2018

DIGITAL PHASE LOCK LOOP DESIGN FOR LINEARIZATION OF CHIRPED LASER

Stefan Knipp
University of Rhode Island, sknipp@my.uri.edu

DIGITAL PHASE LOCK LOOP DESIGN FOR LINEARIZATION OF CHIRPED LASER

BY

STEFAN KNIPP

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND

2018

MASTER OF SCIENCE THESIS

OF

STEFAN KNIPP

APPROVED:

Thesis Committee:

Major Professor Tao Wei

Jien-Chung Lo

Otto J. Gregory

Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2018

ABSTRACT

The purpose of this study is to create an all digital solution for controlling the relatively cheap and small form-factor, all-semiconductor Distributed Feedback Lasers (DFBs). The objective is to generate a frequency sweep as linear as possible, which enables usage in applications like Coherent Optical Frequency Domain Reflectometry (C-OFDR). This is done utilize the input current dependency of the output frequency.

An Mach-Zehnder Interferometer (MZI) is used to measure the speed of frequency change. Next to that and a simple first order RC-filter, all components are implemented digitally using an Field Programmable Gate Array (FPGA) to allow for high adaptability as one of the many benefits of an all digital solution. XOR and Phase Frequency Detector (PFD) are evaluated as phase comparators.

Within this study, the XOR Phase Comparator (PC) together with a variable gain of the Loop Filter (LF) and a high capacitance RC-filter is found as the best solution.

ACKNOWLEDGMENTS

I would like to thank a number of people for enabling me to conduct this study here at the University of Rhode Island.

First, I would like to thank my major professor, Dr. Tao Wei, for his highly motivated and kind supervision and assistance. This inspired me and kept motivation up even during periods of struggle.

This is also true for the fellow students of the NextLab, particularly Zheyi Yao, whom I want to thank for friendly assistance and always being open to introduce me to the lab and all equipment when needed.

I would also like to thank my Master's Program Committee and the whole ECBE department.

Next to that, I want to say thanks to the exchange coordinators of both URI and TU Darmstadt for having me provided with the opportunity of this one year dual degree exchange program and great lifetime experience.

TABLE OF CONTENTS

ABSTRACT ii

ACKNOWLEDGMENTS iii

TABLE OF CONTENTS iv

LIST OF FIGURES vi

CHAPTER

1 Introduction 1

1.1 Laser modulation 1

1.2 PLL basics 3

2 Approach 4

2.1 General setup 5

2.2 Used Tools and Hardware 6

2.3 Implementation of basic PLL 14

3 Experiment 18

3.1 Iteration 1 19

3.2 Iteration 2 21

3.3 Iteration 3 27

3.4 Iteration 4 and implementation of gain change 29

4 Discussion 31

5 Conclusion / Future 35

LIST OF REFERENCES 38

BIBLIOGRAPHY 39

LIST OF FIGURES

Figure		Page
1.1	Operation principle of C-OFDR [1]	2
1.2	Block diagram of a PLL	3
2.1	Complete schematic of the setup	5
2.2	Schematic of an MZI	11
2.3	CD4046 output frequency over DAC code	15
2.4	CD4046 output signal compared to reference clock	17
3.1	Noisy MZI signal recorded with oscilloscope after PD	20
3.2	Signal captured from the $2m$ MZI	21
3.3	Linear laser driver control voltage rise with step in the middle, and FFT of the MZI output	22
3.4	PD output versus LD input	24
3.5	MZI output, regular start (left); with $120mV$ jump function in the beginning (right)	27
3.6	PD signal as seen by the FPGA after AD-conversion (top); with additional TTE 750kHz low-pass (bottom)	28
3.7	Output, using PFD together with $70\mu F$ laser and $3\mu F$ DAC-filter capacitance	28
3.8	4 points of threshold increase +5	30
4.1	Frequency output of MZI and phase error for $22\mu F$ DAC-filter capacitance	33
4.2	Phase error for $10\mu F$ DAC-filter capacitance	34
4.3	FFT of the MZI signal and phase error for $82\mu F$ laser capacitance	34

CHAPTER 1

Introduction

Lasers are more and more becoming part of everyday life, mostly used invisible to the end user. In the past decades, lasers have undergone a vast change in cost, availability and down-scaling in form factor. While for laboratory environments it is feasible to have cryogenic, shock-sensitive and large setups for high performance lasers, they mainly found their way into everyday life's applications by the availability of small, rigid, and low-cost semiconductor lasers.

One such emerging application with many use cases is Coherent Optical Frequency Domain Reflectometry (C-OFDR). This has certain requirements for the laser and its properties, that the laser can be tuned towards in order to get best results. One of them is a broad, continuous output frequency range of the laser. This is mostly defined by the physical capabilities of the used laser technology. Another one is that sweeps through those frequencies are as linear as possible. This can be influenced by finding smart ways to control the laser.

This work focuses on controlling comparatively cheap DFB lasers to reach that goal with as few as possible external analog components to reach a low-cost, digital solution.

1.1 Laser modulation

As stated above, C-OFDR is based on sweeps of the laser's output frequency. Those are called chirps, as the laser is repeatedly cycled being kept off or at low output power for a certain amount of time, and then swept in frequency. Two factors are important for spatial resolution. On the one hand, the maximum tunable range determines the width of the Fourier Transform of the reflected and received signal. On the other hand, a purely linear frequency sweep translates to an

even distribution of spatial points on the frequency axis of the Fourier Transform, thus not decreasing resolution by compressing it in some parts, and requiring computing intense post-processing compensation [1]. Figure 1.1 shows a basic C-OFDR setup.

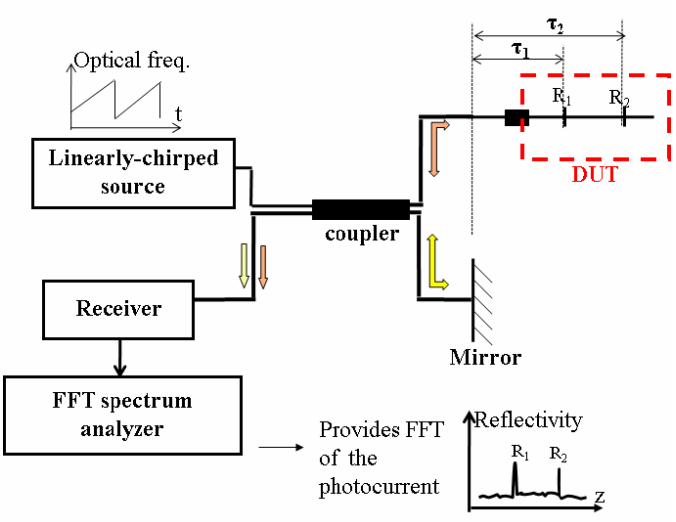

Figure 1.1. Operation principle of C-OFDR [1]

External Cavity Tunable Lasers (ECLs) are a good choice for a big tuning range of about $100nm$, but require external optical components and are therefore not as cheap and compact.

Distributed Feedback Lasers (DFBs) on the other hand can be pure semiconductor lasers. Tuning is accomplished by changing physical characteristics of the laser by modulating the temperature, which is highly determined by the driving current. [2] Driving current and output frequency are not linearly correlated though. Hence, a more complex current-control is needed than just a linear ramp signal.

This work tries to find a way to improve the linearity of frequency over time for a low cost, low energy and low scale, yet robust setup. Therefore this is done in a digital way, using only few external non-value-critical analog passive filters.

As described in section 2.1, the setup uses a Mach-Zehnder Interferometer (MZI) to measure the speed of frequency change of the laser. The output of this is ideally a sine wave of constant frequency, indicating a linear frequency change of the laser. So the goal is to regulate the laser current to achieve this output. This can be done using a Phase Lock Loop (PLL) to lock the MZI signal to a given reference clock.

1.2 PLL basics

A Phase Lock Loop (PLL), as the name suggests, tries to lock the phase difference between the input signal and a given reference signal to a constant value. This implies a perfect frequency lock. It consists of three elements, the Phase Comparator (PC), which compares the two input clocks and generates a phase error signal. This is fed into the Loop Filter (LF), which generates the appropriate control voltage to feed into the Voltage Controlled Oscillator (VCO).[3] The LF consists of zero or more integrator stages, defining the type of the PLL, and filter stages, defining the order of the PLL.

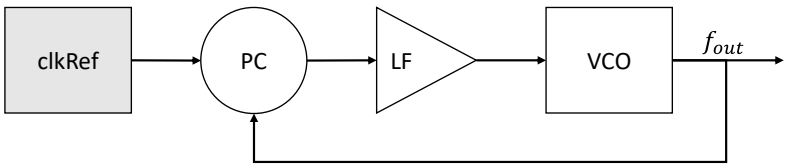

Figure 1.2. Block diagram of a PLL

An important characteristic of PLLs is the pull-in range [4], defining how far a initial frequency offset can be for the PLL to reach lock.

For digital PLLs, the two common types of PC are XOR and Phase Frequency Detector (PFD). While the PFD can pull-in to any frequency, the pull-in range of the XOR is very small.

CHAPTER 2

Approach

The goal of this work is to generate a laser frequency sweep as linear as possible. With a wavelength of $1550nm$, thus a frequency of $f = \frac{c}{\lambda} \approx 194THz$, this frequency is far from being converted to an electronic signal and measurable in exact and cheap ways. Thus it can not be used as a direct input to a PLL to be locked to a reference clock with the desired sweep characteristics.

For our purpose, there is no need to know the exact frequency though. Instead, the derivative of it, the change of frequency over time, is important. This can be extracted by using a MZI. The output of this is a light-wave modulated in intensity according to the change of frequency of the input. The frequency of this modulation is dependent on a) the change of frequency over time of the input wave, and b) of the design of the MZI itself and can be easily modified to be in a range of $100kHz$ or $1MHz$ for the desired frequency sweep. [5] So this can be converted to an electric signal using a photodetector and afterwards be digitized using Analog-to-Digital Converters (ADCs). This can be used then to lock to a reference clock using a PLL.

As I come from another background, I did not have any experience with PLLs when I started this work. So it was a good idea to start out familiarizing with the subject by building a basic PLL first, before getting into laser control involving all the optical components. This was also useful to get used to the tools and hardware used in this experiment, even though they partly changed and were expanded over time.

After the basic PLL was working, I continued with moving towards the final setup.

2.1 General setup

The setup consists of the DFB diode together with a voltage controlled current driver. This is controlled by an Field Programmable Gate Array (FPGA) with a $12bit$ Digital-to-Analog Converter (DAC) and $14bit$ ADC attached. The sampling rate of the ADC is $40MHz$, so there is enough headroom for the MZI target frequency of $500kHz$, resp. the former $1MHz$ or $100kHz$. All analog electrical components are interconnected using coaxial cables with SMA or BNC connectors and using 50Ω termination. The output of the laser is split using an 95/5 coupler. 95% of the signal are available to be used for the applications like C-OFDR. The 5%-branch is fed in the MZI whose output is connected to a photodetector. This translates the input light intensity back to an analog voltage which then gets sampled by the ADC. All optical components are interconnected using single mode optical fiber. The complete setup is depicted in figure 2.1.

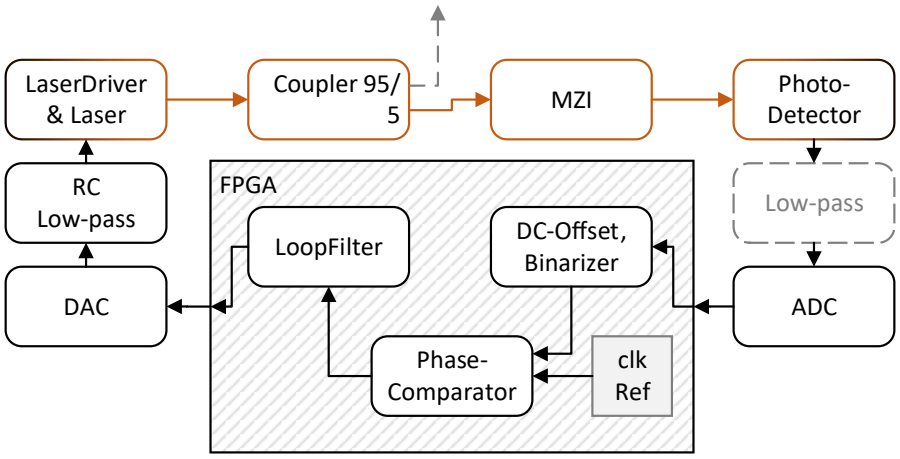

Figure 2.1. Complete schematic of the setup

The voltage regulated laser driver, together with the laser and followed by the MZI act together similar to a VCO. The difference is that a constant voltage will not output a certain frequency, but a voltage change will. The frequency is controlled by the derivative of the input voltage.

2.2 Used Tools and Hardware

The exact components are as follows.

Digital control logic

The main component of this PLL is the digital logic component that implements all of a PLL's functions phase comparison and loop filter. For the digital control logic, a FPGA is used. This was chosen over a microcontroller, as it is unlikely to reach the performance limitations as all tasks can run in parallel, versus a purely sequential signal analysis, processing and control with a microcontroller. Verilog was used as Hardware Description Language (HDL).

Because of already being used to the *Xilinx* environment with Integrated Development Environments (IDEs) like *ISE* and *Vivado*, and having powerful Integrated Circuits (ICs) in their repertoire, a development board with a Xilinx Zynq-7000 series System on Chip (SoC) was prospected for. The Zynq series of devices have the advantage of also having an *ARM* processing core on-chip, next to the FPGA part. This opens up even more possibilities, combining the strengths of both FPGA and microcontroller.

Next to that, the requirement for the development board was to be easy to extend with peripheral boards for A/D and D/A conversion. There are two well established standards, next to one new one. The well established being the *Samtec FMC* connectors, which provide high performance and high pin counts (20–200 I/Os). They are pretty expensive though. Another standard, that is open, is the *Digilent Pmod* interface. This features low pin counts (4–8 I/Os) for very low prices though. As there is a huge gap in between, the rather new OpalKelly Syzygy open standard is evolving. This features a in between pin count (10–32 I/Os) of performance capable interconnection for an average price.[1]

[1] http://syzygyfpga.io

This lead to the choice of the *OpalKelly brain1* FPGA board. This has 4 Syzygy connectors. There is an extension board available which translates a Syzygy port to Pmod connectors, so other periphery can be used. Xilinx Zynq 7012S is used as FPGA, next to 1GB DRAM, SD-Card slot, USB-C and Ethernet connectivity.

Laser module and laser driver

As a laser source, the following DFB diode module is used.

Alcatel A 1905 LMI

Part No.	3CN 00462 DZ
Wavelength	1550nm
Output power	$20mW$
Threshold current max.	$40mA$
Absolute max current	$350mA$
Spectral width	typ: $2MHz$, max: $5MHz$

The output power of $20mW$ is reached at an input current of $190mA$. A Thermoelectric Cooler (TEC) element is included in the case and can be used, together with the integrated Thermistor, to keep the diode at a constant temperature.

In this work, the laser is powered up to a peak current of 240mA. This can be done without damaging the diode, because of the short periods of time in which the current ramps up to the full load, followed by a time with minimum current.

As a laser driver, ***Koheron CTL100-A-400*** [2] was used. This features both DC and AC coupled current control voltage inputs with a crossover frequency of $100kHz$. There are different gain factors for the DC and AC inputs, which

[2] https://www.koheron.com/photonics/ctl100-laser-controller

complicates transferring eventually arising DC components on the AC side to the DC input. Gain factor for the DC side can be set to 2, 20, or $200\frac{mA}{V}$. The aimed at sweep duration is in the range of $1ms$ to $10ms$. Those factors makes the AC input both hard to use, but also not necessary for this use case. The laser diode itself is mounted on a big aluminum base plate used for dissipating the heat from the laser's integrated TEC element, which is also regulated by the *CTL100* to keep the laser diode's temperature constant.

Photodetector

Koheron PD200T [3] is used as a photodetector to convert the MZI output back to an electrical signal. It also provides an adjustable Schmitt-triggered digital, binary TTL output. Due to the increasing output power of the laser within each cycle though, only the analog output can used and directly sampled by the ADC. This way, no analog Automatic Gain Ccontrol (AGC) circuitry is needed to keep the signal level at constant amplitude. Binarization is done in the FPGA after the signal passed AC coupling and undergone additional, digital DC component removal.

DAC and ADC modules

As a necessity for digital control of analog systems, both DAC and ADC modules had to be used for interfacing the analog components.

[3] https://www.koheron.com/photonics/pd200t-photodetection

	DAC	ADC
IC	TI *DAC121S101*	AD *LTC2264-12*
Aboard of	*PmodDA2*	*POD-ADC-LTC2264x*
Resolution	$\frac{2^{12bit}}{3.3V}$	$\frac{2^{12bit}}{2V}$
Sample rate	$\frac{40MSa}{s}$	variable, up to $\frac{16.5MSa}{s}$
Interface	$160Mhz$ DDR $2bit$-serial LVDS	$31.25MHz$ SPI

The Digital-to-Analog Converter module ***Digilent PmodDA2*** [4] was used for the training PLL , as it is an official module for the Digilent ZedBoard, using the open standard Pmod interface, thus good availability, price and specifications that fit this application.

The two *Texas Instruments DAC121S101* ICs, only one of them is used, are interfaced using the Serial Peripheral Interface (SPI) bus. Maximum frequency is specified as $30MHz$. In the main experiment they are used with $31.25MHz$ which was chosen although slightly over specifications for the following reasons: First of all, the data transfer between FPGA and DAC should happen as fast as possible, resulting in a minimal delay. With the ZedBoard, the master clock of $100MHz$ was divided by 4 to get 25MHz SPI clock. When migrating to the brain1 platform, the master clock changed to $125MHz$. This could have been compensated with increasing the clock divider. As this has to be an even number, the next choice of 6 would have lead to a clock speed of 20MHz, which is only two thirds of the specified frequency.

The output settling time is specified as approximately $10\mu s$ aka 100kHz between each fully settled sample. Measured combined transfer plus settling time is only $2.3\mu s$, though. This consists of $0.8\mu s$ time from initiating a transfer until the output voltage starts to change, and an additional $1.5\mu s$ for the output voltage to

[4] https://reference.digilentinc.com/reference/pmod/pmodda2/start

settle after a change of almost half the output range, which was measured from code `0x800` to `0xF00` with the training setup, using $25MHz$ clock speed. Because the DAC is used in a constant rising slope only, the settling time and the slew rate of $1\frac{V}{\mu s}$ were considered fast enough to satisfy the needs for this application. As the laser driver is 50Ω terminated, the $3.3V$ output range of the *PmodDA2* are divided down by the factor of 2, basically increasing resolution by a factor of 2.

As a possible addition, a more high speed DAC was bought. This was when transitioning to the brain1 platform, so the OpalKelly *POD-DAC-AD9116* with Syzygy Connector was chosen, making use of an *Analog Devices AD9116* $12bit$ DAC. Specifications are $125MSPS$ and $11.5ns$ settling time. As this is mostly designed for Radio Frequency (RF) purposes and thus is AC coupled, though, it is hard to use it for the reasons stated in the laser driver paragraph.

The ADC is a peripheral board and part of the OpalKelly brain1 demo repertoire. The *POD-ADC-LTC2264x* consists of a *Linear Technology LTC2264-12* $12bit$, $40MSPS$ ADC. The inputs are transformer coupled and high-pass filtered with high enough cut-off frequency, so DC components resulting from imperfect MZI characteristics are sufficiently filtered for this application. The *LTC2264-12* is connected to the FPGA via a $2bit$ $160Mhz$ Double Data Rate (DDR) serial Low-Voltage Differential Signaling (LVDS) interface, so each of the $16bit$ frames can be transmitted to the FPGA in real time. The analog-digital conversion itself is done pipelined with a latency of 6 clock cycles. This makes, together with transmission, up for a total delay of $(6+1) * \frac{1}{40MHz} = 175ns$.

Optical components and Mach-Zehnder Interferometer

First component after the light exits the laser is an optical isolator to prevent light reflections from splices to feed back into the laser, decreasing quality of the

output.

After that, the light passes the 10% side of a 10/90 coupler, where the 90% are separated for the actual application.

The MZI is the optical component enabling to measure the laser's sweep velocity, which is expressed by the MZI's optical intensity output frequency. Built from 2 optical fiber branches with different lengths, one branch acts as a delay element for the light wave due to a longer path. When the two light waves are brought back together, they differ in phase and frequency and interfere with each other. This causes a resulting light intensity of sinusoidal shape. The frequency is dependent on how much the input frequency changed while the light traveled through the delay-branch. So it is dependent on the length of the delay τ, and the laser's sweep velocity.

$$\tau = \frac{\Delta d}{c} = \frac{\Delta d}{200000 \frac{km}{s}} \tag{2.1}$$

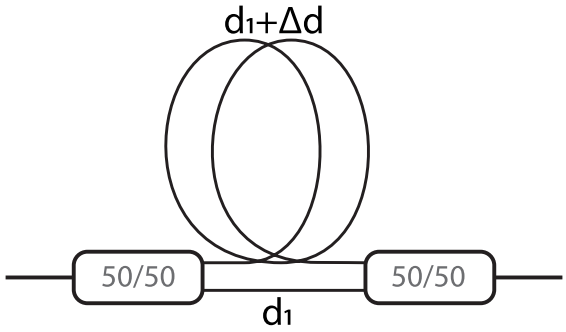

Figure 2.2. Schematic of an MZI with common branch length d_1 and delay length Δd

The laser frequency is given by

$$\omega(t) = \omega_0 + \xi t \tag{2.2}$$

with ξ as the sweep's slope. Phase is given by

$$\Phi(t) = \Phi_0 + \omega_0 t + \frac{1}{2}\xi t^2 \tag{2.3}$$

[6]

After splitting the light beam in the two branches of the MZI, we get

$$\begin{aligned} E_1 &= E_0 * \cos\big(\Phi(t)\big) \\ E_2 &= E_0 * \cos\big(\Phi(t-\tau)\big) \end{aligned} \tag{2.4}$$

where E_1 is the electric field at the end of the short branch and E_2 the electric field at the end of the longer delay-branch, where both are brought together. The constant factor of 0.5 from the $3dB$ input coupler is neglected here, as it is the same for both branches and changes nothing but the scale of amplitude. The common length between short and long branch is ideally infinitesimally short and, as it influences the phase at the end of both branches in the same way, can be neglected.

$$\begin{aligned} \mathbf{i_{PD}} \div \kappa &= |E|^2 \\ &= (E_1 + E_2)^2 = E_1^2 + E_2^2 + 2E_1E_2 \end{aligned} \tag{2.5}$$

with κ as the Photodetector (PD)'s responsivity.

$$\begin{aligned} \mathbf{i_{PD}} \div \kappa = E_0^2 \frac{1 + \cos\big(2\Phi(t)\big)}{2} + E_0^2 \frac{1 + \cos\big(2\Phi(t-\tau)\big)}{2} \\ + 2E_0^2 \cos\big(\Phi(t)\big) * \cos\big(\Phi(t-\tau)\big) \end{aligned} \tag{2.6}$$

Because the $\cos(2\Phi(t-\tau))$ frequency components are off limits to be detected by the PD, they average out to 0. This leads to

$$\mathbf{i_{PD}} \div \kappa = E_0^2 + E_0^2 \Big[\cos\big(\Phi(t) - \Phi(t-\tau)\big) + \cos\big(\Phi(t) + \Phi(t-\tau)\big) \Big] \tag{2.7}$$

$$\mathbf{i_{PD}} = \kappa * \cos\big(\Phi(t) - \Phi(t-\tau)\big) \tag{2.8}$$

All components are connected using fiber optic connectors for easy replacement of single parts of the setup. For example using another coupler ratio for directing more optical power to the photo detector can be done easily by replacing just the 10/90 coupler.

Other lab equipment and tools

Next to all the above mentioned experiment setup, standard laboratory equipment was used.

A digital storage oscilloscope was used for getting real time insights for debugging purposes. Also the Fast Fourier Transform (FFT) function was useful for getting a first impression of MZI output frequencies and locking quality.

For more in-depth analysis of the results, a *pico Technologies PicoScope* PC oscilloscope was used. Because of the big memory, this allows to record complete sweep cycles and transfer them to a computer for analysis using *MathWorks Matlab*.

For debugging of the FPGA logic, *Xilinx ILA* was used, which is accessed via *JTAG* from within *Vivado*.

For first tests of the laser and optical setup, an arbitrary waveform generator was used, together with a fiber optic power meter. Using this, one could verify that all used components and optical splices met the requirements.

Training PLL components

The components used in the preliminary PLL experiment, described in section 2.3, were as follows.

As the FPGA development board, Digilent ZedBoard was used due to its great extendability via both FMC and many Pmod connectors. This also uses a SoC of the Xilinx Zynq-7000 series. It also contains components useful for runtime reconfiguration and debugging, as LEDs, buttons and switches.

To replace the optical setup, the CD4046[5] VCO was used. As this generates a digital clock signal, no analog to digital conversion was needed in this experiment. To generate the analog control voltage, the same DAC as in the main experiment was used.

2.3 Implementation of basic PLL

Coming from a more digital background, I chose to get used to PLLs in general at first, without involving the optical setup.

After first research, I chose the *CD4046* IC as a VCO, and the $100kHz$ version of the *Si5xx* series *Silicon Labs* Voltage Controlled Crystal Oscillator (VCXO) options. After measurements with the VCXO, I decided that the measured tuning range of only $200ppm$ over the whole input voltage range would be too small for this experiment, though, so the VCO was chosen.

The next step was to choose a target frequency and select external components accordingly. I chose 20kHz as frequency, as this was well within the specifications of the CD4046. Values could only be roughly estimated, as none of the datasheets found provided exact diagrams or tables displaying the correlation of capacitors and resistors to frequency and tuning range. After assembling the components

[5] http://www.ti.com/product/CD4046B , 2018-06-24

on a breadboard and 2 iterations of updating the capacitor value, the targeted frequency was within tuning range.

Afterwards, I plotted to output frequency over all control voltage input values. This can be seen in figure 2.3. Not that the x-axis depicts the codes of the $12bit$ over $3.3V$ DAC. The correlation is not very linear. Almost all of the output frequency range is spread over only the upper half of input voltage range.

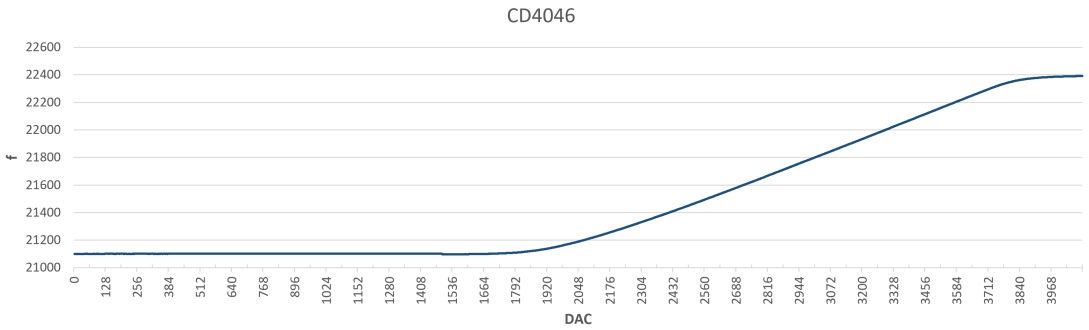

Figure 2.3. CD4046 output frequency over DAC code

The next step was to set the FPGA and DAC part up. The DAC module, as part of the *Digilent* repertoire, already came with a Verilog demo application, so this needed only minor changes to be used here. The main effort here was clearly to elaborate the PLL's architecture.

After some research in PLLs and their main components, phase comparator and loop filter, I had to choose which architecture I wanted to go with.

There are two main ways of doing phase comparison. This is using an XOR gate to combine reference and external clock, or using a PFD , each with its own advantages and disadvantages. Constant differences in phase offset are completely irrelevant throughout this work.

The XOR PC is a simple XOR-logic between reference clock and external

clock. This means that the output is constantly 0 for signals that match phase and frequency. For signals that are the opposite of each other, e.g. 180° phase difference, the output is constantly 1. It is task of the LF to assess this output and control the VCO accordingly. Usually one tries to hold the high-time of the XOR-output at 50%. This gives the maximum margin towards the extrema of 0% and 100% and allows to judge whether to speed up or slow the VCO down.

The PFD consists of 2 D-type FlipFlops and an AND-gate [7]. As the name suggests, it compares both phase and frequency of the two inputs. This makes it a good candidate for clocks with very different frequencies in the beginning. The pull-in range is infinite. In contrast to that, the output of the XOR can only be interpreted as to whether the average high-time is 50%. This is also the case for double and half frequencies e.g. . The output of the PFD is a up and a down signal indicating the needed frequency change for the VCO. The LF can be very simple.

As the XOR has both rising and falling edges as reference points, the output is changing for clocks that have a pulse width other than 50%. As there are twice as many reference points, a XOR-based PLL can react faster to aberrations of the VCO frequency and therefore keep the phase noise lower. This is depending on the performance of the LF though.

Because of the duty-cycle, which is not quite 50% for the CD4046, and initial problems finding lock using the XOR PC, I decided to go with the PFD. Similar to the work by *A. M. Fahim* [8], a fast start sequence for faster initial pull-in was implemented. At startup, the DAC value is modified by ±128, until the phase error is less than 1% of the period time, e.g. $500ns$ for the $20kHz$ reference clock.

Figure 2.4 shows the jitter of about $50ns$ for the $50000ns$ clock period.

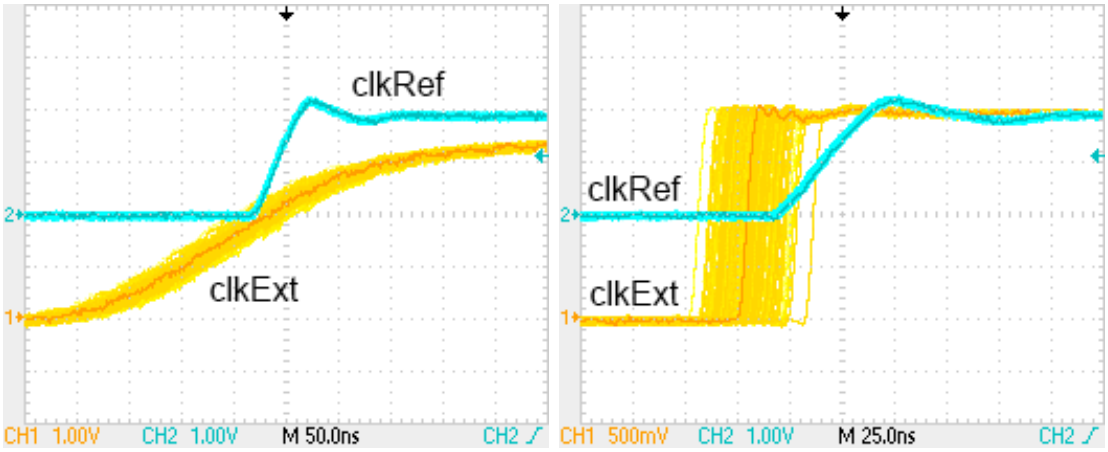

Figure 2.4. CD4046 output signal (orange) compared to reference clock (blue). As seen and output by FPGA on the right.

CHAPTER 3

Experiment

Starting point for the main experiment was the setup described in chapter 2 and the work done in section 2.3.

The CD4049 part was replaced by the optical setup. Therefore the MZI had to be designed. Work by *Zhen Chen et al.* reported an MZI frequency of $150kHz$ for a delay-branch length difference of $2m$ and a sweep time of a similar DFB diode of $9ms$ [5]. This leads to $9ms * 150kHz = 1350$ cycles of the sine wave resulting in 1350 reference points for an PFD PC resp. 2700 reference points for an XOR PC. I decided to try to go for higher frequencies in order to have more reference points for the PLL. In order to reach that, the MZI was designed with a delay branch length of 20m more than the reference branch, which should result in a 10 times higher output frequency than for the 2m MZI.

After some training splices, the MZI was built and all other components were spliced with connectors and the connected.

As the DAC module was already used in the previous experiment, only the ADC control needed to be taken care of now. As this module was part of the OpalKelly open source brain1 environment, I was able to use the demo application as a good reference for that. Notice that the sampling rate of the ADC is $40MHz$, whereas the FPGA runs off a master clock of $125MHz$. This means a new sample is ready approximately every 3rd clock cycle.

Next was adapting the output of the DAC to match the 50Ω input resistance of the laser driver's DC input. Therefore a 49Ω resistor was inserted between DAC and the SMA connector. This leads to a resistive voltage divider of factor 2, so the maximum voltage for the DAC running off a $3.3V$ supply is $1.65V$. Looking at

the gain options of the laser driver, the only viable option was the $200\frac{mA}{V}$ one, to reach the specified current of $190mA$. Within the Verilog code, I set 3000 as the maximum code for the 12bit DAC, which leads to a laser current of:

$$\mathbf{I_L} = \frac{3000}{4096} * \frac{3.3V}{2} * 200\frac{mA}{V} = 242mA \tag{3.1}$$

To always keep the laser above threshold current, the minimum code, as used during dead time, was set to 500 for $40mA$. This gives $3000 - 500 = 2500$ DAC steps for the sweep. On average, or assuming a $10ms$ purely linear rise, this gives an update interval, resp. sample rate of:

$$\frac{10ms}{2500} = 4\mu s \Rightarrow 250kHz \tag{3.2}$$

This is over the measured $2.5\mu s$ half range data transfer and output settling time of the DAC and so confirms the previous assumption of providing adequate performance.

Now was the time to test all components. The ADC was tested using a signal generator. The optical setup was tested using signal generator and measuring the optical output power after each component using a fiber optic power meter and comparing to mathematically derived values.

3.1 Iteration 1

I decided to start out sticking to the PFD implementation used in section 2.3. Because a rising control voltage is needed for the optical setup, in contrast to a constant DC offset voltage for the CD4046, the first idea was to accomplish that by creating a sawtooth Verilog module. This creates the 12bit DAC value by linearly incrementing the output by 1 every $4\mu s$, as calculated before. To reach phase lock now, this uses the PFD outputs up and down to decrease

resp. increase that cycle time and so adapt to the needs of laser current inclination.

Evaluating the recorded ADC output data showed that this is far from a working solution using this setup though. As can be seen in figure 3.1, the PD output is very noisy with higher frequency components of large amplitude, making it impossible to extract the main frequency component and binarize noise-free. This lead to malfunction of the PFD.

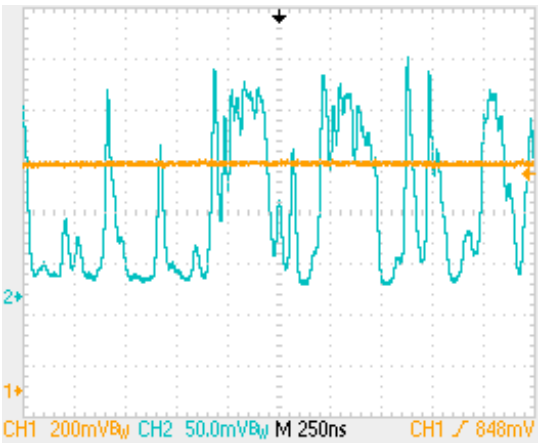

Figure 3.1. Noisy MZI signal recorded with oscilloscope after PD

To get a clue about whether there was a flaw in the whole setup, a proven 2m MZI was used in between to verify that the measurements were solid. This did, indeed, lead to much better results, as seen in 3.2. No phase lock was reached in this short try before going back to the 20m MZI, but all components other than the MZI were proven to work as expected.

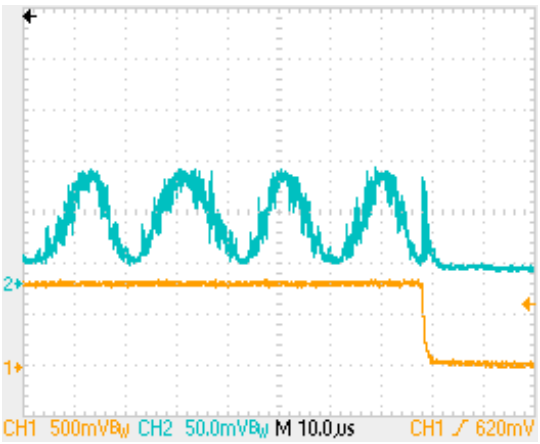

Figure 3.2. Signal captured from the $2m$ MZI

3.2 Iteration 2

After this first run and observations, the causes for the bad outcome with the 20m MZI were sought and some assumptions were made.

One was, that the coherence of the laser output might be too low and thus is the cause for other frequency components in the MZI. Furthermore this might be reduced by reaching a good and stable lock condition. If this was true after all, though, this cannot be avoided with the used laser without finding lock at first which degrades this to a possible explanation for the noise, but without any beneficial outcomes for designing the PLL.

The next assumption was that reflections caused by bad splices lead to additional interferences causing the higher frequency components. Using a visible laser fiber fault locator, faults and splices with high loss were sought to be replaced. This had no clear outcome, so just some splices were replaced without any positive effects on signal integrity, though. Afterwards, a optical power measurement was run in every branch and output of the MZI. This also indicated that there is no loss as high as to explain the noise amplitude.

Next step was analyzing the DAC step size's influence on the laser frequency. Therefore, the signal generator was connected to the input of the laser driver. The arbitrary signal generator was set to a waveform replicating a linear rising slope, with one point right in the middle, where the voltage is not increased in $1mV$ steps, but takes a $2mV$ step. The outcome can be seen in figure 3.3.

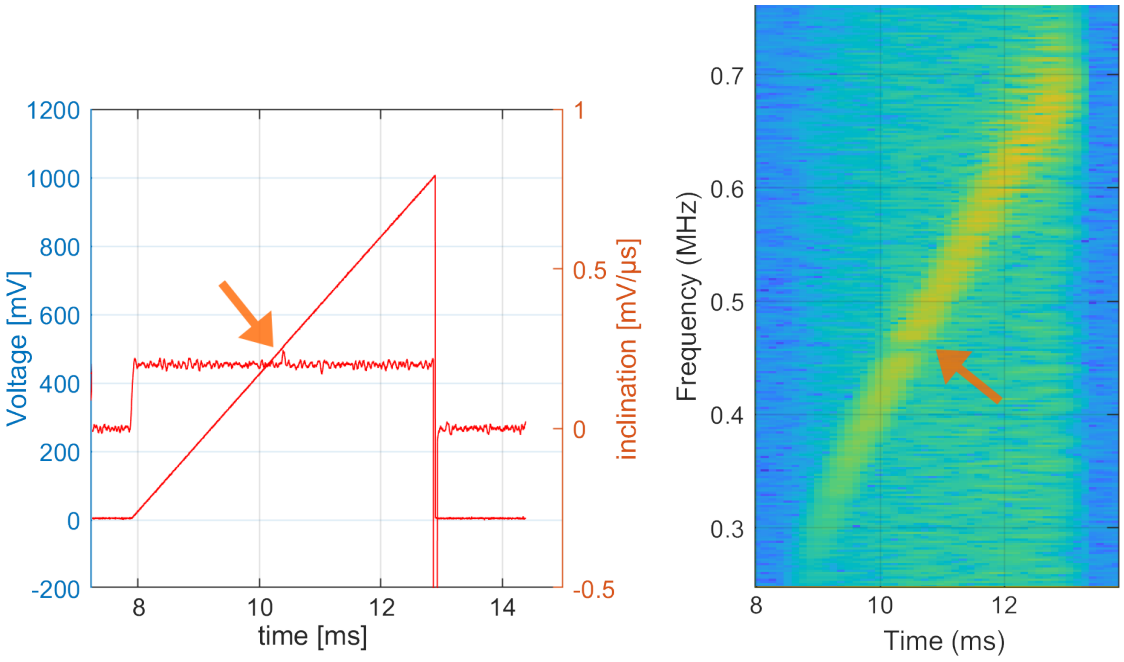

Figure 3.3. Linear laser driver control voltage rise with step in the middle, and FFT of the MZI output

As this is a big jump in frequency and thus ways for smoothening the DAC output voltage jumps were seeked. Where the easiest is certainly just increasing the resolution of the DAC, this will decrease, but not eliminate those voltage steps. Instead, an RC-filter was used. This can be placed at two locations. Either placing a capacitor directly in parallel with the laser will introduce some kind of current smoothing, but is relying on an unknown output impedance of the voltage-to-current amplifier of the laser driver. Next to that, the already existent 50Ω resistor for impedance matching after the DAC can be used as part of an RC-filter by

placing a capacitor behind it. The value of the capacitor was initially calculated using the formula for the time constant when charging a capacitor $\tau = R * C$. Against the common usage of 5τ for defining a capacitor as fully charged, for this application 2τ, which is almost 90% was defined to be the maximum that should be reached. This is, because the capacitor is charged exponentially by

$$V_C = V_S(1 - e^{-\frac{t}{RC}}) \tag{3.3}$$

So whenever the DAC changes to the next higher output voltage, the capacitor should not be fully charged to the last voltage yet, to further smoothen the waveform.

Again, assuming a required linear slope, the time step between each new DAC value is $4\mu s$. Setting 2τ to this value leads to a capacitance of $C = \frac{4\mu s}{2*50\Omega} = 40nF$, so the next standard capacitor value of $56nF$ was used. Next to that, additional capacitors of $100nF$ and $560nF$ were placed to have more headroom. All capacitors were in series to a switch to connect or disconnect them.

On the other hand, using equation 3.4, this results in a cut-off frequency of $57kHz$ for the $56nF$ capacitor, which is below the laser drivers crossover frequency.

$$f_c = \frac{1}{2\pi RC} \tag{3.4}$$

The delay from changing the laser driver's control voltage, via optical output, to the PD where it is converted back to an electric signal, was measured by applying a sinusoidal control voltage waveform to the input and measuring this, together with the PD's output with an oscilloscope at different frequencies. This was also done with a capacitor of $47\mu F$ connected in parallel with the laser diode to see its influence on the reaction time. Notice the capacitor value needs to be

much higher here compared the the DAC RC-filter, because the laser driver's impedance is much lower. Figure 3.4 shows that the delay is minimal even though using a capacitor.

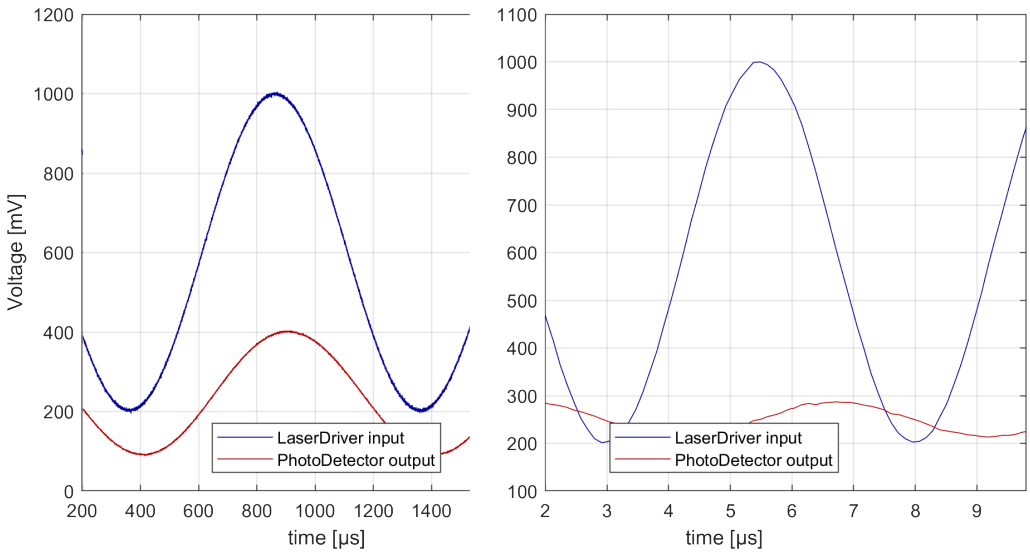

Figure 3.4. PD output versus LD input

Next, the total delay of the feedback loop was evaluated to see if counter measures needed to be implemented. The loop delay consists of the following elements:

- $2\mu s$ DAC transfer and settling
- $4\mu s$ DAC RC-filter delay at $C = 56nF$
- $1.5\mu s$ laser, laser driver and PD
- $0.1\mu s$ optical propagation
- $0.2\mu s$ ADC pipeline latency and transmission
- negligible FPGA control logic

This totals to $3.8\mu s$, neglecting the delay introduced by the necessary RC-filter. This equals about 4 cycles for the $1MHz$ reference clock, 2 cycles for the later used $500kHz$ clock. This is certainly a delay worth noting, but was not considered harmful to getting at least close to finding initial lock, which did not happen until here.

After those steps did not lead to an effective reduction of noise in the MZI signal, the PC was changed from PFD to XOR architecture. With XOR as a PC with different phase offset and output signals, the LF needed to be adapted. Assuming a MZI frequency near the reference clock, the duty cycle, or high-time, of the output indicate whether to accelerate or decelerate the laser control voltage climb, with 50% indicating a perfect inclination. The LF can be a integrator, basically counting the ADC cycles where the XOR is high and incrementing the DAC code and resets itself once a given limit is reached. This integration threshold sets the gain of the LF. A bigger threshold decreases the gain as the XOR output needs to be high for a longer time in order to result in a DAC code increase, and vice versa.

Considering the ADC update rate of $40MHz$, and reference clock of $1MHz$, the calculated threshold for the 50% XOR duty cycle, is $\frac{40MHz}{1MHz} * 0.5 = 20$. Trying out different integrator thresholds in the range $10 - 30$, with $56nF$ and 156nF DAC filter, still did not result in a phase lock.

This lead to the shortening of the MZI delay branch by half, leading to a delay branch of $10m$. By doing that, the noise was considerably reduced, but still high in amplitude, to cause many flaws in the binarizer. In order to mitigate that, it was tried to set the binarizer's Schmitt Trigger thresholds farther away from the mean value.

Also an additional digital DC offset correction filter was implemented. This was implemented as a First In, First Out (FIFO) register, holding all ADC values of the last three reference clock cycles. Adding them to the accumulator on FIFO write operation and subtracting the read FIFO values. This has the problem though, that it works only flawless, if the cycle period of the MZI clock is equal to the reference clock, hence is known. For unknown periods, as they occur in the initial unlocked condition, this is not the case. As it turned out shortly after implementation, the DC components of the sampled and digitized signal was indeed completely free of DC components, other than the Least Significant Bit (LSB), though. So this digital filter was dropped again shortly after.

Next, a stabilizing capacitor in parallel to the laser was tested. As the output impedance of the laser driver was assumed to be rather low, with $47\mu F$ a bigger value was chosen here. After removing the DAC filter and running with different integrator thresholds, this time around the value of 40, because of the halved MZI delay, leading to a halved frequency of $500kHz$.

This time, the PLL found partly lock for about one quarter to one third of the full sweep. Increasing the threshold to higher values moved the locking to a later phase of the sweep.

As seen in figure 3.5 on the left, even with rather low threshold values, the locking needed some time to establish, MZI frequency slowly rising against the reference. Therefore, a jump function was introduced in the beginning of each sweep cycle, helping to hive the laser output frequency increase after a long time of constant current closer to targeted levels, as seen on the right of figure 3.5.

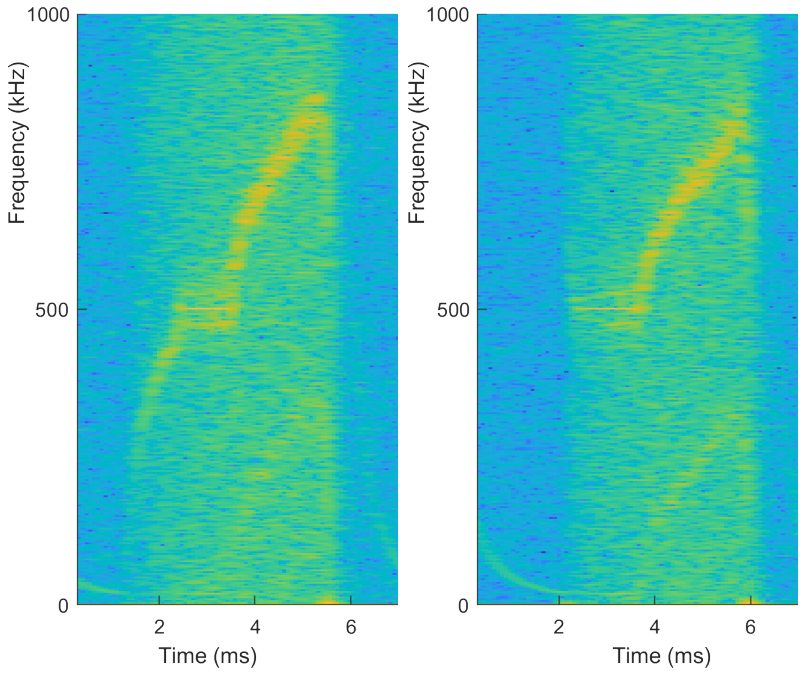

Figure 3.5. MZI output, regular start (left); with $120mV$ jump function in the beginning (right)

3.3 Iteration 3

After achieving better results with the shortened, noise-reduced version of the MZI in iteration 2, further decrease of the noise was achieved using a low-pass filter after the PD. Therefore a $750kHz$ cut-off frequency eleven-pole passive low-pass filter, type *TTE J97-.75M-50-720A* was used.

This did not lead to an improvement of XOR-PC performance, but finally enabled the use of a PFD based PC now. Due to heavy reduction of high frequency noise, an almost high frequency noise-free shape can be recorded by the ADC and successfully binarized without excessive edges caused by the noise. This is depicted in figure 3.6.

This time, after proven successful in iteration 2, the integrator kept being used as LF. So only the PFD's up-output was used in this iteration. DAC code increment step size was implemented dependent on the high time of the last clock

cycles XOR output. This was in order to be able to increment faster, to not lose lock to higher frequencies as experienced before.

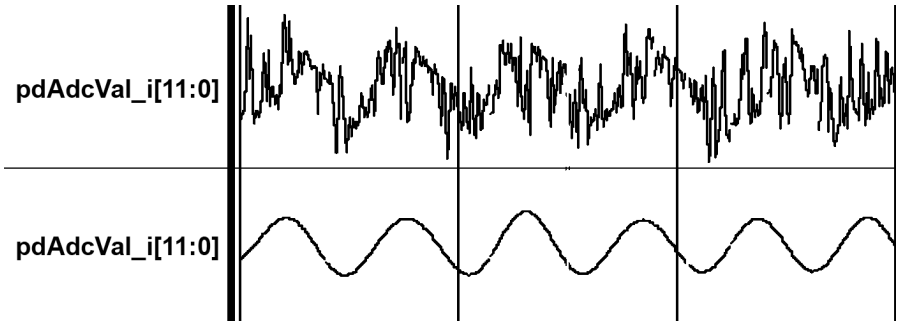

Figure 3.6. PD signal as seen by the FPGA after AD-conversion (top); with additional *TTE* 750kHz low-pass (bottom)

The best lock that could be reached using the PFD is only a very approximate frequency lock as seen in figure 3.7. High filter capacitance of $70\mu F$ laser-side and $3\mu F$ DAC-side was needed to reach that.

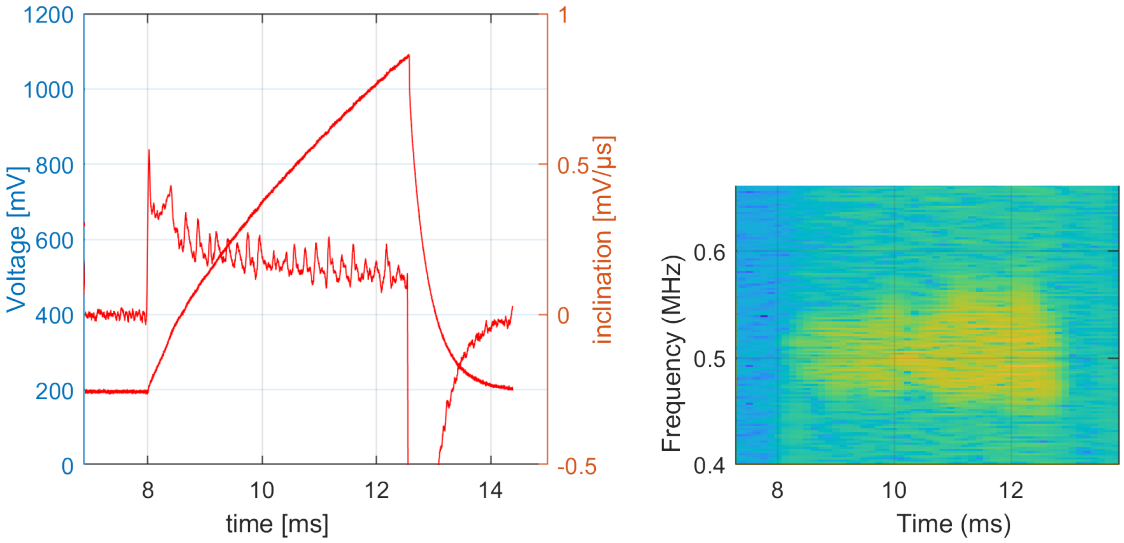

Figure 3.7. Output, using PFD together with $70\mu F$ laser and $3\mu F$ DAC-filter capacitance

Because of the inability to reach real phase lock, this experiment was stopped shortly after, to focus on improving the XOR based solution.

3.4 Iteration 4 and implementation of gain change

As iteration 2 and 3 have shown, correct capacitor values and placement is crucial for proper function of the XOR based PLL, not so much the signal to noise-ratio of the MZI signal, though. For this reason, in this last iteration, more capacitor values were tested on both laser side, as well as for the DAC filter. Switchable capacitor banks were used on laser and DAC filter side, reaching up to $22\mu F$ for the DAC filter and up to $100\mu F$ in parallel with the laser. With values this high, those components should be considered as a separate integrator as part of the LF.

The most important necessity for change, though, was learned from the fact that changing the LF integrator's threshold, lead to different locking points. An gain change over the duration of a sweep cycle was implemented, gradually increasing the integrator threshold with time advancing. Rough estimates of the values can be read from the results of the experiments of iteration 2. These values were not affected much by changing the filter-capacitor's values.

At first, gain change was implemented as 4 specific points spread evenly over the slope, that increased the threshold by 5. With the right gain settings, this lead to locking over the full sweep cycle. For unlocked conditions though, due to wrong initial gain settings, figure 3.8 clearly shows visible characteristics in the MZI's frequency spectrum, caused by those abrupt jumps.

This is, why this frequency adaption was made steady, incrementing by 1 at a arbitrary number of points, evenly distributed over the sweep. Good results are achieved with the number of increase steps set to around 20 ± 5.

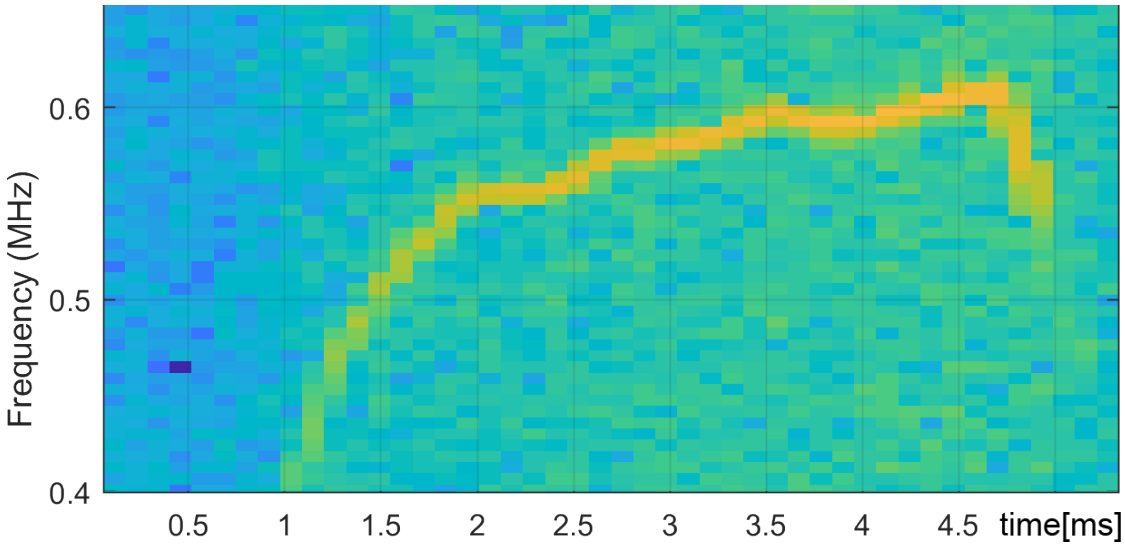

Figure 3.8. 4 points of threshold increase +5

CHAPTER 4

Discussion

Increasing the capacitance and implementing LF gain change lead to a working solution. This was tuned by adapting these values accordingly.

There is a trade-off to be made between locking time and locking quality. Figure 4.1 shows the locking with a DAC-filter capacitance of $22\mu F$, which is very high, and thus leads to a later lock because of the dampened initial jump function and slope, even if tried to increase on FPGA-side. The initial jump is completely removed and the inclination of the laser driver (LD) input voltage keeps on rising until almost $t = 3ms$, when lock is reached and the inclination starts falling.

In contrast tot that, figure 4.2 shows the resulting phase error signal using only a $10\mu F$ capacitance. Here, the lock is reached $1ms$ earlier at the expense of phase error noise.

Figure 4.3 shows the outcome using only a parallel laser capacitance of a total of $82\mu F$. The PLL catches lock almost immediately, because this capacitor almost does not limit the initial jump due to the low impedance of the laser driver. However, the sidebands are a more pronounced and the phase noise is high.

The achieved frequency sweep can be calculated by the formula [6]

$$\Delta f_{Laser} = \frac{f_{MZI}}{\tau} * t_{sweep} \tag{4.1}$$

With the lowest lock time shown here of $3ms$, this gives a total frequency sweep of $30GHz$, while the long locking time of $4.5ms$ results in a $45GHz$ sweep.

For an approximation of the sweep in terms of wavelength, the laser's nominal wavelength of approximately $1.55\mu m$ has to be considered. This equals

a frequency of $f = \frac{c}{\lambda} = \frac{300Mm/s}{1.55\mu m} \approx 193.55THz$. Adding the sweep frequency of $45GHz$ leads to a frequency of $194.00THz$. This, in turn gives a wavelength at the end of the sweep of $\frac{300Mm/s}{194.00THz} = 1.5464\mu m$. This gives a wavelength sweep of $\Delta\lambda = |1.5464\mu m - 1.55\mu m| = 3.6nm$.

Long time, I have relied on the thought that the DAC resolution would not be a limiting factor. This was because of the MZI target frequency of $500kHz$ at $5ms$ sweep-duration, leading to 2500 sine pulses. This equals the 2500 usable DAC steps. Whereas, due to the non-linearity of the laser, certainly not one step per sine pulse, I relied on the thought that a little interpolation with a small RC-filter would be sufficient. As experiments and figure 3.3 show, even tiny voltage jumps have a larger than expected impact on the frequency. It shows, that much larger capacitor values, and therewith delay in the feedback loop, is needed than expected.

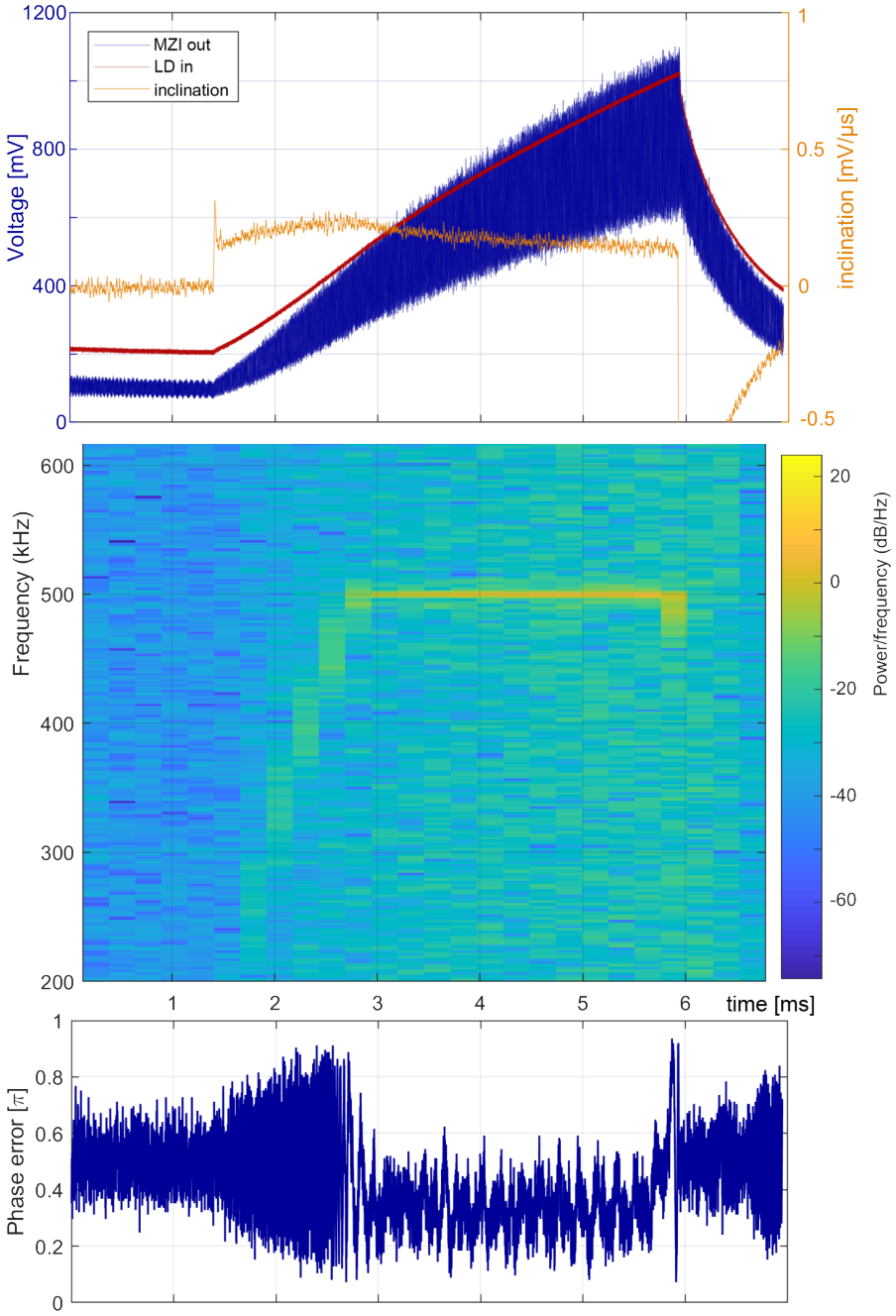

Figure 4.1. Frequency output of MZI and phase error for $22\mu F$ DAC-filter capacitance

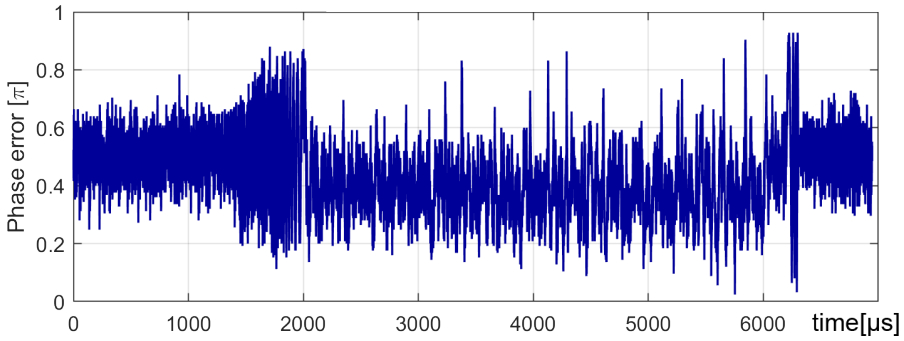

Figure 4.2. Phase error for $10\mu F$ DAC-filter capacitance

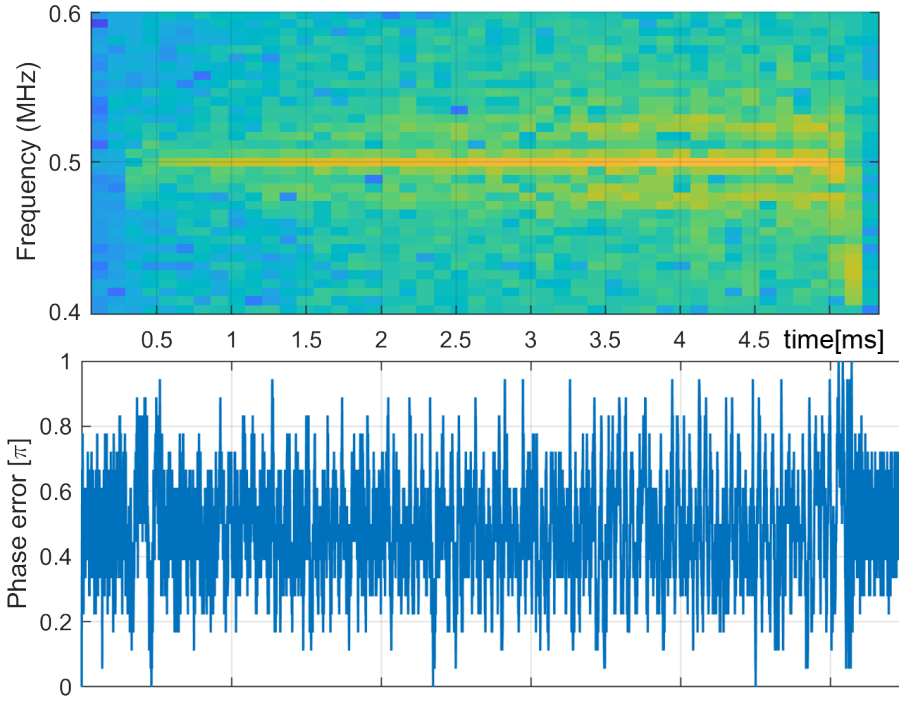

Figure 4.3. FFT of the MZI signal and phase error for $82\mu F$ laser capacitance

CHAPTER 5

Conclusion / Future

Within this study, a digital PLL using an XOR-PC was developed for controlling a DFB sweep over a maximum of $3.6nm$, or with a minimum MZI signal phase noise of 0.4π.

This was achieved using a preprogrammed gain change of the Loop Filter to adapt the feedback to the non-linear characteristics of the laser. Whereas this is certainly a minimal solution, it demonstrates the great versatility of digital implementations and leaves headroom for improvement and many FPGA resources unused.

As noted in chapter 4, the use of a faster and higher resolution DAC would lead to less filtering efforts and thus less loop delay and more direct control. Both quality and locking time could probably be improved.

Next to that, the developed sawtooth generation module using variable timesteps from iteration 1 might be a good choice for applying predistortion. This could learn the behavior of the laser over multiple cycles and approximate a near ideal control voltage by measuring the outcome and continuously adapt the used timesteps.

LIST OF ACRONYMS

ADC Analog-to-Digital Converter

AGC Automatic Gain Ccontrol

C-OFDR Coherent Optical Frequency Domain Reflectometry

DAC Digital-to-Analog Converter

DDR Double Data Rate

DFB Distributed Feedback Laser

ECL External Cavity Tunable Laser

FFT Fast Fourier Transform

FIFO First In, First Out

HDL Hardware Description Language

IDE Integrated Development Environment

FPGA Field Programmable Gate Array

IC Integrated Circuit

LF Loop Filter

LSB Least Significant Bit

LVDS Low-Voltage Differential Signaling

MZI Mach-Zehnder Interferometer

PC Phase Comparator

PD Photodetector

PFD Phase Frequency Detector

PLL Phase Lock Loop

RF Radio Frequency

SoC System on Chip

SPI Serial Peripheral Interface

TEC Thermoelectric Cooler

VCO Voltage Controlled Oscillator

VCXO Voltage Controlled Crystal Oscillator

LIST OF REFERENCES

[1] K. Yuksel, M. Wuilpart, V. Moeyaert, and P. Megret, "Optical frequency domain reflectometry: A review," in *2009 11th International Conference on Transparent Optical Networks*, June 2009, pp. 1–5.

[2] S. Søgaard, J. Henningsen, and J. E. Pedersen, "Wavelength modulation of fibre lasers- a direct comparison with dfb lasers and extended cavity lasers." 2002.

[3] F. Gardner, *Phaselock techniques*. Hoboken, NJ: John Wiley, 2005.

[4] P. E. Allen. Georgia Institute of Technology. "Lecture 070 – digital phase lock loops (dpll)." Accessed: 2018-07-11. 2003. [Online]. Available: http://pallen.ece.gatech.edu/Academic/ECE_6440/Summer_2003/L070-DPLL(2UP).pdf

[5] Z. Chen, G. Hefferman, and T. Wei, "Digitally controlled chirped pulse laser for sub-terahertz-range fiber structure interrogation," *Optics Letter*, vol. 42, no. 5, pp. 1007–1010, Mar 2017. [Online]. Available: http://ol.osa.org/abstract.cfm?URI=ol-42-5-1007

[6] N. Satyan, "Optoelectronic control of the phase and frequency of semiconductor lasers," Ph.D. dissertation, California Institute of Technology, 2011.

[7] S. Palermo. Texas A&M University. "Lecture 7: Phase detector circuits." Accessed: 2018-07-05. 2007. [Online]. Available: http://www.ece.tamu.edu/~spalermo/ecen620/lecture07_ee620_phase_detectors.pdf

[8] A. M. Fahim, "A compact, low-power low-jitter digital pll," in *ESSCIRC 2004 - 29th European Solid-State Circuits Conference (IEEE Cat. No.03EX705)*, Sept 2003, pp. 101–104.

BIBLIOGRAPHY

Allen, P. E. Georgia Institute of Technology. "Lecture 070 – digital phase lock loops (dpll)." Accessed: 2018-07-11. 2003. [Online]. Available: http://pallen.ece.gatech.edu/Academic/ECE_6440/Summer_2003/L070-DPLL(2UP).pdf

Chen, Z., Hefferman, G., and Wei, T., "Digitally controlled chirped pulse laser for sub-terahertz-range fiber structure interrogation," *Optics Letter*, vol. 42, no. 5, pp. 1007–1010, Mar 2017. [Online]. Available: http://ol.osa.org/abstract.cfm?URI=ol-42-5-1007

DiLazaro, T. and Nehmetallah, G., "Large-volume, low-cost, high-precision fmcw tomography using stitched dfbs," *Optics Express*, vol. 26, no. 3, January 2018.

Fahim, A. M., "A compact, low-power low-jitter digital pll," in *ESSCIRC 2004 - 29th European Solid-State Circuits Conference (IEEE Cat. No.03EX705)*, Sept 2003, pp. 101–104.

Gardner, F., *Phaselock techniques*. Hoboken, NJ: John Wiley, 2005.

Palermo, S. Texas A&M University. "Lecture 7: Phase detector circuits." Accessed: 2018-07-05. 2007. [Online]. Available: http://www.ece.tamu.edu/~spalermo/ecen620/lecture07_ee620_phase_detectors.pdf

Satyan, N., "Optoelectronic control of the phase and frequency of semiconductor lasers," Ph.D. dissertation, California Institute of Technology, 2011.

Satyan, N., Vasilyev, A., Rakuljic, G., Leyva, V., and Yariv, A., "Precise control of broadband frequency chirps using optoelectronic feedback," *Optics Express*, vol. 17, no. 18, August 2009.

Søgaard, S., Henningsen, J., and Pedersen, J. E., "Wavelength modulation of fibre lasers- a direct comparison with dfb lasers and extended cavity lasers." 2002.

Silicon Laboratories Inc., "An687: A primer on jitter, jitter measurement and phase-locked loops," May 2012.

Yuksel, K., Wuilpart, M., Moeyaert, V., and Megret, P., "Optical frequency domain reflectometry: A review," in *2009 11th International Conference on Transparent Optical Networks*, June 2009, pp. 1–5.