranging performance of an optical communication link. The objective of the experiment is to answer, in part the second, third and fourth sub-question from section 1.2.

The system setup shown in section 6.2, shows the physical setup of the FPGA with an optic fiber cable, which length can be varied. The fiber is looped back into the system itself and the ToF of a RCID is computed in post-processing on a PC.

The system design, presented in section 6.3, starts with the presentation of a top-level system model displayed in figure 6.3. It shows all the components, the SFP drivers, GTH Transceivers, signal generation firmware, data capture firmware, reset synchronization and VIO control. The system clocking structure is elaborated upon next in subsection 6.3.2. The signal generation and data capture firmware operate in a different clocking domain than the GTH Transceivers do, they obtain their clock from the user clock provided by the GTH transceiver. This clock is then provided to the signal generation firmware as well as an MMCM to generate a 200 MHz phase locked clock provided to the data capture block. Within the GTH transceiver there are several clocking domains at play that are automatically scaled by the Xilinx Ultrascale Transceiver IP, which are further elaborated upon in subsection 6.3.3.

The GTH itself is configured through Xilinx IP. This is common as this firmware is relatively versatile and allows for enough freedom in terms of configurability. Projects for the Large Hadron Collider (LHC) by CERN commonly use the IP at the core of their FPGA designs such as [24]. It does not, however, allow the user to manually configure signals from within the transceiver itself as a measure of precaution. Aspects that are configured are the use of a buffer for the transmitter and receiver. From the trade-off in table 6.1, the criteria provided by [26] are listed to aid in the trade-off between using a buffer for transmitter and receiver. In the end, the removal of the buffer on both ends makes the system more deterministic with a lower latency.

Section 6.4 shows the methods employed to generate a placeholder signal, calculate the ToF from this data stream, perform calibration and finalize the measurements with post-processing. In a proposal from National Aeronautics and Space Administration (NASA) to the CCSDS, on optical telemetry stream has embedded RCIDs in the data stream on which the range can be computed [6]. A simplified version of such a signal is constructed such that it only consists of RCIDs. The time of transmission is recorded as well as the time of arrival of the identical frame to extract the ToF. To calibrate these measurements, similar tests are done using loopback devices, presented in figure 6.14 and figure 6.15. These devices provide a near-zero ToF of the signal, such that the internal delay of the FPGA can be calibrated for. Finally the post-processing of the data is required to filter out outliers that occur due to the sampling clock and the user clocks not being synchronized.

The RCIDs are generated using a 16-bit counter with a sufficiently large rollover time such that multiple instances of an RCID appearing in a captured window is unlikely. These RCIDs are 8b/10b line encoded by the GTH Transceiver and transmitted over optical fiber.

For verification in section 6.5, the system is tested with three different line rates, 500 Mbps, 1 Gbps and 1.25 Gbps with the use of a post-implementation functional simulation from Vivado ML. Nominal system functionality was observed as well as the correct clock frequencies and the correct RCID increments were observed. The post-implementation functional simulation can be considered to be relatively accurate, and from these simulations the anticipated internal delay can be deduced as well. The internal delays from the simulation were found to be 518.737 ns, 252.212 ns and 209.755 ns respectively. The relation between these values and the DRP frequency was that the internal delay is proportional to approximately thirteen DRP clock cycles.

From the first validation system tests presented in section 6.6, it can be concluded that a glitch exists in the system. Semi-regularly the receiver is shut down for a period of time, after which it is rebooted. During this process, the phase between the transmitter and receiver clock is locked arbitrarily by the internal control of the GTH Transceiver IP. The frequency at which this occurs cannot be determined due to limitations in the data capture method. The phase which the system locks to is measured by the sampler clock, this discretizes the phase positions that are possible depending on the ratio between the DRP frequency and the sampler frequency, which in turn is dictated by the line rate. For an increase in line rate, the ratio between these frequencies decreases, so does the variance in the measurements. This phenomenon has been given the name clock phase drift.

From the calibration tests, it could be concluded that the average time delay measurements largely correspond to the time values found during verification. The delay imposed by the optic fiber loopback adapter is the most representative for calibration. Furthermore the calibration displays that the internal behaviour of the MGT is constructed in a deterministic part representing the internal latency and a stochastic part representing the jitter of the clock phase drift.Listed in order from low to high line rate, the internal delay values found were 509.316 ns, 255.418 ns and 206.525 ns, with standard deviations of 15.782 ns, 8.433 ns and 6.678 ns.These values are only several nanoseconds off from the verification value, confirming that the system operates well despite system resets.

Tests with the FPGA are performed at the three line rates, each with 5 different optical fiber lengths, ranging from 10 to 50 m with 10 m increments in section 6.7. The range is computed for each cable length and the results are presented in table 6.9, table 6.10 and table 6.11 respectively. Due to the clock phase drift noise, the internal delay may vary accordingly, causing the range measurement to float around the range in a normally distributed pattern, nearly identical to the normal distribution found during calibration, with its mean shifted upwards however. The average of this measurement closely coincides with the cable length, increasing the precision of the measurement, the accuracy however remains uncorrelated as there is no evident pattern when observing the relative errors with respect to the line rates. The results are summarized in table 6.17.

Table 6.17: Summary of the test results

r_{target}(m)	μ(m) @ 500 Mbps	μ (m) @ 1 Gbps	μ (m) @ 1.25 Gbps	σ_{fiber} (m) @ 500 Mbps	σ_{fiber} (m) @ 1 Gbps	σ_{fiber} (m) @ 1.25 Gbsp	σ(ns) @ 500 Mbps	σ (ns) @ 1 Gbps	σ (ns) @ 1.25 Gbps
10	9.927	8.758	9.804	3.228	1.591	1.146	15.804	7.791	5.611
20	19.652	18.802	19.891	3.199	1.588	1.203	15.666	7.775	5.893
30	29.847	28.962	29.778	3.243	1.585	1.217	15.884	7.759	5.957
40	39.968	38.911	40.024	3.331	1.594	1.289	16.311	7.807	6.312
50	50.064	50.178	50.001	3.268	1.657	1.275	16.003	8.116	6.245

To grasp the in-orbit performance of these systems better, the performance metrics are substituted into a simulation made by [7] in section 6.8. The simulations are made using the Tudatpy kernels, for the SCs LUMIO and LPF orbiting the Moon. The goal of such this is not to create the most accurate and representative orbital simulation, but rather to serve as a means of comparison with a validated framework. Hence the spherical harmonic gravity of the Earth and the Moon is modelled with a maximum order of 2 and 20. The measurement rate of the system is relatively high, with 25, 50 and 62.5 million measurements per second. Embedding the simple signal in a real signal would still yield measurement rates well within the hundreds of thousands measurement per second that could be made. These large measurement rates cannot be simulated due to discretization errors with respect to time in the syntax of the code. The integration time step is set to 5 s and the measurement time step is taken to be 5 minutes with a total simulation time of 12 days. The positional accuracy of both LUMIO and LPF both converge to an accuracy below 10 m with a higher line rate converging faster. The system has a large enough bandwidth to accommodate for a large quantity of range measurments per second.

To answer the second sub-question:

What methods can be utilized to generate a stable and consistent signal on an FPGA?

The Xilinx Ultrascale GT Wizard IP is a powerful and robust piece of firmware that is commonly used for engineering projects that require and FPGA [24]. The data required to be transmitted is automatically serialized according to a selected encoding scheme from within the transceiver wizard. In the interface of the wizard the user is guided to selecting all parameters, line rate, reference clock source and frequency. This is accompanied by Xilinx Clocking Wizard IP that facilitates a free-running clock source required for the internal reset switches of the IP.

The GTH IP outputs two clocking sources that operate in two different clock domains called `TXUSRCLK2` and `RXUSRCLK2`. All firmware components that are related to the transmissions shall operate in the `TXUSRCLK2` domain and components related to the receiver domain shall operate in the `RXUSRCLK2` domain, to ensure signal stability.

The answer to the third sub-question:

How can the internal propagation delay of the FPGA be quantified?

The internal delay was found through the use of a bridged SFP and an optic fiber loopback adapter. These internal delays were variable dependent on the settings of the GTH as well as internal working of the GTH itself. For a lower line rate, the internal delay was found to be higher on average compared to a higher line rate. Moreover the internal delays are distributed, the magnitude of the standard deviation depends on the sampling frequency over the DRP frequency. This distribution may be caused by the frequent internal reset glitch that currently exists, however it is expected that the internal delay will not remain constant and is subject to noise coming from the GTH.

The fourth sub-question:

How does the line rate of the communication influence the range measurements in the FPGA?

table 6.17 shows the influence of the line rate on the accuracy of range measurements conducted with optic fiber. From the internal delay calibration it was found that the ToF is not constant and normally distributed around a mean with a standard deviation that depends on the line rate as well as the sampling frequency. Naturally for an increase in line rate, an increase in overall accuracy was observed.

6.10. Discussion

Overall the results are consistent for each line rate and the measurement are normally distributed roughly around the optic fiber length, due to the clock phase drift, indicating that a range measurement is possible, however it is currently surrounded by a large uncertainty. Ideally, for every transmission, the relative phase between the data clock and recovered clock is fixed and constant based on the distance between transmitter and the receiver producing range measurements as presented in chapter 4. This phase could then be compared to identify a more fine range as well as determine underlying components that contribute to the noise. Currently, the distance is only based on the ToF of an RCID as the underlying phase data itself are the largest contributor to the noise, which is less than ideal.

Assuming the phase between the clocks is locked, the resolution of the system could be expressed as multiples of the sampling frequency over the refractive index of the medium. For vacuum and a sampling frequency of 200 MHz, the resolution would be multiples of 1.5 m, as the FPGA is not able to distinguish changes in distance smaller than this. In an ideal situation where there would be no uncertainty at all, the distance measurement would increase by the respective resolution of the system when a threshold is met, meaning that for 10 m, the distance the system would measure would be 9 m, as the previous step in resolution is this value, and the next discernible value would be at least 10.5 m with the current system.

Theoretically the system, as is, is capable of performing millions of measurements per second. Realistically, the RCIDs would be embedded into the normal data stream reducing the measurements rate to several thousand to several hundred thousand measurements per second if required for orbital determination but it is scalable to a much lower measurement rate, as high measurement rates are often susceptible to noise. The orbital simulation in section 6.8 proves this in part. Although the methods employed to create the simulation do not provide a foundation of an accurate orbital estimation. The way the observations are made are continuous creating unnecessary correlation. The validity of the simulation can be disputed. Nonetheless it does provide for a comparison of the systems.

To further improve the accuracy however, the serious bottleneck of this system should be addressed. Whichever avenue is explored with the FPGA, the factor that limits it all is the sampling of the signals within the FPGA. Currently, continuous sample is only possible due to a loop that captures windows of a predetermined length, what happens in between these windows remains unknown. Many projects using FPGAs manage to extract these signals with an oscilloscope that has a much higher bandwidth. This is a brilliant method for clocking signals, but not for a multi-bit data sequence like the RCIDs. There is simply not enough capacity in a oscilloscope to measure this, and thus a more sophisticated, continuous method has to be developed with the FPGA itself.

There is certainly room for this within the FPGA, as the KCU105 has MGTs capable of reaching upwards of 16 Gbps [23], meaning that there is plenty of bandwidth for a custom written continuous data sampling firmware that directly transmits the RCIDs to an external device. This does however create the issue of correctly time tagging the RCIDs, the mechanism of which may ultimately need to be located on the FPGA as well as a sort of latching mechanism.

Concretely, what needs to happen to finalize this system is: firstly, re-configure the Ultrascale GT IP to troubleshoot the resetting bug creating a deterministic system. Secondly, manually take control over frame alignment as done in various projects from CERN. Thirdly stabilize the phase of the transmitter and receiver clock using DDMTD. Fourthly export these clocking signals through SMA connectors to measure them using an oscilloscope. Finally, improve the export of the signal data such that it is not limited by the 200 MHz by latching the time difference on the FPGA and continuously transmitting the data over a high speed link including the latched time tag. The time tag latching ultimately results in a time resolution of 1 over $811 \cdot 10^6$ seconds, and if there is a deterministic phase relation between transmission and receiving, this measurement can be complemented by the readings of the DDMTD such that a final potentially millimeter level accuracy can be achieved.

6.11. Recommendations

In this section, recommendations for future work are made based on the results of this chapter.

- **Frame Alignment**
  Organizations such as CERN often make use of the embedded Rx Slide feature in the Ultrascale GT transceiver IP [24], [31], [33]. This way the frame alignment is more deterministic as there is more control over the way frames are aligned. One could create a more refined data structure in which a custom alignment sequence is embedded, and based on the alignment sequence a controller can be built that feeds back a signal into the Rx slide input of the GT transceiver IP. This way the system is not dependent on the automatic alignment of the 8b/10b alignment circuit. This does not mean that the automatic circuit is bad under any circumstance, using the Rx slide feature simply provides the user with more control and oversight on what happens during alignment that may or may not have implications in terms of ranging.
- **Investigate the cause of the occasional receiver reset**
  For this thesis, due to lack of experience with programming and configuring MGT transceivers on an FPGA, lack of information due to the confidential nature of the IP, the behaviour of the receiver is discovered, noted and accepted as is. To create a fully fledged system however, this will need to be fixed to reduce the overwhelming clock phase drift uncertainty, to narrow down on more fine uncertainty sources that can be isolated from the data clocks.
- **Revise data capture mechanisms**
  The current data capture method is easy to use, quick and robust. It is, however, limited to a 200 MHz sampling clock rate, allowing for a resolution of 5 ns. To increase the accuracy, this sampling clock speed needs to be increased. Currently only a connection over JTAG is realized, potentially limiting this sampling speed. The alternative is to use the same firmware, but instead transmit over Ethernet which theoretically allows for a larger bandwidth. The maximum internal clock frequency that can be generated is 811 MHz, which could potentially be realised when configuring Ethernet. This does not however take away the BRAM limitations of the board. The data capture system uses a large amount of BRAM, depending on the amount of signals that are

to be captured and the size of the capture window. In case a larger sampling frequency is used, the fixed window size is filled more quickly, providing a higher resolution, but has a smaller observation time. Ideally, this would be compensated by simply increasing the window size, but this is, in turn, limited by the BRAM. There also exist more sophisticated methods, such as programming a custom serial AXI stream over either UART or Ethernet, which may circumvent the limitations of the MATLAB data capture IP, they do however require more extensive knowledge on FPGAs and data protocol in general.

- **Realize a system with two terminals**
  A second terminal needs to be added to create a complete system such as illustrated at a top level in figure 6.9. Currently the FPGA operates on a loopback, in essence creating a reflection of the original signal, in theory replacing terminal B by a terminal with no internal delay. In a full system, the signal would be received on the opposing end, processed and a response would be sent. This creates a complex system where not only the clocking mechanisms of the master terminal would have to be taken into account, but also from the slave terminal.

- **Implementation of an external DDMTD board**
  Under the assumption that the transceiver reset bug is fixed and the data stream is continuous for extended periods of time, clock synchronization is the next important step. In order to increase timing accuracy, the transmitter and receiver clock will need to be stabilized and furthermore the magnitude of the phase difference needs to be determined in order to make accurate range measurements. To compute this phase to sub-picosecond level accuracy, a DDMTD circuit can be attached on the FMC connector of the KCU105. The output of this circuit can then internally be used to synchronize clock phase to sub-picosecond level accuracy. The constraint to this method again is that these signals must remain observable in order to compute range. For the applications of CERN [8], [24], [31], [33], only phase stability itself is required, the magnitude of the phase is irrelevant to their projects. In order for this phase to be measurable, the data packets themselves will still have to be readable, which is currently limited to 200 MHz. The phase of the clocks will also have to be exported from the DDMTD to be measured. This measurement is now not limited to any internal clocking, instead it is limited by the bandwidth of the oscilloscope the beating clocks are attached to.

- **Analyse Free-Space performance**
  The board employs SFP modules which are versatile and a lot of accessories relating to these modules are available. In essence, the system is plug and play, as long as there is an SFP driving an optical transmitter and receiver for free-space. Such a setup can be constructed with an SFP, and the FPGA could be used to drive the setup. Naturally, there would be more losses compared to using optic fiber, but it would represent a more realistic scenario. An example of this would be the first experiment where a laser transmitter pointed to a retro-reflector in chapter 4, instead of the laser transmitter an SFP could be used instead, with the necessary adaptations in place.

- **Create dedicated SFP PCB**
  The MGT firmware and hardware provided by Xilinx are robust and easy to use for networking applications when the user knows how to configure such devices. However this process adds a lot of features that introduce a deterministic latency as well as a stochastic part to this latency. Furthermore, there is no direct control over what is being transmitted by the SFP as it is driven by the GTH IP. This is not inherently bad as CERN uses the same IP in addition to their own firmware, but this its application is accurate networking, not range measurements. Preferably as with the experiment from TNO in chapter 4, direct control over the laser transmitter and receiver are preferred. This would require special hardware where the modules could be mounted onto, which can be connected to the FPGA that remains as a controller for the clocks. The data can then in theory be directly exported from the dedicated hardware, circumventing the limitations imposed by the FPGA.

- **Identify clock errors**
  It is highly likely that there is unwanted errors due to the clocks. The clocks have a MGT reference clocks and there is synchronization issues with respect to the transmitter and receiver clock, which may introduce some jitter that is currently not detectable, but may be affecting the bottom line accuracy, in the event that the clock drift noise is resolved and the observation resolution is

increased.

7

Laser Signal Phase Comparison using DDMTD

As initial part of the thesis, the implementation of an external DDMTD PCB was envisioned. As part elaborated in the recommendations of section 6.11, the DDMTD would be responsible for sub-picosecond clock synchronization. Due to several factors and the time constraints of the thesis, the manufacturing and implementation of the DDMTD could no be realised.

Nevertheless, the PCB Design and Schematics are produced as part of this thesis with Fusion 360 Eagle. Having these designs available provides a basis for future work.

First, the objectives and goals are discussed in section 7.1. Here, the potential of the device is listed and what could be realized when implementing it in the current system. Next in section 7.2 the system design will be briefly elaborated upon. Finally some recommendations are provided in section 7.4.

7.1. Objective and Goals

From the previous chapters, the main aspects to improve the performance in the future are to first stabilize the current system as is, expand to two terminals instead of one, and implement the DDMTD in this system.

The objective of this chapter is to provide an answer to the fifth sub-question:

1. How can the performance of the system be improved to millimeter-level?

A concrete numerical answer to this sub-question however, can unfortunately not be provided. What can be provided is a hypothetical implementation based on the work from the previous chapters.

The goals of this chapter are therefore to:

- Present a possible system design using a DDMTD.
- Deliver schematics and PCB designs based on the work of [10] and [8].

It should be noted that, to function properly, the issues identified with the current GTH configuration (section 6.11) need to resolved to implement a system consisting of multiple terminals.

7.2. System Design

In this section, the proposed system design is presented. This system design however is based on the following assumptions:

- The system reset glitch from chapter 6 is debugged and fixed.
- The appropriate clock signals can be extracted and provided to the both FPGA through SMA connectors and the appropriate firmware is written for this.

- The data capture capabilities are increased by, for example, a custom AXI serial stream of the data to a PC.
- The PCB design is finalized, tested and fully validated before implementation.

These obstacles have to be overcome before attempting to implement the PCB into the design. The occasional reset causes a strange behaviour with respect to the clocks, which for the loopback range measurements had both advantages and disadvantages. However, for a fully fledged system, consistent and predictable behaviour is required.

Extracting clocks signals and providing them to the FPGA internals seems straightforward. Manually configuring MGT clocking sources however has proven to be a difficult task, requiring more advanced knowledge of the internal workings of such transceivers. There is only a limited amount of differential pair SMA connectors on board, meaning that inputs and outputs are scarce and FMC breakout boards may be required to provide more clock sources to the MGT.

Without the necessary tools to observe larger signals for a longer period of time, it is not recommended to continue. Creating a two terminal system requires a more complex data stream where more bits per packet are involved and the overall latency is significantly increased. The current method of data capture cannot fulfill the requirement as more data has to be captured for a longer period of time. Currently, an increase in one means a decrease in the other.

Finally, the PCB design is derived from a schematic provided by [8] [1]. The files are created from the schematics provided, however the design itself is not manufactured or tested. It may require several revisions to obtain the desired functionality and remove hardware bugs.

7.2.1. System Model

A top level system model is illustrated in figure 7.1. The system model is an expanded version of the model illustrated in figure 6.3. The signal generation is expanded and made more complex to fit the methods described in chapter 2 proposed by [6], [16].

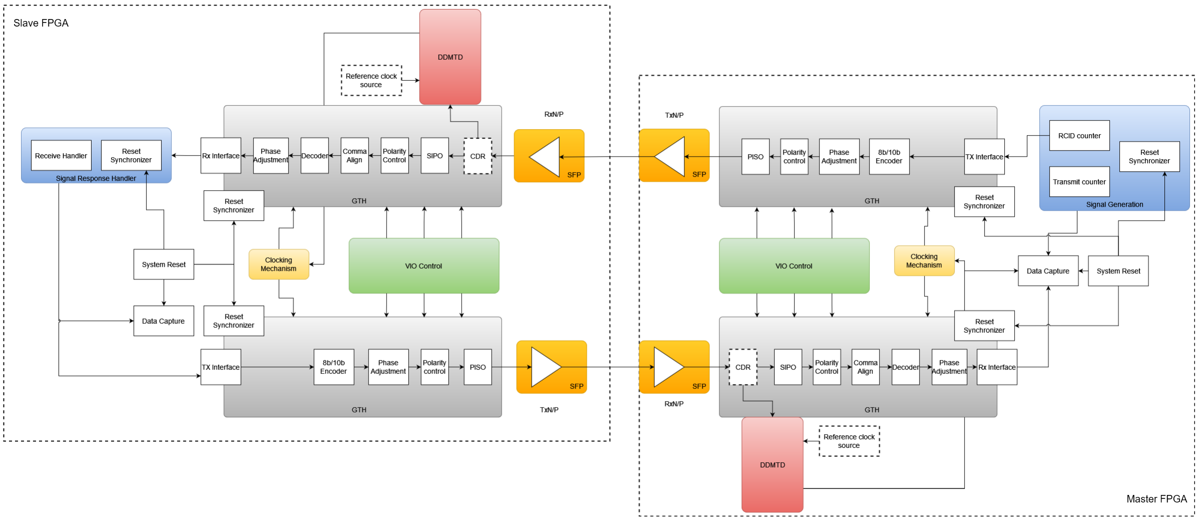


Figure 7.1: System model block diagram of the proposed system that includes full end-to-end synchronous optical communication ranging between two FPGAs with the implementation of DDMTDs

First of all, a signal is generated consisting of an RCID counter. This counter may consist of more than 16 bits compared to what is implemented in chapter 6, this is to overcome the larger latency of the system. For example, if this system was operating at a line rate of 500 Mbps, the internal delay of one terminal would be roughly 510 ns. This is naturally doubled for two terminals in addition to the RTLT between the two terminals, meaning that a 16-bit counter is likely to be reset before a response

[1] `https://gitlab.cern.ch/rsaradhy/ddmtd/-/tree/master`

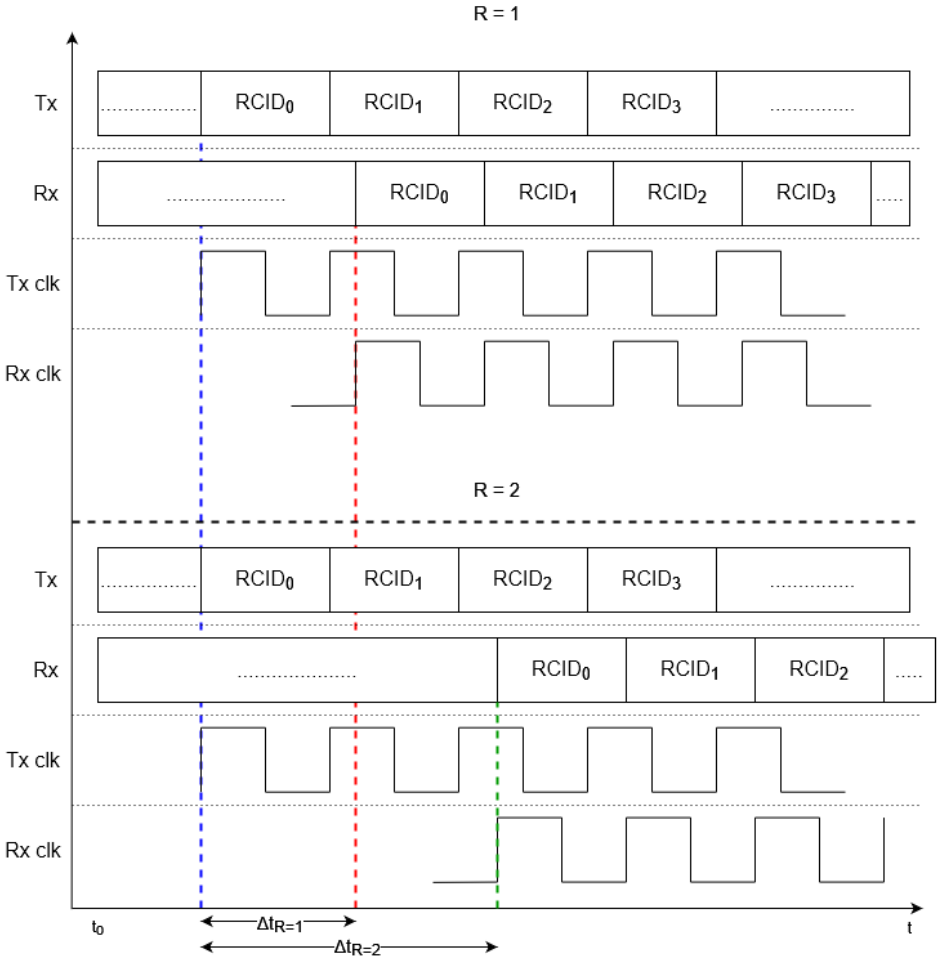


Figure 7.2: Desired behaviour of the data and the phase of the data clocks with respect to range *R*.

is received creating ambiguity. The RCID is transmitted and a timer is captured, time stamping the transmission of an RCID.

This is transmitted to the slave FPGA where the CDR recovers the transmission clock from the data. This is then provided to the DDMTD which computes a phase difference in comparison to a reference clock. The phase difference is then fed into a clocking mechanism that creates a clock signal that is phase-locked to sub-picosecond level to the reference clock. This clock signal is then provided for both the transmitter and receiver, synchronizing the two transceivers. The data is handled through necessary firmware and re-transmitted back to the master FPGA.

Upon arrival at the master FPGA, the data clock is recovered again, where it is compared to the reference clock. Here, a phase difference can be observed between the reference clock and recovered clock, which can be sampled. The time of arrival is recorded and the ToF can be accurately determined as both terminals are synchronized with respect to one and other.

For this to work the relation between the data, data clocks and range must be locked as illustrated in figure 7.2. The system is required to have a predictable relation between these parameters, every time the system is instantiated, with the relation predictable changing when the range between the terminals increases or decreases. In this figure the range is denoted by unit *R*, depicting an arbitrary range. Whenever the range is increased, the sequence is expected to change as illustrated.

7.2.2. PCB Design

The PCB circuit schematics are directly taken from [8]. The schematic images are converted to functional schematics in Fusion360 Eagle. The schematics are presented in Appendix B.

From here, a PCB design file is created, all the individual components are modelled and all electrical component values are quantified. To be noted is that the PCB is modelled after the design from CERN[2]. Figures of these layouts can be found in Appendix C.

7.3. Conclusion

In section 7.2 a concept system design is presented that illustrates the potential method to implement DDMTD in an end-to-end optical communication system. Here DDMTD is used to measure the phase relation between the transmitter and receiver, which must be locked, which can be achieved by using Tclink [25] for instance.

A first iteration of the DDMTD PCB is made in subsection 7.2.2 with Fusion360 Eagle, the schematics of which are provided in Appendix B and Appendix C. Ther are based of the schematics from [8] made for the White Rabbit project from CERN.

To answer the fifth and final sub-question:

> How can the performance of the system be improved to millimeter-level?

The main limit in the system is the rate at which the signals can be observed in the FPGA in order to determine the ToF. In order to circumvent this, DDMTD can be implemented in the system to first of all stabilize the data clock phase of the transmitter and receiver. This phase data, under the condition the system data structure behaves as illustrated in figure 7.2, can be extracted as the signals are available on the external DDMTD.

This information with regards to the phase is, if the DDMTD is configured accordingly, potentially accurate to the femtosecond. The ToF measurements can be complemented by this phase data to obtain highly accurate range measurements.

The implementation of the DDMTD may not be straightforward and will require more analysis and experience with FPGAs. The performance can potentially be improved in an end-to-end system by the exceptional timing distribution the device allows for, in combination with the appropriate FPGA firmware [8] [10].

7.4. Recommendations

The following recommendations are given

- **Improve data capture before implementation of the DDMTD**
  As mentioned in the recommendations of chapter 6, the current data capture reaches its limits as soon as the signal data width is increased or a longer window is to be captured. A different method, through for example a custom AXI stream must be devised before attempting to implement the DDMTD.
- **Implement deglitcher in FPGA**
  The computational resources of the FPGA can be utilized much more efficiently. A DDMTD will require deglitching [10] which can be offloaded to the FPGA, reducing the amount of post-processing and may ultimately be necessary in order to phase lock a clock.
- **Use an FMC breakout board**
  This system will likely require the export and import of various single-ended and differential clock pairs. The KCU105 has a limited amount of differential pair clock inputs (figure 1.1). To circumvent this issue, a breakout board could be attached to one of the free FMC slots on the board, providing more possibility for input and output through SMA connectors.
- **Adapt the DDMTD PCB to extract the beating clocks directly**
  To relax the constraints on the data acquisition on board of the FPGA, it is a smart idea to adapt the PCB such that it is easy to export the phase data to an oscilloscope directly by implementing dedicated probe points for example. This way the FPGA only has to capture the ToF data, which can then be complemented by the oscilloscope measurements.

[2] https://gitlab.cern.ch/rsaradhy/ddmtd/-/tree/master

8
General Conclusion

To conclude the thesis, a general overview will be provided of all the activities with their respective results as well as the links between them. Furthermore the main research question will be concretely answered.

The first test in order to formulate an answer to the research question is conducted in chapter 4. Here a free-space laser ranging experiment with the use of the setup provided by TNO described in detail in section 4.2 is executed. It considers a laser transmitter and receiver located at the Meteotoren in Scheveningen and a retro-reflector located at the top of the TNO building in The Hague, separated by approximately 2442 m of atmosphere. The goals of this experiment were obtain range measurements using the ToF of laser, quantify the uncertainties in the setup and identify bottom-line constraints.

The way the system operates is that a set of 40 laser pulses are transmitted at a frequency of 5 MHz with a duty cycle of 25%, lasting for a total of 8 µs. After the last pulse, no pulses are transmitted for 20 µs, after which the cycle repeats itself. Within this period of silence, the reflected signal is expected to be observed on the receiver side of the setup. These signals are sampled using a oscilloscope with a sampling rate of 50 MS/s, triggered at the first rising edge of the transmitted pulse, resulting in 100 captures. From these measurements a range could be recovered according to the method described in section 4.3. First the data is filtered statistically such that clear boundaries with respect to the start and end of both transmitted and received pulses are present in the data, regardless of the SNR of the signal. Based on these times, the average range could be calculated in section 4.4, which resulted between 2459.73 m and 2463 m with a standard deviation of 1.26 m. The uncertainty of this measurement remains 6 m due to sampling.

The major issue with this test setup was that proper calibration for ranging purposes were lacking due the fact that this test setup was not built by TNO with laser ranging in mind. Hence there is roughly 20 m residual to obtain a measurement of roughly 2442 m. The exact distance to the retro-reflector remains unknown and it is difficult to determine exactly as well. The major limit in terms of resolution was the poor sampling frequency of the oscilloscope, resulting in a high uncertainty. Furthermore, the range measurements were not based on the data structure but rather of the registered arrival and transmission of photons. Hence the main points that need to be addressed in the systems that follow are proper calibration, and allowing for the implementation of a continuous data stream such that laser communication ranging can be conducted.

In the next chapter, chapter 5, the system that addresses these shortcomings is described. An system built on an FPGA is devised with its primary goal to accurately map the internal delay of the system such that this can be used as a value of calibration for the range measurements. Moreover, it provides a great opportunity to gain experience in programming an FPGA and familiarizing with internal clocking mechanisms and the development environment. The system design, presented in section 5.2, shows the system design to measure the internal delay. The firmware written routes a 10 MHz signal to a GPIO pin on-board, which ought to mimic the location of the transmitter. This is connected with a small piece of copper wire to another GPIO pin, which is routed back to the FPGA. Based on the phase difference between the transmitted and the received signal at the same time instance, a time delay can be computed.

This creates two additional problems, one being the ability to measure the phase difference with sufficient enough accuracy, the second being able to capture signal data from within the FPGA. For the first issue, DDMTD is utilised, which is a method employed in networking solutions by CERN, explained in detail in section 2.4. The digital circuit is implemented into the FPGA such that the maximum obtainable resolution of the phase difference is 0.01 ns. The second issue is resolved by using IP from MATLAB, which allows the user to sample a window of signal data from within the FPGA that can be exported to a PC. The maximum achievable sampling rate of this firmware however is limited to 200 MHz, which is already a 4 times improvement over the acquisition rate of the previous experiment.

The method of obtaining the internal delay is described in section 5.3. The data captured by the MATLAB IP can be analysed using Python. When routing the transmitted and received signal to the DDMTD it generates two beating clocks with a much lower frequency in the kHz range. From these signals a time difference can be computed. Before the measurements are made, the system is verified using a post-implementation functional simulation in section 5.4, which is an advanced simulator included in the Vivado ML software package that accurately mimics the actual hardware the firmware is loaded on. From this simulation could be concluded that a bias in the DDMTD is expected as the measured phase difference is always slightly below what is the actual phase difference programmed into the verification firmware.

In section 5.5 this behaviour is validated by running a custom version of the firmware on the FPGA, where specific phase differences are requested by the on-board PLL. It could be concluded that the DDMTD exhibits a constant bias, where it underestimates the phase difference with 281.4 ps on average. This is taken into account in when measuring the internal delay in section 5.6. From the results an average internal delay of 15.83 ns with a standard deviation of 275.7 ps was found, which translates to a range of 4.75 m with a standard deviation of 8.27 cm.

Remarkable is that the resolution at which these measurements are made is significantly higher than the sampling rate physically allows due to the presence of the DDMTD. If a XOR gate would be implemented as digital phase frequency detector, the resolution would be limited to 5 ns. This shows the potential of the implementation of DDMTD in ToF measurements.

From the results of this experiment could be concluded that the internal delay found was not representative for a signal that travels through the MGT. Hence the research questions this chapter aimed to answer, remained unanswered and the system had to be re-calibrated whenever optical communication was inevitably implemented. The goals of the chapter are met however, a strong foundation in terms of knowledge and experience with respect to programming an FPGA is layed.

The next step is to implement communication and obtain a range measurement through the data structure. To achieve this the MGT on board of the FPGA need to be utilised, which is done in chapter 6. First the system is re-calibrated, followed by range measurements over optic fiber are conducted with the system. In general, the system aims to abide by the method of optical telemetry ranging [6], [16], the data structure of which is outlined in subsection 2.3.1. The experimental setup is provided in section 6.2, consisting of the FPGA, SFP modules and single-mode optical fiber cables, each with a length of 10 m, combined to create lengths ranging from 10 m to 50 m.

The system design is described in section 6.3. Here a system model is presented that incorporates the GTH of the FPGA to transmit and receive a custom signal, composed of RCIDs at line rates of 500 Mbps, 1 Gbps and 1.25 Gbps. The line rate is a definition commonly referred to by Xilinx to denote the data rate that exists in the medium in which information is relayed. The lower limit on the line rate is imposed by the lowest the MGT is allowed to operate and the upper limit is bound by the maximum attainable rate of the SFP modules. The identifiers are simple 16-bit integer numbers generated by a counter. The signal is 8b/10b line encoded to allow for an easy to use communication system that automatically aligns the serial data into packets.

The method of determining the range is described in section 6.4. The time of arrival and time of transmission of each RCID is compared in order to measure the ToF and therefore calculate the range or in this case the fiber optic cable length. Before an accurate range measurement can be made however, the re-calibration needs to be addressed. This is done through two devices, a fiber optic loopback adapter consisting of a very small piece of optic fiber (~2 cm) and a bridged SFP adapter, that directly bridges the transmitter and receiver drivers in the SFP. This creates a negligible external delay, essentially only leaving the delay that can be attributed to the internal mechanisms of the FPGA.

From these calibration measurements it could be concluded that the internal delay is constructed of a deterministic and a stochastic part, the magnitude of both reducing with an increase in line rate. The deterministic part of the latency can be attributed to the MGT, which is also confirmed in validation, while the stochastic part is largely contributed by a system bug, which causes the MGT to periodically reset, causing the clocking mechanisms to periodically change phase which is discretely measured by the data capture IP.

Furthermore post-processing of the data is required due to synchronization issues between the data and the clock driving the acquisition firmware. This is an unfortunate byproduct of this data capture method as it can only operate in one clock domain and will have to be tolerated. Most of these artifacts can be resolved using post-processing.

A VHDL testbench is written to verify the firmware with a post-implementation functional simulation in section 6.5. With these simulations a virtual optic fiber cable with zero length is attached to the terminals, in essence creating an instant loopback. This method allows for verification values with respect to the calibration as well as verify the correct functioning of the system. The simulation is done using the aforementioned line rates and internal delays of 518.737 ns, 252.212 ns and 209.755 ns were obtained from the low to high line rates respectively.

Comparing this to the system validation from section 6.6, the values are similar. The optic fiber loopback adapter was attached to the SFP measurements with this configuration were made. From this, the deterministic part, or average internal delay, spanning from low to high data rate was found to be 509.316 ns, 255.418 ns and 206.525 ns. These values are within several nanoseconds of the numbers obtained through simulation. The behaviour the simulation did not register however is the stochastic part of these measurement, which have σ values of 15.782 ns, 8.433 ns and 6.678 ns respectively. These standard deviations can be largely attributed to the fact that a system bug exists, which could not be resolved given the time span of the thesis.

Naturally this stochastic part is inherited in the range measurements made with the system, the results of which are presented in section 6.7. A summary of the results is presented in table 6.17. From the results can concluded that the variance in the measurements is nearly identical, usually within 1 ns of the standard deviation found during calibration, and is simply part of the system behaviour as of writing the report. Overall the system is capable of measuring the length of the optic fiber cables. With the largest relative error of 4.989% for a line rate of 1 Gbps.

To visualize the performance of the system in-orbit and compare the various line rates to each other, an orbital simulation using Tudatpy kernels from [7] is used and is presented in section 6.8. Here the orbits of LUMIO and LPF are simulated and estimated. The standard deviations expressed in meter, found in section 6.7, are inserted in the simulation and an orbital estimation can be made. Due to the high data rate of the system, theoretically millions of range measurements per second can be conducted. This is however not realistic and a measurement rate of 10 Hz was chosen, as higher measurement rates will result in floating point errors in the simulation. The final positional errors are presented in table 6.16. The orbital simulation serves as a comparison between different system configurations, and is not intended to be a highly accurate representation of reality.

From this system the major limiting factor that is identified was the limit on the frequency at which measurements can be made. In chapter 5, this was circumvented using DDMTD, which proved to be an elegant method and relatively easy to implement to measure clock phase difference. Given the time limits imposed on the thesis, the implementation of DDMTD in the optical communication could not be realized. An improvement to the internal DDMTD is to make an external PCB with the physical circuit printed on it, as used in projects of CERN such as White Rabbit[8].

A start to this is made in chapter 7. A first iteration PCB design based of the schematics from CERN. The PCB design is made using Fusion360 Eagle, the schematics of which can be found in Appendix B and Appendix C. A suggestion on the implementation is proposed in subsection 7.2.1, where the external DDMTD is connected to the FMC of the FPGA board to take care of stable phase computations between the receiving and transmitting clock potentially improving the accuracy in the end-to-end system. The requirement for this to work is that the relation between data, data clock phase and range is deterministic, which in the system in chapter 6 due to the system bug.

To conclude the thesis the main research question is to be answered. In this chapter a concise answer to this question will be provided. The main research question is stated first:

> What ranging performance can be achieved in a satellite laser communication system, with the use of commercial off-the-shelf Multi-Gigabit Transceivers on an Field Programmable Gate Array?

The ranging performance on the FPGA depends on several factors:

1. The line rate of the optical communication link.
2. The sampling frequency of the data capturing method.
3. The relative phase between the transmitter and receiver clock in the GTH.

First of all the line rate. The line rate ultimately determines the amount of range measurements per second that can be used for orbital estimation. If a system is capable of making more measurements per second the positional uncertainty drops significantly. This large measurement rate relaxes the constraints for the orbital determination department of a space mission as the range measurement rate can is salable from fractions of a hertz to several hundred kilohertz.

Secondly the system as has a standard deviation dependent on the sampling frequency over the DRP frequency. The lower this ratio is, the lower the standard deviation in this measurement. This is an artifact that does however disappear whenever the reset bug is fixed. This will cause the system to be more deterministic.

Upon link establishment, the transmission clock and the recovered clock inhibit a specific phase offset. This phase offset is not constant and is random every new link establishment. Institutions such as CERN solve this problem by implementing their TC link [24] such that phase stability is guaranteed. TC link has a built in DDMTD which is limited by the floating point clock divider from the Xilinx MMCM as elaborated in subsection 5.2.4. An external DDMTD could be employed to obtain a higher level of accuracy due to dedicated oscillator on the PCB.

Finally, with the current method the signal data needs to be retrieved from the FPGA at a particular frequency resulting in an inherent uncertainty that is equal to the speed of light of the medium through which the signal travels multiplied by the sampling period. Furthermore the aforementioned relation between the sampling frequency and the DRP frequency plays an important role for the current range measurements.

Concretely, the standard deviation of the range measurements made for line rates of 500 Mbps, 1 Gbps and 1.25 Gbps are 3.235 m, 1.618 m and 1.226 m respectively.

References

[1] D. M. Boroson, J. J. Scozzafava, D. V. Murphy, B. S. Robinson, and M. I. T. Lincoln, “The lunar laser communications demonstration (llcd),” in *2009 Third IEEE International Conference on Space Mission Challenges for Information Technology*, 2009, pp. 23–28. DOI: 10.1109/SMC-IT.2009.57.

[2] G. Cai, J. Sun, G. Li, *et al.*, “Self-homodyne free-space optical communication system based on orthogonally polarized binary phase shift keying,” *Appl. Opt.*, vol. 55, no. 17, pp. 4514–4521, Jun. 2016. DOI: 10.1364/AO.55.004514. [Online]. Available: https://opg.optica.org/ao/abstract.cfm?URI=ao-55-17-4514.

[3] J. Ma, K. Li, L. Tan, S. Yu, and Y. Cao, “Performance analysis of satellite-to-ground downlink coherent optical communications with spatial diversity over gamma&#x2013;gamma atmospheric turbulence,” *Appl. Opt.*, vol. 54, no. 25, pp. 7575–7585, Sep. 2015. DOI: 10.1364/AO.54.007575. [Online]. Available: https://opg.optica.org/ao/abstract.cfm?URI=ao-54-25-7575.

[4] L.-X. Meng, L. Li, L.-z. Zhang, S.-s. Zhao, and H.-l. Jiang, “Research on optic antenna of space laser communication networking,” in *International Symposium on Photoelectronic Detection and Imaging 2013: Laser Communication Technologies and Systems*, K. E. Wilson, J. Ma, L. Liu, H. Jiang, and X. Ke, Eds., International Society for Optics and Photonics, vol. 8906, SPIE, 2013, p. 890 627. DOI: 10.1117/12.2034892. [Online]. Available: https://doi.org/10.1117/12.2034892.

[5] W. Wu, M. Chen, Z. Zhang, X. Liu, and Y. Dong, “Overview of deep space laser communication,” *Science in China Series F: Information Sciences*, 2018. DOI: 10.1007/s11432-017-9216-0.

[6] M. Sánchez Net and J. Hamkins, “Optical telemetry ranging,” *The Interplanetary Network Progress Report*, vol. 42-221, pp. 1–23, May 2020. [Online]. Available: https://ipnpr.jpl.nasa.gov/progress_report/42-221/42-221B.pdf.

[7] T. Tanis, “Autonomous orbit determination in cislunar space,” Available at https://repository.tudelft.nl/islandora/object/uuid:4e0f0a32-9028-413c-b815-2639ce360745/datastream/OBJ/download, Master’s Thesis, University of Technology Delft, 2022.

[8] R. Saradhy, E. Frahm, E. B. Mendes, and R. Rusack, “A sub-picosecond digital clock monitoring system,” *Journal of Instrumentation*, vol. 18, no. 01, T01003, Jan. 2023. DOI: 10.1088/1748-0221/18/01/t01003.

[9] P. Moreira and I. Darwazeh, “Digital femtosecond time difference circuit for cern’s timing system,” Tech. Rep., 2011.

[10] P. M. O. F. Moreira, “Timing signals and radio frequency distribution using ethernet networks for high energy physics applications,” Ph.D. dissertation, Department of Electrical and Electronic Engineering, University college of London, 2014.

[11] F. Zhu, S. Lu, J. Sun, R. Zhu, X. Hou, and W. Chen, “Inter-satellite laser-ranging based on intradyne coherentdetection,” *Applied Optics*, vol. 60, no. 28, pp. 8930–8938, Oct. 2021. DOI: 10.1364/AO.434807. [Online]. Available: https://opg.optica.org/ao/abstract.cfm?URI=ao-60-28-8930.

[12] T. Palings, “Literature study: Review of laser communication ranging,” Mar. 2023.

[13] J. B. R. Battat, T. W. Murphy, E. G. Adelberger, *et al.*, “The apache point observatory lunar laser-ranging operation (apollo): Two years of millimeter-precision measurements of the earth-moon range1,” *Publications of the Astronomical Society of the Pacific*, vol. 121, no. 875, p. 29, Jan. 2009. DOI: 10.1086/596748. [Online]. Available: https://dx.doi.org/10.1086/596748.

[14] B. S. Sheard, G. Heinzel, K. Danzmann, D. A. Shaddock, W. M. Klipstein, and W. M. Folkner, “Intersatellite laser ranging instrument for the grace follow-on mission,” *Journal of Geodesy*, vol. 86, pp. 1083–1095, 2012. DOI: 10.1007/s00190-012-0566-3. [Online]. Available: https://doi.org/10.1007/s00190-012-0566-3.

[15] J. O. Dickey, P. L. Bender, J. E. Faller, *et al.*, “Lunar laser ranging: A continuing legacy of the apollo program,” *Science*, vol. 265, no. 5171, pp. 482–490, 1994. DOI: 10.1126/science.265.5171.482. eprint: https://www.science.org/doi/pdf/10.1126/science.265.5171.482. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.265.5171.482.

[16] M. Sánchez Net, "Optical ranging: Synchronous-mode concept, prototype, and validation.," *The Interplanetary Network Progress Report*, vol. 42-229, pp. 1–47, May 2022. [Online]. Available: https://ipnpr.jpl.nasa.gov/progress_report/42-229/42-229B.pdf.

[17] J. J. Degnan, "Laser transponders for high-accuracy interplanetary laser ranging and time transfer," in *Lasers, Clocks and Drag-Free Control: Exploration of Relativistic Gravity in Space*, H. Dittus, C. Lammerzahl, and S. G. Turyshev, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 231–242, ISBN: 978-3-540-34377-6. DOI: 10.1007/978-3-540-34377-6_11. [Online]. Available: https://doi.org/10.1007/978-3-540-34377-6_11.

[18] "Optical communications physical layer," Consultative Committee for Space Data Systems, Tech. Rep. CCSDS 141.0-B-1, Aug. 2019.

[19] "Optical communications coding and synchronization," Consultative Committee for Space Data Systems, Tech. Rep. CCSDS 142.0-B-1, Aug. 2019.

[20] Y. Hu, D. Tso, S. Kapai, and M. Feldman, "D-DMTD: Digital dual mixer time difference.," Tech. Rep. SAND2017-10097 672174, Sep. 2017. DOI: 10.2172/1494164. [Online]. Available: https://www.osti.gov/biblio/1494164.

[21] J. W. Marini and C. W. Murray, "Correction of laser range tracking data for atmospheric refraction at elevations above 10 degrees," NASA Goddard Space Flight Center, Greenbelt,MD, Tech. Rep. NASA Report X-591-73-351, Nov. 1973.

[22] P. Neyezhmakov and A. Prokopov, "Using the average air temperature along the trace for atmospheric correction in laser ranging: Accuracy analysis," *Ukrainian Metrological Journal*, Apr. 2023. DOI: 10.24027/2306-7039.4.2022.276320.

[23] *Kcu105 user guide*, English, version Version 1.10, Xilinx, 128 pp. [Online]. Available: https://www.xilinx.com/content/dam/xilinx/support/documents/boards_and_kits/kcu105/ug917-kcu105-eval-bd.pdf, February 6, 2019.

[24] E. Mendes, S. Baron, C. Soos, J. Troska, and P. Novellini, "Achieving picosecond-level phase stability in timing distribution systems with xilinx ultrascale transceivers," *IEEE Transactions on Nuclear Science*, vol. 67, no. 3, pp. 473–481, 2020. DOI: 10.1109/TNS.2020.2968112.

[25] E. B. D. S. Mendes, S. Baron, and M. Taylor, "TCLink: A Timing Compensated High-Speed Optical Link for the HL-LHC experiments," *PoS*, vol. TWEPP2019, p. 057, 2020. DOI: 10.22323/1.370.0057. [Online]. Available: https://cds.cern.ch/record/2724958.

[26] *Ultrascale architecture gth transceivers*, English, version Version 1.7.1, Xilinx, 518 pp. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ug576-ultrascale-gth-transceivers, August 18, 2021.

[27] Xilinx, *Kcu105 gth ibert design creation*, Available at https://www.xilinx.com/member/forms/download/design-license.html?cid=d77a4e72-c593-4d2d-9ff4-93221c7fe0c5&filename=xtp346-kcu105-gth-ibert-c-2017-3.pdf, Oct. 2017.

[28] *Xilinx power estimator user guide*, English, version Version 2023.1, Xilinx. [Online]. Available: https://docs.xilinx.com/r/en-US/ug440-xilinx-power-estimator, May 16, 2021.

[29] K. Schouhamer and L. Patrovics, "Performance assessment of dc-free multimode codes," *IEEE Transactions on Communications*, vol. 45, no. 3, pp. 293–299, 1997. DOI: 10.1109/26.558690.

[30] *Versal acap gty and gtyp transceivers architecture manual*, English, version Version 1.2, Xilinx. [Online]. Available: https://docs.xilinx.com/r/en-US/am002-versal-gty-transceivers, May , 2021.

[31] M. B. Marin, S. Baron, S. Feger, *et al.*, "The gbt-fpga core: Features and challenges," *Journal of Instrumentation*, vol. 10, no. 03, p. C03021, Mar. 2015. DOI: 10.1088/1748-0221/10/03/C03021. [Online]. Available: https://dx.doi.org/10.1088/1748-0221/10/03/C03021.

[32] M. Sánchez Net, "Optical ranging: Asynchronous-mode concept, prototype and validation," *The Interplanetary Network Progress Report*, vol. 42-232, pp. 1–18, Feb. 2023. [Online]. Available: https://ipnpr.jpl.nasa.gov/progress_report/42-232/42-232A.pdf.

[33] E. Mendes, S. Baron, D. M. Kolotouros, C. Soos, and F. Vasey, "The 10g ttc-pon: Challenges, solutions and performance," *Journal of Instrumentation*, vol. 12, pp. C02041–C02041, Feb. 2017. DOI: 10.1088/1748-0221/12/02/C02041.

A

Gaussian distribution characteristics of each individual trace

Table A.1: Gaussian distribution parameters

Trace No.	μ	σ^2
1	-1.267	0.060
2	-1.250	0.004
3	-1.275	0.068
4	-1.258	0.132
5	-1.257	0.013
6	-1.263	0.024
7	-1.255	0.011
8	-1.264	0.031
9	-1.256	0.012
10	-1.284	0.214
11	-1.253	0.005
12	-1.255	0.015
13	-1.272	0.082
14	-1.257	0.025
15	-1.261	0.040
16	-1.260	0.017
17	-1.263	0.013
18	-1.274	0.115
19	-1.269	0.037
20	-1.277	0.075
21	-1.267	0.168
22	-1.270	0.054
23	-1.264	0.174
24	-1.258	0.010
25	-1.265	0.016
26	-1.296	0.161
27	-1.276	0.250
28	-1.253	0.003
29	-1.273	0.233

Continued on next page

Table A.1 – continued from previous page

Trace No.	μ	σ^2
30	-1.254	0.002
31	-1.266	0.033
32	-1.264	0.119
33	-1.266	0.048
34	-1.259	0.003
35	-1.258	0.026
36	-1.251	0.005
37	-1.257	0.010
38	-1.261	0.018
39	-1.291	0.443
40	-1.254	0.002
41	-1.261	0.018
42	-1.281	0.132
43	-1.254	0.002
44	-1.274	0.052
45	-1.265	0.046
46	-1.265	0.080
47	-1.258	0.006
48	-1.269	0.150
49	-1.257	0.016
50	-1.266	0.014
51	-1.286	0.134
52	-1.266	0.123
53	-1.285	0.249
54	-1.268	0.085
55	-1.255	0.006
56	-1.272	0.205
57	-1.271	0.026
58	-1.258	0.010
59	-1.253	0.003
60	-1.273	0.049
61	-1.269	0.076
62	-1.256	0.006
63	-1.268	0.040
64	-1.253	0.003
65	-1.258	0.050
66	-1.281	0.085
67	-1.278	0.378
68	-1.275	0.100
69	-1.265	0.028
70	-1.251	0.002
71	-1.284	0.282
72	-1.257	0.007
73	-1.259	0.008
74	-1.275	0.062
75	-1.263	1.030
76	-1.254	0.005
77	-1.252	0.002

Continued on next page

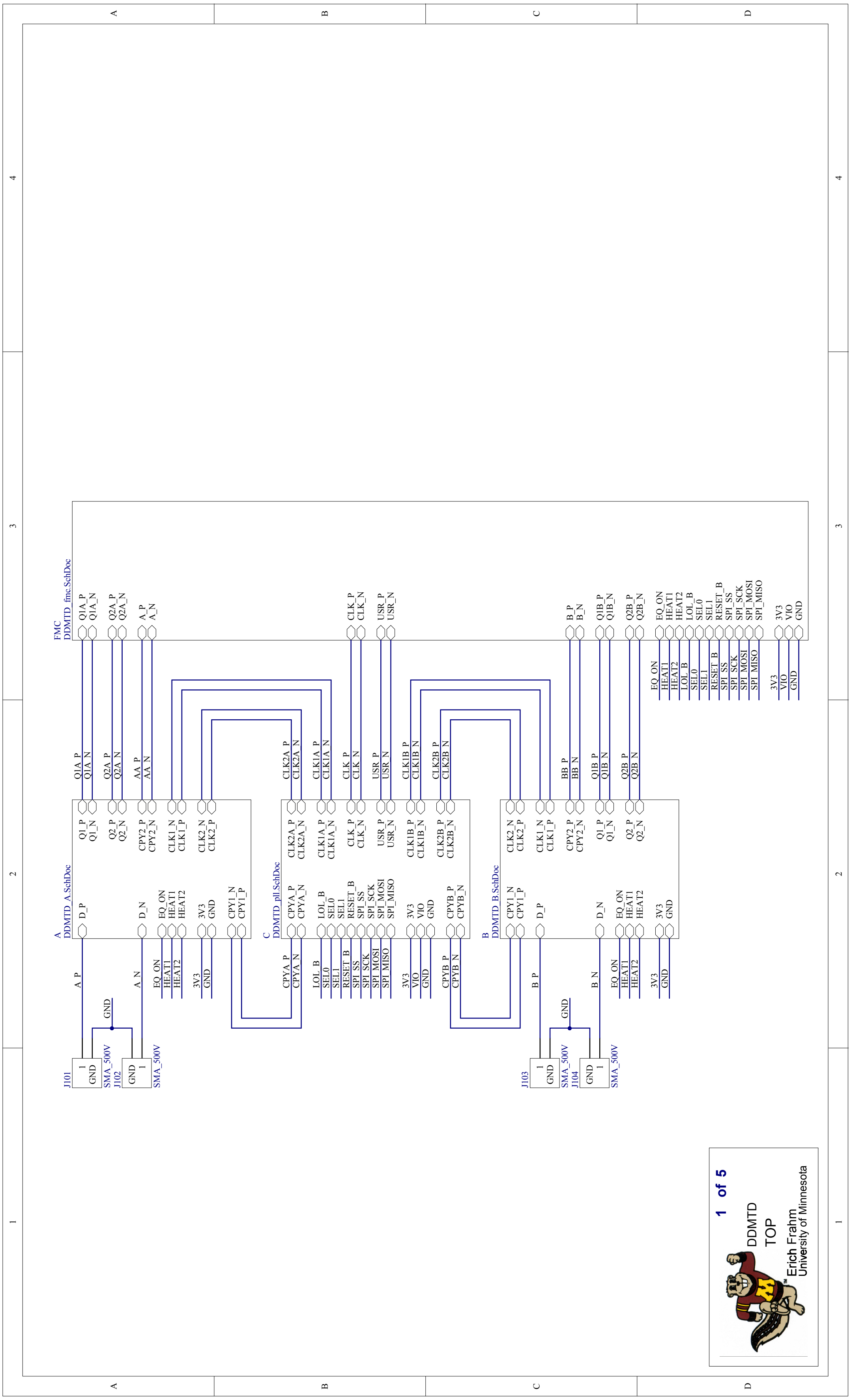
Table A.1 – continued from previous page

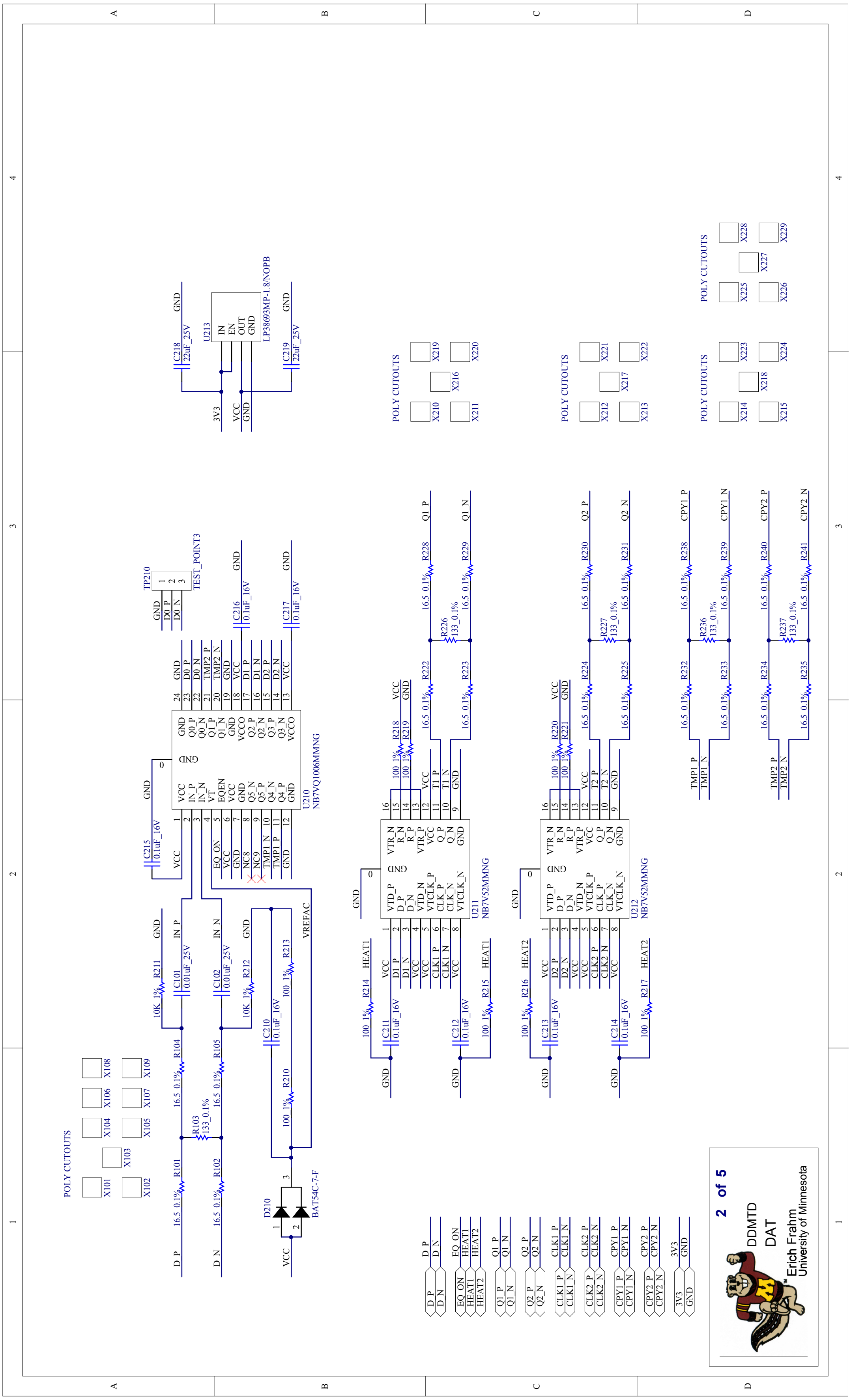
Trace No.	μ	σ^2
78	-1.265	0.013
79	-1.278	0.103
80	-1.267	0.030
81	-1.257	0.042
82	-1.265	0.037
83	-1.266	0.063
84	-1.274	0.276
85	-1.260	0.043
86	-1.257	0.054
87	-1.282	0.158
88	-1.272	0.107
89	-1.264	0.021
90	-1.263	0.102
91	-1.263	0.074
92	-1.293	0.301
93	-1.253	0.005
94	-1.261	0.021
95	-1.263	0.071
96	-1.280	0.222
97	-1.284	0.130
98	-1.264	0.014
99	-1.267	0.026
100	-1.261	0.024

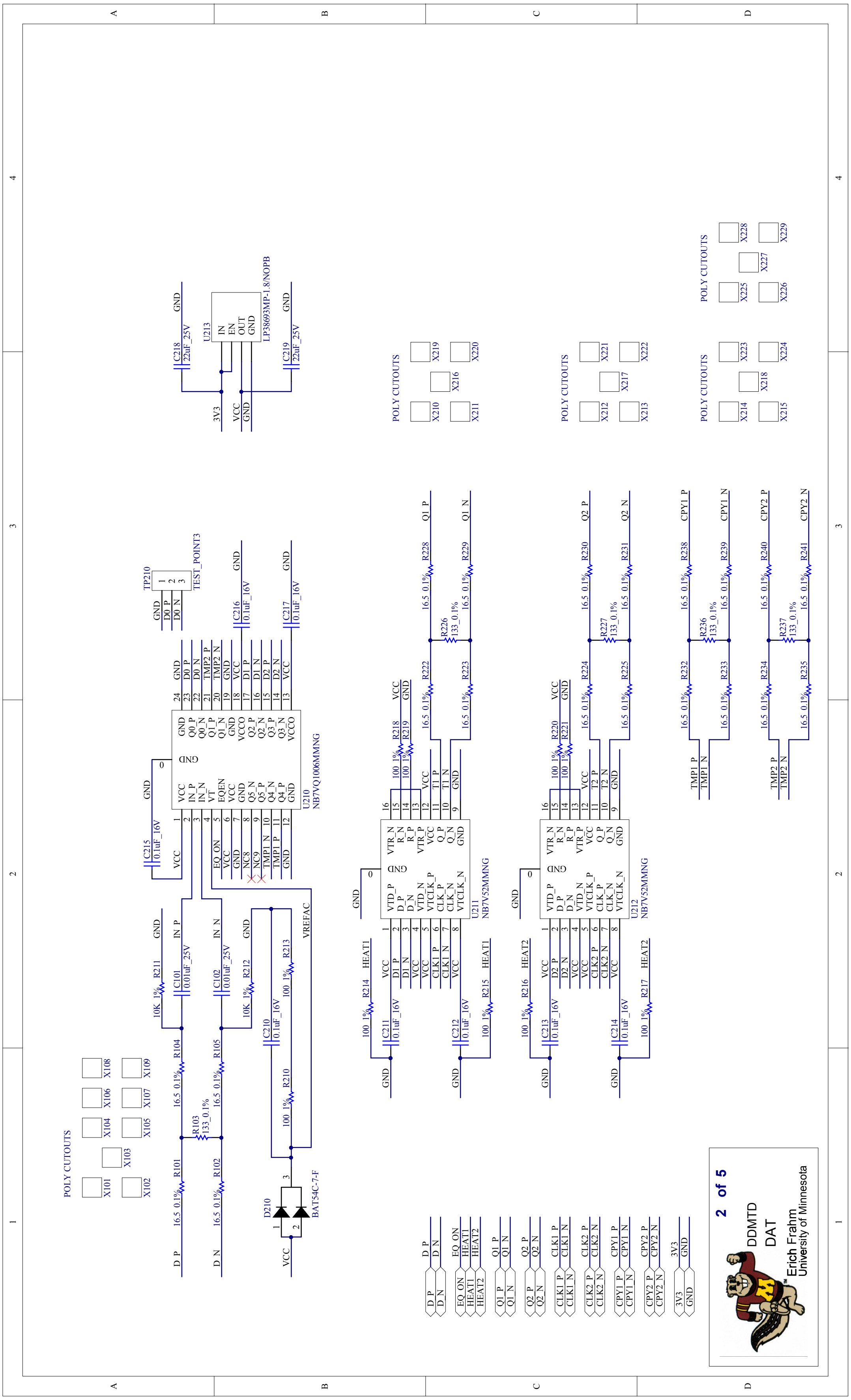
B

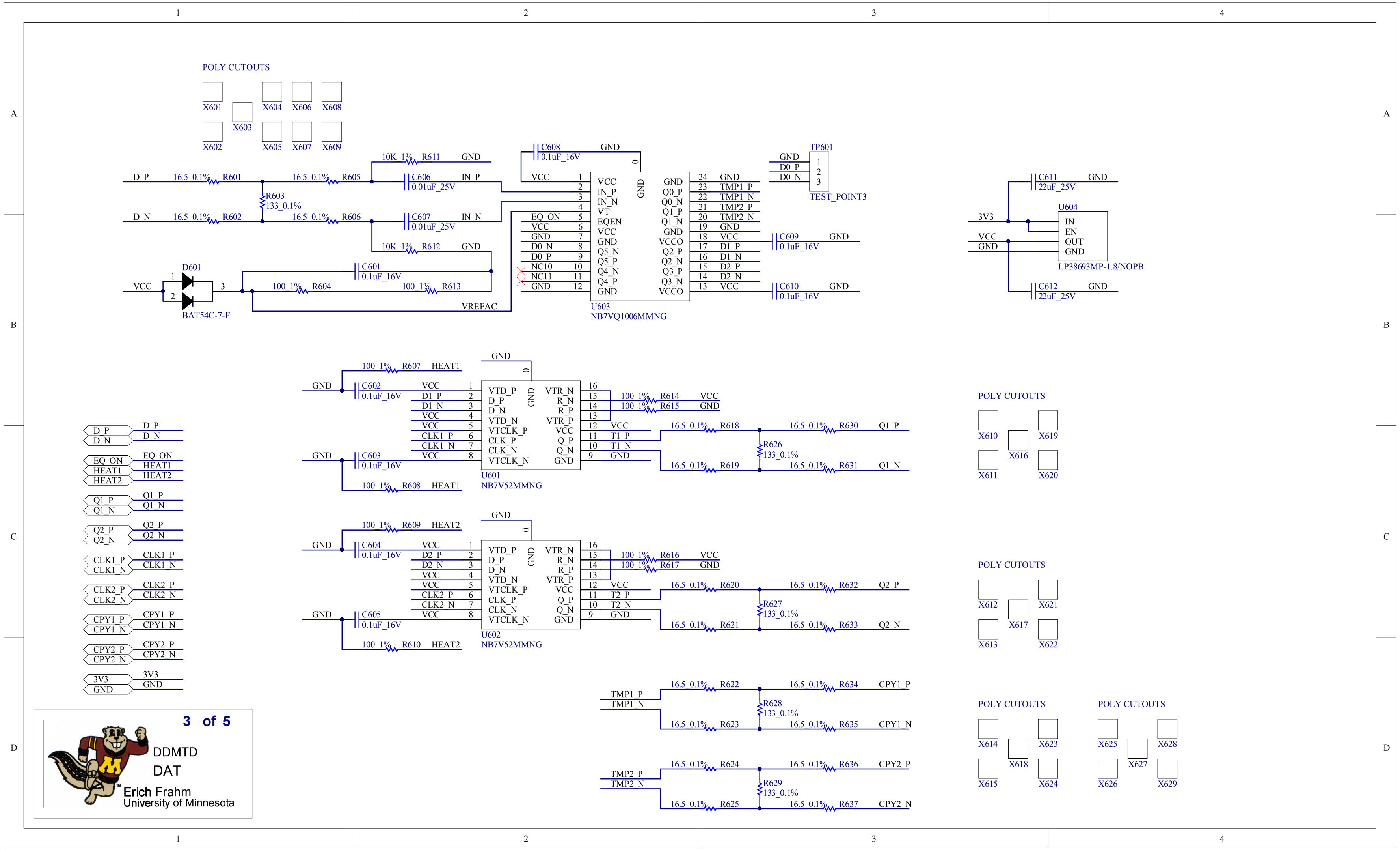
DDMTD Schematics

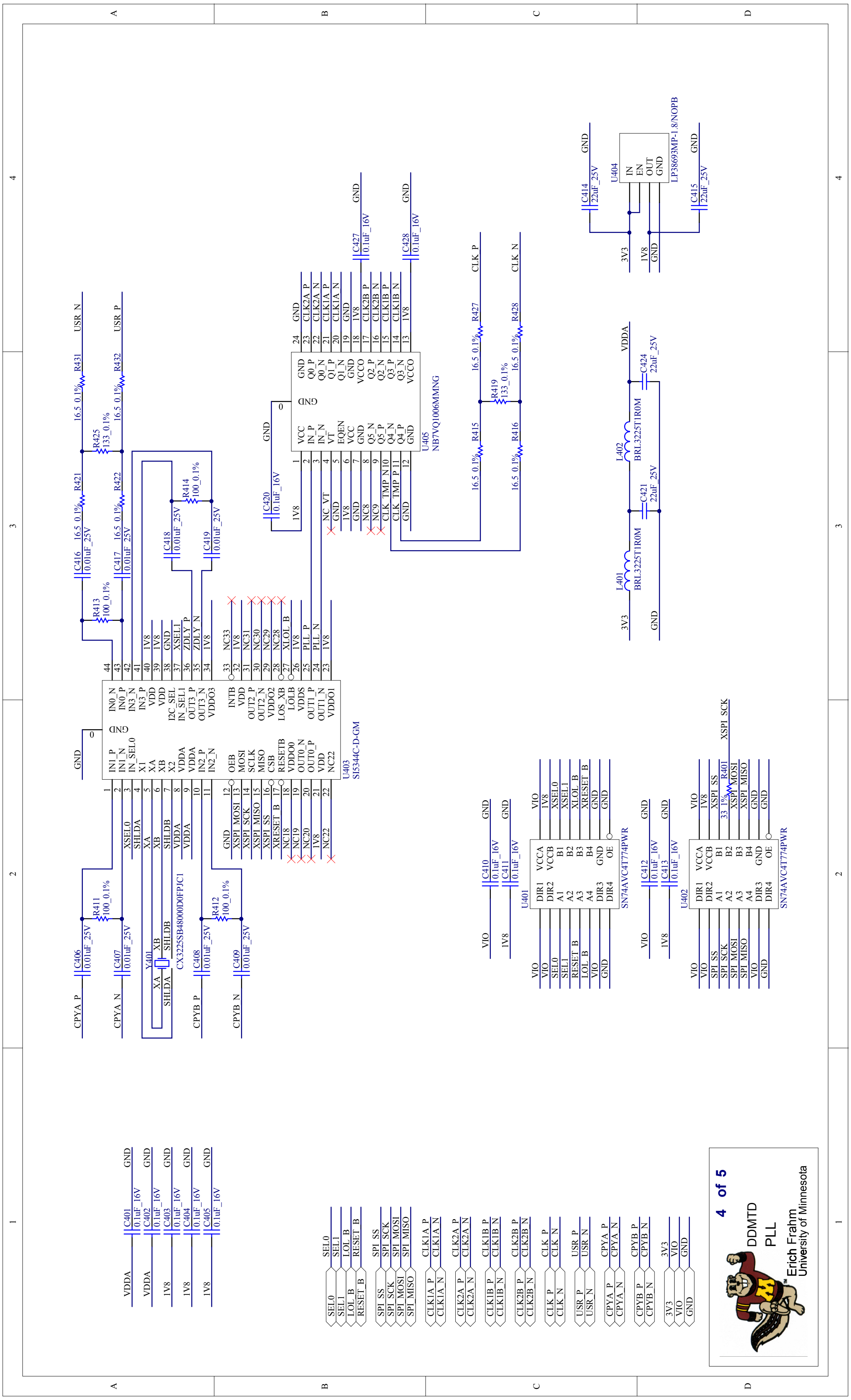
The DDMTD schematics are illustrated on the next pages.

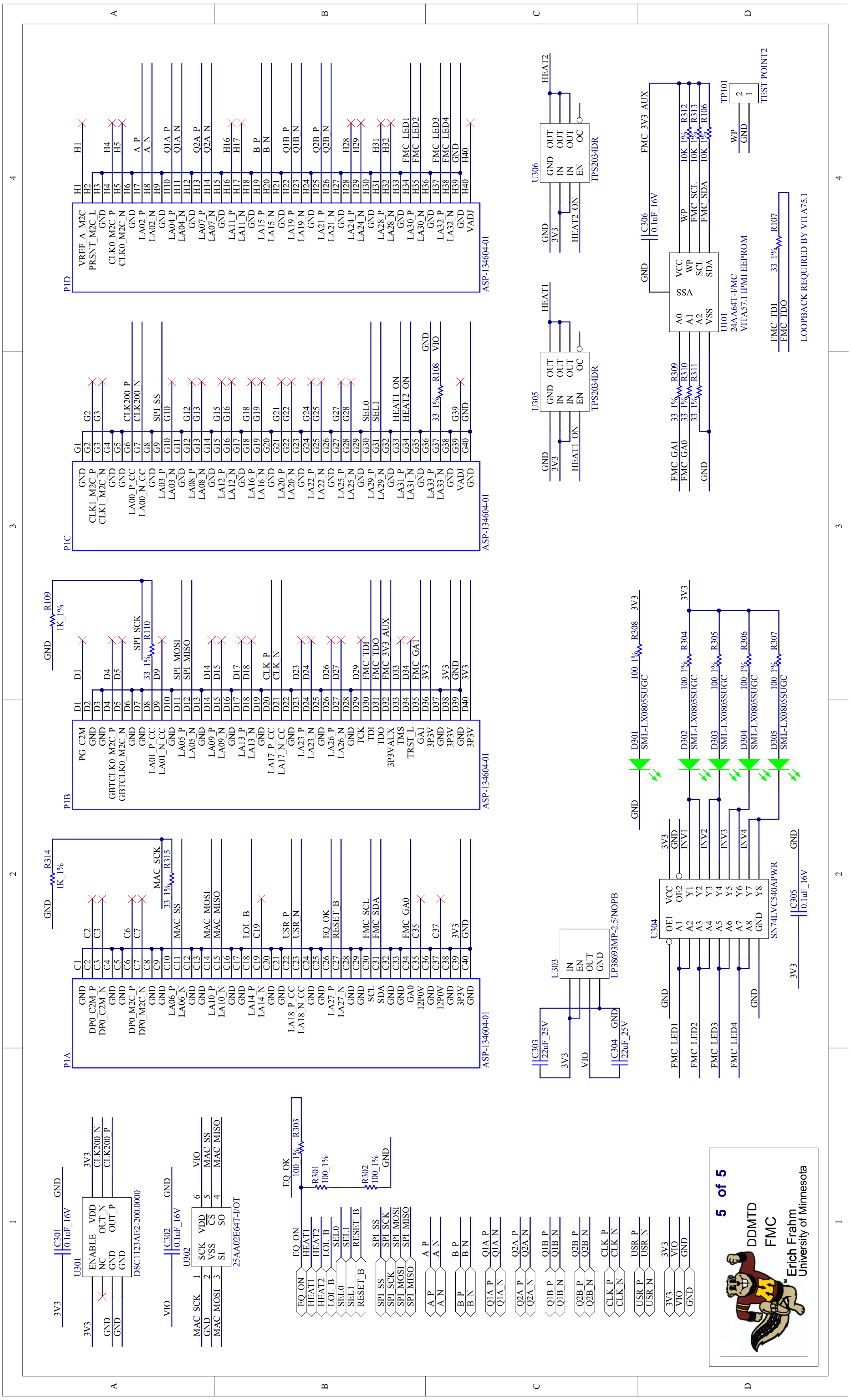

FMC
DDMTD_fmc.SchDoc
Q1A_P
Q1A_N
Q2A_P
Q2A_N
A_P
A_N
CLK_P
CLK_N
USR_P
USR_N
B_P
B_N
Q1B_P
Q1B_N
Q2B_P
Q2B_N
EQ_ON
HEAT1
HEAT2
LOL_B
SEL0
SEL1
RESET_B
SPI_SS
SPI_SCK
SPI_MOSI
SPI_MISO
3V3
VIO
GND
A
DDMTD_A.SchDoc
Q1_P
Q1_N
Q2_P
Q2_N
CPY2_P
CPY2_N
CLK1_N
CLK1_P
CLK2_N
CLK2_P
D_P
D_N
CPY1_N
CPY1_P
AA_P
AA_N
C
DDMTD_pll.SchDoc
CLK2A_P
CLK2A_N
CLK1A_P
CLK1A_N
CLK1B_P
CLK1B_N
CLK2B_P
CLK2B_N
CPYA_P
CPYA_N
CPYB_P
CPYB_N
B
DDMTD_B.SchDoc
BB_P
BB_N
J101
J102
J103
J104
SMA_500V
1
GND
1 of 5
DDMTD
TOP
Erich Frahm
University of Minnesota
A
B
C
D
1
2
3
4


2 of 5
DDMTD
DAT
Erich Frahm
University of Minnesota
POLY CUTOUTS
X101
X102
X103
X104
X105
X106
X107
X108
X109
D_P
D_N
R101
R102
16.5_0.1%
R103
133_0.1%
R104
R105
R210
100_1%
R211
10K_1%
R212
R213
C101
0.01uF_25V
C102
C210
0.1uF_16V
IN_P
IN_N
GND
VCC
VREFAC
D210
BAT54C-7-F
C215
U210
NB7VQ1006MMNG
EQ_ON
TMP1_N
TMP1_P
D0_P
D0_N
D1_P
D1_N
D2_P
D2_N
TMP2_P
TMP2_N
C216
C217
TP210
TEST_POINT3
U213
LP38693MP-1.8/NOPB
C218
C219
22uF_25V
3V3
U211
NB7V52MMNG
U212
HEAT1
HEAT2
CLK1_P
CLK1_N
CLK2_P
CLK2_N
R214
R215
R216
R217
R218
R219
R220
R221
R222
R223
R224
R225
R226
R227
R228
R229
R230
R231
R232
R233
R234
R235
R236
R237
R238
R239
R240
R241
C211
C212
C213
C214
Q1_P
Q1_N
Q2_P
Q2_N
CPY1_P
CPY1_N
CPY2_P
CPY2_N
X210
X211
X212
X213
X214
X215
X216
X217
X218
X219
X220
X221
X222
X223
X224
X225
X226
X227
X228
X229
A
B
C
D
1
2
3
4


2 of 5
DDMTD
DAT
Erich Frahm
University of Minnesota
U210
NB7VQ1006MMNG
U211
NB7V52MMNG
U212
NB7V52MMNG
U213
LP38693MP-1.8/NOPB
D210
BAT54C-7-F
TP210
TEST_POINT3
POLY CUTOUTS
VREFAC
3V3
VCC
GND
D_P
D_N
EQ_ON
HEAT1
HEAT2
Q1_P
Q1_N
Q2_P
Q2_N
CLK1_P
CLK1_N
CLK2_P
CLK2_N
CPY1_P
CPY1_N
CPY2_P
CPY2_N
TMP1_P
TMP1_N
TMP2_P
TMP2_N
C218 22uF_25V
C219 22uF_25V
C215 0.1uF_16V
C216 0.1uF_16V
C217 0.1uF_16V
C101 0.01uF_25V
C102 0.01uF_25V
C210 0.1uF_16V
C211 0.1uF_16V
C212 0.1uF_16V
C213 0.1uF_16V
C214 0.1uF_16V
A
B
C
D
1
2
3
4


POLY CUTOUTS
X601
X602
X603
X604
X605
X606
X607
X608
X609
D_P
D_N
16.5_0.1% R601
16.5_0.1% R602
R603 133_0.1%
16.5_0.1% R605
16.5_0.1% R606
10K_1% R611
10K_1% R612
GND
C606 0.01uF_25V
C607 0.01uF_25V
IN_P
IN_N
VCC
D601
BAT54C-7-F
C601 0.1uF_16V
100_1% R604
100_1% R613
VREFAC
C608 0.1uF_16V
U603
NB7VQ1006MMNG
VCC
IN_P
IN_N
VT
EQEN
Q5_N
Q5_P
Q4_N
Q4_P
GND
Q0_P
Q0_N
Q1_P
Q1_N
VCCO
Q2_P
Q2_N
Q3_P
Q3_N
EQ_ON
D0_N
D0_P
NC10
NC11
TMP1_P
TMP1_N
TMP2_P
TMP2_N
D1_P
D1_N
D2_P
D2_N
TP601
TEST_POINT3
C609 0.1uF_16V
C610 0.1uF_16V
3V3
U604
IN
EN
OUT
LP38693MP-1.8/NOPB
C611 22uF_25V
C612 22uF_25V
U601
NB7V52MMNG
U602
NB7V52MMNG
VTD_P
D_P
D_N
VTD_N
VTCLK_P
CLK_P
CLK_N
VTCLK_N
VTR_N
R_N
R_P
VTR_P
Q_P
Q_N
C602 0.1uF_16V
C603 0.1uF_16V
C604 0.1uF_16V
C605 0.1uF_16V
100_1% R607
100_1% R608
100_1% R609
100_1% R610
HEAT1
HEAT2
CLK1_P
CLK1_N
CLK2_P
CLK2_N
100_1% R614
100_1% R615
100_1% R616
100_1% R617
T1_P
T1_N
T2_P
T2_N
16.5_0.1% R618
16.5_0.1% R619
16.5_0.1% R620
16.5_0.1% R621
16.5_0.1% R622
16.5_0.1% R623
16.5_0.1% R624
16.5_0.1% R625
R626 133_0.1%
R627 133_0.1%
R628 133_0.1%
R629 133_0.1%
16.5_0.1% R630
16.5_0.1% R631
16.5_0.1% R632
16.5_0.1% R633
16.5_0.1% R634
16.5_0.1% R635
16.5_0.1% R636
16.5_0.1% R637
Q1_P
Q1_N
Q2_P
Q2_N
CPY1_P
CPY1_N
CPY2_P
CPY2_N
X610
X611
X612
X613
X614
X615
X616
X617
X618
X619
X620
X621
X622
X623
X624
X625
X626
X627
X628
X629
3 of 5
DDMTD
DAT
Erich Frahm
University of Minnesota


A
B
C
D
1
2
3
4
U403
SI5344C-D-GM
U405
NB7VQ1006MMNG
U401
SN74AVC4T774PWR
U402
SN74AVC4T774PWR
U404
LP38693MP-1.8/NOPB
Y401
CX3225SB48000D0FPJC1
L401
BRL3225T1R0M
L402
BRL3225T1R0M
C401 0.1uF_16V
C402 0.1uF_16V
C403 0.1uF_16V
C404 0.1uF_16V
C405 0.1uF_16V
C406 0.01uF_25V
C407 0.01uF_25V
C408 0.01uF_25V
C409 0.01uF_25V
C410 0.1uF_16V
C411 0.1uF_16V
C412 0.1uF_16V
C413 0.1uF_16V
C414 22uF_25V
C415 22uF_25V
C416 0.01uF_25V
C417 0.01uF_25V
C418 0.01uF_25V
C419 0.01uF_25V
C420 0.1uF_16V
C421 22uF_25V
C424 22uF_25V
C427 0.1uF_16V
C428 0.1uF_16V
R401 33_1%
R411 100_0.1%
R412 100_0.1%
R413 100_0.1%
R414 100_0.1%
R415 16.5_0.1%
R416 16.5_0.1%
R419 133_0.1%
R421 16.5_0.1%
R422 16.5_0.1%
R425 133_0.1%
R427 16.5_0.1%
R428 16.5_0.1%
R431 16.5_0.1%
R432 16.5_0.1%
VDDA
1V8
3V3
VIO
GND
CPYA_P
CPYA_N
CPYB_P
CPYB_N
USR_P
USR_N
CLK_P
CLK_N
PLL_P
PLL_N
XSEL0
XSEL1
XLOL_B
XRESET_B
XSPI_SS
XSPI_SCK
XSPI_MOSI
XSPI_MISO
SEL0
SEL1
LOL_B
RESET_B
SPI_SS
SPI_SCK
SPI_MOSI
SPI_MISO
CLK1A_P
CLK1A_N
CLK2A_P
CLK2A_N
CLK1B_P
CLK1B_N
CLK2B_P
CLK2B_N
ZDLY_P
ZDLY_N
SHLDA
SHLDB
XA
XB
IN1_P IN1_N IN_SEL0 X1 XA XB X2 VDDA VDDA IN2_P IN2_N OEB MOSI SCLK MISO CSB RESETB VDDO0 OUT0_N OUT0_P VDD NC22
IN0_N IN0_P IN3_N IN3_P VDD VDD I2C_SEL IN_SEL1 OUT3_P OUT3_N VDDO3 INTB VDD OUT2_P OUT2_N VDDO2 LOS_XB LOLB VDDS OUT1_P OUT1_N VDDO1
VCC IN_P IN_N VT EQEN VCC GND Q5_N Q5_P Q4_N Q4_P GND
GND Q0_P Q0_N Q1_P Q1_N GND VCCO Q2_P Q2_N Q3_P Q3_N VCCO
NC_VT CLK_TMP_N CLK_TMP_P
DIR1 DIR2 A1 A2 A3 A4 DIR3 DIR4 VCCA VCCB B1 B2 B3 B4 GND OE
IN EN OUT GND
4 of 5
DDMTD
PLL
Erich Frahm
University of Minnesota


5 of 5
DDMTD
FMC
Erich Frahm
University of Minnesota
ASP-134604-01
LOOPBACK REQUIRED BY VITA75.1
24AA64T-I/MC
VITA57.1 IPMI EEPROM
TPS2034DR
SN74LVC540APWR
LP38693MP-2.5/NOPB
DSC1123AE2-200.0000
25AA02E64T-I/OT

C

DDMTD PCB

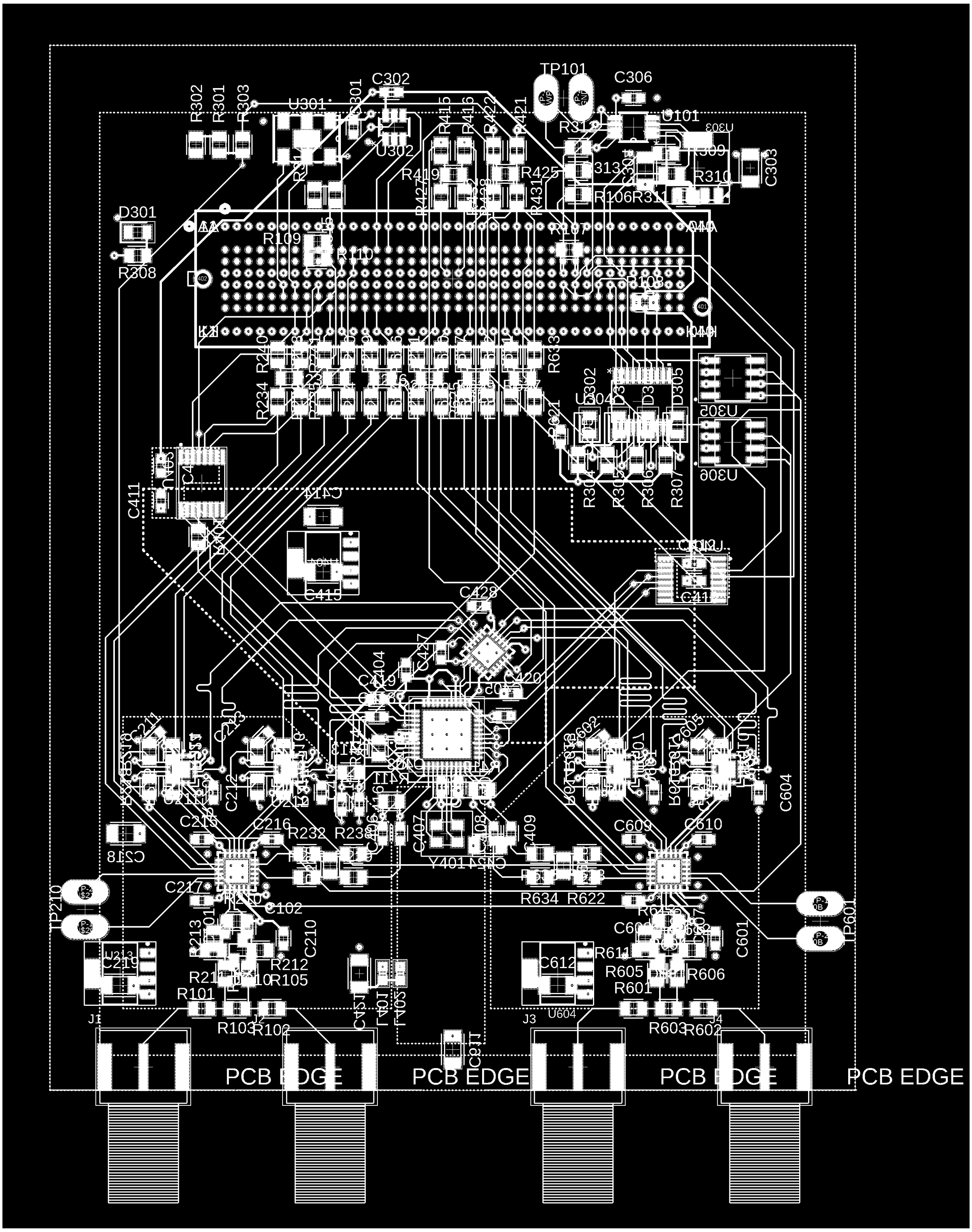


Figure C.1: DDMTD PCB layout

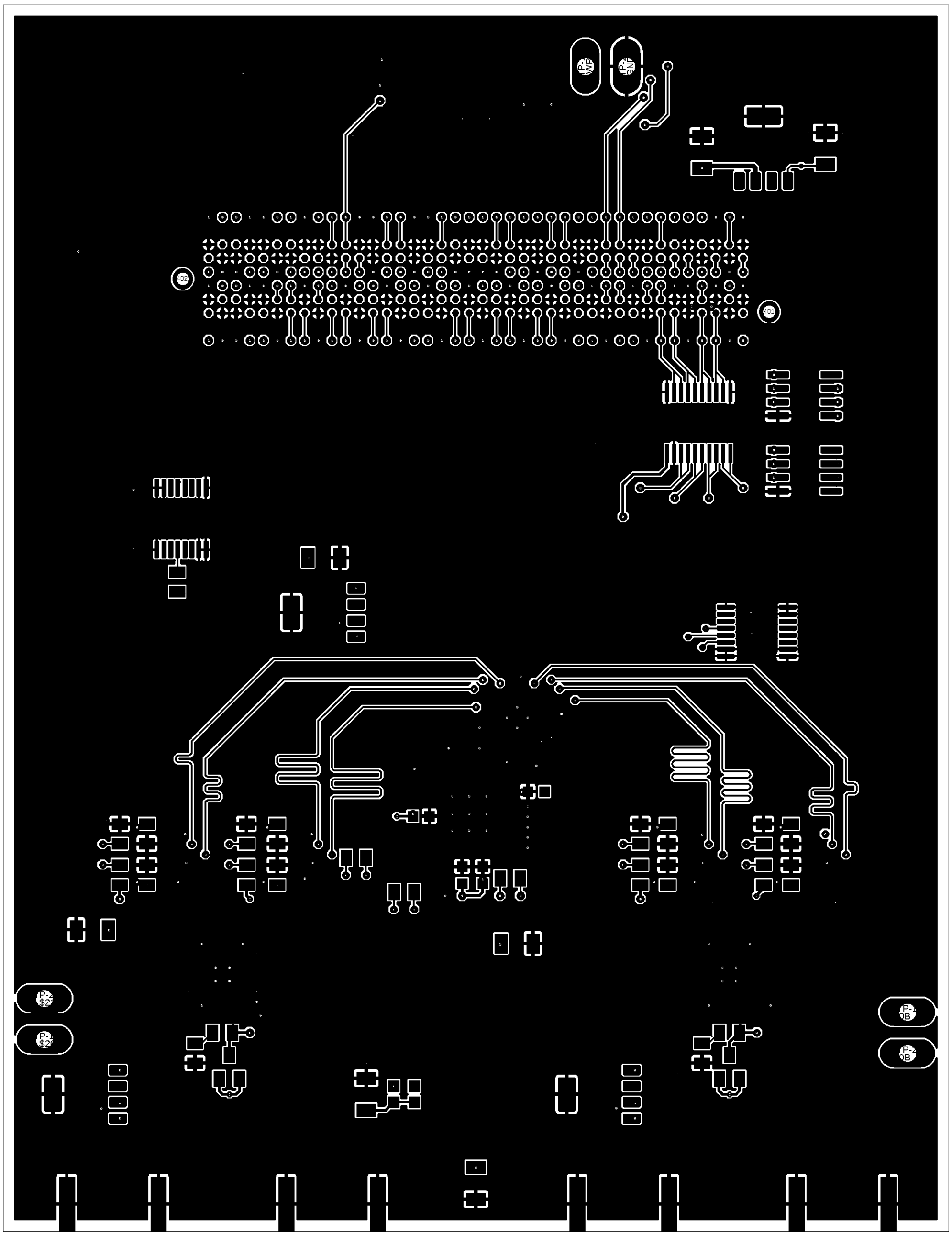


Figure C.2: DDMTD PCB layout