POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in Ingegneria Elettronica

Optimized digital platform for real-time control of integrated photonic circuits

Relatore: Prof. Marco Sampietro
Correlatori: Prof. Giorgio Ferrari
Ing. Francesco Zanetto

Tesi di Laurea Magistrale di:
Vittorio Grimaldi
Matricola: 875829

Anno Accademico 2017-2018

Contents

List of Figures

Sommario

La *Silicon Photonics* è una promettente tecnologia che permette la realizzazione di circuiti ottici integrati di notevole complessità. Il suo sviluppo è stato però ostacolato dall'estrema sensibilità di queste architetture a cambi di temperatura e tolleranze di processo. Queste limitazioni vengono superate grazie all'introduzione della CLIPP, un sensore innovativo e di facile integrazione che permette una misura estensiva e non invasiva del segnale luminoso, aprendo così le porte per un'azione di controllo in tempo reale che permetta la stabilizzazione di circuiti fotonici.

Questo lavoro di tesi è basato su una scheda multifunzione che permette la lettura a basso rumore di 16 CLIPP in contemporanea e il controllo di 16 attuatori termici. Un FPGA controlla tutte le funzionalità della board, ed è stato quindi programmato in VHDL per lo scopo. L'FPGA gestisce anche la comunicazione con un software appositamente realizzato, il quale consente una riconfigurazione dei parametri del sistema e una comoda visualizzazione dei risultati ottenuti.

Attraverso l'introduzione di una demodulazione in due passaggi e lo sviluppo di precise tecniche di processing digitali, la sensitività del sistema è stata aumentata di un fattore 6 rispetto a soluzioni precedenti. Particolare cura è stata posta nella realizzazione di appositi filtri per la cancellazione delle armoniche ad alta frequenza risultanti dalle demodulazioni svolte.

Infine, alcuni algoritmi di controllo sono stati programmati e testati su un circuito fotonico integrato, dimostrando l'efficacia della piattaforma nello stabilizzare e controllare il punto di lavoro di più dispositivi in simultanea. In

particolare, gli esperimenti hanno provato che, anche in seguito all'applicazione di ingenti disturbi, l'algoritmo è in grado di riportare il sistema nella sua condizione di equilibrio in pochi secondi.

Abstract

Silicon Photonics is a promising technology that allows the realization of photonic integrated circuits of outstanding complexity. Its development has been hindered by the extreme sensibility of these architectures to temperature drift and fabrication tolerances. This limitation can be overcome with the use of the CLIPP (ContactLess Integrated Photonic Probe), an innovative and easily integrable sensor that allows non-invasive monitoring of light signals, potentially paving the way for real-time control and stabilization of photonic circuits.

This thesis has addressed a multi-purpose motherboard that allows the low-noise readout of up to 16 CLIPP simultaneously and the control of 16 thermal actuators. The FPGA controlling all the functionalities of the board has been programmed in VHDL for the purpose. It also communicates with a specifically designed software interface, allowing an easy reconfiguration of the many parameters of the system and a convenient visualization of the obtained results.

The sensitivity of the system has been boosted by a factor 6 with respect to previous realizations through the introduction of a two-step demodulation and optimized digital filtering techniques. Special care was taken in the realization of digital filters that allow the cancellation of all the high-frequency unwanted harmonics resulting from the demodulations performed to extract the signal.

Eventually, embedded control algorithms have been coded and successfully tested on a real photonic integrated chip, demonstrating the effectiveness of the platform in stabilizing and controlling the photonic layer through closed-

loop control. The experiments have proven that, even after the application of severe disturbances, the algorithm is able to bring back the system into its equilibrium condition in few seconds.

Chapter 1

Introduction

The improvements in the fabrication process of electronic devices have lead, through the years, to the creation of high-density and complex circuits. In particular, technology scaling has allowed to create faster circuits with smaller components, allowing then the possibility of having more devices in the same area. This has made the role of connections more and more important. Metal connections are in fact hindering further improvements, due to their maximum data-rate supported (less than 10 GHz), the big power dissipation introduced and the sensibility to crosstalk and external disturbances.

A promising solution for this issue is the use of optical connections. Optical connections are already widely used for data-transmission, especially for long distances. Recently, the interest in short distance optical connections has grown, suggesting their use on a single board or even on a single chip.

An optical connection is indeed able to solve many of the traditional connection problems described above. Nonetheless, these connections are more expensive and more complex than the traditional ones.

A promising technology in this field is the "CMOS Integrated Silicon Photonics", since it allows to integrate, on the same chip, optical and electronic components. This technology can be used to create low power and high data-rate optical interconnections. At the same time, fabrication costs are kept low since it's possible to take advantage of the standard well-known CMOS

processes. This technology can be also exploited to realize not only simple optical connections, but even more complex devices.

Unfortunately, the integration of many optical devices on the same chip has been hindered by the lack of non-invasive method to monitor the light inside the circuits. Real-time monitoring is indeed fundamental to realize closed-loop control systems, that allow to compensate and stabilize process tolerances, temperature fluctuations and crosstalk effects, inevitable in complex systems. The non-invasive sensor CLIPP (*ContacLess Integrated Photonic Probe*) allows to overcome this limitation.

1.1 Silicon Photonics

Silicon Photonics is a technology that allows the creation of integrated optical circuits. In the last years, it has become one of the most promising photonic platforms, thanks to the excellent optical properties of the silicon and to the compatibility with the CMOS standard fabrication process. The purpose of the Silicon Photonics is the integration of optical and electronic devices into the same chip, allowing then the realization of hybrid systems with advanced functionalities and high performances.

In particular, SOI (*Silicon on Insulator*) technology offers an excellent support for the realization of photonic devices. A key advantage of this technology is the big difference between the refractive index of the silicon ($n \simeq 3.45$) and the silicon oxide one ($n \simeq 1.45$). This allows to create good rectangular Si/SiO_2 waveguides with excellent optical characteristics. Indeed the optical modes are well confined inside the guide, allowing the possibility of scaling the core up to 0.1 μm^2, in the direction of a wide integration of these devices. Moreover, silicon is transparent for wavelengths between 1100 nm and 7000 nm, since the energy of a single photon is not sufficient to promote electrons from the valence to the conduction band. This means that silicon waveguides work perfectly in the IR spectrum (1300 nm - 1550 nm), widely used in the telecommunications industry.

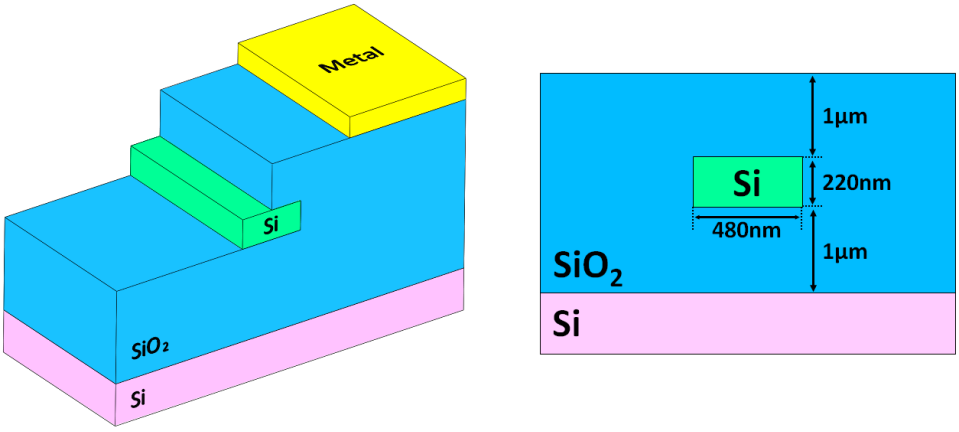

Figure 1.1: 3D representation and cross-section of a rectangular waveguide in SOI technology.

The main problem of this technology comes from the fact that many optical devices are highly sensitive to temperature fluctuations and process tolerances. For instance, it has been proven experimentally that a single degree centigrade temperature variation causes a drift in the frequency response of an optical interferometer of about 10 GHz [11]. To overcome this issue, it's necessary to create closed-loop systems in order to compensate these fluctuations, allowing to stabilize the working point of each device with precision in order to guarantee the correct functioning of the system.

1.1.1 Waveguide

The basic component of the Silicon Photonics is the waveguide, through which the optical signals are routed from one point of the circuit to the other. The guide is made with a central silicon core, usually lightly p-doped, surrounded by a silicon oxide cladding, as depicted in figure 1.1. The propagation of the light inside the waveguide relies on the total internal reflection principle. The big difference between the refraction indices of the two materials guarantees that the light is well confided inside the guide and allows the propagation of the signal with minimal losses.

The main source of losses is the absorption of photons due to the fabrication

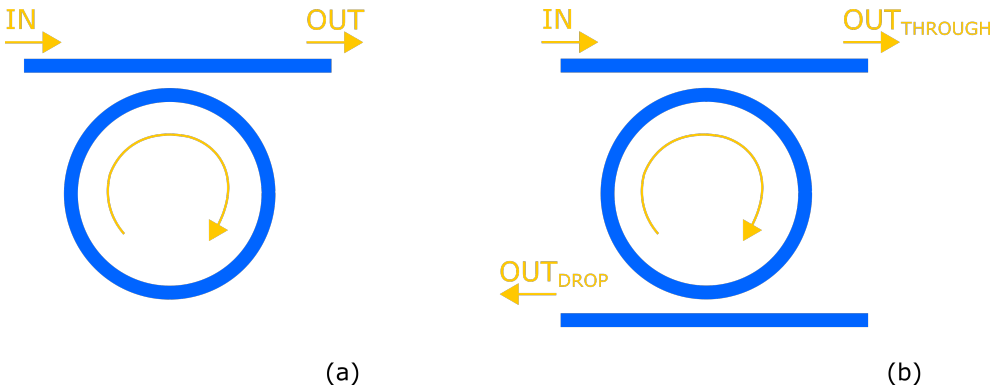

Figure 1.2: Simple ring resonator (a) and add/drop filter (b).

imperfections at the interface between core and cladding. This losses are exploited by the CLIPP sensor in order to measure the quantity of light inside the guide without extracting any photons, as further explained in the following paragraphs.

1.1.2 Microring resonator

An important optical component realized with this technology is the microring resonator. A ring resonator is basically a waveguide closed in loop on itself, as shown in figure 1.2(a). In this way, when the length of the optical path is an integer multiple of the wavelength, the light returns exactly in phase with itself after a roundtrip, generating constructive interference and making the ring resonate. In this condition, the light at that wavelength that passes through a waveguide in close proximity to the ring is transferred to the ring itself through the evanescent field and released after many roundtrips. The final effect is a drop of the optical power that reaches the output, due to the losses inside the ring.

Instead, if another waveguide is placed on the opposite side of the ring, the optical power is transferred there. This structure depicted in figure 1.2(b). In this way, a microring resonator can be used as an add/drop filter for specific wavelengths.

The typical transfer function of a microring resonator is depicted in figure

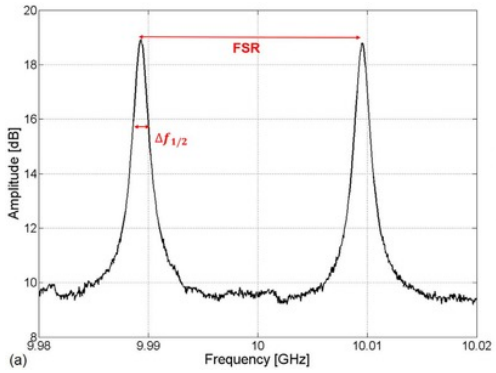

Figure 1.3: Typical transfer function of a microring resonator.

1.3. Notice how the device has multiple resonances at all the integer multiples of the wavelength. The distance between resonance peaks is called "Free Spectral Range" (FSR), and it's often used as a figure of merit of the device.

The microring resonator has a high sensitivity to temperature fluctuations and fabrication tolerances. An error of 1 nm in the length of the ring shifts the resonance frequency of more than 100 GHz [5].

The working point of the device can be tuned by changing the length of the optical path. The optical path is defined as

$$L_{opt} = d \cdot n \tag{1.1}$$

where d is the geometric length of the ring and n is the refractive index of the mean. Since d is clearly fixed by the geometry of the system, the only possibility to vary L_{opt} is necessary by changing the refractive index. This is possible since the refractive index depends on the temperature of the material.

This means that, with a thermal actuator, it's possible to precisely control the phase shift introduced on the signal, changing, in this way, the resonance frequency of the device.

1.1.3 Thermal actuators

As already said, many optical devices show high sensitivity to temperature fluctuations because of the high thermo-optic coefficient of the silicon. This

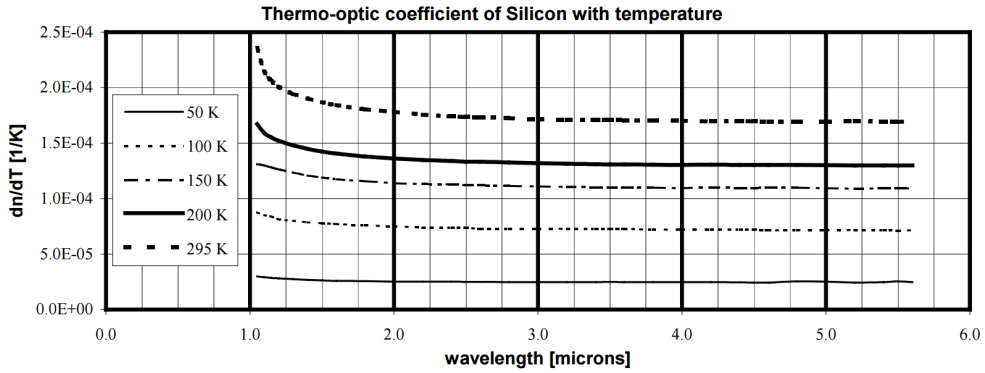

Figure 1.4: Dependence of the refractive index of the silicon as a function of the wavelength and the temperature.

functional dependence is depicted in figure 1.4. As a consequence, it's necessary to stabilize precisely the temperature to guarantee a correct functioning of the optical circuit. At the same time, this high sensitivity can be exploited to actively calibrate the working point of each and every device.

For this purpose, some thermal actuators, called heaters, are integrated in strategic points in the optical circuits. They are basically small resistances obtained by deposing a small layer of metal on top of the oxide that covers the guide. The ease of integration allows the actuators to be placed in virtually any point of the circuit.

The average response time of a heater is around some µs. This means that the device can be used to compensate the slow thermal drifts that affect the circuit, but also to introduce a thermal modulation of the light, up to hundreds of kHz. This modulation is usually called "tone" and can be used to create complex control algorithms.

Notice that, since the temperature of a heater is proportional to the power dissipated, the variation of the refractive index depends quadratically on the voltage applied to the actuator.

1.2 CLIPP sensor

The creation of complex photonic systems has long been hindered by the fact that truly non-invasive method to monitor light weren't really available. Traditional methods rely on the use of photodetectors that extract a small quantity of light to measure it. This method is conceptually easy, but it can be applied a limited number of time, since each measure slightly attenuates and disturbs the light signal. Also, it's not easy to properly extract photons to perform this measurement, since special structures are needed. With the advent of Silicon Photonics, the number of devices integrated on the same chip has grown, and with it the necessary observation points, making the use of traditional methods impractical.

CLIPP (*ContactLess Integrated Photonic Probe*) is an innovative sensor that allows to measure in a non-perturbative way the optical power in a silicon waveguide, by exploiting the fact that the light locally changes the electrical characteristics of the guide. The sensor is simply realized by deposing two electrodes on top of the cladding of the waveguide. For this reason, it is highly integrable and can be then used to measure the intensity of light in virtually any point of the circuit.

1.2.1 Theory of operation

To describe the working principle of this sensor, in the following a rectangular waveguide in SOI is considered, as the one depicted in 1.1. From an electrical stand point, the core of the guide behaves basically as resistor. The value of this resistance can be easily estimated as

$$R_{WG} = \frac{1}{\sigma} \frac{L}{W\dot{H}} \tag{1.2}$$

where σ is the conductivity of the silicon, W and H are the dimension of the core of the guide, and L is the distance between the electrodes of the sensor. Assuming a waveguide as the one depicted in figure (W = 480 nm, H= 220 nm), standard doping level for the wafer SOI (that is about $10^{15}/\text{cm}^3$ p-type)

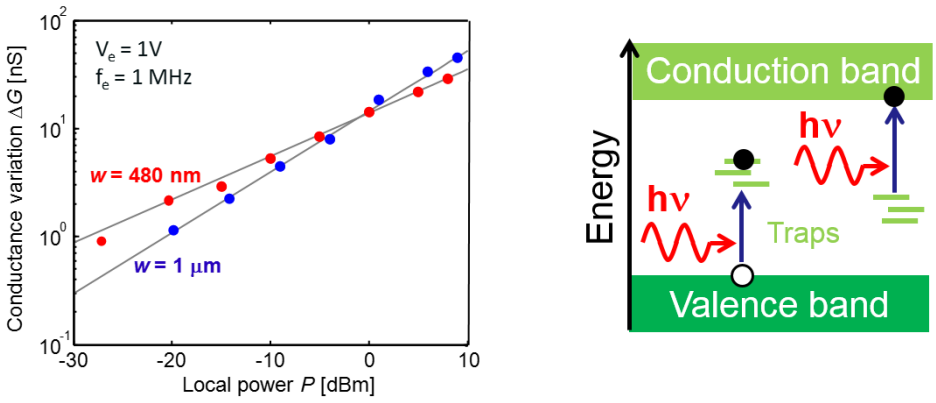

Figure 1.5: On the left, conductance variation as a function of the optical power through a single mode waveguide (in red) and a multi-mode waveguide (in blue). On the right, representation of the Surface-State Absorption mechanism that causes the variance of conductance of the waveguide.

and a distance between the electrodes of L = 100 µm, the estimated resistance is R = 125 MΩ, or, equivalently, the conductance is G = 8 nS.

To measure the optical power, the CLIPP exploits the variation of the resistance that the presence of light induces naturally in the guide, as depicted in figure 1.5(a). Since the geometric parameters are fixed, this variations are due to variations of the conductivity.

The electrical conductivity is defined as

$$\sigma = q(\mu_n N_n + \mu_p N_p) \tag{1.3}$$

where q is the charge of the electron, μ the mobility and N is the concentration of free carriers per unite volume. It has been observed that, while modest variations of the mobility do happen, conductivity variations can be mainly attributed to the increment of the concentration of free carriers.

Many phenomenons contribute to the variation of the concentration of free carriers, but the *Surface-State Absorption* (SSA) mechanism has been identified as the main responsible. This mechanism relies on the interaction of photons with the intra-gap energetic states at the interface between core and cladding. In fact, at the considered wavelengths (between 1300 nm and

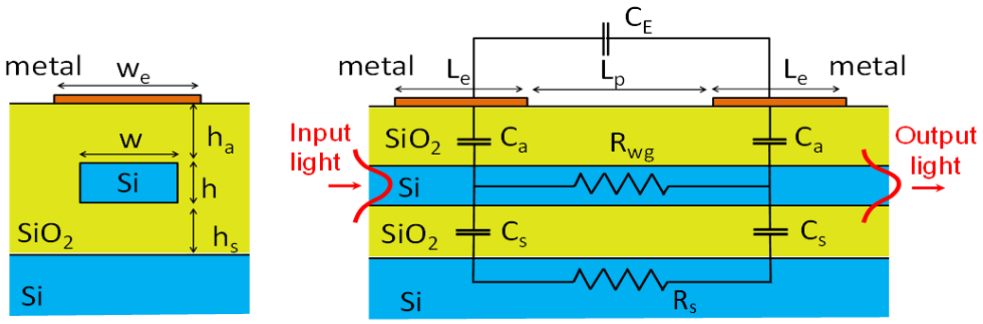

Figure 1.6: Electrical model of a CLIPP.

1550 nm), the energy of the photons is not enough to promote an electron from the valence to the conduction band. Nonetheless, this energy is enough to induce the promotion of carriers to intermediate states, given the smaller energetic gap. In the same way, a photon can interact with an electron already in this intermediate state and promote it, in turn, to the conduction band. This mechanism is depicted in figure 1.5(b). The extra carrier generated increases locally the conductivity of the guide. It has been experimentally demonstrated that the conductivity variations due to the SSA are proportional to the square root of the optical power [21].

Even though there are some examples of sensors that measure the resistance of the waveguide by directly contacting the core [13], in order to perform a truly non-invasive measurement the CLIPP accesses the core capacitively, avoiding a direct contact with it.

The electrodes are realized through a 200 nm gold layer deposited on the cladding, at about 1 µm from the core. At that distance, considering a wavelength of 1550 nm, the signal has an optical intensity about 80 dB less than its peak value in the guide. This means that the electrodes do not cause a noticeable variation of the propagation losses and do not perturb the signal, as intended.

1.2.2 Electrical model of the sensor

The electrical model of the sensors is depicted in figure 1.6. The first contribution that has to be highlighted is the access capacitance of the sensor C_A. The value of this capacitance can be estimated, neglecting the fringe effect, as parallel-plates capacitor, leading to an approximate value of

$$C_A = \varepsilon_o \varepsilon_{ox} \frac{W_e L_e}{h_a} \tag{1.4}$$

where ε_o is the vacuum permittivity, ε_{ox} the relative permittivity of the silicon oxide, $h_a = 1$ µm is the oxide thickness and W_e and L_e are the electrodes dimensions. Assuming $W_e = 480$ nm $L_e = 200$ µm, the estimated C_A value is between 2 and 5 fF. The intended path for the signal is the series of the two access capacitances C_A and the waveguide resistance R_{WG}, already estimated to be 125 MΩ.

The information desired might be hidden due to the presence of a competing path through the parasitic capacitance C_E. This parasitism is due to the direct coupling between the electrodes and the connection wires and can be estimated to be around 10-100 fF, depending on the sensor geometry. Its contribution is then way bigger than the one due to the series of the waveguide resistance and the two access capacitances. One should consider that this value strongly depends on the layout of the electrodes and the connection wires, so extra care should be put in the design of the CLIPP [6].

The C_s - R_s path through the substrate is another competing path for the signal. Luckily, its contribution is visible only at frequencies higher than the ones usually chosen. For this reason, it will be neglected from now on in the simplified model of the sensor. Notice that this is true only considering this first simple geometry of the sensor. For more complex geometries, its effect should be always investigated and discussed [12].

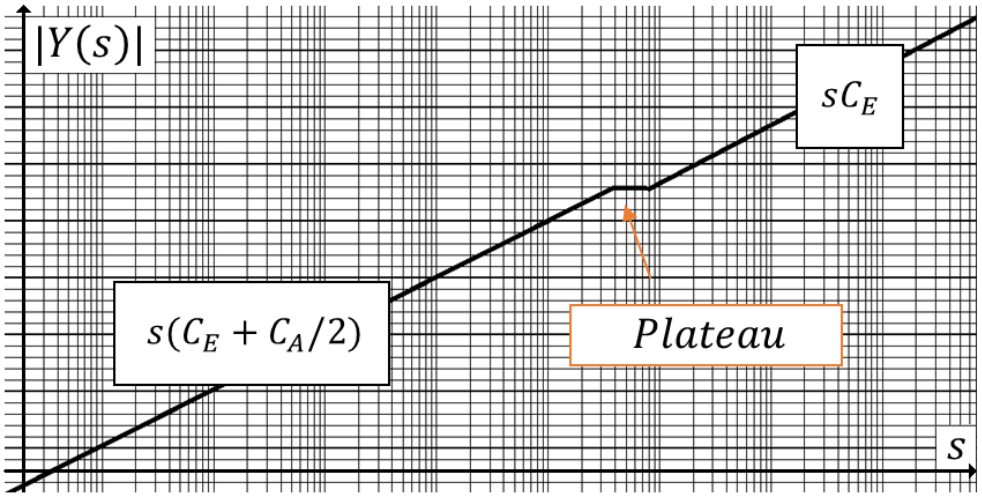

Figure 1.7: Spectrum response of the admittance of a CLIPP.

1.2.3 Frequency analysis

Considering the electrical model depicted in figure 1.6, the admittance of the guide can be written as

$$Y(s) = \frac{s(2C_E + C_A)(1 + s\frac{2C_E C_A}{2C_E + C_A}\frac{R_{WG}}{2})}{2(1 + sC_A\frac{R_{WG}}{2})} \tag{1.5}$$

From this expression, it's possible to obtain the plot in figure 1.7. Notice the plateau region due to the pole and the zero of the transfer function, where the information of the waveguide resistance is located.

This admittance can be splitted into its real and imaginary part, for which $Y = \Re(Y) + \Im(Y)$. In particular

$$\begin{aligned} \Re(Y) &= \frac{\omega^2 C_A^2 \cdot \frac{R_{WG}}{2}}{2\left(1 + \omega^2\left(C_A\frac{R_{WG}}{2}\right)^2\right)} \\ \Im(Y) &= \frac{\omega\,(2C_E + C_A)\left(1 + \omega^2\left(\frac{2C_E C_A^2}{2C_E + C_A}\left(\frac{R_{WG}}{2}\right)^2\right)\right)}{2\left(1 + \omega^2\left(C_A\frac{R_{WG}}{2}\right)^2\right)} \end{aligned} \tag{1.6}$$

For frequencies above the pole $f_p = \frac{1}{2\pi C_A\frac{R_{WG}}{2}}$, the real part of the admittance becomes proportional to the conductance of the waveguide, i.e.

$$Re(Y) \simeq \frac{1}{R_{WG}} \tag{1.7}$$

Consequently, when stimulating the sensor at sufficiently high frequency, by extracting the real portion of the admittance it's possible to measure the information about the waveguide resistance, hence about the local optical power. The position of the pole depends on the value of R_{WG} and C_A. It's possible to control these quantity by properly designing the dimension of the electrodes and the distance between them. In this way it's possible to guarantee an adequate plateau extension to perform the measurements. With the geometry considered so far, $f_p \simeq 1$ MHz.
It should be noticed that the value of R_{WG} is dependent on the quantity of light inside the sensor. This means that, for different working conditions, the pole might assume different values. In particular, the more the optical power, the lower the resistive value, the higher the frequency of the pole.

Effect of feed-through capacitance

The effect of the parasitic capacitance C_E should be carefully discussed. In fact, the time constants of the pole and the zero of the admittance are

$$\tau_p = C_A \frac{R_{WG}}{2} \quad \text{and} \quad \tau_z = \frac{2C_E C_A}{2C_E + C_A} \frac{R_{WG}}{2} \tag{1.8}$$

Since C_E is usually larger than C_A, the zero and the pole are really close to each other. This means that the extension of the plateau is quite limited. The admittance of the sensor is then mainly due to the feed-through capacitance, and the information of interest is masked. Extra effort should then be put in the design and layout of the sensor in order to minimize this effect.

1.2.4 Impedance measurement

The easiest way to stimulate this passive sensor is with a sinusoidal voltage $V_{stim} = V_0 \sin(\omega_{stim} t)$ applied to one electrode. By doing so, a current flows into the waveguide, with amplitude proportional to the light intensity. The best way to read such current and convert it into a voltage is with a transimpedance amplifier (TIA) connected to the second electrode, as depicted in figure 1.8.

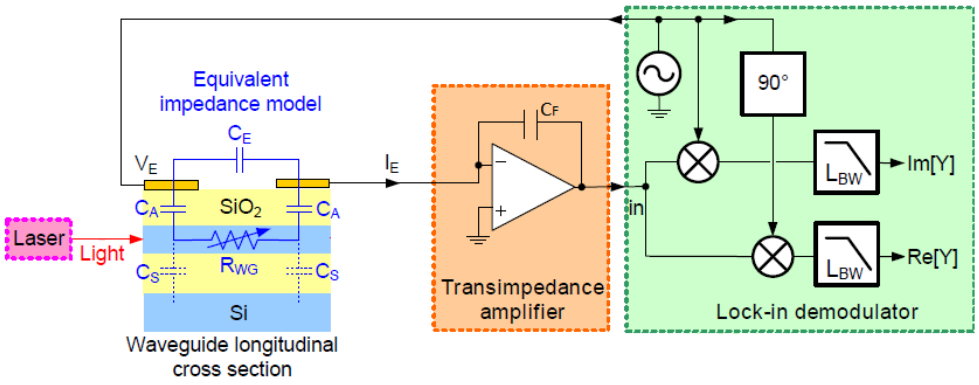

Figure 1.8: Typical circuitry used for the reading of a CLIPP.

The information of the waveguide resistance is extracted with the lock-in technique, by demodulating the voltage at the output of the TIA to obtain the real and imaginary part of the admittance. The use of this technique is the natural choice considering the fact that the forcing signal has already to be at high frequency to properly bypass the entrance capacitances.

In addition, the lock-in technique allows to perform a high resolution measurement, by avoiding the 1/f noise of the readout chain. The signal obtained after the demodulation is then filtered with a narrow low-pass filter to extract the information of R_{WG}. The bandwidth of the LPF can be chosen as narrow as needed to reduce the white noise contribution and obtain the necessary resolution.

1.3 Dithering technique

The dithering technique is performed by superimposing a small sinusoidal signal to the actuation voltage applied to the heater that controls the behavior of an optical device. This causes a modulation in the introduced phase shift, that will lead, in turn, to a modulation of the output optical power. If the applied signal is sufficiently small, the amplitude of the modulation directly depends on the first derivative of the transfer function in that working point, as depicted in figure 1.9. This implies that, by demodulating the output signal

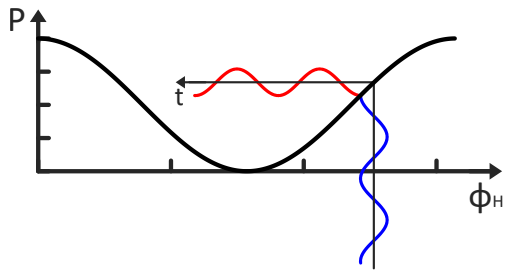

Figure 1.9: Effect of the phase shift modulation on the output power in a microring resonator.

at the dithering frequency, it's then possible to extract the information about the first derivative of the transfer function directly by the physical system. The obtained signal has also the information about the sign of the derivative: in fact, if the curve has a negative slope, then the phase of the optical modulation is opposite with respect to the phase of the signal used for the modulation.

1.3.1 Effect of the dithering on the input signal spectrum

The modulation of the optical power inside the waveguides causes a modulation of the conductance of the guide as well. In particular, G_{WG} can be written as

$$G_{WG} = G_{WG,0} + G_{WG,dith}\sin(\omega_{dith}t) \tag{1.9}$$

The generated current that flows inside the guide is then

$$I_{WG} = V_{stim}G_{WG} = V_0\sin(\omega_{stim}t)G_{WG,0} + V_0\sin(\omega_{stim}t)G_{WG,dith}\sin(\omega_{dith}t) \tag{1.10}$$

Recalling that, for Werner's formula

$$\sin\alpha\sin\beta = \frac{1}{2}[\cos(\alpha-\beta) - \cos(\alpha+\beta)] \tag{1.11}$$

the overall current is

$$I_{WG} = V_0G_{WG,0}\sin(\omega_{stim}t) + V_0\frac{G_{WG,dith}}{2}\{\cos[(\omega_{stim}-\omega_{dith})t] - \cos[(\omega_{stim}+\omega_{dith})t]\} \tag{1.12}$$

leading to the spectrum depicted in figure 1.10.

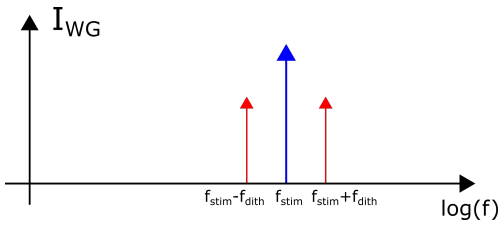

Figure 1.10: Spectrum of the current flowing into the waveguide thanks to the forcing sinusoid.

To extract the dithering information using the lock-in system described in 1.2.4, it's necessary to introduce a further demodulation after the first one. This operation is performed by feeding, to a second mixer, the same signal used to generate the thermal modulation in the first place.

Chapter 2

Overview of the control system

In this chapter, a multichannel electronic system is presented. The ultimate purpose of this system is to demonstrate the possibility of a closed-loop control of complex photonic circuits, where it's necessary to tune the working point of many devices simultaneously. The system was conceived with a modular architecture, designed to read the signals coming from many CLIPP sensors (up to 32), and drive many thermal actuators (up to 16).

The CLIPP signal is read with a tailored ASIC to perform a low-noise preamplification. The output of the ASIC is then brought to the main multichannel electronic board. For geometric reasons, a second interface board was introduced, with the only task of connecting the ASIC with the main board. The signal is then processed, amplified and digitized.

To manage the complexity of such system, a digital architecture based on a FPGA has been chosen. Thanks to the reconfigurability and the high performances of the FPGA, it's possible to implement the control algorithm most suitable for each application. Eventually, a custom user interface is used to configure and monitor in real time the behavior of the whole system.

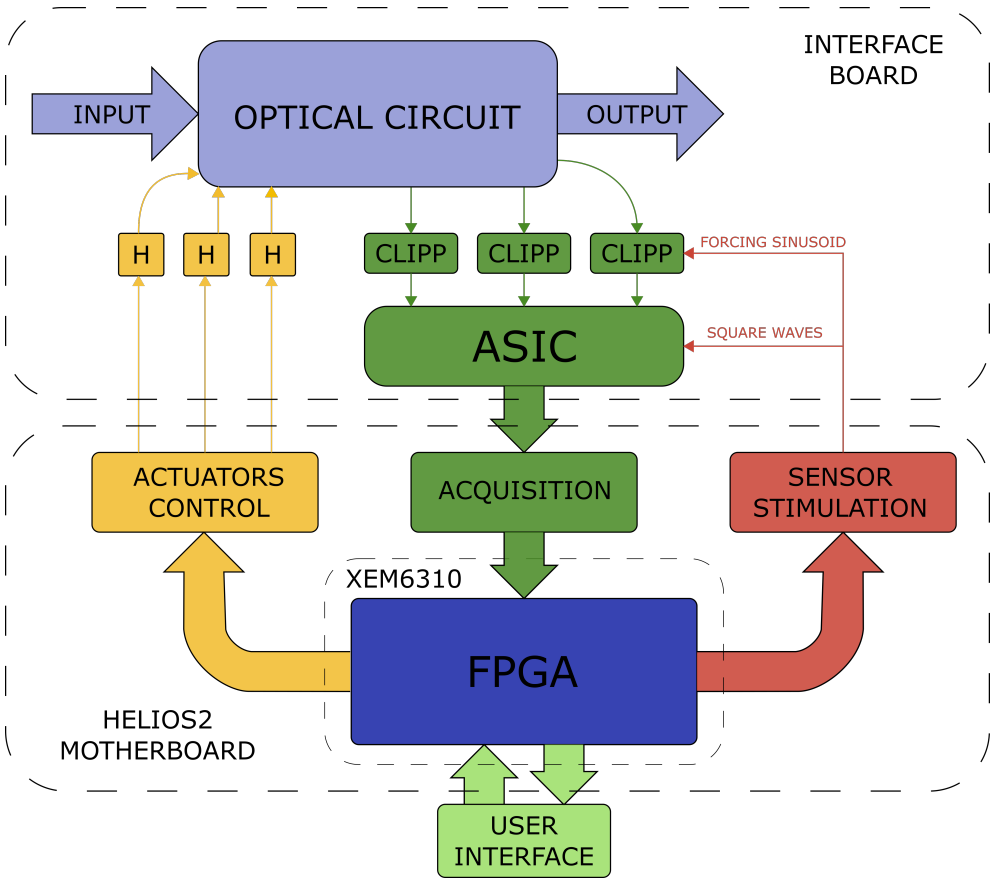

Figure 2.1: Schematic view of the control platform.

2.1 Hardware

The hardware architecture of the system is conceived in a modular way, in order to optimize the overall performance while trying to keep the complexity as low as possible. The CLIPP current signal is brought to a tailored ASIC that performs a first low-noise preamplification and then demodulates it with a switch-based mixer. Then, the signal is brought on the main multichannel electronic board. This board will be addressed as "motherboard" from now on. For geometric reasons, it was introduced a second board whose only task is to connect the ASIC and the photonic chip with the motherboard. The signal is then processed, amplified and digitized. Each part of the system will be now discussed in detail.

2.1.1 ASIC

As already discussed in chapter 1, the signal of interest is intrinsically low, in the order of few pA, and is hidden by a much larger spurious component, in the order of hundreds of nA, due to the stray capacitance between the electrodes. For this reason the system needs to perform an ultra low-noise reading of the output current signal of the sensor. To do so, a tailored ASIC has been previously designed, and it's schematically depicted in figure 2.2.

Given the extreme simplicity of the CLIPP, a complex photonic chip might easily contain a high number of them. For this reason, a single ASIC should be designed to be connected to as many sensors as possible. For this reason, the front-end ASIC was designed with 4 independent readout chains, each of them featuring an 8:1 input multiplexer, in order to connect up to 32 CLIPPs (4 in parallel), while keeping limited the dimension of the chip.

The output of each of these multiplexers is connected to a transimpedance amplifier, in order to provide a first amplification of the signal and to convert it into a voltage. Then, the high-frequency voltage signal is demodulated with a mixer that works with two square waves in quadrature with each other. In this way it's possible to extract both the real and the imaginary part of the signal that comes out from the CLIPP.

In a good design, the noise of the preamplification stage should be the only one limiting the sensitivity of the system. The output noise power spectral density of this chip is then an important figure of merit, and it has been measured to be $\simeq (30\ \mathrm{nV}/\sqrt{\mathrm{Hz}})^2$ [26].

Multiplexer

At the input of the ASIC there are 4 multiplexers with 8 inputs each. Each of these multiplexers is controlled with 3 bits, chosen from the end user, who selects which CLIPP has to be read. In this way, the multiplexers allow to read up to 32 sensors without having to build 32 independent acquisition chains.

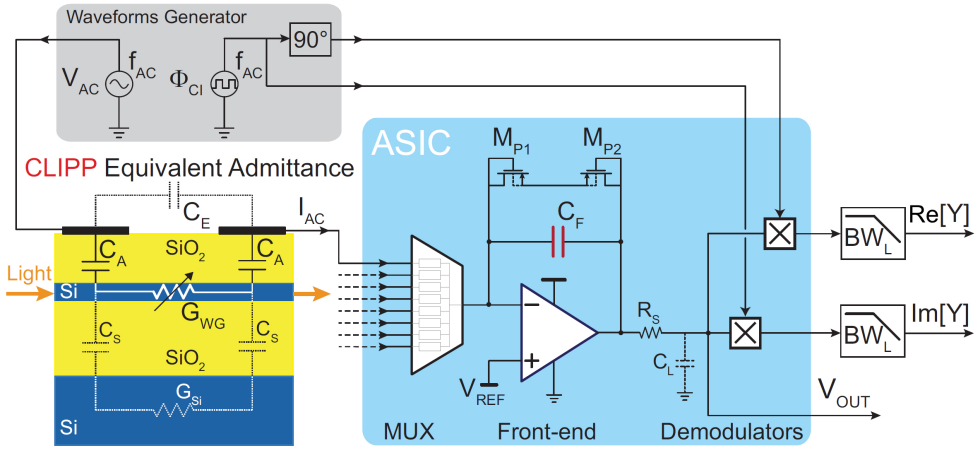

Figure 2.2: Schematic view of a single channel of the ASIC

The multiplexer was designed to avoid that the CLIPPs that are not selected might inject spurious signals in the virtual ground of the connected transimpedance amplifier. A simple multiplexer with a single-MOSFET switch wouldn't be sufficient for this application. In fact, the CLIPPs in the photonic chip are stimulated all together. So, due to the drain-source parasitic capacitance of the MOSFET ($\simeq 1\,\mathrm{fF}$), at an operating frequency of around 1 MHz the disconnected CLIPPs would inject a not negligible current in the amplifier.

A simple solution to limit this feedthrough effect is to add a low-impedance path towards ground that is active when that channel is not selected. To do so, an extra switch is added to the structure, as shown in figure 2.3. This transistor turns on to connect the pads of the off CLIPPs to ground, thus limiting the current that reaches the virtual ground of the transimpedance amplifier.

To further reduce this effect, the final solution chosen is a cascade of two of these 2-MOSFET structures, resulting in the 4-MOSFET architecture in figure 2.3. The on-resistance of the multiplexer increases, but it is still negligible compared to the high value of impedance of the CLIPP.

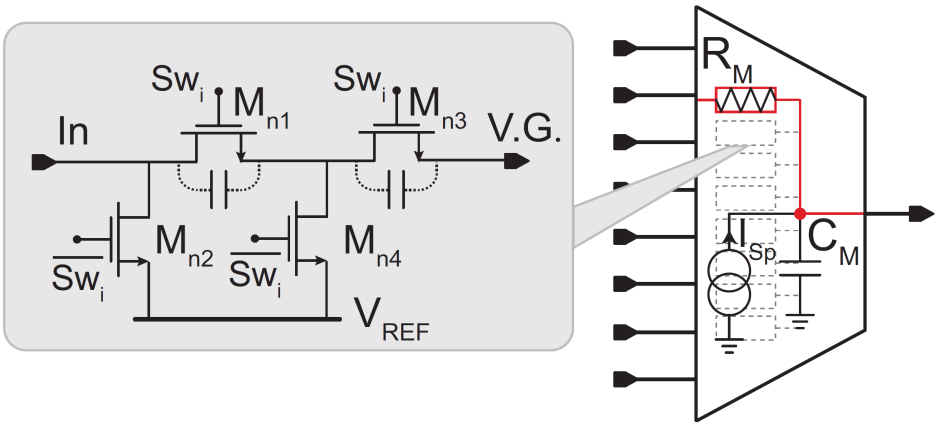

Figure 2.3: Circuital view of one of the four multiplexers present on the ASIC

Transimpedance amplifier

At the output of the multiplexer the current signal has to be amplified and converted into a voltage. A transimpedance stage is then perfect for the purpose, since the virtual ground pin is an excellent current reader. Traditional transimpedance amplifiers with a resistive feedback are not suitable though. In fact, the bandwidth of such stage, assuming that the amplifier is properly designed, is limited by the value of the resistance and the parasitic capacitance in the feedback branch. This leads to a pole at $f_p = 1/2\pi C_f R_f$. One could then decide to lower the value of R_f as much as needed in order to increase the bandwidth, but this would come at the expense of an increased input current noise power spectral density, being $S_{in,R_f} = 4kT/R_f$. The system described so far needs both a high bandwidth, given the high working frequency of the sensor, and a low input noise spectral density, given the small intensity of the signal that has to be measured.

To break this trade-off, a popular solution is to use an active integrator stage, i.e. using a capacitance in the feedback branch rather than a resistance, as depicted in figure 2.2. In fact the capacitance is virtually noiseless, breaking in this way the noise-bandwidth trade-off.

The choice of the value of this capacitance is limited by the swing of the

output node. Given a 3 V power supply, and being 1.5 V the output voltage of the amplifier when no input is applied, the output swing is then limited at around ± 1.3 V. The biggest contribution to the signal comes from the parasitic capacitance C_E between the two electrodes of the sensor. This means that the output voltage can be computed as

$$V_{out} = -V_{stim}\frac{C_E}{C_f} \tag{2.1}$$

As a worst case scenario, a CLIPP with $C_E = 50$ fF is considered. The system is designed to be used with a CLIPP forcing sinusoid up to $V_{stim} = \pm 10$ V, so to ensure that the amplifier doesn't clamp to the supply the minimum value of C_f is around 385 fF. For this design, a safe value of 500 fF was chosen.

Ideal active integrator are not feasible in practice though. In fact, any DC current injected into the virtual ground node would charge the feedback capacitor and cause the output to saturate after a certain amount of time. To avoid this issue it's necessary to add a resistive path in parallel to the capacitive. To avoid falling back into the trade-off above, a high value of resistance has to be chosen.

In IC it's difficult to implement big resistances though. For this reason, a tailored pseudoresistor was designed, that exploits subthreshold transistors to implement high resistive values. The topology also allows to regulate the value by changing its biasing voltage. The resulting resistance ranges from a value of 20 MΩ up to 20 GΩ [26].

Mixer

Once the signal is preamplified by the transimpedance stage, it can be demodulated. The output of each amplifier is thus connected to a switch-based mixer, whose architecture is depicted in 2.4. Two square waves from an external source are fed to the mixer to drive the switches. These square waves are in quadrature with each other in order to extract both the real ($V_{out,I}$) and the imaginary ($V_{out,Q}$) part of the signal. The reference voltage for the mixer is provided from an external source as well.

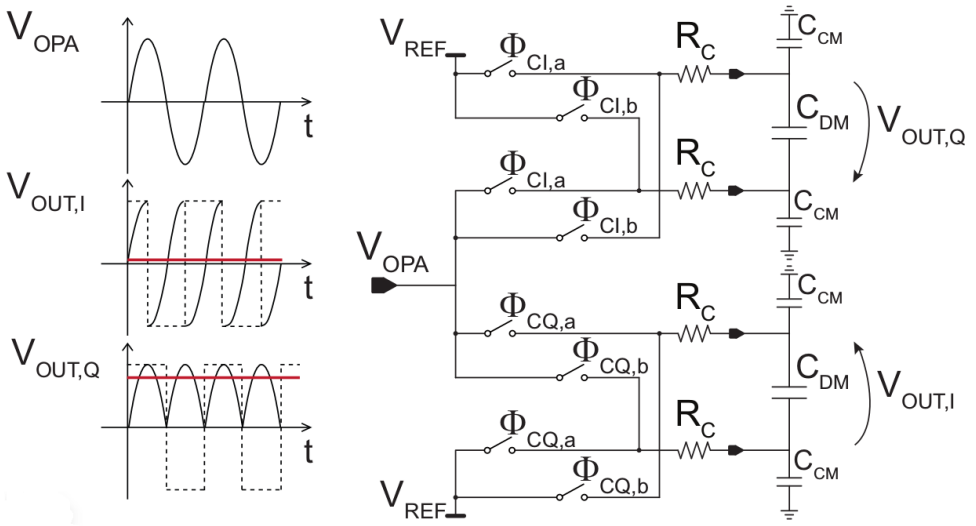

Figure 2.4: Circuital view of one of the four switch-based demodulators

Even if square-wave demodulation would indeed lower the signal-to-noise ratio by a factor $2\sqrt{2}/\pi$, that is $\simeq 11\%$ [8], this solution was preferred over a true analog multiplier, because of its simplicity and lower area occupation, in addition to the lower 1/f noise.

For diagnostic reasons, it has been provided also the possibility of extracting from the ASIC the output of the transimpedance amplifier before the demodulation.

2.1.2 Interface board

The input pads of the ASIC described above are connected to the photonic chip through chip-to-chip wire-bonding in order to minimize the parasitic capacitance of the connection wires. For this reason it's necessary that the two chips are placed close to each other on the same PCB. Also, the photonic chip has to be placed on the optical bench in order to be properly connected with the fibers and all the optical instrumentation. At the same time, the board used to perform the reading and the control action has to feature many acquisition and actuation channels to manage complex photonic chips. It would be then impractical to realize a board with all the characteristics above.

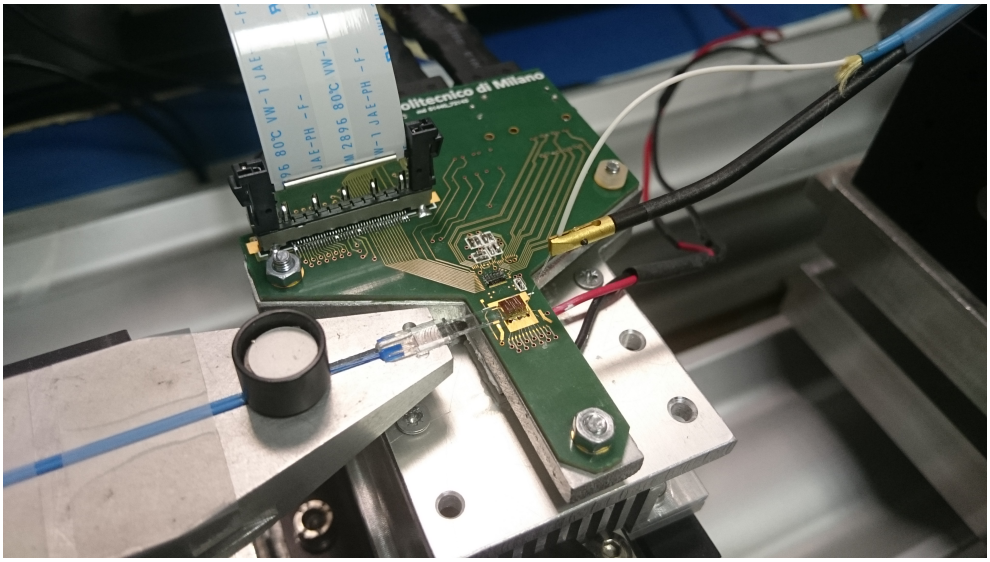

Figure 2.5: Interface board on an optical bench. Thanks to the geometry of this board, it's possible to easily bring the optical fibers to the photonic chip.

The simplest solution is to rather use two PCBs : one wide, general-purpose motherboard with all the necessary electronics on it; one simple and small board, whose only task is to connect the previous one with the output of the electronic chip without impairing the access to the photonic chip. This little board has been called "mantis", due to its shape. The overall setup is shown in 2.5. The metal support on which the board is placed ensures structural integrity.

Moreover, a Peltier cell is placed below the metal support of the PCB, in order to keep constant the temperature of the photonic chip. In this sense, the metal support also works as a thermal accumulator and helps stabilizing the temperature of the system.

To connect this interface PCB to the motherboard, the 2821385-1 by TE Connectivity shielded cables are used.

2.1.3 HELIOS2 motherboard

The core of the system is the board in figure 2.6. Thanks to the aforementioned cable, it can be connected to any photonic chip through the interface

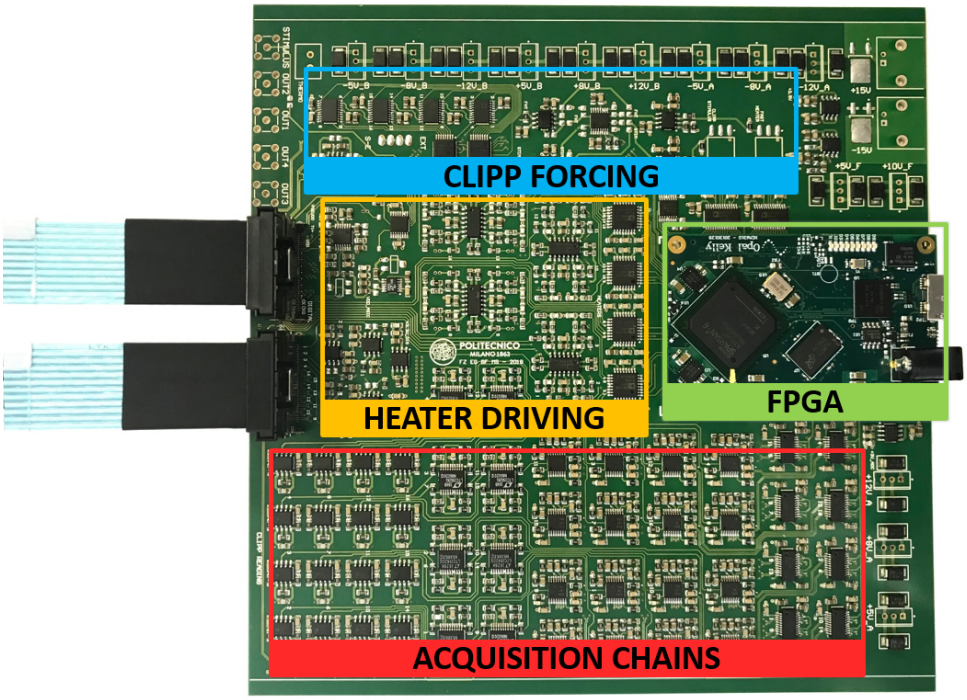

Figure 2.6: Photo of the motherboard. All the chains described are highlighted with different colors.

PCB. The board is designed to perform four main tasks:

- **Acquisition:** the board has to read and process the signals at the output of the ASIC. To be compatible with future developments of the ASIC, the system features 16 parallel acquisition chains, in order to acquire the real and imaginary parts of up to 8 sensors in parallel.

- **Actuation:** the board has to control the actuators in order to steer the behavior of the photonic chips. In particular, there are 16 DACs, each of them individually connected to one heater. The voltage applied to the heaters is either selected from the user or generated by a control algorithm. It's also possible to generate a low-frequency sinusoid (up to few hundreds kHz) to be superimposed to any voltage fed to the heater. In this way it's possible to perform a low-frequency thermal modulation;

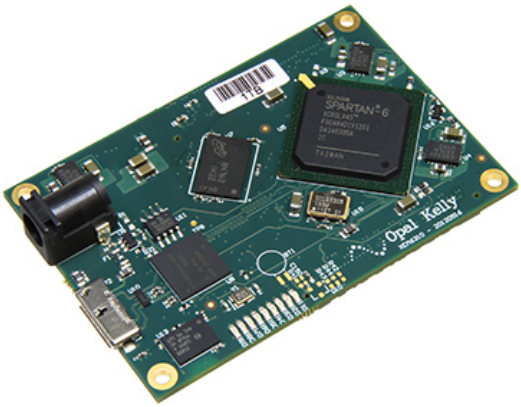

Figure 2.7: Board XEM6310, provided by Opal Kelly. The Spartan-6 FPGA is mounted on it.

- **Stimulus:** the CLIPPs have to be stimulated with a sinusoid at sufficiently high frequency to properly bypass the access capacitances. So, a section of the board is dedicated to the generation of this sinusoid with tunable amplitude and frequency. It also generates the square wave that is used by the ASIC to perform the on-chip demodulation. Eventually, the possibility of summing this fast sinusoid to the heater voltage is provided, in order to perform a high-frequency thermal modulation.

- **ASIC control**: the board has also to control the ASIC connected to the photonic chip. For this purpose it features some voltage regulator, a digital bus to set the multiplexers and a DAC to properly bias the pseudoresistor.

Each of these functionalities will be discussed in details in the next chapter.

2.1.4 FPGA

The complexity of the described system requires a digital core in order to individually control each component. Moreover, since the board was not designed for a specific task, but rather to properly read the signals from the CLIPPs and drive the thermal actuators, a digital core can also be exploited to perform any desired control action. Given the size of the system and the

necessity of a fast control action, a FPGA was chosen over a simple microcontroller.

FPGA stands for "Field Programmable Gate Array", and it is an integrated circuit designed to be programmed and configured by the user. This is possible thanks to the presence of "configurable logic blocks", that are basically logical cells that can be connected to each other by properly programming the FPGA. Each of these logic blocks has a certain amount of LUTs (look-up tables) inside, that can be mapped as any truth table and can, in this way, unravel any logic function.

FPGA are classified as "spatial-computing" devices, since they don't execute the instructions one after the other, as a processor would do, but they rather translate the code into a piece of hardware. Each circuit would then be mapped individually into a certain region of the FPGA and work in parallel to the other ones.

The FPGA chosen for the application is the Spartan-6 LX45, produced by Xilinx. A commercial module was preferred, as it integrates many other useful functionalities, such as an USB 3.0 interface, that allows to easily connect the FPGA to a computer. The board chosen is the XEM6310, produced by Opal Kelly [15] and shown in figure 2.7. It features:

- a FPGA Spartan-6 (by Xilinx);
- 128 MiB of DDR2 SDRAM;
- 16 MiB of non-volatile flash memory, in order to store the configuration file of the FPGA when it's powered off;
- a SuperSpeed USB 3.0 interface (Cypress FX3) for configuration and data transfer;
- eight LEDs, especially useful for debugging purposes.

This board is connected both to the motherboard with two high-density connectors that feature 80 pins each, and to the PC through an USB 3.0

interface. The connection with the PC is especially useful to read, save and display the result of the acquisition chain. Since the USB communication is bidirectional, this connection also allows to change some parameters of the FPGA without having to unplug the whole system and re-compile it.

2.2 VHDL and communication protocols

In this section, a quick overview of the VHDL language is presented, through which the FPGA was programmed. It is also described the communication protocols through which the FPGA can communicate with the components on the motherboard and with the PC. This task is heavily simplified by the use of the commercial module described above, that integrates many useful built-in functionalities.

2.2.1 VHDL

An FPGA is a piece of hardware that can be programmed thanks to appropriate languages. These are not actual programming languages, but rather "Hardware Description Languages" (HDL), and they allow to describe the functionality of an electronic circuit without explicitly stating the components that have to be used. The code is then translated into a real circuit netlist by appropriate synthesis tools.

It's worth pointing out that a HDL can be used to describe any kind of circuit, even an analog one. In this case the synthesis tool would generate a netlist that would be translated into the necessary masks to create the ASIC. In this project, the HDL was instead used to program a FPGA, so it was intended for digital logic circuits.

The two main HDLs for digital logic circuits are Verilog and VHDL, that stands for "VHSIC Hardware Description Language", where VHSIC stands, in turn, for "Very High Speed Integrated Circuits". In this project, the latter was chosen.

The design is split into different functional blocks. Each of them constitutes

an "entity". Once an entity is compiled, it can be instantiated every time needed, and can then be used as a black-box that simply relates inputs and outputs. Each entity is coded with an "architecture" in order to describe its intended behavior.

The Xilinx tools necessary to compile the FPGA also allow the synthesis of some proprietary functional blocks, called "IPCores". These cores can be configured through a graphical interface and, once generated, they can be instantiated exactly as the entities described above. It's good practice to use them whenever possible, since they both simplify the design and they provide an optimized use of resources.

2.2.2 Communication protocols

The communication protocols between the FPGA, the hardware of the motherboard and the PC are here described. As already said, the FPGA can communicate with the control board through the expansion connectors of the Opal Kelly module. The communication with the PC is instead managed by the USB controller, also present on the module.

Communication FPGA - motherboard

To properly handle the communication between the FPGA and the devices on the motherboard, it's necessary to respect the communication protocol indicated on the datasheet of each component. Since the FPGA module features a high number of pins, there is no need to complicate the design by choosing 2-pin protocols, such as the I2C. For this reason the components chosen feature a simple protocol, that is often the SPI (Serial Peripheral Interface).

In the SPI protocol the transmission happens between a device working as "master" and one or more "slaves". The master controls the bus line, generating the clock signal and deciding when to start and when to end the communication. In this design, the master is always the FPGA, while the slave are the components that execute the instructions specified by the FPGA.

In general, the SPI has the following characteristic:

- it's serial, since the data are transferred serially, one bit per clock pulse;
- it's synchronous, since there's a clock that handles the transmission and determines the transmission speed;
- it's full-duplex, since the communication can technically happen simultaneously in transmission and reception.

As shown in figure 2.8, the SPI usually has three or more lines for each device:

- **Serial Clock (SCLK)**: generated from the master, it's used to coordinate the data transmission and reception;
- **Master Output Slave Input (MOSI)**: it's the line where the master send the data to the slave, one bit per time, synchronously with the clock pulses;
- **Master Input Slave Output (MISO)**: it's the line where the slave send the data to the master, one bit per time, synchronously with the clock pulses;
- **Slave Select (SS)**: this line is needed to activate the desired slave for data reception. In fact, both the MOSI and the MISO might be shared between different slaves. This line allows then the master to select the desired device to exchange the data with. In many components, this is also used to synchronize the transmission.

It should be pointed out that the slave devices usually have additional pins, that may be synchronous or asynchronous to the clock. The most common one is the reset pin, that, once driven high (or low, depending on the single component), brings back to a certain status the circuit and the registers within.

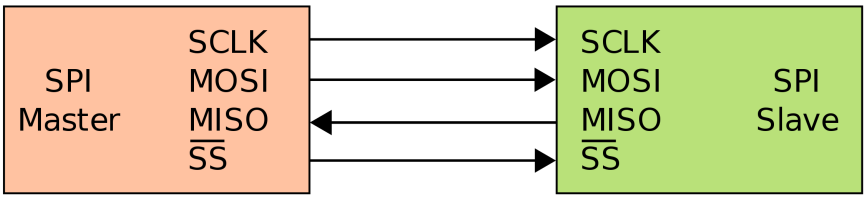

Figure 2.8: Basic SPI bus example: single master to single slave

Communication FPGA - PC

The FPGA has also to communicate with the PC in order to be configured, to plot the acquired and processed data, save them if necessary and set some parameters of the system. The communication is handled through the peripheral controller (Cypress FX3) mounted on the Opal Kelly board. The FPGA interacts with it through some reserved pin. Some proprietary structures have to be included in the code in order to properly handle the communication, as thoroughly explained in the user manual of the board [14].

In particular, the basic necessary entity is the "Host Interface", that handles the data transfer between the FPGA and the controller. Moreover, a specific structure called "endpoint" has to be added for each communication channel needed. There are many different endpoints available:

- **Wire**: it transfers 16-bit data in an asynchronous way. It's useful to send working parameters for internal structures. There are two types of wire endpoints, according to the direction of the communication: *WireIn*, from the PC towards the FPGA, and *WireOut*, from the FPGA towards the PC;

- **Trigger**: it sends an impulse-like signal, that only lasts a period of the clock it is synchronous with. It is useful to control signals that have to be asserted every time a certain action has to be performed, such as a *reset* or a *start* signal. Also in this case, they're available in both directions (*TriggerIn* and *TriggerOut*);

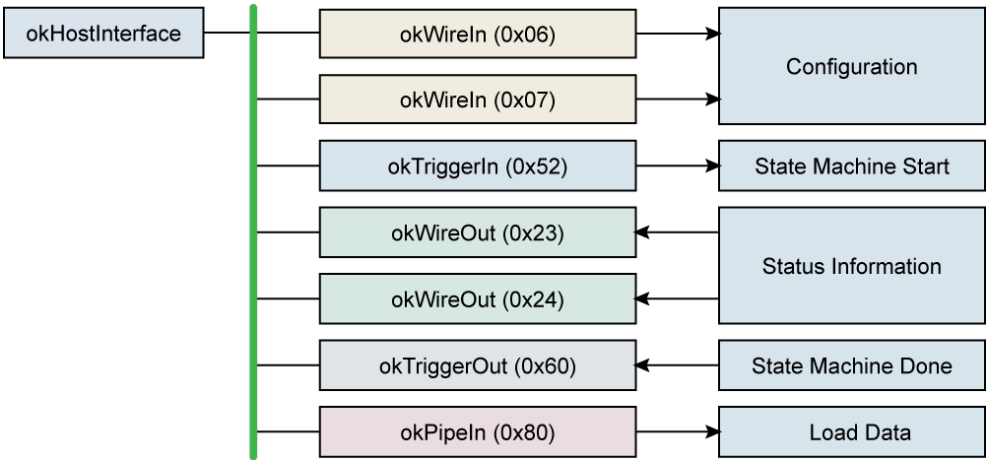

Figure 2.9: Block diagram of a typical communication interface in a FPGA

- **Pipe**: these endpoints are meant to transmit big quantity of data. They're typically used to send the data acquired from the ADCs to the PC. They might need auxiliary memory structures, such as a FIFO, in order to handle the data-transfer between regions that work at different clock frequencies. Also in this case, they're available in both directions (*PipeIn* and *PipeOut*).

Figure 2.9 shows, as an example, the block scheme of a typical communication interface. The Host Interface is instantiated only once, and it is connected both to the reserved FPGA pins in order to communicate with the peripheral controller, and to each of the instantiated endpoints. Each endpoint's instance has a personal address and can be used independently from the other ones.

It's important to point out that each endpoint can be used only a finite number of times. For instance, *WireIn* endpoints are limited to 32 instances. Therefore, they're not sufficient for systems with a big number of configuration parameters, such as this one. To solve this issue, a memory-like structure has been implemented in the FPGA. In particular, one *WireIn* endpoint is used to transfer the data, a second one to transfer the address at which the data has to be saved. Then, a third synchronous signal, generated with a *TriggerIn* endpoint, tells the FPGA that there's a new data that has to be stored. In this way, by trading FPGA resources with used endpoints, it is possible to set

and save a bigger number of parameters.

2.3 User Interface

The possibility of interfacing the FPGA to a personal computer allows to implement a custom software to configure and monitor the behavior of the full system in real time. A C# user interface was thus implemented, using the environment "Visual Studio", provided by Microsoft. The software runs on any Windows 64-bit machine and communicates with the board thanks to an USB 3.0 interface, as explained in the previous section.

The software is used to easily set and tune some parameters of the system without having to recompile the VHDL code. In particular, the main functionalities that the software performs are:

- configure the FPGA and check the connection with the board;
- configure the ASIC multiplexers and pseudo-resistors;
- set amplitude, frequency and phase of the stimulation and demodulation signals;
- acquire the data from the USB interface and provide a graphical interface to properly display and possibly save them;
- set the voltage of the heaters and the dithering frequency and amplitude;
- perform some simple and low-speed control algorithm, as a preliminary step before implementing them directly in the FPGA.

Chapter 3

Motherboard architecture

The CLIPP sensor allows a simple non-invasive reading of the intensity of the light inside a waveguide. This allows the possibility of an extensive monitoring of a photonic chip. It is then convenient to create a general control motherboard, that is able to perform a measurement and a control action, regardless of the specific photonic chip considered.

This board has to feature the electronic to generate the wave to properly stimulate the CLIPP. In the following, this portion of the system will be called "stimulation chain". Moreover, the system needs to process and acquire the data coming from the ASIC, responsible for the preamplification and demodulation of the CLIPP signal. This portion of the system will be called "acquisition chain". The final goal of the system is to perform some control action, acting on the specific actuators (i.e. the heaters), so it's also necessary to have an "actuation chain". Eventually, the motherboard has also to control some parameters of the ASIC. This portion will be called "ASIC control chain". All the components are highlighted in figure 3.1.

A detailed description of each of these sections of the motherboard is presented in this chapter, with a focus on the hardware, the VHDL code to control the components and the user interface developed.

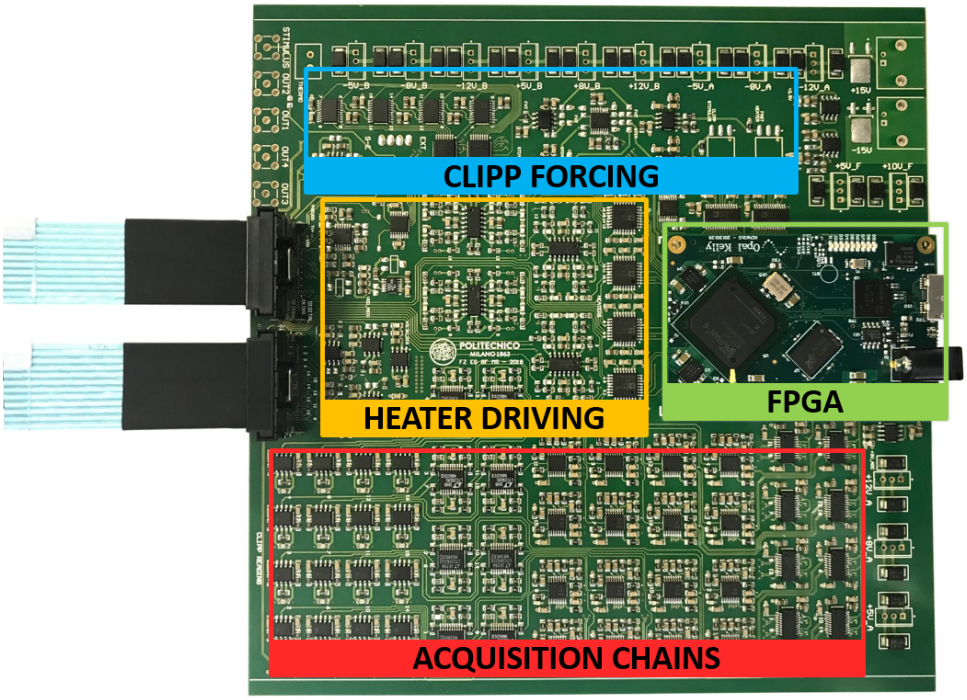

Figure 3.1: Picture of the motherboard. All the chains described are highlighted with different colors.

3.1 Acquisition Chain

The purpose of the acquisition chain is to properly process and digitize the data coming from the ASIC. To do so, the signal has to be properly amplified in order to exploit the whole ADC full scale range. The amplitude and frequency of the CLIPP signal can vary for many reasons: it depends in fact on the quantity of light present into the waveguide, on the geometry of the CLIPP and on process variations. For this reason the gain of the chain has to be programmable, in order to be able to properly amplify the signal in every working condition and even with different photonic chips.

Before being digitized, the signal has also to be filtered with an anti-alias filter. This is necessary to remove the high-frequency harmonics resulting from the on-chip demodulation, that would be otherwise aliased and might result in high spurious signals. Moreover, this filter is also useful to reject high-

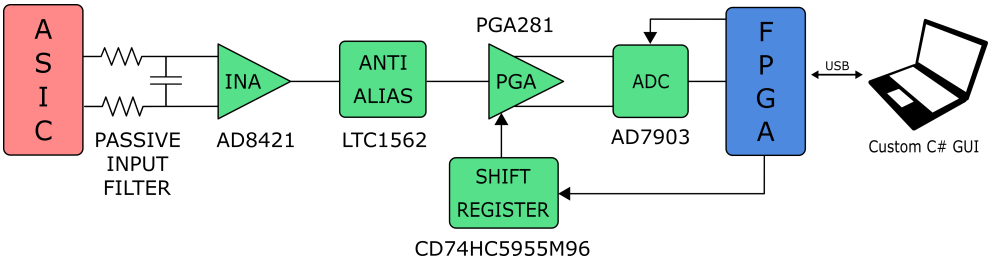

Figure 3.2: Schematic view of one of the 16 acquisition chains

frequency disturbances that might be coupled to the CLIPP or the electronic itself.

The schematic view of one hardware chain is provided in figure 3.2. It's worth recalling that each CLIPP has 2 of these channels dedicated on the board: one to extract the real part of the signal, one the imaginary.

3.1.1 Input RC network

The output of the ASIC is differential and is brought to the motherboard through shielded cables. The ASIC has an output resistance of $R_{ASIC} = 5\,\mathrm{k}\Omega$ that can be exploited to add a first pole to reduce the high-frequency results of the on-chip demodulation. To do so, it's possible to add a differential capacitance on the board, before the input of the first amplifier. The pole frequency is set to leave unaffected the dithering tones that might be superimposed to the input signal. For this reason, it was set to be

$$f_{pole} = \frac{1}{2\pi C R_{ASIC}} \simeq 200\mathrm{kHz} \tag{3.1}$$

that leads to an overall capacitive value of $C = 160\,\mathrm{pF}$. The equivalent circuit is shown in figure 3.3. The value of the C_{PCB} that has to be mounted on the board has to be computed taking into account also the capacitance of the shielded wires. The measurement of the wire capacitance was performed using the "E4980A" LCR meter, produced by Agilent. It showed that the cables adds a parasitic capacitance towards ground of $C_{WG} = 60\,\mathrm{pF}$ and a differential parasitic capacitance between the two wires that bring the differential signal of $C_{WW} = 20\,\mathrm{pF}$.

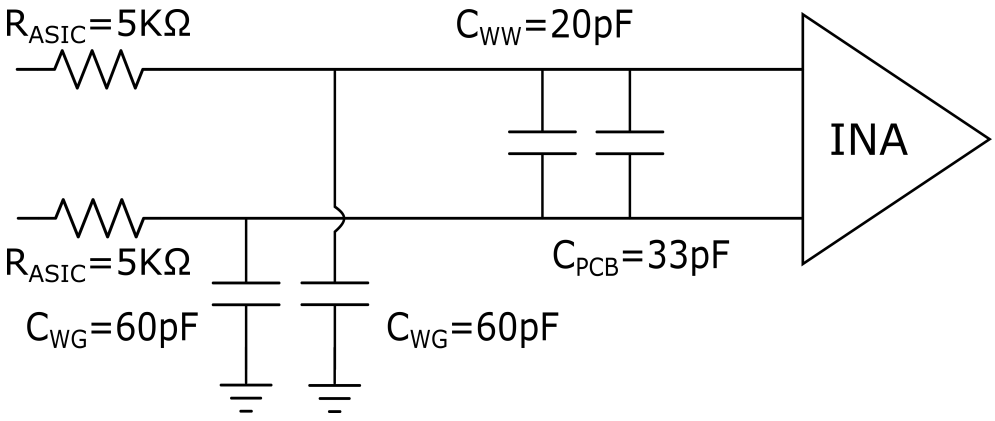

Figure 3.3: Equivalent circuit of the input network of each acquisition channel.

Therefore, because of the cables, each line experiences already an equivalent capacitance of 100 pF. To reach the target value of 160 pF, it's necessary to add 60 pF more. Since the capacity is mounted between the two differential lines, the actual value of the capacitance has to be halved. For these reasons, a capacitance of $C_{PCB} = 33\,\text{pF}$ was chosen.

3.1.2 INA

Aside from this first filtering action, the first thing that the motherboard has to do is to further amplify the signal in order to ensure that the noise of the following stages is completely negligible. To do so, the AD8421 instrumentation amplifier of Analog Devices was chosen, especially for its small low-frequency input noise power spectral density. In this way, the differential signal is amplified and converted into its single ended version, that is easier to be processed. The gain of this stage is set by a resistance R_G according to the equation

$$G = 1 + \frac{9.9\ \text{k}\Omega}{R_G} \tag{3.2}$$

as stated in the datasheet [18].

For this design, a resistance of 330 Ω was chosen, leading to $G = 31$, leading to an input referred noise spectral density of $\simeq (5\ \text{nV}/\sqrt{\text{Hz}})^2$

3.1.3 Anti-Alias

After the first amplification, it is necessary to filter out the unwanted harmonics. To do so, the LTC1562 active filter of Linear Technology was chosen. The frequency response of this filter can be changed by connecting different value of resistors to its pins, as indicated in the datasheet [28]. For this design, it was chosen to use a 4th-order Butterworth filter. This ensures a good flatness in the pass band, avoiding the necessity of a digital compensation of the attenuation. On the other side, the phase shift introduced by the filter is not negligible even at frequencies much lower than the pole. Since the second demodulation needed to extract the dithering signal is performed in the digital domain, and since the lock-in technique is phase-sensitive, this effect has to be taken into account. A detailed discussion will be provided in chapter 4.

The bandwidth of the filter is chosen, once again, to leave unaffected eventual dithering tones superimposed to the input signal. The maximum bandwidth of this component is actually limited to 150 kHz, so this is the final value chosen. The output of this filter is still single-ended.

3.1.4 Programmable Gain Amplifier

Gain programmability is a useful feature of the acquisition chain, since the amplitude of the signal to be measured might change for the reasons explained before, and every signal should exploit the ADC full scale range. To do so, a programmable gain amplifier (PGA) was added to the chain. The component chosen is the PGA281 of Texas Instruments. As stated in the datasheet [24], it has 4 digital pins that can be controlled to select the desired gain. The possible gains range from 0.125 up to 128, doubling from one configuration to the other.

To reduce the pins necessary on the FPGA, the 4 gain pins of the 8 amplifiers of the channels responsible to extract the real portion of the signal are all connected together. The same happens for the ones in the imaginary channels.

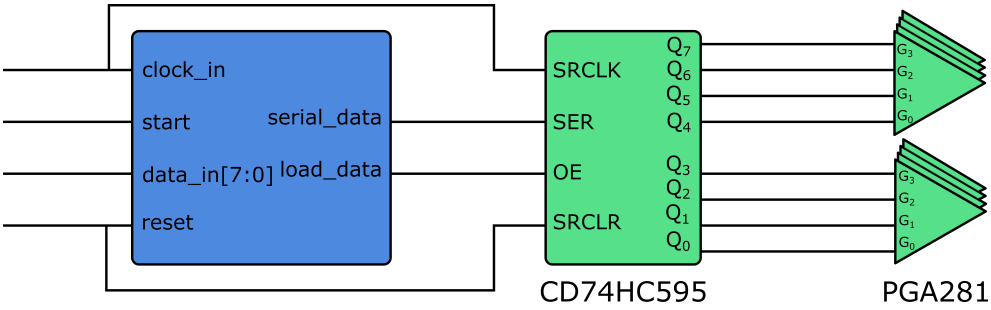

Figure 3.4: Structure of the control of the PGA281. In blue the entity inside the FPGA, in green the physical components mounted on the board.

Moreover, a 8-bit Serial Input Parallel Output (SIPO) shift register was preferred over a paralleled connection, therefore saving even more pins. The shift register chosen is the CD74HC595 of Texas Instruments and it communicates with the FPGA with a SPI protocol.

As stated in the datasheet [3], it features a pin for the master clock, one for the reset, one for the serial transmission of the configuration bits from the FPGA, and one chip select used to synchronize the transmission. In this way, only 4 connections are used to control all the 16 PGAs on the board.

VHDL code

The VHDL code to control the shift register is a single-process finite state machine (FSM). The FSM checks the status of a start *TriggerIn* signal. Once this trigger is asserted, the FPGA drives low the sync signal of the shift register and starts updating the value of the SDI line, one bit per clock cycle. Everything is synchronous to the master clock and has to respect the timing diagrams indicated in the datasheet [3].

Also, the FSM checks the status of a reset *TriggerIn* signal. When asserted, it causes the FSM to go into the "idle" state and to reset the value of the internal signals. The FPGA also drives low the active-low reset pin of the shift register, setting in this way all its output pins to zero. The whole structure is depicted in figure 3.4.

3.1.5 ADC

Once the signal is properly amplified and filtered, it can be digitized.

The output of the PGA is differential, so a differential-input ADC has been chosen, in particular the AD7903 of Analog Devices. It is a 16-bit successive approximation ADC with an input dynamic from $-V_{ref}$ to $+V_{ref}$. This reference voltage can be chosen freely from 2.4 V up to 5.1 V. For this design, the reference was set to 3 V. In this way, the LSB is

$$LSB = \frac{FSR}{2^{no_bit}} = 91.55 \text{ µV} \tag{3.3}$$

The ADC communicates to the FPGA with a SPI protocol. A serial protocol was chosen in order to limit the number of pins needed to connect the ADCs to the FPGA. In this way, each ADC needs just one data line, instead of the 16 needed with a parallel transmission protocol. This feature is particularly useful since 16 ADCs are present on the board. Each ADC has also a chip select pin used to synchronize the transmission. Finally the master clock, provided by the FPGA, is shared between all the ADCs, to ensure a synchronous acquisition even between different channels.

According to the datasheet [9], the clock frequency can be up to 80 MHz. A more relaxed frequency of 40 MHz ($T_{clk} = 25$ ns) was chosen for this design.

This ADC has a worst-case conversion time of 710 ns. This means that, before starting to send out the data, it's necessary to wait for $\lceil 710 \text{ ns}/25 \text{ ns} \rceil = 29$ clock cycles. In addition, to send the data to the FPGA, the ADC needs 16 clock cycles for the serial output, plus one more to communicate that a new conversion can start. Therefore, it's $17 * 25$ ns $= 425$ ns.

Overall, this setup allows a maximum sampling frequency of

$$f_{samp,MAX} = \frac{1}{725 \text{ ns} + 425 \text{ ns}} \simeq 870\text{ksps} \tag{3.4}$$

In this design, the sampling frequency is lowered to 625 ksps, that is exactly a factor 2^8 lower than 160 MHz, for reasons that will be clarified in chapter 5.

VHDL code

The VHDL code to control the ADC is again a single-process FSM. It checks the status of a *TriggerIn*, used as start signal. Once it's asserted, the FPGA drives high the sync pin of the ADCs and starts the conversion and acquisition, that keeps running until another *TriggerIn*, used as stop signal, is asserted.

The FSM, depicted in figure 3.5, is designed to have the following states:

- **Idle:** while waiting for the start *TriggerIn*, the FSM is inactive;
- **Wait for conversion:** once the start *TriggerIn* is asserted, the FSM tells the FPGA to drive high the sync pin to start the conversion of the sampled data, as stated in the timing diagrams in the datasheet [9]. To properly convert the analog value, the FSM has to stay in this state for at least 29 clock cycles, as explained above. Actually, to lower the sampling frequency to the desired value, the FSM stays in this state for 47 clock cycles. After that, the sync pin is driven low. This communicates to the ADCs to start sending out the converted data from the next clock cycle. The FSM can now move the "acquisition" state;
- **Acquistion:** the FSM stores the serially transmitted bits into 16 shift registers, one for each ADC. The ADCs send the data on the falling edge of the clock, so a correct reading is ensured by simply designing the process to be synchronous to the rising edge. The FSM has to stay in this state for exactly 16 clock cycles. After that, it moves to the "end of acquisition" state;
- **End of acquisition**: the acquired data is brought to the following stages of the digital acquisition chain to perform the intended digital signal processing. A *data_ready* signal is driven high, and it's used as a clock signal for the following stages, that are then synchronous to the sampling frequency of the ADCs. Then, if the stop *TriggerIn* has been asserted during this procedure, the FSM goes into the "idle" state.

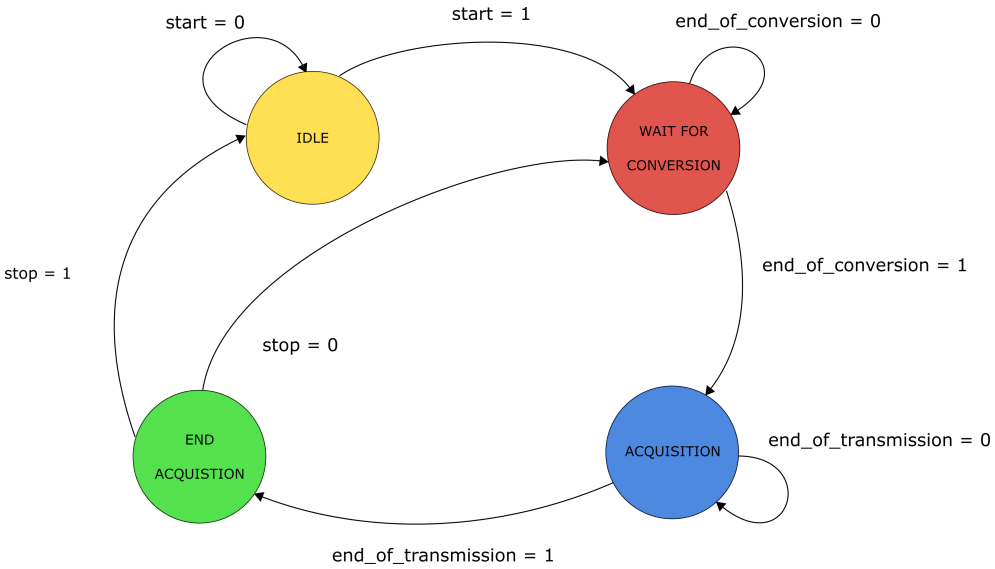

Figure 3.5: State diagram of the FSM of the ADC controller.

Otherwise, it goes back into the "wait for conversion" one, and keeps running in cycle.

Overall, 64 clock cycles are needed to go through the whole FSM, meaning that the sampling frequency is $\frac{1}{64}40$ MHz $= 625$ KSPS.

An asynchronous hard reset *TriggerIn* resets the internal registers and stops the FSM in any moment, bringing it to the "idle" state.

Quantization noise of the ADC

The ADC introduces a certain quantization noise that has to be discussed. Given an LSB value of 91.55 µV, the RMS value of the quantization noise is

$$\sigma_q = \sqrt{\frac{LSB^2}{12}} = 26.43 \text{ µV} \tag{3.5}$$

This is not the actual RMS noise experienced by the signal though. In fact, since the bandwidth of the signal is lower than the Nyquist frequency, it's possible to add a digital low-pass filter after the acquisition, that will effectively reduce the RMS value of the quantization noise. In fact, the input noise spectral density can be computed as

$$S_{q,in} = \frac{\sigma_q^2}{f_s/2} = (23.64 \text{ nV}/\sqrt{\text{Hz}})^2 \tag{3.6}$$

This noise spectrum is then filtered with an equivalent bandwidth BW_{eq}, that leads to a RMS noise of

$$\sigma_{out} = \sqrt{S_{q,in} BW_{eq}} = \sqrt{\sigma_q^2 \frac{BW_{eq}}{f_s/2}} = 507.5 \text{ pV} \tag{3.7}$$

assuming $BW_{eq} \simeq 6$ Hz.

As a consequence, even if the reference voltage and the number of bits have been fixed, the RMS value of the quantization noise can be reduced with respect to the original one by choosing a high $\frac{f_s}{2BW_{eq}}$ ratio. This quantity is often referred to as "oversampling factor", and the general technique is known as "oversampling". In this specific case, the narrow bandwidth of the signal and the high sampling frequency reflect in a high oversampling factor, which in turn makes the quantization noise negligible with respect to the electronic noise of the ASIC, even with a low PGA gain.

The implementation of the digital low-pass filter will be addressed in the chapter 5.

3.1.6 Data transfer via USB 3.0

To properly acquire a big amount of data, Opal Kelly provides a specific endpoint, that is the *PipeOut*. The USB interface is fast enough to properly acquire all the data, but it works only when the scheduler of the operative system allocates the resources to the task that controls it. Therefore, this operation is basically asynchronous with respect to the sampling frequency of the ADC. The integrity of the transmitted data should always be guaranteed though. Because of the different working frequencies between the two interfaces, it's then necessary to save and store the data in the FPGA before transferring them to the PC. Since the *PipeOut* doesn't allow the possibility of storing the data by itself, a memory structure, such as a FIFO, should be used in between, as suggested by the producer himself [14].

The FIFO is provided by Xilinx as an IPCore. Thanks to the optimized implementation of the proprietary FIFO, the amount of resources used is reduced to the minimum, even though one memory per channel is needed.

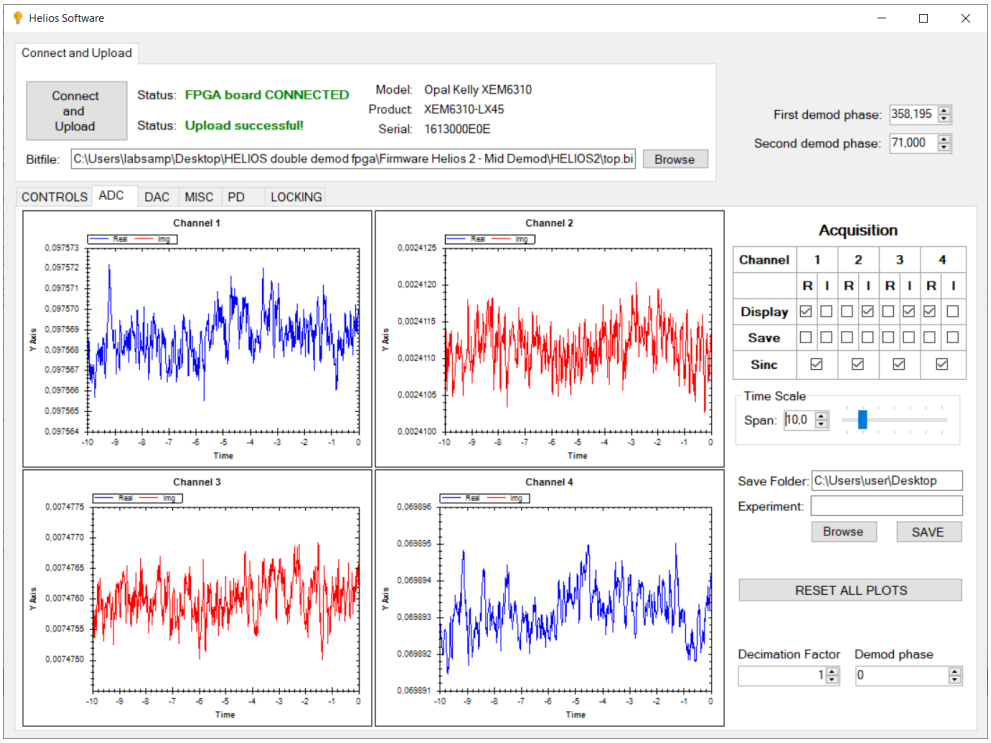

Figure 3.6: UI to control the acquisition chain.

On the FPGA side, the *PipeOut* interface has a 32-bit word width. On the other hand, the *PipeOut* data are transferred over the USB interface in 8-bit words. This means that the user interface has to reconstruct the integrity of the transmitted signal before doing any operations.

3.1.7 User Interface

Figure 3.6 provides a screenshot of the UI designed to properly operate on the acquisition chains.

In particular, custom controls allow to choose whether to display the real and/or the imaginary part for each channel individually. The value displayed in the graph is the value of the signal at the input of the board, i.e. at the output of the ASIC. This means that, each sample, before being plotted or stored, has to be normalized by the gain of the acquisition chain. Notice also how there are only 4 graphs since, with the ASIC considered, it's possible to read only 4 CLIPPs simultaneously.

The interface also allows the user to choose which channels to save to a *.csv* file. The "sinc" box instead allows to activate an additional filter that will be further discussed in chapter 5.

It's also possible to add a decimation action to the data transferred through the USB. This is simply implemented by discarding out each sample except one every decimation factor. This feature is especially useful when a long acquisition has to be performed. Indeed, without decimation the amount of data that would be saved would be too big to be swiftly processed.

3.2 Stimulation chain

The CLIPP has to be stimulated with a high-frequency sinusoid in order to properly bypass the access capacitances. The frequency of this signal may vary between 100 kHz and 10 MHz to adapt to different geometries of the sensor. The maximum amplitude of the stimulus should be as high as possible in order to maximize the current signal in the sensor. This is true as long as the electromagnetic field generated does not interfere with the signal that passes through the waveguide. Such effects are yet to be observed with the available circuitry. So, at the moment, the limit is the power supply of the amplifiers used in the stimulation chain, that was designed to be able to generate a 10 V sinusoid.

Given the different nature of the CLIPP sensors available and the different working conditions of the photonic chips, the stimulation chain should be designed to have the frequency and the amplitude of the sinusoid programmable through the FPGA. The most flexible option is to generate the sinusoid inside the FPGA through a DDS (Direct Digital Synthesizer) and convert it into an analog signal through a DAC.

Given the high frequency nature of the stimulus wave, the DAC has to be intrinsically fast. For this reason, a DAC featuring parallel input pins was chosen over one with a serial communication protocol. The trade-off speed vs. pins usage is evident. The output of this DAC is properly conditioned and

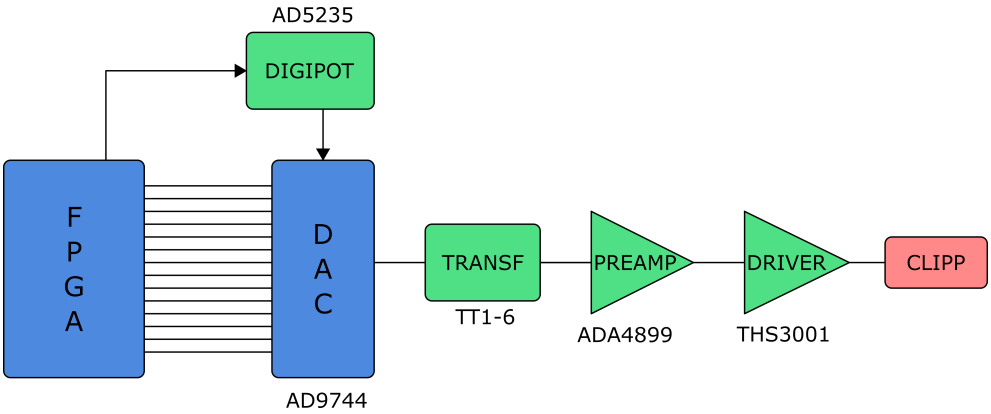

Figure 3.7: Schematic view of one of the 2 stimulation chains.

amplified by the analog chain in order to reach the desired amplitude value, that is sent to the sensors.

All the CLIPPs are stimulated with the same waveform. This means that virtually as many CLIPPs as desired can be connected to a single stimulus chain. The only limitation comes from the maximum capacitive load that can be connected to the driver output at the end of this chain.

The board also features another identical stimulation chain. This was conceived to generate once again a programmable sine wave that can reach up to 10 V and 10 MHz. The purpose of this sinusoid is to be summed to the voltage provided by the actuation chain, as explained in the next paragraphs. In this way it's possible to perform a thermal modulation faster than the one allowed from the DACs in the actuation chain, that work at a limited speed.

3.2.1 DDS

The very beginning of the stimulation chain is the creation of the digital values that will be fed to the DAC to be converted in analog values and used as stimulus. The waveform that has to be produced is a sinusoid. To do so, a simple solution is to use a DDS (Direct Digital Synthesizer).

In its simplest form, a DDS can be implemented from a reference clock, an address counter and a memory, as depicted in figure 3.8 [4]. The digital words that correspond to a whole cycle of a sine wave is stored into this memory. The

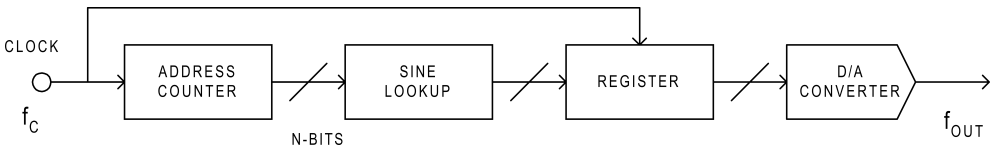

Figure 3.8: Schematic view of a DDS architecture based on an address counter and a sine look-up table.

address counter points to the memory cell that contains the current amplitude value. In this sense, the memory works as a look-up table of a sinusoid. The address counter is then increased every clock cycle.

This simple structure does not provide any flexibility. In fact, once the memory is programmed, it would generate always the same sine wave, without the possibility to tune neither the frequency nor the phase.

A solution is to substitute the address counter with a phase accumulator register and a phase-to-address converter, that addresses the proper memory cell to eventually create the desired wave. The frequency of the sinusoid is set by choosing a phase increment word, that the accumulator keeps summing every clock cycle. The bigger this phase increment value, the steeper the growth of the accumulator result, the higher the speed at which the DDS goes through the whole memory. In this way it's possible to change the synthesized frequency without changing the clock fed to the structure, but rather by acting on the input phase increment. The generated wave frequency can be computed as

$$f_{out} = \frac{\Delta\Theta}{2^N} f_{clk} \tag{3.8}$$

where $\Delta\Theta$ is the phase increment provided by the user, N is the phase width (i.e.the number of bits of the phase increment), and f_{clk} is the system clock frequency.

Usually, in hardware structures, the LUT is directly programmed using some configurable logic blocks. This means that the phase-to-address converter and the memory are conceptually and practically embedded into one "phase-to-amplitude" block. This structure is depicted in figure 3.9.

It's also worth noticing how most DDS architectures exploit the symmet-

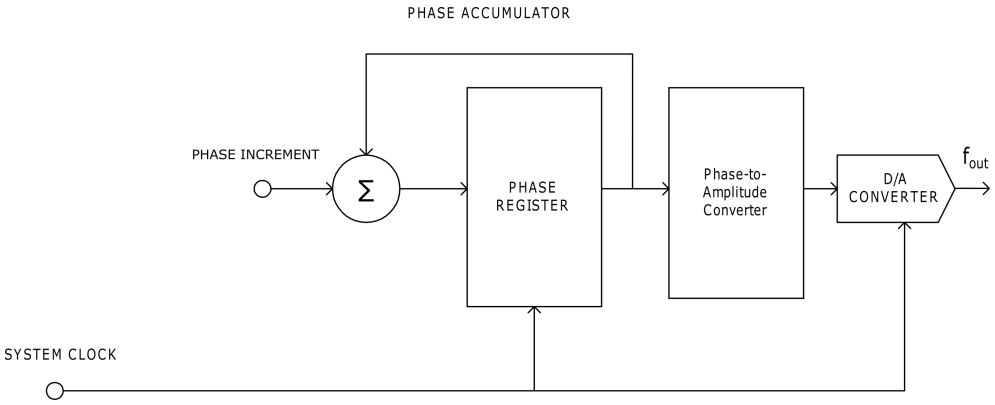

Figure 3.9: Schematic view of a DDS architecture based on a phase accumulator and a phase-to-amplitude converter.

rical nature of a sine wave and store only 1/4 of a complete sine into memory, and then use some mapping logic to synthesize the whole wave.

Practical Implementation

Rather than coding from scratch the DDS used in the project, it was chosen to use an IPCore provided by Xilinx. This allows an easy yet flexible design, besides an optimized use of resources. DDSs are in fact really resources-demanding and, given the complexity and the size of this platform, an optimized solution is advised. The tool generates a component whose entity is depicted in figure 3.10. Some of these ports are optional, meaning that they have to be activated in the GUI of the IPCore generator if needed. In particular, the following ports are present:

- *aclk* input port. It's the master clock of the DDS. The clock provided has to match the one indicated in the GUI of the IPCore generator;
- *aclken* input port. When driven low it invalidates the output of the DDS and sets it to zero, but it doesn't stop the phase accumulator. Optional;
- *aresetn* input port. When driven low for at least two clock cycles, it resets the DDS. Once released, the DDS starts working from the beginning. Optional;

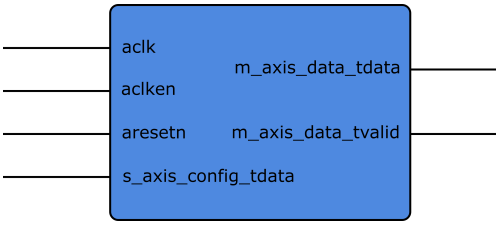

Figure 3.10: Schematic view of a single DDS entity.

- *s_axis_config_tdata* input port. This port is optional and is active if the phase increment and/or the phase shift (referred to as "phase offset" in the datasheet [16]) parameters are programmable. The length of this vector depends on the phase width designed and on whether both the frequency and the phase shift are set to programmable, and is then extended to fit a byte boundary, as depicted in 3.11(a);

- *m_axis_data_tdata* output port. The DDS is able generate automatically both sine and cosine waves, that are grouped into a single output word, as depicted in figure 3.11(b). The sine and cosine words are sign extended, so that the output always fits a byte boundary;

- *m_axis_data_tvalid* output port. It goes high if the DDS doesn't encounter any errors in the generation of the sinusoid.

Some more optional ports can be configured in the tool, but they are not described here since they were never used in this design.

For this design, it was chosen to use a 160 MHz clock and 16-bit phase width. In this way the frequency resolution achieved is

$$\Delta f = \frac{f_{clk}}{2^{16}} \simeq 2.44\,\text{kHz} \tag{3.9}$$

The output is 14-bit wide, since it has to be fed to a 14-bit DAC.

This DDS is useful to create the stimulus wave fed to the CLIPP but also the ones used for the on-chip demodulation. Actually, since the demodulation is performed with a switch-based mixer, the signals that have to to be provided to the chip are a square wave and its quadrature version. To realize them it's

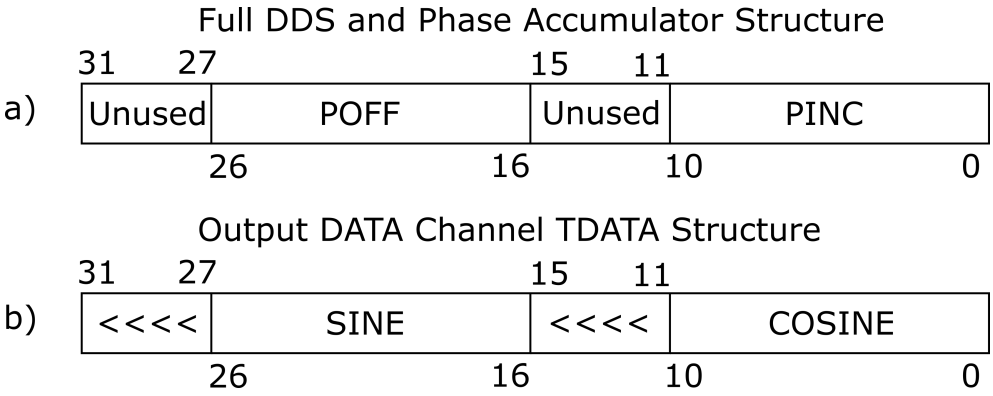

Figure 3.11: Structure of the input configuration port when the phase increment and the phase offset are 11-bit long (a). Structure of the output port when the output signal is 11-bit long (b). "«" indicates sign extension.

possible to simply extract the MSB of the output sine and cosine of the DDS and map them directly outside the FPGA.

This implies that, in the first version of the system, there are two DDS: one to generate the stimulus, with fixed phase offset set to zero, and one to generate the two demodulation square waves, that instead has programmable phase offset in order to compensate the phase shift introduced by the electronic components in the stimulation chain. This problem will be further addressed in the next chapter.

3.2.2 Digital to analog converter

The digital word generated by the DDS has to be fed to a high-bandwidth DAC. The DAC chosen for this purpose is the AD974 of Analog Devices [1]. It features a parallel 14-bit input and a low conversion time, that allows a maximum operating frequency of 210 MSPS. To ensure synchrony between the DAC and the DDS, the same 160 MHz clock of the DDS is used to update the DAC.

The DAC has a current mode differential output, whose full scale range can vary from 2 mA to 20 mA. The value of the full scale range is set thanks to an internal voltage reference and an external resistor, that generates a reference

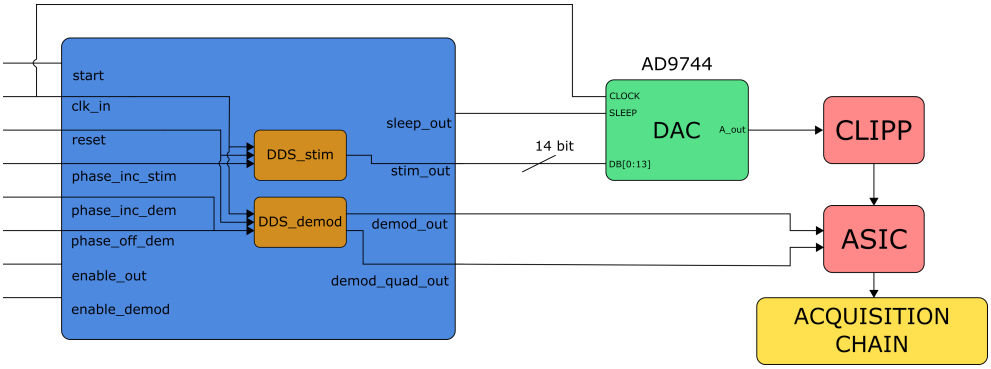

Figure 3.12: Structure of the control of the whole stimulation chain. In blue is depicted the entity of the FPGA, in green the physical components on the motherboard, in red the ones in the integrated circuits.

current for the DAC. Rather than using a fixed resistor, it was preferred to use a digitally controlled potentiometer. In this way, by changing the value of the resistor, it's possible to change the reference current, hence the gain of the chain.

VHDL code

Given the simplicity of the control of the DAC, a single FSM was designed to handle both the DAC and the DDS. The overall structure is depicted in figure 3.12.

The FSM, when in idle state, checks the status of a start *TriggerIn*. When asserted, it sends to the DDS the selected phase increment and phase offset and turns the DAC on through the *sleep* port. There's also a reset *TriggerIn*, that resets and stops the DDS and turns off the DAC.

For diagnostic reasons, two more control signals are added: one *enable_stimulus* that keeps to zero the output of the DAC even if the DDSs are running, one *enable_demod* that keeps to zero the demodulation waves fed to the mixer, even if the forcing signal is active.

Since the DAC updates its output on the rising edge of the clock, the process that controls the FSM has to be synchronous to the falling edge of the clock in order to satisfy the timing constraints.

The entity used to control the second DAC for the other stimulus chain is similar but way simpler than this, since it has only one DDS and does not need any extra control signals. It is not described any further here to avoid redundancy.

3.2.3 Digital potentiometer

The digital potentiometer chosen to generate the reference current of the DAC is the AD5235 of Analog Devices [22]. This component works with a simple SPI protocol, with one serial-input pin, one master clock, and one chip select that works as sync signal.

VHDL code

The VHDL code is again a simple FSM that controls the status of a start *TriggerIn*, and sends the sync signal and the configuration bits to the component, one bit per clock cycle. There's also a reset *TriggerIn*: when asserted, the FSM is brought to idle state and all the internal signals are resetted.

3.2.4 Transformer

The output of the DAC AD9744 is a differential current that has to be converted into a single-ended voltage. To do so it was chosen to use a transformer, as suggested on the datasheet of the DAC itself [1]. This allows a virtually noiseless AC coupling and a direct conversion of the signal into a voltage. The TT1-6 transformer of Mini-circuits was chosen. Small transformers do not work well with low-frequency signals, but since this chain is meant for high-frequency sinusoids, this issue does not impair a correct stimulation of the CLIPP. In fact, the transformer chosen works properly down to 4 kHz [25], lower than the minimum frequency intended to be used.

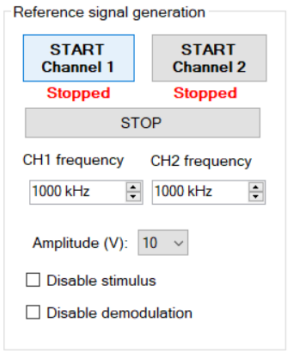

Figure 3.13: UI of the stimulation chain.

3.2.5 Preamplifier and driver

The output of the transformer TT1-6 has to be amplified in order to reach the desired amplitude. The ADA4899 of Analog Devices was chosen for its low noise performances [27]. This preamplifier is used to perform a first amplification by a factor 3, to make the noise of the following stage negligible. A larger gain is not possible, since the component has an output voltage dynamics of $\pm$ 3.5 V. A second amplification stage is thus needed. The THS3001 operational amplifier of Texas Instruments was chosen. It features a $\pm$ 12 V power supply and a high slew-rate, allowing a correct stimulation of all the CLIPPs with a sinusoid of up to 10 V and 10 MHz. Moreover, this stage can drive a high capacitive load without suffering from instability [2], so it's suitable to drive many CLIPPs simultaneously.

3.2.6 User Interface

The stimulation chain does not require a complicated user interface. As shown in figure 3.13, it needs two buttons to activate and stop the generation of the sinusoids. The frequency of both the chains can be programmed with custom controls. A drop-down menu can be used to program the amplitude of the wave. Through the C# code, the potentiometer is properly programmed according to the selected amplitude in order to generate the correct reference current for the DAC. This is completely transparent to the end user, that only

selects the desired amplitude.

3.3 Actuation Chain

The ultimate purpose of this system is to perform a control action. To do so, some actuators have to be properly controlled. As already stated, the actuators are here heaters that, by locally changing the temperature in the circuit, change the equivalent optical path of the guide. The system must then be able to control all the heaters simultaneously and independently from one another. To do so, 16 parallel actuation chains were designed. The main component of each chain is a DAC that translates the digital level of the desired voltage into its analog equivalent.

To perform a dithering modulation, a DDS is used, similar to the ones discussed in 3.2.1. Given the high numbers of actuation chains, it was not possible to use DACs with a parallel input word, but serial input DACs were preferred. Since the digital word has to be sent bit by bit, the update frequency of the DAC is intrinsically lower, limiting in this way the maximum achievable bandwidth, that is around few hundreds kHz. For this reason, the clock frequency used for these DDSs is lowered to 5 MHz.

As already stated in 3.2, if a fast dithering action is needed, a second stimulation chain was designed. This fast sinusoid can be summed to the output of any of the DACs in the actuation chain.

Because of the high current that has to be provided to the heaters, it's not possible to connect them directly to the output of the DACs. A suitable driver is then added after each DAC to provide the necessary power. The whole structure is depicted in figure 3.14.

3.3.1 DAC

Given the high number of DACs needed, it was important to choose a component with many converters in the same package. The choice was for the AD5764R of Analog Devices [7]. It features 4 16-bit DACs in the same package,

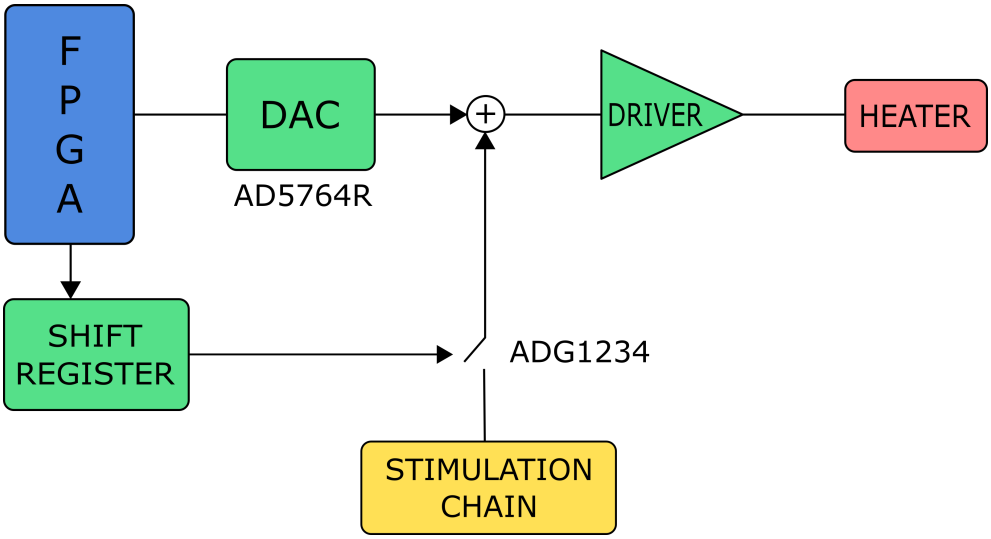

Figure 3.14: Schematic view of 1 of the sixteen actuation chain.

each of them individually programmable with a common SPI interface. The output has a nominal range of ± 10 V, so enough for the purpose. Since there are 16 parallel actuation chains, there are then 4 chips on the board.

Each chip has a pin for the chip select that works as sync signal, one for the serial data in and one for the master clock, that can go up to 30 MHz. The DAC is designed to accept only 24-bit long instruction words, where the 16 LSBs are the value that has to be converted, while the 8 MSBs are the instruction bits that select to which DAC the data have to be fed. The interface needs 3 extra clock pulses between a conversion and the following one. Overall, it's then possible to send an instruction to the chip every 27 clock pulses. It was chosen to feed to the DAC a 27 MHz clock, allowing to update a single channel at 1 MHz, or all of them, in turn, at 250 kHz each.

VHDL code

To better handle the communication protocol of the DAC, it was chosen to separate the design in 3 functional blocks, as depicted in figure 3.15. So, the following 3 entities are used:

- **Data generation:** the first entity has to create the 16-bit word that

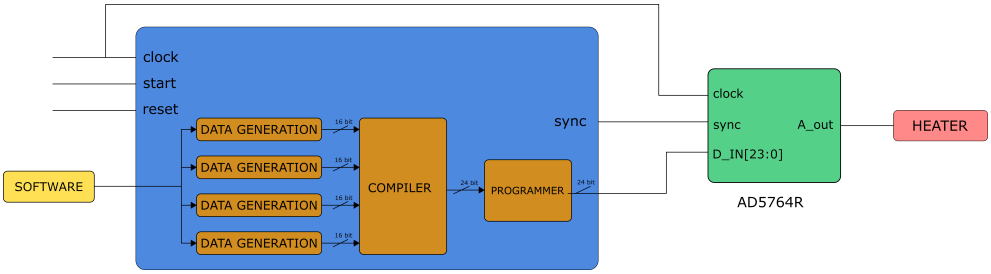

Figure 3.15: Structure of the control of one chip, hence 4 actuation chains.

has to be fed to each single DAC. This is the sum of the DC voltage chosen and the eventual sinusoid superimposed to it. This structure is replicated 16 times, 4 for each chip;

- **Compiler:** this second entity creates the 24-bit instruction that has to be fed to the chip. This instruction is the concatenation of the 16 bits generated in the data generation entity, and the correct address of the DAC to be updated. This structure has to be replicated only 4 times, one for each chip;

- **Programmer:** the last entity handles the communication protocol and the buses of each chip. There are 4 of them, one for each chip.

All these functional blocks are embedded inside an entity that controls everything with a single-process FSM. Therefore, there is one entity per chip. This FSM checks the status of a start *TriggerIn*, and, once started, turns on the corresponding chip. A reset *TriggerIn* is used to stop the FSM and to set to zero the value of the internal registers. The actuation voltage, the dithering amplitude and frequency and other control signals all come from the custom GUI.

3.3.2 Driver

The maximum output current of each DAC is only $\pm$ 3 mA. However, the impedance of the heater is very low, around few hundreds of Ω. Since the maximum value of the voltage across the heater is 10 V, the current needed

is in the order of tens of mA per heater. For this reason, a driver that can provide such high output current has been connected to the output of each DAC. For the purpose, that AD8513 of Analog Devices was chosen. According to the datasheet [23], the output current can go up to $\pm$ 70 mA, when properly biased.

The amplifier is used in inverting configuration. In this way, the virtual ground can be used as summing node to easily add the fast sinusoid of the second stimulation chain to the DC value of the heater.

3.3.3 Switches

A switch is used to choose whether to connect the output of the second stimulation chain to the virtual ground of the driver, as depicted in figure 3.14.

These switches have to withstand a high voltage, since the high-frequency sinusoid might be up to $\pm$ 10 V, and have to work at high frequencies, up to 10 MHz. The ADG1234 of Analog Devices was chosen in this design, since it respects all the characteristics above [17].

Since there's no need to update the working condition of these switches in real time, it was chosen to save some pins of the FPGA by using two Serial Input Parallel Output (SIPO) 8-bit shift registers. The component is again the CD74HC595 of Texas Instruments, the same used in the acquisition chain [3]. Therefore, also the VHDL code is exactly the same explained before.

3.3.4 User Interface

To configure each heater independently from the others, 16 control interfaces are implemented in the software, as shown in figure 3.16. In this way, many parameters can be individually tuned. The user can select the DC value of the heater, enable the dithering sinusoid, choose its frequency and amplitude and activate the sum of the fast sinusoid. It's also possible to drive the heaters with a voltage ramp, that can be useful to scan the transfer function of certain photonic devices.

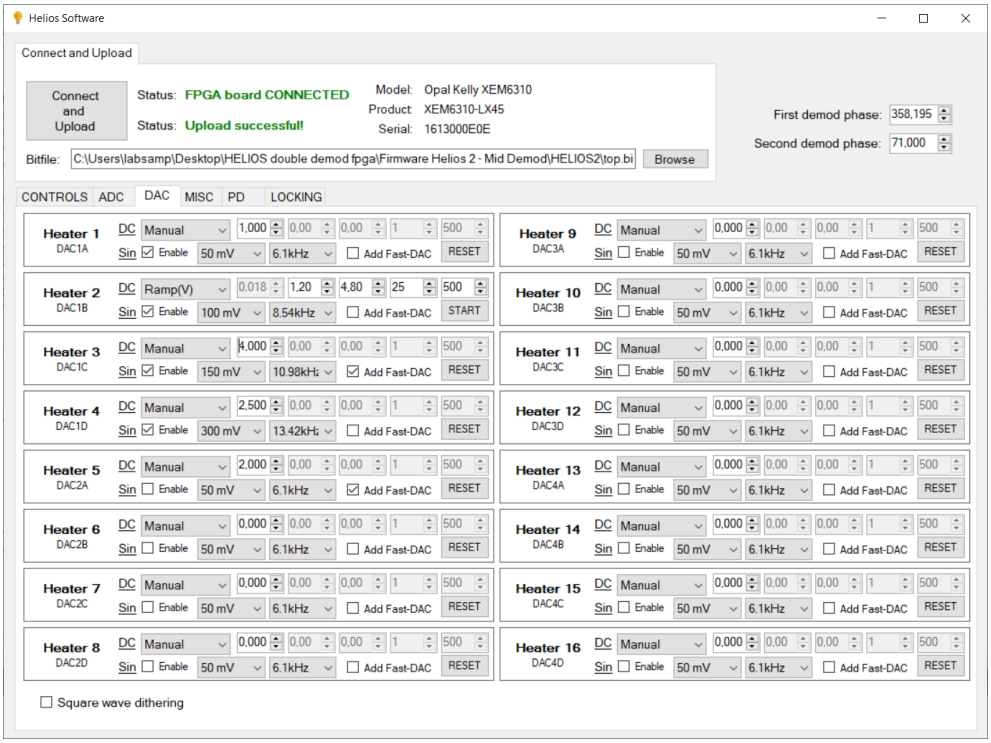

Figure 3.16: UI to control all the actuation chains

3.4 ASIC control chain

The motherboard also controls the custom integrated circuit used to perform the preamplification and the demodulation of the signal. In particular, as shown in figure 3.17 , this ASIC control chain has to:

- send the selection bits to configure each multiplexer. There are 4 8-input multiplexers, so overall 12 control bits are needed. These are mapped on 12 pins of the FPGA. In the graphic interface the end user selects the desired input for each multiplexer;
- send the square waves to drive the mixers, as already discussed in the paragraph 3.2.1;
- control the DAC that sets the biasing voltage of the pseudoresistor. This is a single voltage signal that doesn't have to be updated in real time. The DAC has then to be as simple as possible. The AD5687R was

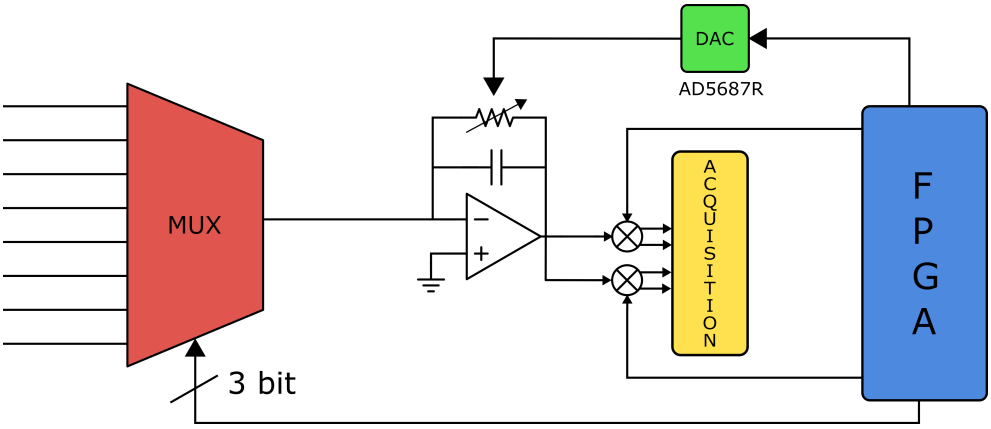

Figure 3.17: Schematic view of the ASIC control chain

chosen for this purpose [10]. It communicates with the FPGA with a SPI protocol, so it has a pin for the serial input, one for the chip select and one for the master clock. The VHDL code is a simple FSM that checks the value of a start *TriggerIn* signal, and updates the serial output once it's asserted. A reset *TriggerIn* is also used to reset both the FSM and the DAC.

Chapter 4

Two-step demodulation technique

The presented system is intended to read very small signals. For this reason, a tailored ASIC with very low input noise power spectral density was designed to perform a first preamplification. The ASIC has also to perform an on-chip demodulation, that was intended to bring the signal in baseband. After that, the output of the ASIC is fed to the acquisition chain, as described in the previous chapter.

A first problem of this readout strategy arises from the fact that the motherboard has a non-negligible DC offset, slightly different in each channel. This offset has then to be individually measured and compensated.

Another issue is that the signal, once in baseband, experiences all the $1/f$ noise of the components in the acquisition chain. Even if all the gain stages are properly set, the high intensity of the $1/f$ noise might be the dominant contribution to the overall readout noise, limiting the accuracy of the measurement.

Indeed, in a well-designed acquisition chain, the stages after the preamplification should introduce a negligible noise contribution, so that the input referred noise of the whole chain is dominated by the first stage noise. To reach such a condition in our system, it was necessary to change readout strategy

to escape from the $1/f$ dominant contribution. The idea is then to perform a two-step demodulation. The first one is still performed on the ASIC, but the demodulator works at a different frequency with respect to the stimulus one. In this way, the output of the ASIC is not in baseband, but at a certain intermediate frequency chosen by the user, possibly larger than the 1/f noise corner frequency. This frequency acts as a carrier for the signal, that can be properly amplified without experiencing the $1/f$ noise, and then digitized correctly by the ADCs.

To extract the information of interest a second demodulation is then necessary. This demodulation is performed in the digital domain. Notice that, in this system, it's not possible to avoid the first on-chip demodulation and perform a single digital demodulation, since the frequency of the stimulus wave is too high (up to 10 MHz) to be properly processed and digitized with the chosen hardware.

This solution also allows an easy cancellation of the offset of each acquisition chain, since the information to be recovered is now in the amplitude of the intermediate sinusoid and not in its DC value.

4.1 Noise of the acquisition chain

As stated above, the main problem of the acquisition chain is the presence of a strong $1/f$ noise. To assess only the effect of the motherboard, the ASIC was detached from the motherboard and the "MXA N9020A" spectrum analyzer of Agilent Technologies was used to measure the noise power spectral density of the acquisition chain. The result is depicted in blue in figure 4.1. To test the correct operation of the board, the measurement was also repeated by acquiring the noise directly with the ADCs of the acquisition chain. The acquired signal was then digitally demodulated at different frequencies and low pass filtered with a narrow band (~ 1 Hz) low pass filter, in order to obtain the whole noise spectrum. The result is depicted in red in figure 4.1, showing a very good matching with the previous measurement and demonstrating the

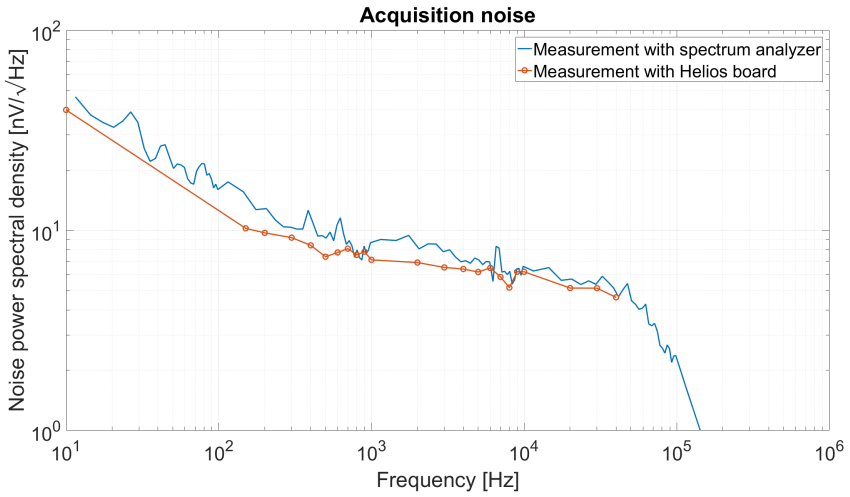

Figure 4.1: Input noise power spectral density of the motherboard. In blue the measure with the net analyzer, in red the one with the motherboard itself. The pole is actually lower than the one introduced by the Butterworth filter because the PGA is set to its maximum gain and has a lower bandwidth.

correct operation of the board.

4.2 Theory of operation

4.2.1 First demodulation

As already stated in the previous chapter, to perform the measurement the user has to select the stimulus frequency in the graphical interface. The software then computes the phase increment that has to be fed to the DDS responsible for the generation of the stimulation signal. With this new approach, the user has also to select another frequency f_{mid}, that is the intermediate frequency desired at the output of the ASIC. The first on-chip demodulation will then occur at $\omega_{dem} = \omega_{stim} - \omega_{mid}$.

In fact, the signal at the output of the TIA can be written as

$$s(t) = Ssin(\omega_{stim}t + \varphi) = Asin(\omega_{stim}t) + Bcos(\omega_{stim}t) \tag{4.1}$$

where A is the amplitude of the real part and B of the imaginary one. These are the quantities of interest that have to be extracted at the end of the

processing chain.

The analog mixer works with two square waves in quadrature with each other that have an angular frequency of ω_{dem}. The in-phase square wave can be Taylor-expanded as

$$d_{mixer}(t) = \frac{4}{\pi}\sin(\omega_{dem}t) + \frac{4}{3\pi}\sin(3\omega_{dem}t)... \tag{4.2}$$

Neglecting the high frequency harmonics and the $\frac{4}{\pi}$ term (compensated in the software), the in-phase output of the mixer is

$$y_{in-phase}(t) = [A\sin(\omega_{stim}t) + B\cos(\omega_{stim}t)]\sin(\omega_{dem}t) \tag{4.3}$$

Recalling the Werner's formulas for which

$$\begin{aligned} \sin\alpha\sin\beta &= \frac{1}{2}[\cos(\alpha-\beta) - \cos(\alpha+\beta)] \\ \cos\alpha\cos\beta &= \frac{1}{2}[\cos(\alpha-\beta) + \cos(\alpha+\beta)] \\ \sin\alpha\cos\beta &= \frac{1}{2}[\sin(\alpha-\beta) + \sin(\alpha+\beta)] \end{aligned} \tag{4.4}$$

it is

$$\begin{aligned} y(t) = \frac{A}{2}\{cos[(\omega_{stim}-\omega_{dem})t] - cos[(\omega_{stim}+\omega_{dem})t]\}+ \\ +\frac{B}{2}\{-sin[(\omega_{stim}-\omega_{dem})t] + sin[(\omega_{stim}+\omega_{dem})t]\} \end{aligned} \tag{4.5}$$

where $\omega_{stim} - \omega_{dem} = \omega_{mid}$, as already stated. The high-frequency harmonics resulting from this demodulation are filtered out by the anti-alias filter in the acquisition chain. So, neglecting also all the scale factors that are compensated via software, the resulting signal at the input of the ADC is

$$y_{in-phase}(t) = A\cos(\omega_{mid}t) - B\sin(\omega_{mid}t) \tag{4.6}$$

Instead, the quadrature component of the signal, obtained with the quadrature square wave fed to the mixer, is

$$\begin{aligned} y_{quad}(t) = [A\sin(\omega_{stim}t) + B\cos(\omega_{stim}t)]\cos(\omega_{dem}t) = \\ = \frac{A}{2}\{\sin[(\omega_{stim}-\omega_{dem})t] + \sin[(\omega_{stim}+\omega_{dem})t]\}+ \\ +\frac{B}{2}\{\cos[(\omega_{stim}-\omega_{dem})t] + \cos[(\omega_{stim}+\omega_{dem})t]\} \end{aligned} \tag{4.7}$$

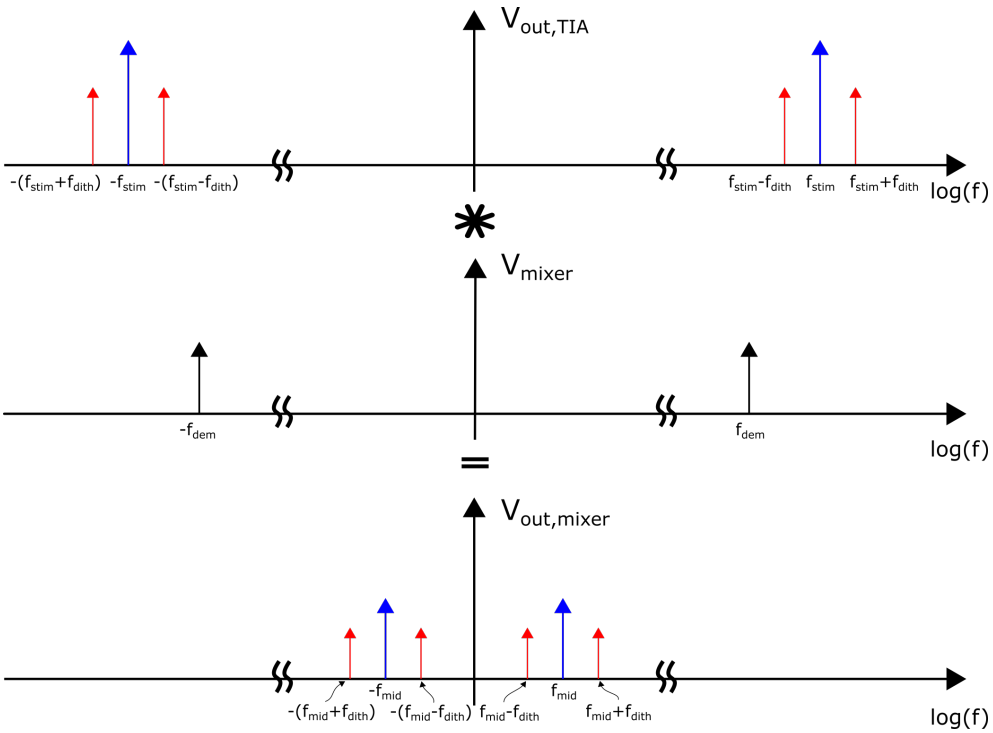

Figure 4.2: Effect of the first demodulation on the output signal of the TIA. In blue is represented the information about the admittance of the waveguide, in red the eventual dithering tones superimposed to it.

Neglecting again the high-frequency harmonics and the scale factors, it is, at the input of the ADC

$$y_{quad}(t) = A\sin(\omega_{mid}t) + B\cos(\omega_{mid}t) \tag{4.8}$$

Notice that it is the same exact result obtained from the in-phase output of the mixer in (4.6) shifted of -90°.

The effect of this first demodulation in the frequency domain is summed up in figure 4.2, where also the eventual dithering tones are depicted.

4.2.2 Second demodulation

The easiest way to extract the information is by exploiting the digitized signal from the in-phase and the quadrature acquisition chains. Both these

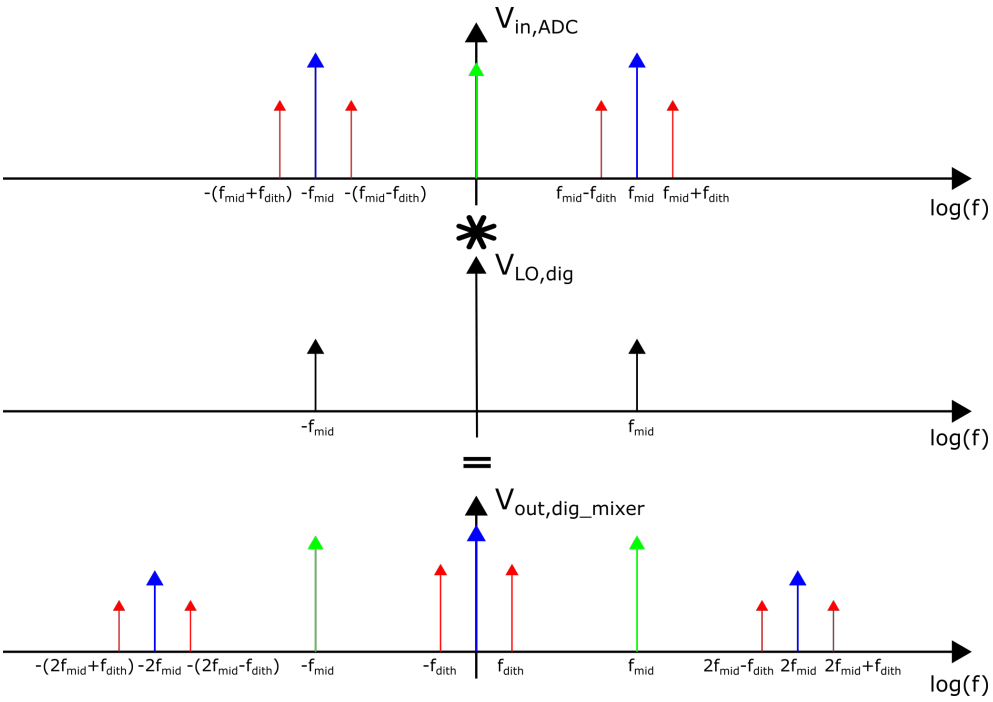

Figure 4.3: Effect of the second demodulation on the digitized signal. In blue the information about the admittance of the guide, in red the eventual dithering tones superimposed to it, in green the offset of the motherboard.

signals can be fed to two identical digital mixers driven by

$$d_{digital}(t) = cos(\omega_{mid} t) \tag{4.9}$$

Notice that, as stated in equation (4.6), the information about the real part is carried by a cosine. So the signal used for the in-phase second demodulation should be a cosine as well. Actually, it was chosen to use a square wave rather than a cosine, in order to simplify the implementation of the digital mixer. The high-frequency harmonics resulting from the demodulation are here neglected, since they are filtered out by the following low-pass filter.

Using the Werner's formula in (4.4), at the output of the digital mixer of the in-phase chain, it is

$$\begin{aligned} z_{real}(t) &= [A\cos(\omega_{mid} t) - B\sin(\omega_{mid} t)]\cos(\omega_{mid} t) = \\ &= \frac{A}{2}\cos(0) - \frac{B}{2}\sin(0) = \frac{A}{2} \end{aligned} \tag{4.10}$$

Notice that the high-frequency harmonics are here neglected since they are filtered out by the following digital processing chain. This chain will be further discussed in the next chapter. With the described operations, the real part of the signal can be properly extracted. The factor 1/2 is once again compensated in the user interface.

Similarly, neglecting once again the high-frequency harmonics, the output of the quadrature chain is

$$\begin{aligned} z_{imag}(t) &= [A\sin(\omega_{mid}t) + B\cos(\omega_{mid}t)]\cos(\omega_{mid}t) = \\ &= \frac{A}{2}\sin(0) + \frac{B}{2}\cos(0) = \frac{B}{2} \end{aligned} \tag{4.11}$$

In this way, the imaginary part of the signal is properly extracted as well.

The result of this second demodulation is depicted in figure 4.3

In-phase and quadrature digital demodulation

The procedure described above is not the only possible solution to extract the desired data. In fact, by looking at equation (4.6), it's possible to notice that the signal at the intermediate frequency still retains the information about both the real and the imaginary part of the TIA output voltage. It's then possible to use a single acquisition chain to extract both the components by properly demodulating in the digital domain. To do so, each acquisition chain passes through two digital mixers, fed with two waves in quadrature with each other. So, using the in-phase acquisition chain, it would be, for the real part

$$d_{digital,real}(t) = cos(\omega_{mid}t) \tag{4.12}$$

that leads, at its output, to

$$\begin{aligned} z(t) &= [A\cos(\omega_{mid}t) - B\sin(\omega_{mid}t)]\cos(\omega_{mid}t) = \\ &= \frac{A}{2}\cos(0) - \frac{B}{2}\sin(0) = \frac{A}{2} \end{aligned} \tag{4.13}$$

as above. Once again, the high-frequency harmonics are neglected and the 1/2 factor compensated by the software.

For the imaginary part, instead, the digital mixer has to be fed with

$$d_{digital,imag}(t) = sin(\omega_{mid}t) \tag{4.14}$$

that leads to

$$z(t) = [A\cos(\omega_{mid}t) - B\sin(\omega_{mid}t)]\sin(\omega_{mid}t) = \\ = \frac{A}{2}\sin(0) - \frac{B}{2}\cos(0) = -\frac{B}{2} \tag{4.15}$$

So the imaginary part is properly extracted as well, aside from the sign, easily corrected in the software.

A major advantage of this second solution is that it only needs half of the physical channels on the board. This means that, in this way, the motherboard is able to read up to 16 CLIPPs simultaneously, twice as before.

Notice that, in this discussion, all the phases have been neglected for the sake of simplicity, but they will be further discussed in the next paragraphs.

Effect on the offset

Another advantage of the two-step demodulation approach is that the signal arriving on the motherboard is modulated at an intermediate frequency f_{mid} and can be easily distinguished from the offset V_{os} of the electronic components, that is instead at DC.

The offset is acquired together with the useful signal and then multiplied by the digital mixer. The effect of this multiplication is the upconversion of the offset to/at f_{mid}, resulting in:

$$z_{os}(t) = V_{os}\cos(\omega_{mid}t) \tag{4.16}$$

This situation is depicted in figure 4.3. This means that the following low-pass filtering action of the lock-in chain is now able to easily remove the modulated offset of the motherboard.

A smarter solution to remove the offset is to introduce a high-pass filter before the digital mixer. In fact, at this point of the acquisition chain, the acquired offset is still at low frequency, so a HPF is able to greatly attenuate it, leaving unaltered the signal that is instead at f_{mid}. The residual offset signal is then modulated at f_{mid} and completely removed by the low-pass filter. Even if this solution requires an additional processing step, it has the advantage of reducing the off-band attenuation requirement of the LPF.

4.2.3 Computation of the phase increment for the first demodulation

Given the presence of two demodulations rather than just one, three DDSs are now needed in the stimulation chain. The first one is still necessary to create the forcing sinusoid. The phase increment fed to this DDS is computed by the graphical interface after the choice of the stimulation frequency by the end user according to

$$f_{stim} = \frac{160\ \text{MHz}}{2^{16}} \Delta\Theta_{stim} \tag{4.17}$$

recalling that this DDS is fed with a 160 MHz clock and a 16-bit wide phase increment.

Another DDS is necessary to perform the first on-chip demodulation. Finally, another one is used to perform the second demodulation in the digital domain. As already discussed, the end user just chooses the frequency of the second demodulation, and the interface computes once again the necessary phase increment according to

$$f_{mid} = \frac{160\ \text{MHz}}{2^{16}} \Delta\Theta_{mid} \tag{4.18}$$

Notice that also this DDS is fed with a 160 MHz clock and a 16-bit wide phase increment.

The remaining DDS has to work at a frequency that is exactly $f_{stim} - f_{mid}$. It follows that the exact phase increment necessary should be computed as

$$f_{dem} = f_{stim} - f_{mid} = \frac{160\ \text{MHz}}{2^{16}} (\Delta\Theta_{stim} - \Delta\Theta_{mid}) = \frac{160\ \text{MHz}}{2^{16}} (\Delta\Theta_{dem}) \tag{4.19}$$

This implies that

$$\Delta\Theta_{dem} = \Delta\Theta_{stim} - \Delta\Theta_{mid} \tag{4.20}$$

Notice that also this DDS is fed with a 160 MHz clock and a 16-bit wide phase increment. Since this choice is consistent with the other DDSs, the equivalent phase increment needed is simply the difference between the two.

The typical value of the intermediate frequency is somewhere above the corner frequency of the noise of the board, that was measured to be around

$f_c \simeq 1$ kHz. It's also better not to go way beyond this frequency. In fact, it should be remembered that there's a 4th order anti-alias filter in the acquisition chain with a bandwidth of 150 kHz. If the intermediate frequency is chosen too close to the pole constellation of this filter, it would be necessary to add a digital compensation of the attenuation. For this reason, the intermediate frequency is usually set around 20 kHz.

4.2.4 Dithering tones

Figure 4.2 shows the complete spectrum of the signal that is acquired by the ADCs of the board, that is the sum of the contributions of the clipp impedance and the dithering tones. Notice that the dithering tones are now at a different frequency with respect to the one at which they're generated. Nonetheless, it's still possible to recover them correctly, either by demodulating two times in the digital domain, or directly with the second demodulation discussed above performed at a different frequency. The latter solution was chosen, since conceptually and practically easier.

The dithering tones and the signal necessary for their demodulation are generated with two DDSs, both fed with a 5 MHz clock and a 16-bit wide phase increment. The DDS responsible for the synthesis of the dithering tone is fed with a phase increment $\Delta\Theta_{dith}$, so the generated frequency is

$$f_{dith} = \frac{5 \text{ MHz}}{2^{no\_bit}} \Delta\Theta_{dith} \tag{4.21}$$

Assuming for simplicity that $f_{dith} < f_{mid}$, the frequencies of the tones after the first demodulation move to $f_{mid} \pm f_{dith}$, as depicted in figure 4.2.

Since the intermediate frequency is generated according to

$$f_{mid} = \frac{160 \text{ MHz}}{2^{16}} \Delta\Theta_{mid} \tag{4.22}$$

it is

$$f_{dem,dith} = f_{mid} \pm f_{dith} = \frac{5 \text{ MHz}}{2^{16}} (32\Delta\Theta_{mid} \pm \Delta\Theta_{dith}) = \frac{5 \text{ MHz}}{2^{16}} (\Delta\Theta_{dem,dith}) \tag{4.23}$$

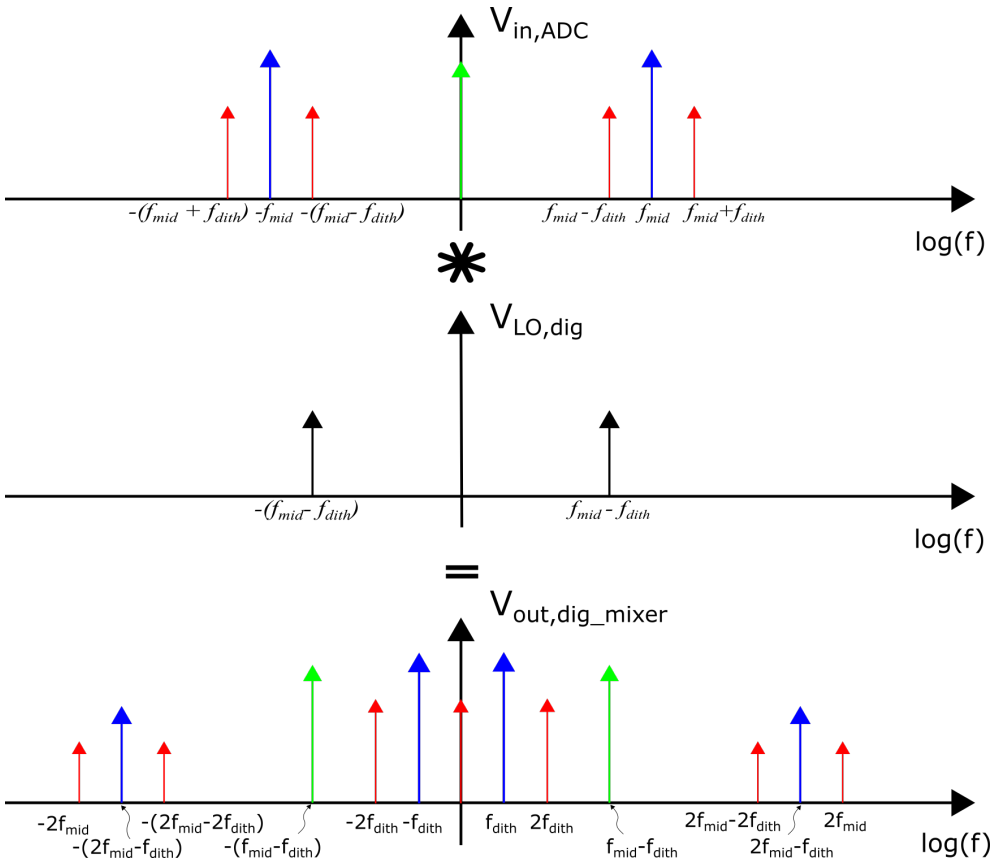

Figure 4.4: Effect of the second demodulation on the digitized signal to extract the dithering tones. In blue the information about the admittance of the guide, in red the eventual dithering tones superimposed to it, in green the offset of the motherboard.

This implies that the phase increment that has to be fed to the DDS responsible of the demodulation of the dithering tone is

$$\Delta\Theta_{dem,dith} = 32\Delta\Theta_{mid} \pm \Delta\Theta_{dith} \tag{4.24}$$

Notice that, to perform this operation correctly, the clock frequency of the dithering DDS has to be an integer submultiple of the one chosen for the stimulation DDS, so that the demodulation frequency is precisely the one needed.

The result of this second demodulation is depicted in figure 4.4

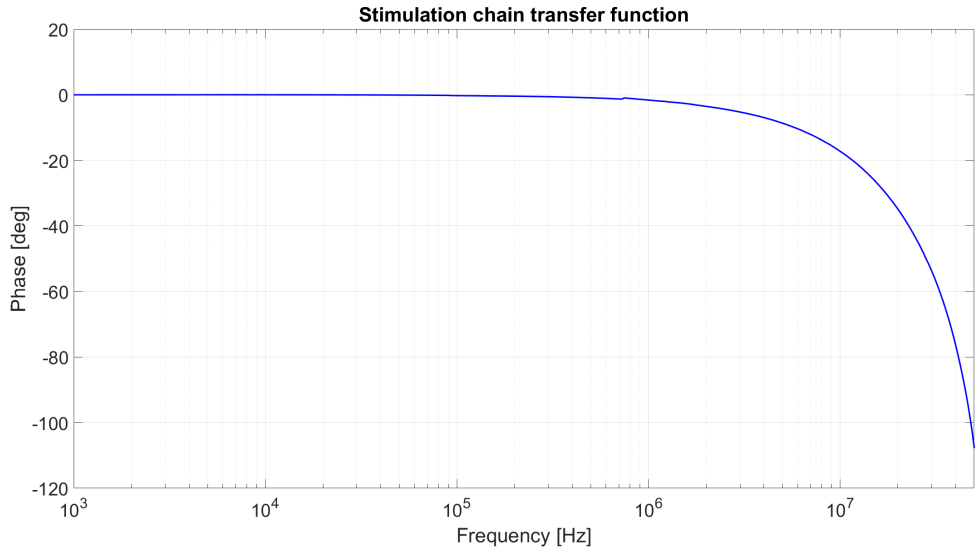

Figure 4.5: Phase shift introduced by the analog stimulation chain on the generated sinusoid.

4.3 Phase shift calibration

All the aforementioned signals have to pass through many electronic components before being acquired and demodulated. This means that they will inevitably suffer from some phase shift introduced by the frequency response of each component. The phase shift added will be clearly dependent on the intermediate frequency chosen, and, since the lock-in technique is phase-sensitive, it will affect the demodulation phase needed to properly extract the real and imaginary part of the acquired signal. For this reason, the phase shift behavior of the chains of interest has to be measured and properly compensated. To do so, the information extracted can be used to implement an auto-calibration of the phase shift fed to the DDSs. In this way, this issue is completely transparent to the end user.

In the following paragraphs, the phase contribution of each part of the system will be explained in detail.

4.3.1 Delay introduced by the stimulation chain

So far, the stimulation signal and the one responsible for the on-chip demodulation have been considered exactly in phase with each others, since they are generated at the very same moment. This is not entirely true though. In fact, the forcing sinusoid suffers from a certain phase shift introduced by the electronic components in the analog chain. On the other end, the square waves for the first demodulation do not follow the same path. In fact, these square waves are generated by extracting the MSB at the DDS output and by mapping it directly on a digital bus that goes from the motherboard to the chip. So, these signals suffer from a very little phase shift, practically zero, and the phase shift introduced by the stimulation chain has then to be compensated. To do so, the DDS responsible for the creation of these square waves is designed with tunable phase. Since the phase shift depends on the frequency of the stimulus, it was necessary to measure the whole transfer function of the chain, as already discussed. To do so, the "E5061B" network analyzer by Agilent Technologies was used. The results are shown in figure 4.5.

This information can be used to properly compensate the shift produced by the analog chain, so that the two signals are exactly in phase and in quadrature with the forcing sinusoid.

DDS delay

Another concern regarding the synchronization of the stimulus with the square waves comes from the fact that they are generated by two different DDSs. To ensure the synchronization of the waves, the two DDSs have to be started at the very same instant. It is then necessary to verify that different DDSs fed with different phase increments have no phase shift between each other.

To do so, the MSBs of the two DDS outputs were mapped directly on two pins of the FPGA, thus generating two square waves perfectly synchronous to the DDS sinusoids. Then, with the "HDO6054A-MS" oscilloscope by Teledyne

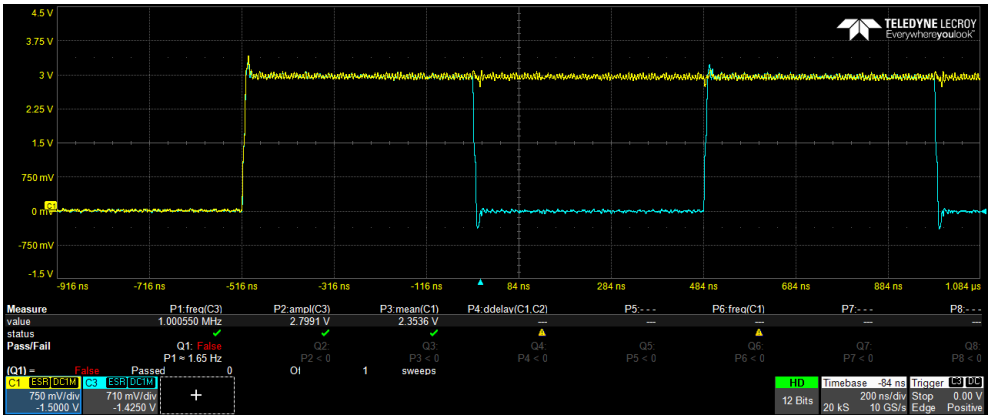

Figure 4.6: Snap picture taken with the oscilloscope of two signals generated by two different DDSs. In this measure, one DDS generates a 1 MHz square wave, the other a 20 kHz one. In can be seen that the two rising edges are perfectly synchronous.

Lecroy, a snap picture of the startup instant of the two DDSs was taken. The result is shown in figure 4.6. It can be seen that the two rising edges are perfectly in phase between each other, so that it's possible to conclude that the synchronization of different DDSs can be obtained, even when they're not generating the same frequency.

4.3.2 Delay introduced by the acquisition chain

In the previous paragraph, equation (4.6) stated that the signal acquired by the ADC of the in-phase chain was:

$$y(t) = A\cos(\omega_{mid}t) - B\sin(\omega_{mid}t) \tag{4.25}$$

This is not entirely true, since the phase shift introduced by the electronic components in the acquisition chain has also to be taken into account. So the correct expression is actually

$$y(t) = A\cos(\omega_{mid}t + \varphi_{board}) - B\sin(\omega_{mid}t + \varphi_{board}) \tag{4.26}$$

As a consequence, the phase shift introduced by the board has to be compensated. This simply means that the signal used for the digital demodulation has to be shifted of the same quantity. Notice that this outcome is true both

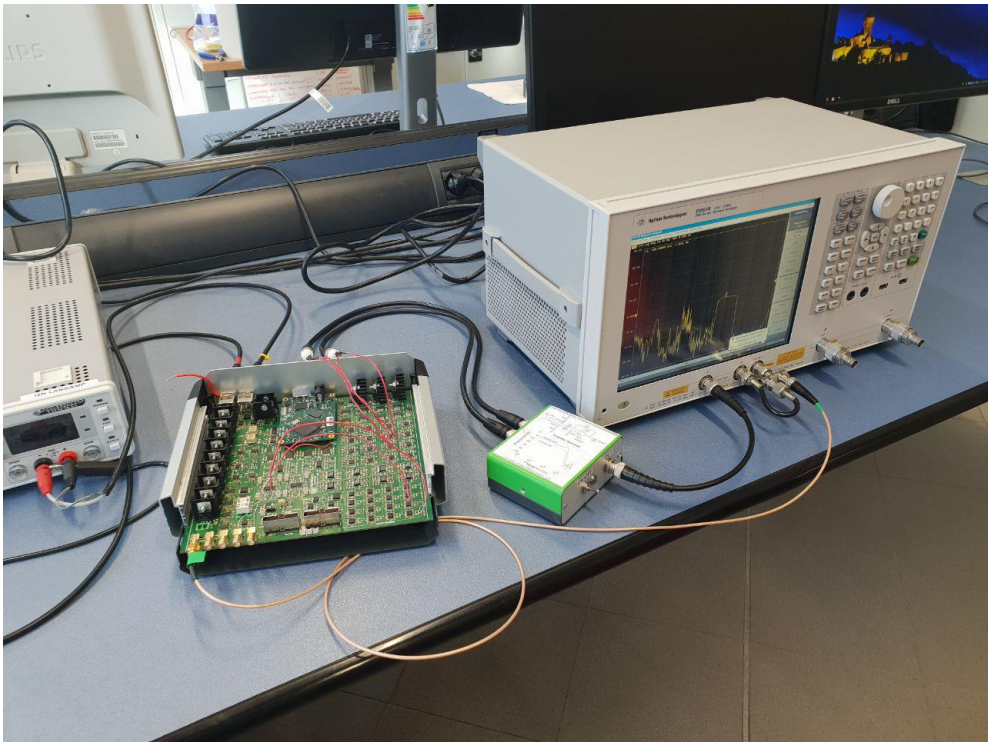

Figure 4.7: Setup used for the measurement of the transfer function of the acquisition chain.

when the in-phase and quadrature demodulation is performed on chip and when they are performed digitally in the FPGA.

So, it's once again necessary to measure the phase shift of the chain in the frequency range of interest. To perform the measurement, the "E5061B" network analyzer by Agilent Technologies was used once again. The setup is shown in figure 4.7.

It should be pointed out that the first pole introduced in the acquisition chain exploits the output resistance of the ASIC and the equivalent capacitance of the shielded cable, as explained in 3.1.1, that were disconnected from the system when performing the measurement. To replicate its effect, a resistance was mounted in series to the wires that bring the differential signal to the input amplifier. This resistance was computed knowing that the pole is at

$$f_p = \frac{1}{2\pi C_{PCB} R_{eq}} \simeq 192 \text{ kHz} \tag{4.27}$$

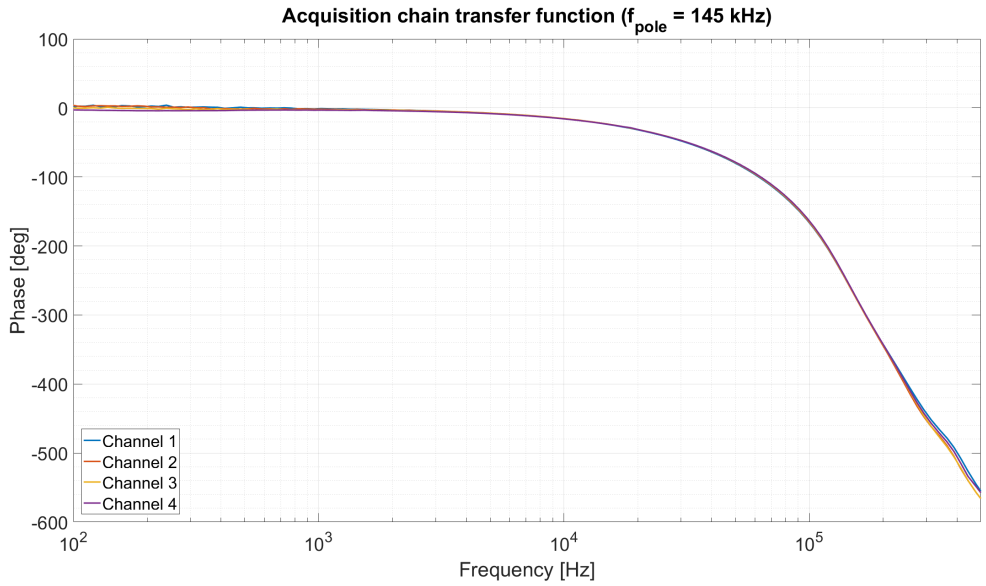

Figure 4.8: Phase shift introduced by the acquisition chain on the ASIC output signal.

leading to a resistance of 13 kΩ.

The result of the measurement is shown in figure 4.8. Once again, with this data, it's possible to introduce an auto-calibration for this phase as well, similar to the one designed for the phase shift introduced by the stimulation chain.

Notice that the phase shift measured is coherent to what expected, considering the first-order pole and the 4th-order Butterworth anti-alias filter in the chain.

Effect of tolerances on the readout

In figure 4.8, the measurement of four different parallel channels is provided. It can be noticed that there's a little difference between the phase shift introduced by one and the other, due to the tolerances in the values of the components. This might seem negligible, but it could easily lead to a non-negligible spurious signal.

For instance, when the system is used with the in-phase and quadrature demodulation performed directly on the ASIC, the signals acquired by the

two acquisition channels used for the real and imaginary parts are digitally demodulated with the same signal. It's then important that they're perfectly in quadrature with each other, otherwise the demodulation will lead to a wrong result.

Assuming that, after the phase shift calibration, the real part in the imaginary channel still suffers from a phase shift of $\Delta\varphi$ w.r.t. the imaginary one, it is

$$\begin{aligned} z(t) = \{A\sin[(\omega_{stim} - \omega_{dem})t + \Delta\varphi] + B\cos[(\omega_{stim} - \omega_{dem})t]\}\cos(\omega_{mid}t) = \\ = \frac{A}{2}\sin(\Delta\varphi) + \frac{B}{2}\cos(0) \simeq \frac{A}{2}\Delta\varphi + \frac{B}{2} \end{aligned} \tag{4.28}$$

since, for small $\Delta\varphi$, $\sin(\Delta\varphi) \simeq \Delta\varphi$.

The problem arises from the fact that, in this system, the real part is an order of magnitude larger that the imaginary one. So, for a $\Delta\varphi \simeq 0.5° \simeq 0.01$ rad, the signal would be grossly wrong of a 10% factor.

This issue can be easily solved by avoiding the use of two hardware chains and a single digital demodulation, and by rather using only one acquisition chain and performing, in the digital domain, both the in-phase and the quadrature demodulation, as described in 4.2.2. In fact, by using only one acquisition chain, the real and imaginary components will be always perfectly in quadrature, thus solving this issue.

4.3.3 Delay introduced by the digital chain

Physical components are not the only source of delay. Indeed it should be considered that the acquired signals have to pass through some registers in the FPGA before getting to the digital mixer. This is due to the presence of some digital processing stages between the ADC acquisition and the demodulator. All these registers are synchronous to the 625 kHz clock cycle, generated by the FSM that controls the ADCs. This implies that the delay introduced by each register is

$$T_{clk} = \frac{1}{625\ \text{kHz}} = 1.6\ \mu\text{s} \tag{4.29}$$

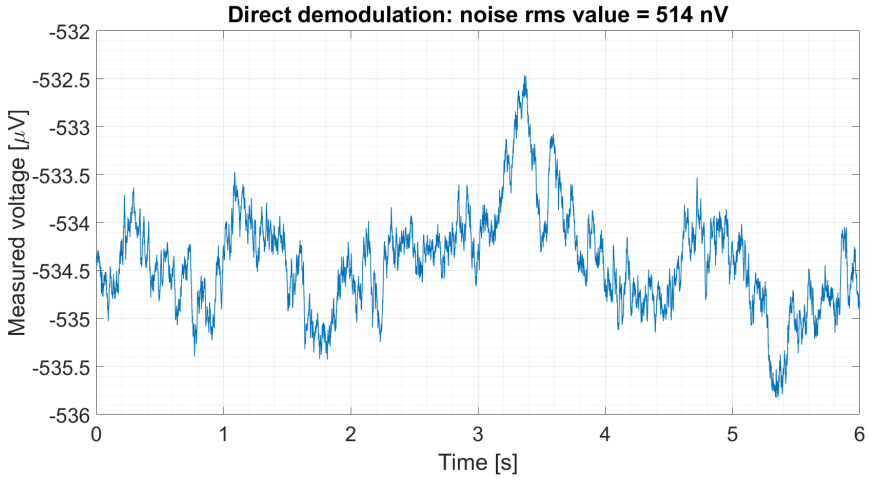

Figure 4.9: Acquisition performed with the direct demodulation approach when no stimulus is applied.

To quantify this delay in terms of phase shift, it's necessary to choose an intermediate frequency. For a value of roughly 20 kHz, that is a period of 50 µs, the phase shift results to be

$$\Delta\varphi_{clk} = \frac{1.6\ \mu s}{50\ \mu s} 360^\circ = 11.52^\circ \tag{4.30}$$

that is absolutely not negligible.

Since this phase shift will depend on the intermediate frequency chosen, rather than compensating it manually during each session, it was preferred to add a delay line to the signal used for the digital demodulation. The length of this line is designed to add exactly the same number of registers experienced by the sampled signal due to the digital processing line. These registers are synchronous to the clock generated by the FSM as well. In this way, the compensation works independently of the intermediate frequency chosen, and even in case the sampling frequency has to be modified for any reason.

4.4 Experimental results

The improvements introduced by the two-step demodulation have been verified experimentally. To do so, it was necessary to acquire the input signal

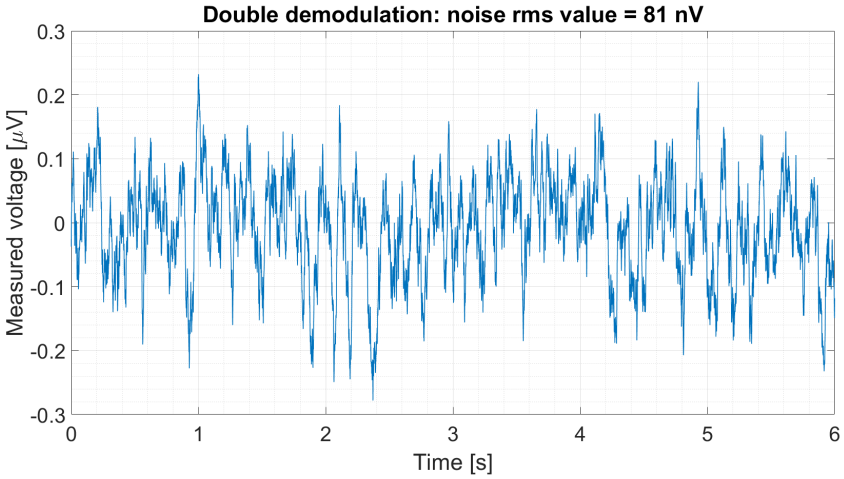

Figure 4.10: Acquisition performed with the two-step demodulation approach when no stimulus is applied.

with the ASIC connected to the board but without providing any forcing sinusoid to the CLIPP. The dominant pole of the system is the one of the digital low-pass filter, set at 6 Hz. A first acquisition without the two-step demodulation was then performed. The result is depicted in figure 4.9. As expected, the signal has a certain DC value, due to the offset of the board. From the acquired data, the equivalent input RMS noise computed is around $\sigma_{in} = 514$ nV. This is higher than the noise expected considering only the white noise contribution of the ASIC, confirming that the board adds non-negligible noise to the system and worsens the overall resolution.

The measurement is then performed once again with the two-step demodulation activated. The acquired noise is shown in figure 4.10. As expected, this time the signal is centered around zero, since the effect of the offset is canceled by the high-pass filtering action introduced before the digital demodulation. The equivalent input RMS noise computed by the measurement is $\sigma_{in} = 81$ nV. This is the value expected considering only the noise spectral density of the ASIC, meaning that the board does not worsen the resolution of the system, that is instead limited by the preamplification stage.

The improvements made can be exploited to either read smaller signals, or to increase the filters bandwidth allowing, in this way, a faster readout.

Chapter 5

Digital signal processing

After being digitized, the data has to be further processed in order to extract the desired information. By choosing a FPGA over a microcontroller, the data can be processed in a parallel way, making it possible to handle many channels at the same time. This means that the information of each sensor can be available almost in real time, and it's then possible to achieve fast control strategies on different optical devices simultaneously.

Specifically, the digital processing chain is structured, for each channel, as shown in figure 5.1. In particular, there is:

- a first-order high-pass filter. This is useful to reduce the DC offset of the board;
- two square-wave demodulators. Both are fed with the output of the HPF above, but the signals used to perform the demodulation are in quadrature with each other, extracting in this way both the real and the imaginary component of the signal, as previously described in 4.2.2.
- two first-order low-pass filters, one for each demodulator. This filter reduces the high-frequency replicas due to the second demodulation, the residual white noise, the modulated offset and eventually the quantization noise of the ADC, as discussed in 3.1.5. The bandwidth of this LPF defines the overall bandwidth of the lock-in;

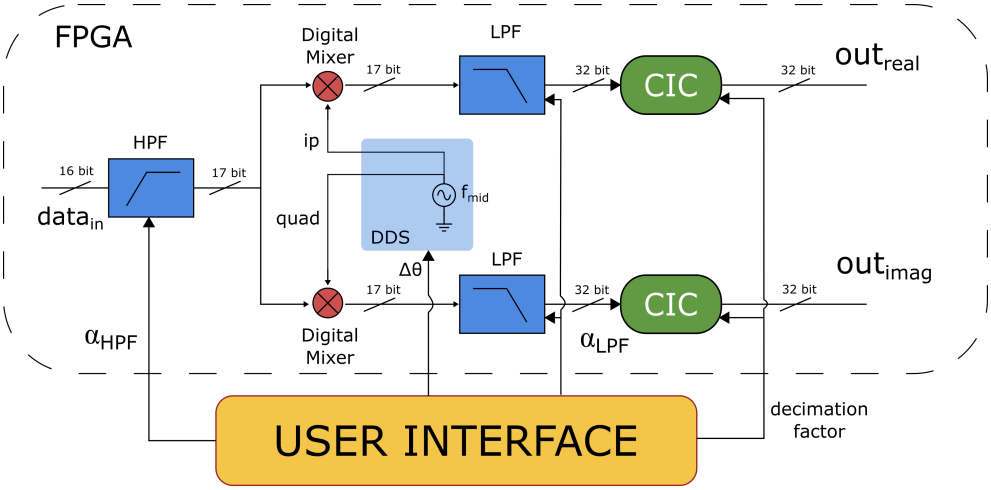

Figure 5.1: Schematic view of one digital processing chain

- two cascaded integrator-comb filters. Thanks to their notching action, they can be carefully designed in order to completely eliminate all the unwanted harmonics that survive to the LPF action.

Each of these blocks will be now individually discussed.

5.1 Fundamentals about digital filtering

5.1.1 Finite Impulse Response Filters

Finite Impulse Response (FIR) filters are the simplest class of digital filters. The general scheme of a FIR filter is depicted in figure 5.2. In order to understand the link between the desired transfer function and the $h[n]$ coefficients, it's necessary to recall that the output of a linear system is the convolution between the input and its weighting function, i.e., in the discrete time domain,

$$y[n] = x[n] * w[n] = \sum_{m=-\infty}^{\infty} x[m]w[n-m] \tag{5.1}$$

So, to compute the output sequence of any linear system, the equation says to flip the time order of an input sample sequence and start stepping the flipped sequence across the sampled weighting function of the system. This is the exact same operation performed by the structure in figure 5.2 through

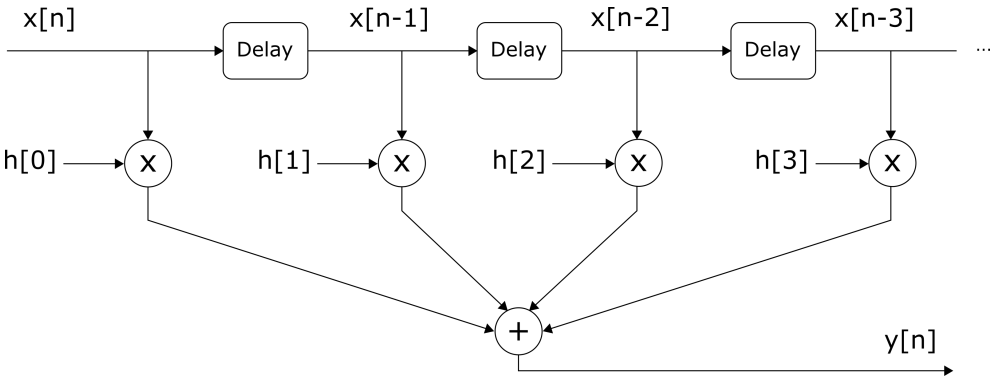

Figure 5.2: Generic scheme of a FIR filter

the use of delay blocks and multipliers. Consequently the $h[n]$ coefficients are simply the samples of the desired weighting function, i.e. $w[n]$ in equation (5.1). These coefficients are often called "taps" in literature.

Rather than computing these taps by hand, it's preferable to use one of the many software routines available for this purpose. In these routines, it's possible to set the order of the filter (i.e. the number of coefficients wanted), the passband and the in-band attenuation, the stopband and the off-band attenuation, and the windowing function. The software automatically computes the correct coefficients to match the specifications.

The design of these filters is then really simple. Nonetheless, they require many multiplications to achieve good filtering performances. Given that multipliers are very resources-demanding in an FPGA, and given that three filters (one HPF and two LPFs) are necessary for each acquisition channel, a more efficient solution that requires fewer resources should be investigated.

5.1.2 Infinite Impulse Response Filters

Another popular class of digital filters are the Infinite Impulse Response (IIR) filters. The difference with respects to the FIR solution is that the filtering actions is performed with the samples of both the input sequence and the output one. This means that there is both a feedforward block, as for the FIR filters, and a feedback one, with two different sets of multiplicative

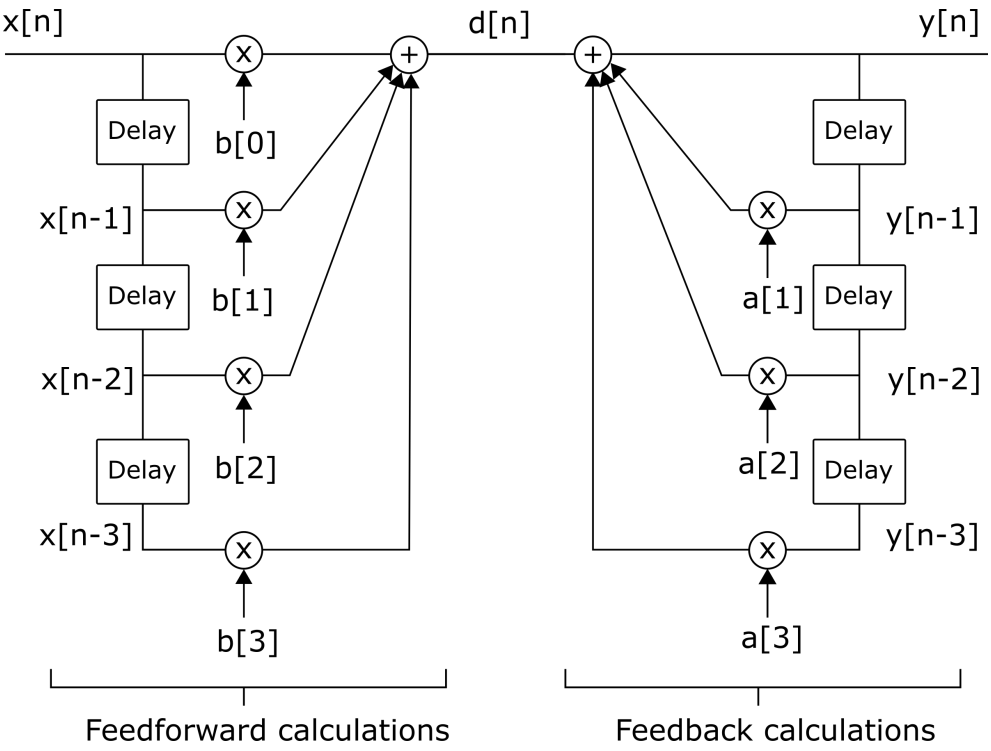

Figure 5.3: Generic scheme of a IIR filter

coefficients. The general scheme is depicted in figure 5.3.

It's easy to imagine how, thanks to the fact that more information is available, less multiplications are needed to achieve the same filtering performances of a FIR filter. This comes at the expense of a more difficult design. In fact there's no direct link between the sets of feedforward coefficients b_i and the feedback ones a_i with the implemented transfer function. Moreover, given the presence of a feedback action, IIR filters might suffer from instability if not carefully designed.

There are some software routines that compute the set of coefficients necessary to implement any filter's shape, even when the closed-form expressions for the filter's Z-transform does not exist and there is no explicit equations to work with (see, for instance, the so-called "optimization methods" [19]). In this case though, being the filters reasonably simple, it was preferred to compute the coefficients in a closed-form solution. To do so, the "bilinear transform" was used. This relies on a first order approximation of the Z-transform, for

which

$$s \leftarrow 2f_s \frac{z-1}{z+1} \tag{5.2}$$

where f_s is the sampling frequency. Through this transformation, any transfer function in the Laplace domain can be transformed in its equivalent in the Z-domain.

5.2 High-pass filter

5.2.1 Transfer function and digital implementation

As already stated, the high-pass filter (HPF) is useful to reduce the acquired DC offset of the board. This operation is possible since the signal is not in baseband, but it's still modulated around the carrier frequency f_{mid}.

To find the discrete-domain representation of this filter, it's necessary to find its Z-domain transfer function and anti-transform it. The transfer function of a HPF in the Laplace domain is

$$H_{HPF}(s) = \mu \frac{s\tau}{1+s\tau} \tag{5.3}$$

where μ is the gain at infinite frequency. Applying the bilinear transform, the Z-transform of the transfer function is

$$H_{HPF}(Z) = \mu \frac{2f_s \frac{z-1}{z+1}}{1+\tau 2f_s \frac{z-1}{z+1}} = ... = \mu \frac{2f_s\tau}{1+2f_s\tau} \frac{1-z^{-1}}{1-\frac{2f_s\tau-1}{2f_s\tau+1}z^{-1}} \tag{5.4}$$

Setting

$$\mu' = \mu \frac{2f_s\tau}{1+2f_s\tau} \quad \text{and} \quad \alpha = \frac{2f_s\tau-1}{2f_s\tau+1} \tag{5.5}$$

the final result can be written as

$$H_{HPF}(Z) = \mu' \frac{1-z^{-1}}{1-\alpha z^{-1}} \tag{5.6}$$

From the Z-transform of the transfer function, it's possible to easily anti-transform it into its representation in the discrete time domain. In fact, recalling that $H(Z) = Y(Z)/X(Z)$ and that the multiplication with z^{-1} represent a one-step delay of the sequence, the relationship between input and output is

$$y[n] = \mu'(x[n] - x[n-1]) + \alpha y[n-1] \tag{5.7}$$

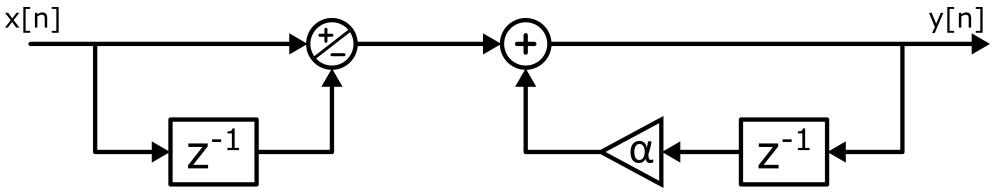

Figure 5.4: Implementation of a I-order HPF in direct form I

Notice that, according to (5.4) and (5.5), the choice of μ' only affects the high-frequency gain of the filter. So, by setting

$$\mu' = 1 \Rightarrow \mu = \frac{1 + 2f_s\tau}{2f_s\tau} \tag{5.8}$$

it's possible to avoid the use of a multiplier, saving resources in the practical implementation of the filter. This comes at the expense of a high-frequency gain that is dependent from the position of the pole. The normalization has then to be properly performed in the software.

Therefore, equation (5.7) is directly translated into the structure shown in figure 5.4, known in literature as "direct form I".

5.2.2 Role of alpha and pole position

According to equation 5.5, it's possible to move the position of the pole by choosing different values for α. Being

$$\alpha = \frac{2f_s\tau - 1}{2f_s\tau + 1} = \frac{f_s - \pi f_p}{f_s + \pi f_p} \tag{5.9}$$

the possible values of α range from +1 to -1 (for $f_p \to +\infty$). The closer to +1 the value chosen, the closer to zero the frequency of the pole obtained.

For the sake of simplicity, only positive values are used in this system. This means that the maximum pole achievable is, for $\alpha = 0$, $f_p = f_s/\pi \simeq 200$ kHz. This is more than sufficient, since the pole should be always kept one decade below the signal frequency, that is around 20 kHz.

Therefore, α is an all-fraction binary number, so it has to be represented in 2's complement fixed-point binary format. To do so, the following steps should be followed:

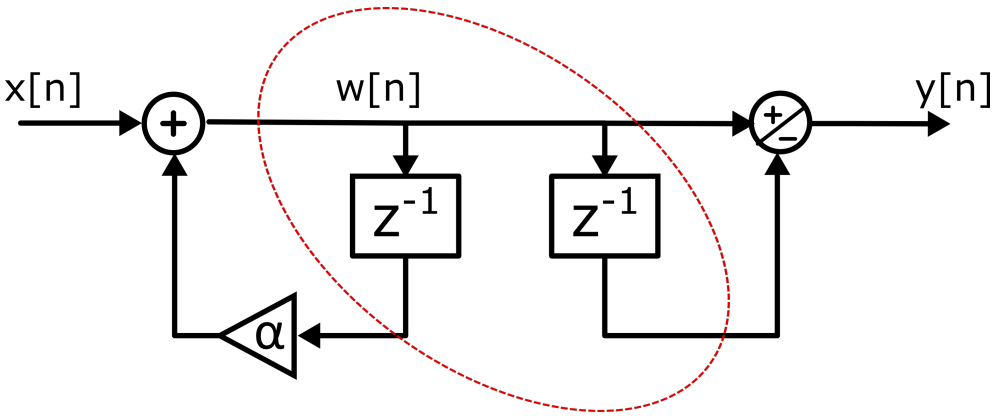

Figure 5.5: Implementation of a I-order HPF in direct form II. The encircled registers can be shared.

1. multiply the absolute value of the intended value of α by 2^{no_bit-1};
2. round the obtained value, using the preferred rounding method;
3. convert the result of the rounding operation to the binary representation and place the binary point to the right of the MSB;
4. if the original value was negative, invert all the bits and add one to the result.

This means that it's necessary to keep track of the decimal part during all the computation.

For this design, it was chosen to use a 16-bit wide α, so that the largest fractional number that can be represented is $\frac{2^{15}-1}{2^{15}}$. This leads to a minimum pole of

$$f_p = \frac{1-\alpha}{1+\alpha}\frac{f_s}{\pi} = 3 \text{ Hz} \tag{5.10}$$

In literature, this number is said to be represented in 1.15 format, meaning that 15 bits (at the right of the dot) are for the decimal part, while 1 bit (at the left of the dot) is for the integer one.

5.2.3 Practical implementation

The feedforward block and the feedback one both perform linear time-invariant operation. This means that they can be swapped, as shown in figure 5.5, allowing the possibility of sharing the encircled registers. This solution is often referred to as "direct form II".

To size the filter, it's necessary to compute its high-frequency gain. This can be done directly from the Z-domain transfer function. Indeed, as suggested by equation (5.2), this can be easily done by setting $Z^{-1} \to -1$, that leads to

$$H(-1) = \frac{2}{1+\alpha} \tag{5.11}$$

Hence, the maximum gain is, for $\alpha \to 0$, 2 at most. This means that, at the end of the chain, only one extra bit is necessary to avoid any overflow errors. This gain will be then normalized in the user interface, as already stated. The following hardware digital signal processing will have 17-bit input word.

A problem arises due to the nature of binary multiplication. In fact, since the output of the multiplier is fed back to its input, the number of bits of the result would increase indefinitely if no action is taken. For this reason it's necessary to truncate the result. The number of bits necessary to avoid overflow has to be computed. In particular, it's necessary to compute the transfer function from the input signal x[n] to w[n], shown in figure 5.5. So it is

$$W(Z) = W(Z) + \alpha W(Z)Z^{-1} \Rightarrow \frac{W(Z)}{X(Z)} = \frac{1}{1-\alpha z^{-1}} \tag{5.12}$$

The highest value of this transfer function is obtained for $Z^{-1} \to +1$, so at low frequencies. Since α is a 16-bit signed fractional number, it is, at most

$$\frac{W[-1]}{X[-1]} = \frac{1}{1-\frac{2^{15}-1}{2^{15}}} = 2^{15} \tag{5.13}$$

So, to properly filter out the low-frequency harmonics of the input signal it's necessary to keep 15 bits more with respect to the input. Notice that these bits are not useful for the final result, but they are necessary to avoid overflow in the middle of the chain. For this reason, they're often referred to as "guard bits".

Implementation of the multiplication

The previous paragraph highlights that, for a 16-bit wide input word, and a 16-bit wide *alpha*, the signal w[n] needs 31 bits. The result of the multiplier has then $31 + 16 = 47$ bits, and, since *alpha* is a fractional number, is in the 32.15 format (32 bits for the integer part, 15 for the decimal one), with two sign bits, due to the multiplication of two signed numbers. Of all these bits, 31 have to be kept. The solution that leads to the smallest rounding error is to discard the extra sign bit, that adds no information, and the 15 bits that represent the decimal part of the result. This operation is called truncation, and it might lead to some non-negligible DC offset [20].

The easiest way to minimize this offset would be to use other strategies to discard the decimal bits. For instance, it's possible to round the number to its closest integer rather than discarding the decimals regardless their value. This would require extra logic, so truncation was still preferred in this application.

The small residual DC offset is not a problem here anyway, since it's modulated by the following mixer and is then filtered out by the low-pass filter.

Notice also that, by discarding all the decimals bits, no residual decimal part has to be handled and the signal can be treated again as an integer value.

5.2.4 Experimental validation

The filter described so far was thoroughly tested. A VHDL testbench was implemented to perform a behavioral simulation. The input stimulus of this testbench was generated with MATLAB and converted in its binary form with a tailored routine. The sinusoid was fed to the input of the testbench, and the output was saved into a *.txt* file, in order to observe the effect of the filter at various frequencies. A second MATLAB routine was used to plot the results.

After this first behavioral simulation, the filter was actually implemented on the FPGA and tested experimentally. Through the "33522A" function waveform generator by Agilent, a sinusoid at a certain frequency was fed to the acquisition chain and digitized by one of ADCs. By measuring the ampli-

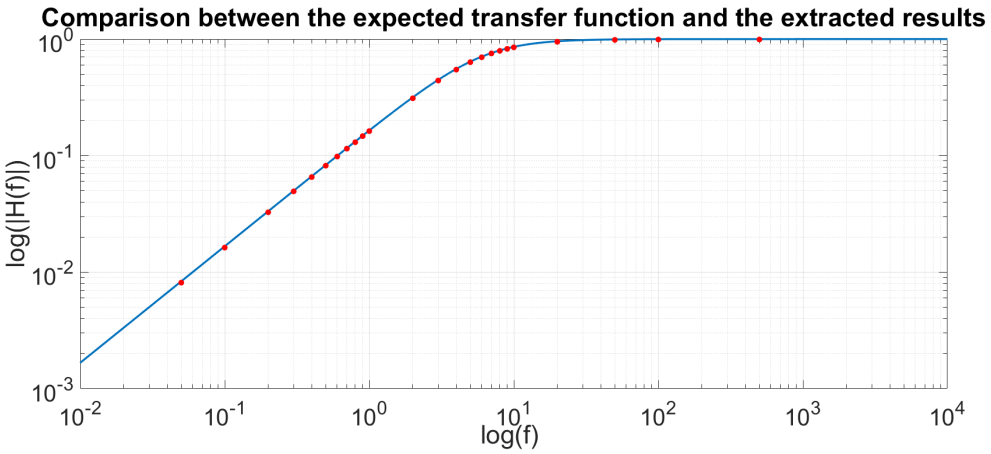

Figure 5.6: Comparison between the data extracted from the filter implemented on the FPGA and the transfer function expected from a HPF

tude value of the filtered sinusoid it was then possible to estimate the transfer function value at that specific frequency. The procedure was repeated with different frequencies. Then, both the expected transfer function and the acquired results were plotted with MATLAB. The experimental results match very well the theoretical behavior, as shown in figure 5.6. For this measure, the pole was set at 6 Hz. The curve is normalized with respect to the high-frequency gain of the filter.

5.3 Digital mixer

The next stage of the digital processing chain is a digital mixer to demodulate the output of the HPF and extract the value of the real and the imaginary part of the signal. This second demodulation is performed with square-wave signals, rather than sinusoidal ones. In this way, the mixers are extremely simple and don't require any multipliers. Since there are two demodulators per channel, a multiplier-based solution would be indeed unfeasible.

All the results stated in the previous chapter where a sinusoidal demodulation was described are still perfectly valid, since the high-frequency harmonics resulting from this modulation are still filtered out by the following narrow-band low-pass filter. The choice of using a square-wave demodulation only

worsens the SNR of the system of a factor $\sim 11\%$ [8].

Practically, the demodulation is performed by directly connecting the input to the output when the square wave has a value of 0 and by inverting the sign of the input when the square wave has a value of 1. The extreme simplicity of this operation was preferred over the more complex sinusoidal demodulation despite the SNR degradation.

As already explained in the previous chapter, the second demodulation is performed at either f_{mid}, if the signal that has to be extracted is the information about the conductance of the waveguide, or $f_{mid} \pm f_{dith}$, if instead the information of interest is the first derivative. The square wave is realized by extracting only the MSB from the output of the specific DDS.

5.4 Low-pass filter

5.4.1 Transfer function and digital implementation

The low-pass filter placed after the digital mixer is useful to reduce the high-frequency harmonics due to the two square-wave demodulations, to further reduce the bandwidth of the white noise and finally to lower the quantization noise introduced by the ADCs.

The steps followed to realize the LPF are very similar to the ones already described for the HPF. The goal is to get a Z-domain transfer function starting from a Laplace-domain one, that, for a LPF, is

$$H(s) = \frac{\mu}{1 + s\tau} \tag{5.14}$$

where μ is the low-frequency gain of the filter. Applying the bilinear transform, it is

$$H(z) = \frac{\mu}{1 + 2\tau f_s \frac{1-z^{-1}}{1+z^{-1}}} = ... = \frac{\mu}{1 + 2\tau f_s} \cdot \frac{1 + z^{-1}}{1 - z^{-1}\frac{2\tau f_s - 1}{2\tau f_s - 1}} \tag{5.15}$$

Setting

$$\mu' = \frac{\mu}{1 + 2f_s\tau} \quad \text{and} \quad \alpha = \frac{2f_s\tau - 1}{2f_s\tau + 1} \tag{5.16}$$

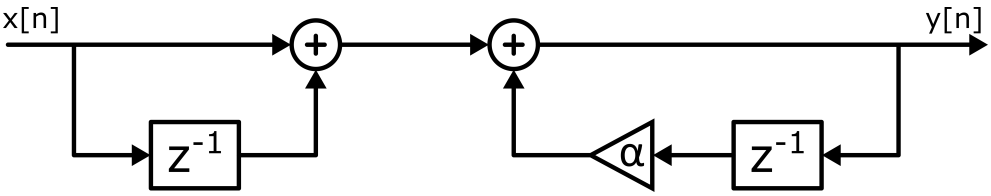

Figure 5.7: Implementation of a I-order LPF in direct form I

the final result can be written as

$$H(Z) = \frac{1 + z^{-1}}{1 - \alpha z^{-1}} \tag{5.17}$$

By anti-transforming the Z-domain transfer function, it's possible to find the discrete-time domain representation. Recalling that $H(Z) = Y(Z)/X(Z)$ and that the multiplication with z^{-1} represent a one-step delay of the sequence, the relationship between input and output is

$$y[n] = \mu'(x[n] + x[n-1]) + \alpha y[n-1] \tag{5.18}$$

Once again, by setting $\mu' = 1$, it's possible to save a multiplier in the feedforward block. This comes at the expense of a low-frequency gain of the filter that is dependent from the position of the pole. The software has again to properly perform the normalization.

This is translated into the structure shown in figure 5.7, known in literature as "direct form I".

5.4.2 Role of alpha and pole position

According to equation 5.16, it's possible to move the position of the pole by choosing different values for α. Being

$$\alpha = \frac{2f_s\tau - 1}{2f_s\tau + 1} = \frac{f_s - \pi f_p}{f_s + \pi f_p} \tag{5.19}$$

as before, all the consideration done in the paragraph 5.2.2 are still perfectly valid. By using 16-bit wide 2's complement fixed-point binary representation, and using only positive values of α, the minimum pole is, for $\alpha = \frac{2^{15}-1}{2^{15}}$, $f_p = 3$ Hz, while the maximum is, for $\alpha = 0$, $f_p = f_s/\pi \simeq 200$ kHz.

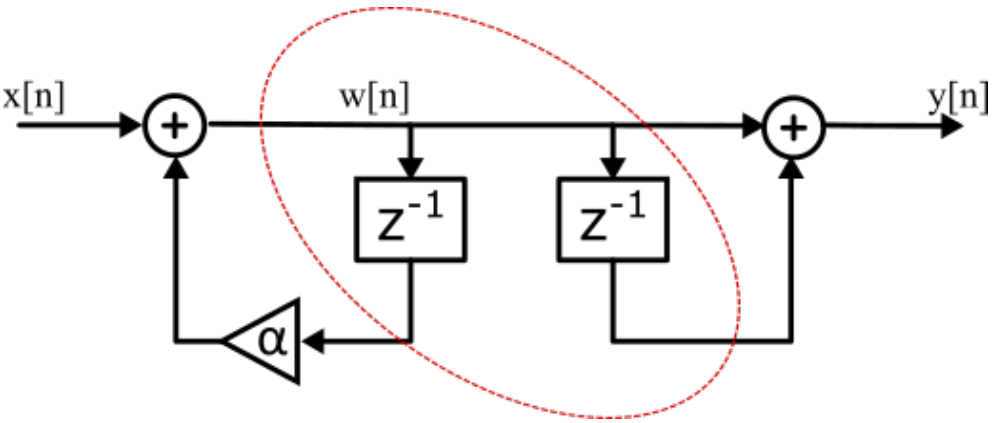

Figure 5.8: Implementation of a I-order LPF in direct form II. The encircled registers can be shared.

5.4.3 Practical implementation

Since the feedforward block and the feedback one both do linear time-invariant operation, they can easily swapped again, as shown in figure 5.8, leading to the "direct form II" of a first order LPF. Also in this case is then possible to share the encircled Z^{-1} registers.

This filter has to be properly sized as well. To do so, it's necessary to compute the maximum gain possible. So, by setting $Z^{-1} \to +1$, it is

$$H(1) = \frac{2}{1-\alpha} \tag{5.20}$$

that is, at most

$$H(1)_{max} = \frac{2}{1-\frac{2^{15}-1}{2^{15}}} = 2^{16} \tag{5.21}$$

So, extra 16 bits should be used. Recalling that, after the HPF, the input word was 17-bit long, the output word will be 33-bit long.

Since the minimum gain of the filter is 2, it's possible to compensate this factor by doing a right shift of the output word. In this way, the output word is only 32-bit long, allowing an easier communication with the software through the USB 3.0 interface.

These bits are also enough to avoid overflow in the intermediate steps, as it can be easily seen by computing once again the transfer function from x[n] to w[n]. Also the multiplier is implemented as described before for the HPF.

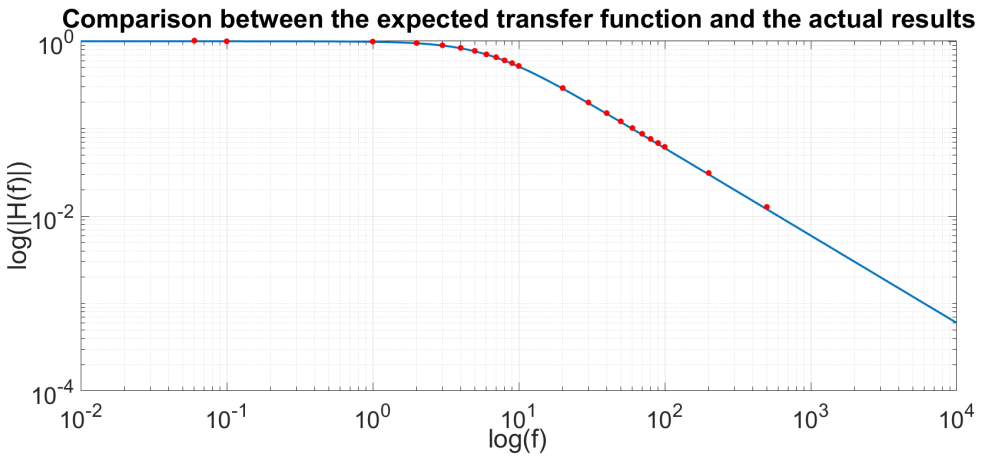

Figure 5.9: Comparison between the data extracted from the filter implemented on the FPGA and the transfer function expected from a LPF

5.4.4 Experimental validation

As for the high-pass filter, the low-pass filter has been thoroughly tested. Another testbench was implemented to perform a behavioral simulation. The input stimulus was generated with MATLAB and converted in its binary form. The output was saved into a file and, through another MATLAB routine, the result was plotted.

The filter was then actually implemented on the FPGA and tested experimentally. With the same procedure described above, the value of the transfer function at different frequencies was extracted and compared, through MATLAB, with the expected behavior. Once again, the experimental results match very well the theoretical behavior, as shown in figure 5.9. For this measure, the pole was set at 6 Hz. The curve is normalized with respect to the low-frequency gain of the filter.

5.5 Notch filter

At the end of the processing chain, there might still be unwanted sinusoids at certain precised frequencies, due to the high-frequency harmonics resulting from the two square-wave demodulations. In fact, the first-order low-pass fil-

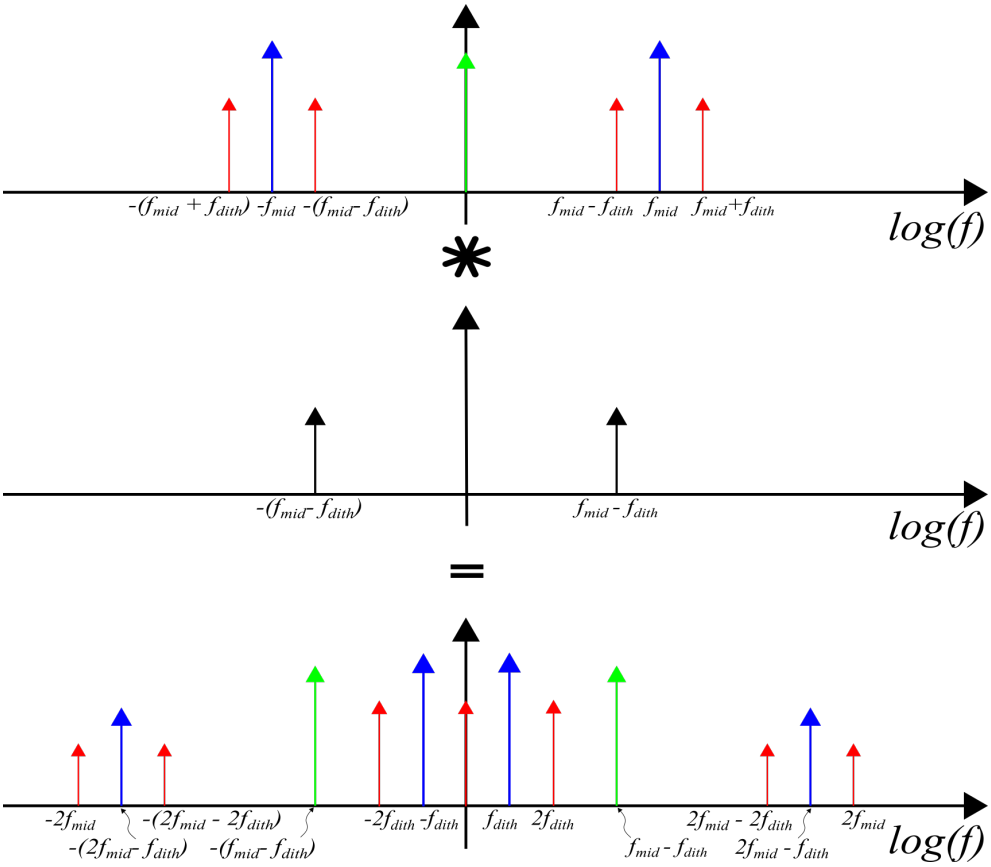

Figure 5.10: Result of the second demodulation performed at $f_{mid} - f_{dith}$ to extract the dithering signal.

ter implemented might not be sufficient to lower the level of these harmonics below the noise of the system, that is around $\sigma_{in} \simeq 100$ nV, as demonstrated in chapter 4. This is especially true when the dithering signal has to be extracted. As shown in figure 5.10, after the second demodulation, the information regarding the conductance of the waveguide is at f_{dith}, that is usually around few kHz. So, even with the lowest bandwidth achievable by the low-pass filter above, that is 3 Hz, the maximum achievable attenuation is $\simeq 10^3$. With a typical signal for the conductance of the waveguide around 100 mV, even after the low-pass filtering action, the sinusoid is still well visible over the noise.

Since these spurious sinusoids are at very precise frequencies, rather than using higher-order IIR filters, a periodic notch filter can be implemented.

5.5.1 Cascaded integrator-comb filter

The simplest implementation of a periodic notch filter is a moving average filter. A moving average filter simply sums up a certain number of samples and divides the result by the number of samples summed. So, in the discrete time domain, it is

$$y[n] = \frac{1}{M}(x[n] + x[n-1] + x[n-2] + ... + x[n-M+1]) \tag{5.22}$$

for a M-point moving averager. This solution is known as "non-recursive moving averager", and its practical implementation is shown in figure 5.11(a). To study the behavior in the frequency domain the transfer function of the system has to be computed. So, applying the Z-transform, it is

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{1}{M}(1 + Z^{-1} + Z^{-2} + ... + Z^{-M+1}) = \frac{1}{D}\sum_{n=0}^{M-1} z^{-n} \tag{5.23}$$

that is a geometric series that converges to

$$H(Z) = \frac{1}{M}\sum_{n=0}^{M-1} z^{-n} = \frac{1}{M}\frac{1-z^{-M}}{1-z^{-1}} \tag{5.24}$$

To evaluate the frequency behavior, one should set $z = e^{j\omega T_S}$. The transfer function results to be

$$H(j\omega) = \frac{1-e^{-j\omega T_s M}}{1-e^{-j\omega T_s}} = \frac{e^{-j\omega T_s M/2}(e^{j\omega T_s M/2} + e^{-j\omega T_s M/2})}{e^{-j\omega T_s/2}(e^{j\omega T_s/2} + e^{-j\omega T_s/2})} \tag{5.25}$$

Recalling Euler's identity for which $2jsin(x) = e^{jx} + e^{-jx}$, the result can be written as

$$H(j\omega) = \frac{e^{-j\omega T_s M/2}}{e^{-j\omega T_s/2}}\frac{2jsin(\omega T_s M/2)}{2jsin(\omega T_s/2)} = e^{-j\pi f(M-1)}\frac{sin(\pi f T_s M)}{sin(\pi f T_s)} \tag{5.26}$$

It's clear how this function has periodic zeros, hence periodic notches, every time that $sin(\pi f T_s M) = 0$, that is

$$\pi f T_s M = k\pi \Rightarrow f_{notch} = \frac{k f_s}{M} \tag{5.27}$$

Unfortunately, this solution is very resources-demanding, since it needs M-1 registers, M-1 summations and a multiplication. A better approach can be

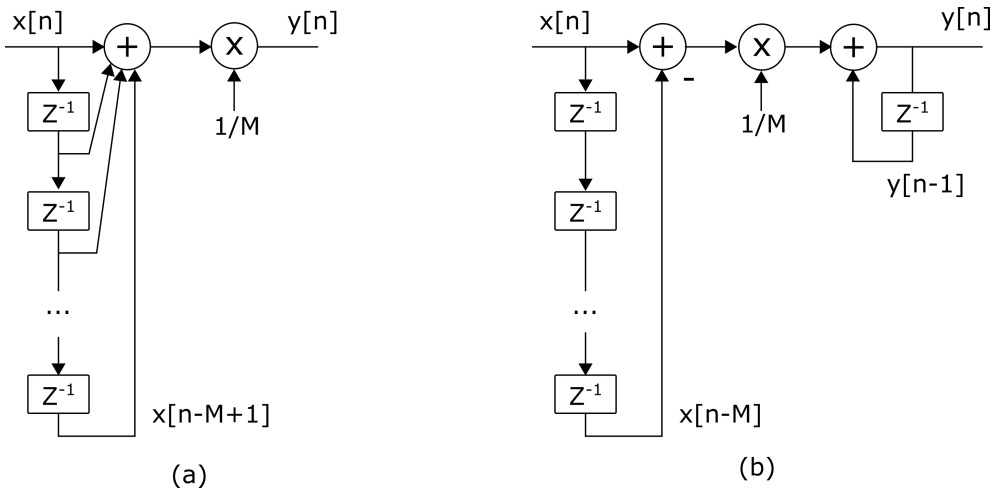

Figure 5.11: (a) Implementation of a M-points non-recursive averager and (b) implementation of a M-points recursive averager

addressed by noticing that

$$\begin{aligned} y[n] &= \frac{1}{M}(x[n] + x[n-1] + ... + x[n-M+1] + x[n-M] - x[n-M]) = \\ &= \frac{1}{M}(x[n] - x[n-M]) + \frac{1}{M}(x[n-1] + ... + x[n-M+1] + x[n-M]) = \\ &= \frac{1}{M}(x[n] - x[n-M]) + y[n-1] \end{aligned} \tag{5.28}$$

This solution only requires 2 additions (regardless of the choice of M), one multiplication and M registers (so 1 more with respect to the previous solution). This is known as "recursive moving averager", and its implementation is shown in figure 5.11(b). Also, since the $1/M$ multiplication stage is only a scaling factor, it can be avoided and compensated in the user interface. The filter obtained in this way is called "Cascaded Integrator-Comb" (CIC) filter.

Actually, to avoid the transmission of long digital words, it's possible to choose M to be a power of 2. By doing so, the gain can be simplified in the FPGA by right shifting the result of the filter of $\log_2(M)$ places. The transmitted word is still only 32-bit long. Notice that M can be chosen as a power of 2 only if the resulting notches are at the correct frequencies required to suppress the residual harmonics in the processed signal. In this case, by choosing properly the sampling and the DDS frequencies, it's possible to obtain

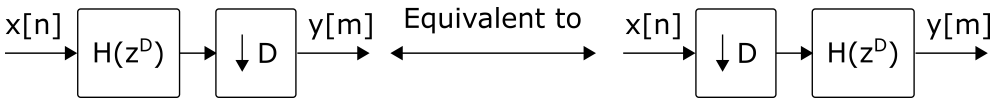

Figure 5.12: Noble identity for the downsampling operation

such condition.

However, this solution still has problems of resource occupation. In fact the first notch has to be at 1.22 kHz for this design, that is the minimum frequency needed to suppress both the second demodulation and the dithering harmonics. Recalling that the sampling frequency is 625 ksps, the system would have to average $625/1,220 = 512$ samples. Each sample is 32-bit wide, and there are two CICs per channel. Overall, it would be $512 * 32 * 2 = 32768$ flip-flops for each channel. This solution is then still too much resources-demanding.

5.5.2 Hogenauer filter

The qualitative idea to overcome this issue is to avoid storing all the samples of the sequence, but rather just one out of every D. This process is often referred to as "downsampling", or "decimation". In this way it's possible to change the equivalent sampling frequency of the sequence, that becomes

$$f_{s,new} = \frac{f_{s,old}}{D} \tag{5.29}$$

Being $z = e^{j\omega/f_s}$, if f_s decreases of a factor D, the new transfer function can be obtained by evaluating the original H(Z) as a function of z^D. This leads to

$$H(j\omega) = \frac{1 - e^{-j\omega T_s DM}}{1 - e^{-j\omega T_s D}} = ... \tag{5.30}$$

that, following the same steps above, leads to periodic notches at

$$f_{notch} = \frac{k f_s}{DM} \tag{5.31}$$

This means that, to get $f_{notch} = 1,22$ kHz, it has to be $625/1,220 = 512 = D * M$, effectively reducing the number of flip-flops needed by a factor D.

The operation described above is known in literature as "noble identity", that simply connects the effect of the reduction of the sampling frequency to the transfer function. This is graphically depicted in figure 5.12.

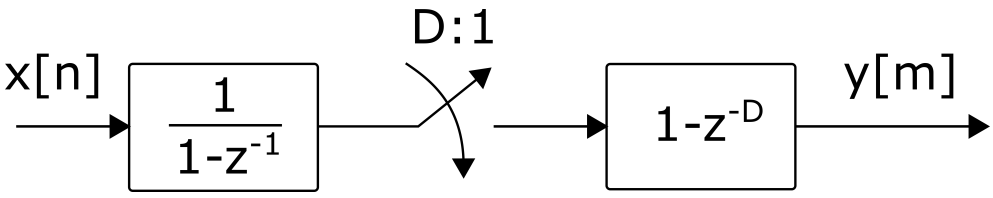

Figure 5.13: Block scheme of a Hogenauer Filter

Applying the decimation to the CIC filter, the final structure is the one depicted in figure 5.13, that is often called "Hogenauer Filter". With this improvement, it's enough to set, for instance, D = 8 and M = 64, thus saving many resources with respect to the standard moving average.

One important thing to take into account when performing decimation are aliasing phenomena. In fact, since the sampling frequency is reduced, folding effect happens in the spectrum of the signal, possibly leading to unwanted in-band noise increase or spurious tones. However, since in this case the decimation is performed after the narrow LPF, this does not happen. Therefore, this approach can be effectively used to improve the performance of the system.

5.6 Resource sharing by time-division multiplexing

The efforts done to minimize the resources used into the FPGA result to be still not sufficient, given the high number of channels that have to be acquired and then processed. The solution to this problem comes from the observation that the sampling frequency (625 kHz) is much lower than the working frequency of every stage, that is synchronous to the 40 MHz clock. This means that each chain is idle for most of the time. It's then possible to implement a smart solution where a single chain is shared between all the channels with a time-division multiplexing approach. This solution works as long as the chain is able to process all the channels before the next samples are available, so within a period of $\frac{1}{625\text{ kHz}} = 1.6$ µs.

Notice that the new approach doesn't introduce any phase delay in the

digital demodulation, since all the signals necessary for the purpose are still updated at 625 kHz.

Notice also that this solution effectively reduces the number of multipliers and adders, that are the most critical resources in the FPGA. Instead, the number of memory elements is fixed by the number of channels, and hence does not change.

This solution complicates the VHDL design of the filters, that, for this reason, are now implemented as FSMs.

5.6.1 Extraction of the dithering signal

Thanks to the resource saving obtained with the shared approach, the functionalities of the FPGA processing can now be expanded. In particular, one of the most useful improvements comes from the possibility to extract simultaneously the value of the average power in the waveguide and the dithering signal. To do so, two more copies of all the stages following the HPF were added, leading to two extra parallel processing chains.

The difference between the two pairs of processing chains is the frequency of the square waves fed to the digital mixers. The first two use two square waves in quadrature with each other at f_{mid}, the other two instead two square waves at $f_{mid} - f_{dith}$.

Notice that this improvement was not possible before the time-division multiplexing of the structure, since it would have almost doubled the resources needed for each channel.

The possibility of having simultaneously the information about the average light power and the dithering signal may be exploited for future sophisticated control algorithms.

Chapter 6

Control of a photonic circuit

Thanks to the presence of the FPGA, the designed platform can perform fast and arbitrary control actions. The specific control action will depend on the specific photonic circuit considered.

Electronic control algorithms rely on the use of sensors, to measure the controlled variable, and then on actuators, that are used to change the value of the control variable. In this specific system, the measurement is performed with the CLIPP. As described, with this sensor it's possible to read both the admittance of the waveguide and, with a help of an external modulation, its first derivative. The actuation is instead performed in the thermal domain, applying the desired voltage across a heater. A scheme of the overall control system is provided in figure 6.1, where all the domains of interest are highlighted.

In principle, these control algorithms can be implemented in the FPGA itself, that can then work without the intervention of the user. In this first test case though, it was preferred to implement the code in C# through the user interface, since it was easier to debug and control. The performances are indeed reduced, but it was still possible to demonstrate the possibility of implementing control actions with the presented platform and that the designed algorithms can actually work.

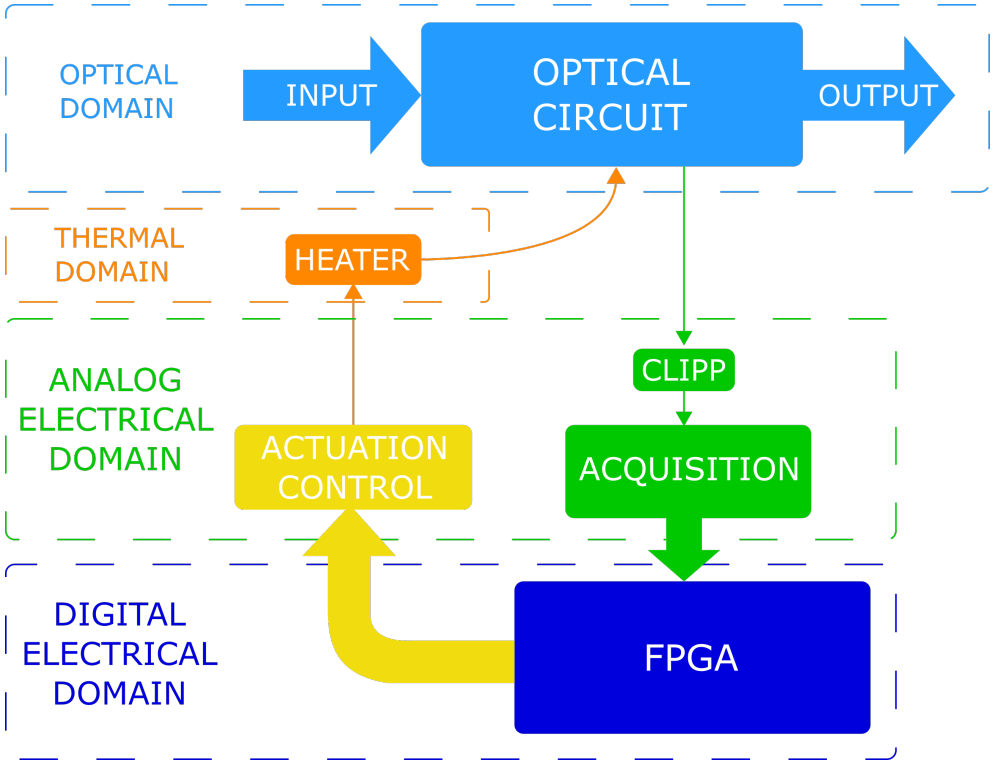

Figure 6.1: Scheme of the control system. All the different domains of interest are highlighted.

6.1 Photonic circuit

The photonic chip used to test the control algorithms is shown in figure 6.2. It features three structures based on microring resonators, each of them controlled with a heater. The first structure is used to modulate the signal intensity in the waveguide, while the second and the third are used as add/drop filters, that are able to transfer the signal at a specific wavelength from one waveguide to the other.

Four CLIPPs are present on the chip: one before the first ring, one right after it, one at the drop port of the first add/drop filter, one at the drop port of the second one. All these CLIPPs can be simultaneously read with the ASIC described in 2.1.1.

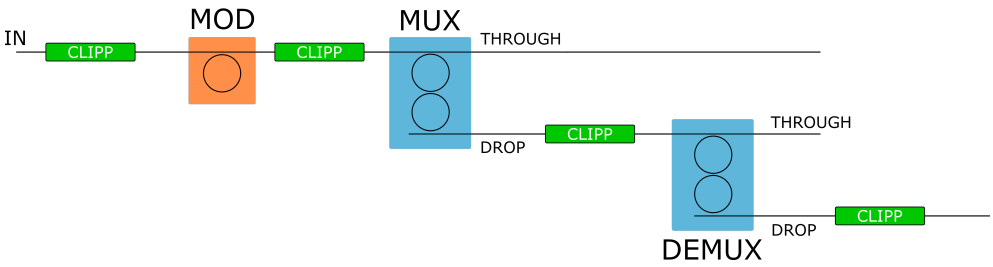

Figure 6.2: Schematic of the photonic test chip.

6.1.1 Modulator

The first microring resonator is used as a ring modulator, meaning it is able to modulate the light traveling in a waveguide to send information coded in bits. As already explained in 1, when the wavelength of the light passing through the waveguide is an integer submultiple of the optical length of the resonator, then part of the signal is transferred, through the evanescent field, into the ring and is trapped there. The light travels in the ring for multiple roundtrips before getting released back into the waveguide. Because of the optical losses of the ring and the multiple roundtrips, the light that reaches the output port at resonance is strongly attenuated, representing a '0' in the transmission of the information. On the other hand, when the ring is not tuned, the input light can reach the output without significant attenuation, representing a logical '1'. By arbitrarily tuning and detuning the ring, it's possible to create a bitstream in the waveguide, to send any kind of information.

The measured transfer function of the device is depicted in figure 6.3(a). This was obtained by linearly sweeping the heater voltage on the modulator and by measuring the signal with the CLIPP right after the modulator. A dithering tone was superimposed to the heater voltage, making it possible to also extract the first derivative, as shown in figure 6.3(b). As already stated, with the use of a heater it's possible to tune the ring, moving the peak wavelength according to the specific necessity. However, it should be recalled that the heater has a response time of few µs, meaning it can handle thermal modulations up to hundreds of kHz. This modulation is useful for control

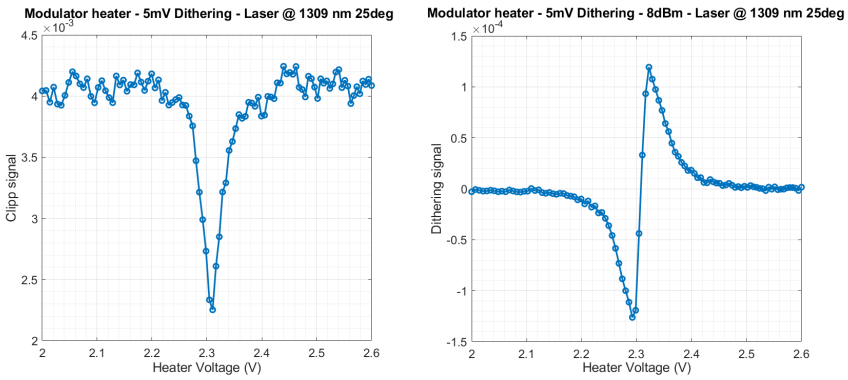

Figure 6.3: Transfer function of the modulator (a) and its first derivative extracted with the dithering technique (b).

purposes, but it's too slow to modulate the information, since it wouldn't exploit all the bandwidth available with optical transmission.

To perform a faster modulation, another actuator made with a p-n junction is integrated in the ring, as depicted in figure 6.4. By contacting the ring to drive the diode in reverse bias condition, it's possible to apply an electric field to the structure. This changes the reticle of the silicon, changing in this way the refractive index of the material. The final effect is similar to the one induced by the heater, allowing then an amplitude modulation of the output light, but at higher frequencies. As a matter of fact, this can be easily performed up to the GHz frequencies, exploiting in this way the high bandwidth available with optical signals. This modulation is not used for any control purposes.

6.1.2 Add/drop filters

The other microring resonators on the photonic chip have instead a whole different function. The difference with respect to the previous ones is the presence of a second waveguide on the other side of the ring. By tuning their resonance peak, these rings can be used to extract all the power at a certain wavelength from the input waveguide and channel it into the second one. By using a heater, it's once again possible to change the optical length of the ring,

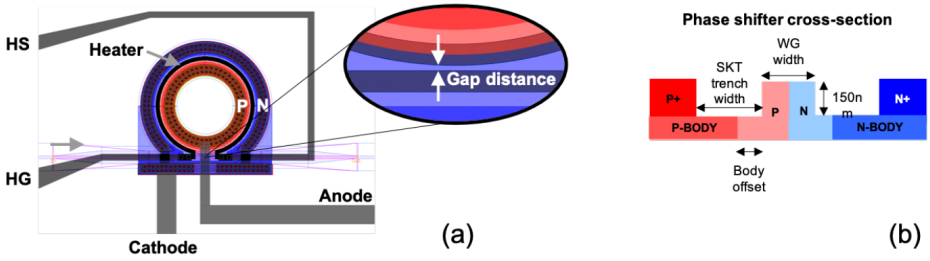

Figure 6.4: Layout and cross section of the ring modulator.

hence the resonance wavelength. This means that the structure can be tuned and used to selectively extract different wavelengths from the input waveguide, thus behaving as a tunable optical filter.

This structure is often called add/drop filter, since it can also be used to add a signal from the second waveguide to the first one, and it's widely used in telecommunication. Conceptually, the whole structure works as a wavelength division multiplexer/demultiplexer (WDM), hence the name "MUX" and "DEMUX" in the figure 6.2.

Practical implementation

The performance of an add/drop filter based on a single-ring structure is limited in terms of neighbor channel rejection for a given channel bandwidth, since the transfer function of a single ring is not narrow nor steep enough. To overcome this issue, higher order microring resonators, made by cascading multiple rings, can be exploited. These kind of add/drop filters have many advantages, such as the larger out-of-band rejection ratio, a flatter resonance peak and a steeper roll-off from the pass-band to the stop-band, as depicted in figure 6.5.

A critical point about ring resonators is the coupling between the ring and the waveguides and, in case of higher order ring resonators, also between two consecutive rings. Due to an error in the design phase, the coupling between the two rings in the considered photonic chip was poorly realized. For this reason, the transfer function of the structure is not the one expected.

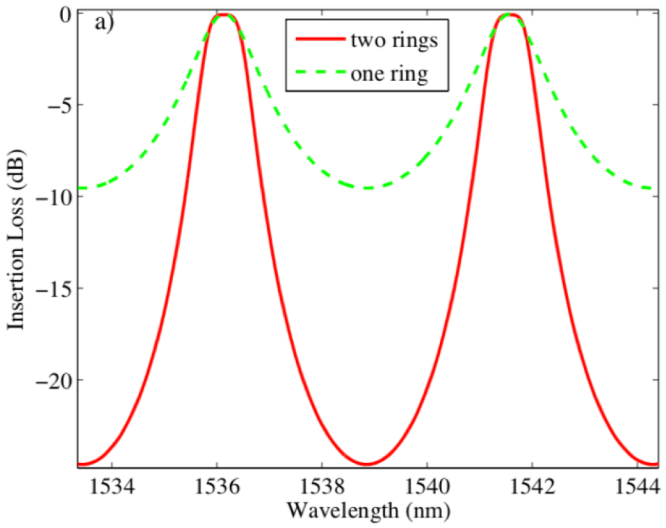

Figure 6.5: Comparison between the transfer function of a first and a second order add/drop filter.

The actual transfer function has been measured once again by linearly sweeping the voltage across the heater on the ring. The result of the measurement is shown in figure 6.6(a). Since the two rings are not properly coupled, instead of a single peak there are two maxima with a local minimum between them.

This implies that also the first derivative of this transfer function will be different from the one depicted in figure 6.3(b). In fact, due to the presence of three critical points, the derivative is expected to cross the zero axis three times. The measurement performed through the dithering technique confirms the expected behavior, as shown in figure 6.6(b). This different transfer function will have implications in the control algorithm performance.

6.2 Control algorithm

6.2.1 Modulator

The intended working point of each of these devices depends then on their functionality. In fact, for the modulator, the target working point is where the transfer function experiences the maximum slope, in order to maximize the

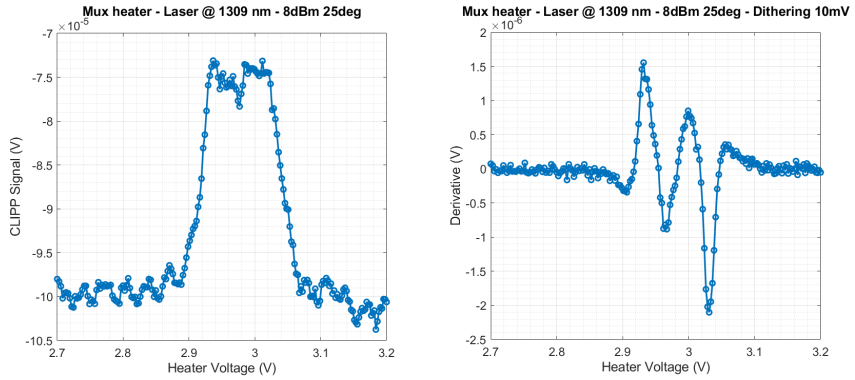

Figure 6.6: Transfer function of an add/drop filter (a) and its first derivative extracted with the dithering technique (b).

effect of the p-n junction modulation on the optical signal. The easiest way to reach this point is by using the information about the first derivative. Indeed, the maximum-slope point is where the derivative is the highest in module, i.e. either the maximum or the minimum of the signal extracted with the dithering technique. The only difference between the two points is that in one case the transmitted bits will be inverted with respect to the other. as a consequence, once one of the two is selected, it's important to be coherent during all the operations.

The algorithm to control the modulator has then to lock on one of the optical working point. To do so, at each iteration, a certain ΔV is summed to the current heater voltage. The sign of ΔV depends on the sign of the difference between two consecutive measurements of the dithering signal. To lock on a maximum, if the difference is positive, the working point is moving in the right direction, hence that the ΔV is correct. If, on the other hand, it's negative, then the sign of ΔV has to be inverted. To lock on the minimum instead ΔV has to be inverted when the difference is positive.

Notice how, once the system reaches the intended working point, the voltage of the heater will keep oscillating between two values across the equilibrium. For this application, this is not considered a problem. Either way, it is possible to reduce the entity of this oscillation by making the increment of the

heater voltage dependent on the measured difference. In this way, it's possible to reduce the oscillations to the minimum at the equilibrium and, at the same time, in case a disturb moves the system from the intended point, it's possible to go back to the equilibrium point within a reasonable number of iterations.

Notice also that the algorithm has a certain region of attraction. In fact, if a certain disturb moves the system too far away from the resonance, then the algorithm won't work, since the measurement of two consecutive value will depend much likely only on noise oscillations. This is not a problem of the algorithm itself, but rather an intrinsic limitation of the ring. In fact, if the system is too far away from the resonance peak, there is no information to be exploited to tune the ring. To overcome this limitation, it's possible to add a control that performs the sweep of the heater voltage if the intended working point is not reached after a certain number of iterations.

Eventually it should be pointed out that the platform doesn't allow the use of negative voltages, that for a heater are equivalent to the positive ones. When this condition occurs, it's possible to reset the heater to its intermediate voltage. From this condition the algorithm will naturally work.

6.2.2 Add/drop filter

For the add/drop filters, the intended working point is instead one of the two maxima of the transfer function in figure 6.6, in order to maximize the quantity of light that the structure can transfer from one waveguide to the other. To reach this point it's possible to use directly the CLIPP admittance signal and lock on one of the two maxima. In this case, the algorithm is exactly the same described before.

This is not the only possibility though, since it's also possible to use the information of the first derivative extracted with the dithering technique. To do so, the target point for the algorithm is either the first or the third zero of the dithering measurement, that correspond to one of the two maxima above. The algorithm to reach this working point has a structure similar to the previous one. At each iteration, it sums a certain ΔV to the current voltage

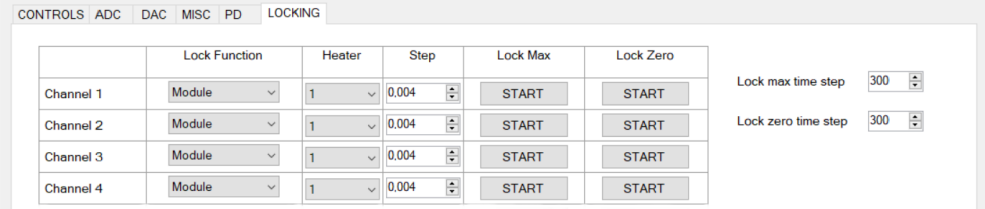

Figure 6.7: User interface for the regulation of the control action.

across the heater. The sign of this ΔV depends on the sign of the measured signal. If the measurement is positive, then the voltage of the heater must be increased, and so ΔV must be positive. On the other hand, when the measurement is negative, ΔV must also be negative, in order to reduce the heater voltage.

All the remarks carried out for the previous algorithm are still valid.

6.2.3 User interface

As already stated, the two algorithms were implemented into the interface programmed in C#. The locking action can be activated by the user thanks to a custom control, shown in figure 6.7. From here, it's possible to choose whether to lock on the zero or on the maximum, which heater has to be controlled, and which measurement should be used to perform the locking, either the real part of the admittance, the imaginary, the module or the dithering. Notice that the algorithm can be activated for any of the four channels simultaneously and independently. It's also possible to set the time step between each iteration of the control algorithms.

6.3 Experimental results

The algorithms described above were then tested experimentally. The setup is shown in figure 6.8. A tunable laser is used in order to generate the optical signal at the intended wavelength that has to be fed to the photonic chip. The signal is brought to the chip through optical fibers that have to be carefully aligned. A photodiode is added at the output of the modulator

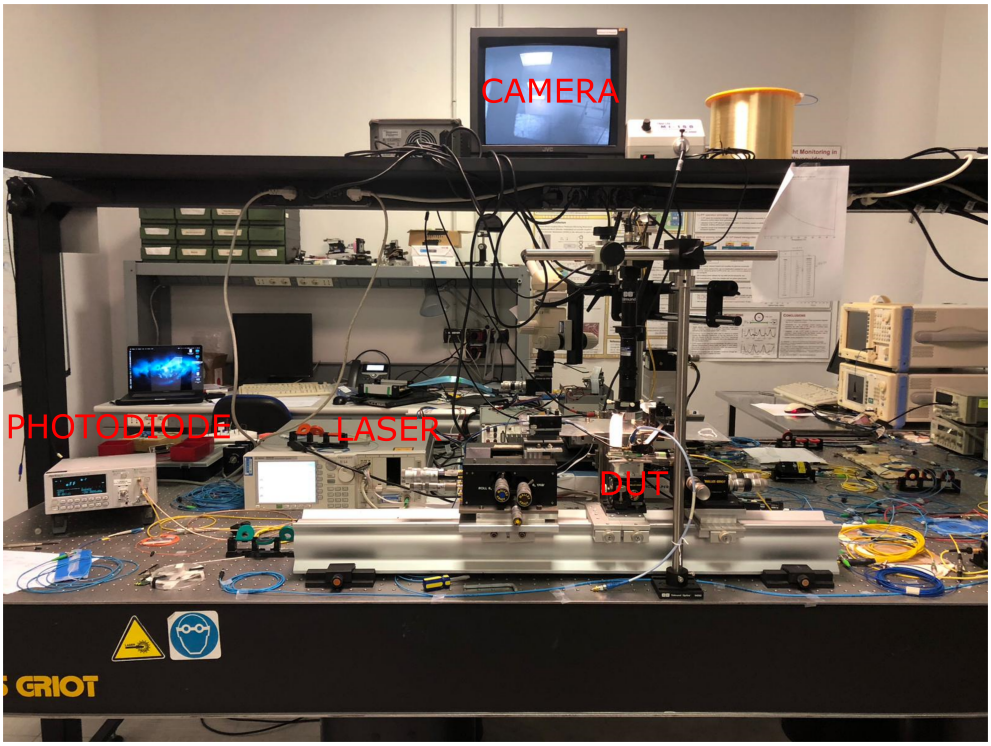

Figure 6.8: Setup for testing of the control algorithms.

waveguide, in order to validate the signal read from the CLIPP. The optical chip is then placed on the interface board with the ASIC. Below the metal support, a Peltier cell keeps constant the average temperature of the board and the chips. Eventually, as described in chapter 2, the interface board is connected to the motherboard, that, in turn, is connected to the PC that runs the user interface.

The input power was set at 8 dBm, and the wavelength at 1309 nm. Also, the dithering amplitude signal was set at 5 mV, and the temperature fixed by the Peltier cell at 25 °C. The bandwidth of the lock-in digital filter was set at 6 Hz, that allows a minimum time step for the algorithms of almost 200 ms.

6.3.1 Control of the modulator

The algorithms were first tested on the modulator. The first part of the experiment demonstrates how, by having the system near the intended working point, the algorithm tunes precisely the bias voltage across the heater to reach

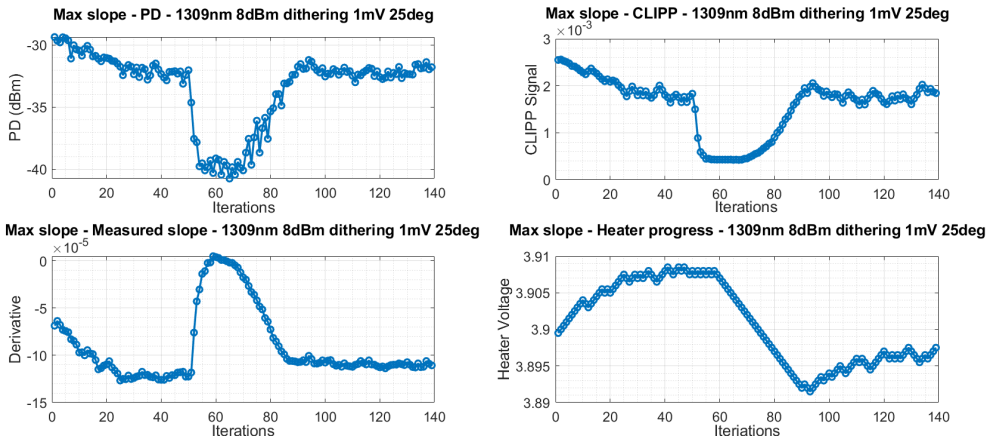

Figure 6.9: Results of the control algorithm locking the modulator to the minimum of the dithering signal.

it. As already stated, the working point has to be at the maximum negative slope of the transfer function, so it was necessary to lock on the minimum of the dithering signal.

In figure 6.9, the result of the control action is shown. After 25 iterations, the system is locked to the correct working point, and the voltage of the heater keeps oscillating around the equilibrium. Then, at the 50th iteration, a disturbance is added by turning abruptly on another heater of the photonic chip with a 4.5 V voltage. Due to the thermal crosstalk between the devices, the temperature of the modulator increases as well, moving the working point to the right in the transfer function in 6.3. This is confirmed by the fact that the signal read from the CLIPP decreases, meaning that the working point has moved towards the peak of the resonance. Notice that this can't be seen by the dithering signal, since, being at its minimum value, it would increase either way.

To compensate the temperature shift, the voltage across the controlled heater has to decrease, as shown in figure 6.9. Notice that, around the 80th iteration, the voltage reaches a minimum and starts increasing again. This happens because of the much slower control action of the Peltier cell that tries to bring the overall temperature of the system back to the correct value. The

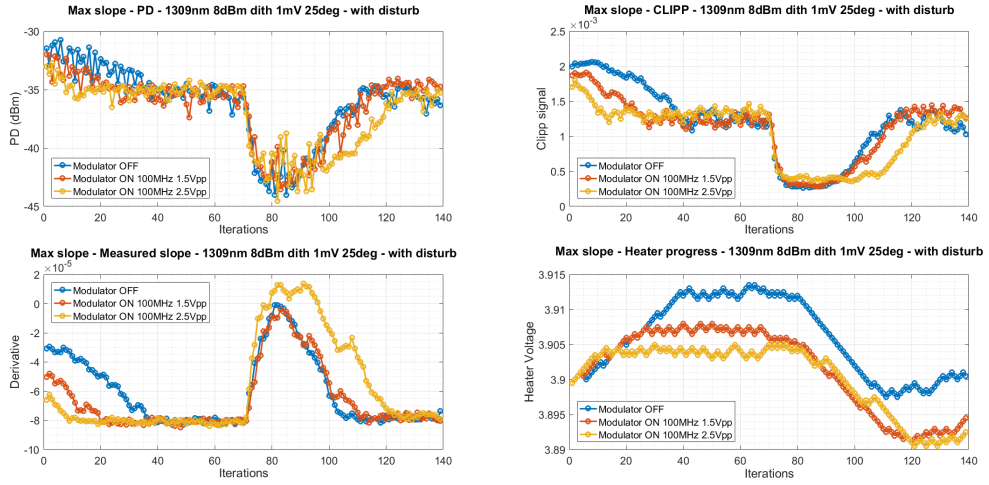

Figure 6.10: Results of the control algorithm locking the modulator to the minimum of the dithering signal with the p-n junction modulation activated.

CLIPP and PD measurements show that the power at the output of the ring is correctly restored to the initial value after the recovery transient, confirming that the algorithm works as expected.

As a second experiment, the control algorithm was tested with also the p-n junction modulation activated. Due to limitations in the setup, the modulation frequency was set at 100 MHz. Nonetheless, the experiment is still valid, because this frequency is much higher than the dithering and the CLIPP readout frequencies. As shown in figure 6.10, the system is still able to reach the equilibrium point. In case of higher peak-to-peak amplitude, the control needs more iterations to reach the equilibrium point. It's also possible to notice that the equilibrium value of the heater voltage is a bit different, meaning that the modulation changes the physical characteristics of the waveguide, causing in this way a shift of the resonance peak.

As a last experiment, the system was locked also on the zero of the dithering signal, that is the minimum peak value of the transfer function. This is not the intended working point of the modulator, and the control action was tested only for diagnostic reason. The results are plotted in figure 6.11. Notice that, at the beginning, the dithering signal first decreases and then goes to zero.

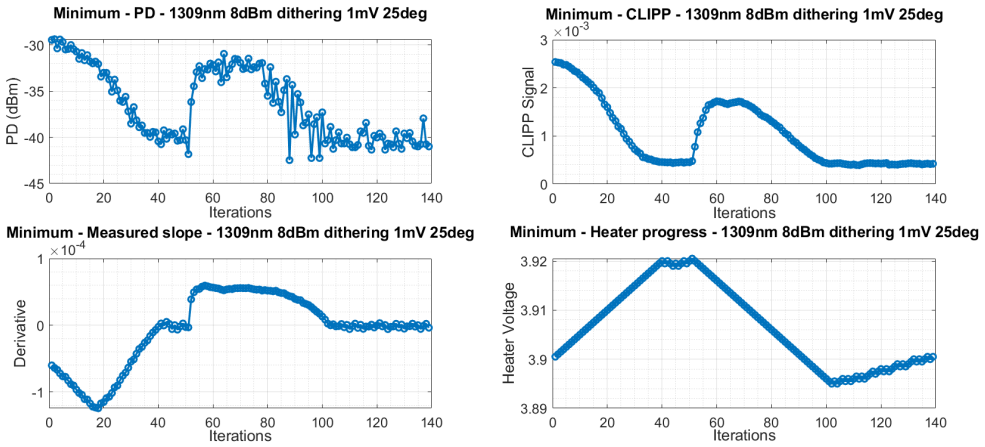

Figure 6.11: Results of the control algorithm locking the modulator to the zero of the dithering signal.

This is correct and it means that the starting point for the control action was on the left of the maximum slope point of the transfer function.

At the 40th iteration, the dithering signal reaches the zero value, and the system is at equilibrium. Then, at the 50th, a thermal disturbance is added. The working points moves to the right of the transfer function in figure 6.3. This is confirmed by the sign of the dithering signal, that is positive as expected, while no information comes from the CLIPP signal, since it would increase either way.

The heater voltage, once again, has to decrease in order to compensate the effect of the other heater. At the end, the system has returned to the equilibrium point, confirming the effectiveness of the control action.

6.3.2 Control of the multiplexer

The control algorithm was tested also on the multiplexer. Notice that the CLIPP used for the measurement is the one on the "drop" port of the multiplexer, while the photodiode is still on the "through" port. This means that the CLIPP and the photodiode will read complementary signals.

The intended working point is the first maximum of the transfer function in figure 6.6, so the first zero in the dithering measurement. As shown in

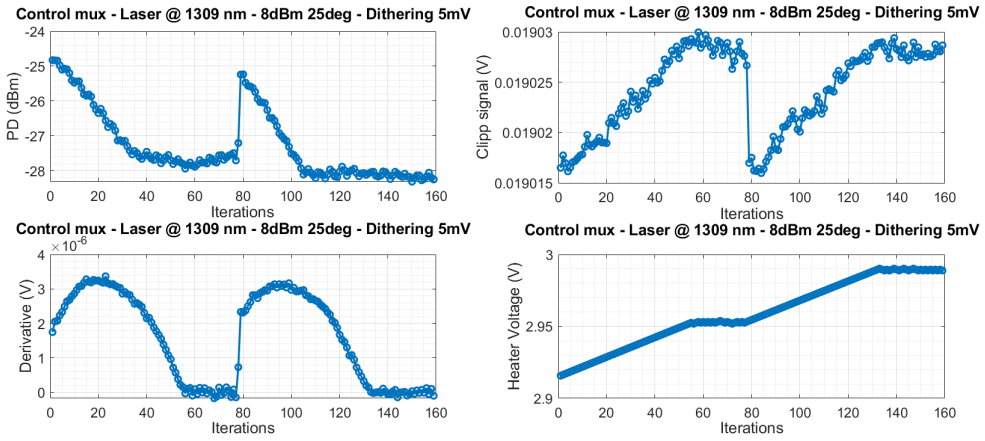

Figure 6.12: Results of the control algorithm locking the multiplexer to the zero of the dithering signal.

figure 6.12, the derivative reaches the equilibrium condition after 50 iterations. Also in this case, before reaching the intended value, the signal increases and then goes to zero, meaning that the starting point was to the left of the first maximum of the dithering function. The heater voltage increases as well, as expected.

After the 80th iteration, a disturbance is added. In particular, this time the wavelength of the input laser is increased by 140 nm, causing the peak of the transfer function to move towards higher temperatures. This is qualitatively confirmed by noticing that the heater has to further increase the refractive index of the silicon in order to make the system reaching the resonance condition.

Indeed, as shown in figure 6.12, the dithering signal increases, meaning that the working point has to move further to the right. The voltage across the heater has then to increase, as expected. After the 130th iteration, the system is stable again.

A second experiment was also conducted, as shown in figure 6.13. This time, after the 80th iteration, the wavelength of the input laser is decreased by 400 nm. The transfer function moves now towards lower temperatures, so the heater has to cool down. The derivative becomes negative, as expected.

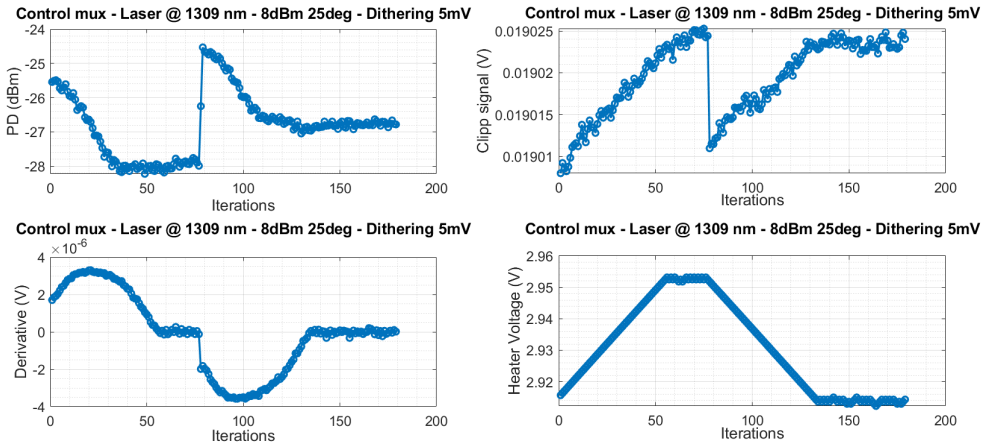

Figure 6.13: Results of the control algorithm locking the multiplexer to the zero of the dithering signal.

Notice that this time the final value read from the CLIPP and the photodiode is different from the initial one. This is correct, since in this case the larger wavelength shift causes the transfer function to move so much to the left that the algorithm ends up locking to the second maximum, that is a bit smaller than the first one.

Conclusions and future developments

This thesis addresses the problem of the lack of methods to realize closed-loop control strategies to stabilize photonic circuits. Indeed, photonic devices suffer from temperature drifts, fabrication tolerances and crosstalk. Their working point is then often different from the one intended, and may also drift during the operations. Therefore, an active control is needed to successfully operate complex photonic systems.

To solve these issues, a breakthrough was the invention of the CLIPP (ContactLess Integrated Photonic Probe). The CLIPP is a sensor that allows non-invasive monitoring of optical signals inside photonic circuits. In addition, being easily integrable, it can be virtually placed anywhere in the circuit to perform an extensive point-to-point monitoring. This paves the way for the implementation of real-time control algorithms. The actuation is performed with the use of heaters that locally change the temperature of the system. By exploiting the dependence of the refractive index of silicon on temperature, it's possible to change the optical path of each device to tune its working point.

Given the low intensity of the signal, a low-noise readout based on lock-in technique has been conceived. At the same time, the system should be designed to be as general as possible, in order to allow the introduction of any control action. For this reason, a modular electronic system was designed. The photonic chip is connected to a tailored ASIC that allows a low-noise preamplification and demodulation of the CLIPP signal. Due to geometric reasons, the

ASIC and the photonic chip are placed on an interface board, whose only task is to connect the ASIC to the motherboard with all the components necessary to close the feedback loop.

The motherboard is then the core of the system. It features:

- a stimulation chain, to generate the forcing signal for the CLIPP and the ones used for its demodulation;
- 16 acquisition chains, to properly amplify, filter and digitize the data from the ASIC;
- 16 actuation chains, to independently control 16 heaters and realize different control actions in parallel;
- an ASIC control chain, to set the working parameters of the ASIC.

A commercial module (XEM6310 by Opal Kelly) that mounts a Xilinx Spartan-6 FPGA is used to control all the electronic components on the board and to handle the communication of the system with a custom C# user interface. A specific VHDL entity has been designed to control each element of the board, usually in the form of a single-process FSM to properly manage the communication protocols. Through the user interface, it's then possible to configure and control the FPGA, set many parameters of the system, and provide a convenient way to visualize and store the acquired results.

The choice of a FPGA over a microcontroller allows a simultaneous reading and processing of many channels, allowing, in turn, truly independent parallel control actions on different devices.

The system was initially conceived to perform the demodulation of the CLIPP signal directly on-chip, by bringing the information in baseband. However, by doing so, the signal experiences all the $1/f$ noise of the acquisition chain after the output of the ASIC, that inevitability degrades the sensitivity of the system. To avoid this issue, a two-step demodulation approach was introduced. A first demodulation is thus performed on-chip at a frequency f_{dem}

different from the stimulation one, in order to bring the signal at an intermediate frequency f_{mid} that is low enough to be correctly acquired by the ADC but still above the 1/f noise corner frequency. A second demodulation is then performed in the digital domain to recover the final information. A reduction of the readout noise of a factor 6 was experimentally observed, confirming the effectiveness of the approach.

Since the lock-in technique is phase-sensitive, the phase behavior of the system was carefully investigated. To do so, the phase shift introduced by the components of both the acquisition and the stimulation chain was measured and then compensated in the user interface.

The two-step demodulation approach also allows a better use of the resources. Indeed, if both the real and the imaginary parts of the signal are extracted on chip, each CLIPP needs two acquisition channels on the motherboard. Instead, with the two-step demodulation, the signal, modulated around the intermediate frequency, still retains the information about both the real and the imaginary parts. By properly demodulating it in the digital domain, it is possible to extract both of them with a single acquisition channel. The number of CLIPPs that can be simultaneously read by the motherboard is therefore doubled.

This architecture also allows an easy cancellation of the offset of the motherboard. Indeed, while the signal is modulated around the intermediate frequency, the offset is still in baseband. A high-pass filter (HPF) can be then introduced before the digital demodulation to remove the offset without impairing the signal. Moreover, after the second demodulation, the residual offset is upconverted to the intermediate frequency, and the low-pass filter of the lock-in will further reduce its effect.

The digital architecture of the system was also improved, in particular for what concerns the processing of the data acquired by the ADCs. The first stage introduced is a HPF to remove the offset of the board, as discussed above. Since there is one filter for each channel, and the resources inside the FPGA are quite limited, a simple first-order filter was chosen. After that, two

square-wave digital mixers are used to extract the real and the imaginary part of the signal, as described above. The choice of square-wave mixers is justified by their extreme simplicity, since they do not require any multipliers. Once the signal is brought in baseband, two tunable low pass filters are used to set the lock-in bandwidth, allowing to perform either high resolution or high speed measurements, depending on the specific application. At the end of the structure, two filters with periodic notches were also introduced. These filters are carefully designed to completely suppress all the high-frequency harmonics resulting from the square-wave demodulations that are not filtered away by the first-order LPF.

A smart solution was also adopted to completely solve the lack of resources inside the FPGA. In fact, since the working frequency of the digital architecture is way higher than the sampling frequency, it's possible to implement a time-division multiplexing of the processing elements. This complicates the design of each of the components described above, but successfully reduces the number of resources needed. Since more resources are now available, it's also possible to expand the digital processing chain to extract other useful information directly on the FPGA.

Eventually, the board was experimentally tested with a specific photonic circuit. Two algorithms were implemented in the user interface. In this way, it was possible to demonstrate the control the working point of two different devices inside the chip. The experimental results show that, even after abrupt disturbances, the system is able to return to its equilibrium condition in a reasonable number of iterations, confirming that it can be effectively used to control a complex photonic architecture.

Future developments

The experiments performed have proven that the board is able to control and stabilize complex photonic circuits. New algorithms can then be designed and tested on new and innovative optical architectures. The specific control algorithm of interest will depend on the specific application considered, and can be initially introduced in the user interface in order to be easily tested. But eventually the control should be implemented into the FPGA to fully exploit the potential of the system.

It should be pointed out that, thanks to the time-multiplexing of the digital processing chain introduced, additional resources are now available in the FPGA, and so the processing chain can be expanded. In this way it's possible to extract useful information from the acquired signal that can be used for future purposes.

If necessary, the sensitivity of the system can be further boosted by implementing a sinusoidal digital mixer. Notice that this is also possible only thanks to the introduction of the time-division multiplexing of the digital chain that has saved many resoruces.

Bibliography

[1] *4-Channel, 16-Bit, 200 kSPS Data Acquisition System*. AD974. Analog Devices. 2011. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ad974.pdf.

[2] *420-MHz High-Speed Current-Feedback Amplifier*. THS3001. Texas Instrument. 2009. URL: https://www.ti.com/lit/ds/symlink/ths3001.pdf.

[3] *8-bit Shift Registers with 3-state Output Registers*. CD74HC595. Texas Instrument. 2004. URL: http://www.ti.com/lit/ds/symlink/cd74hc595.pdf.

[4] *A Technical Tutorial on Digital Signal Synthesis*. Analog Devices. 1999. URL: https://www.analog.com/media/cn/training-seminars/tutorials/450968421DDS_Tutorial_rev12-2-99.pdf.

[5] Isaia Belladelli. "Feedback control of micro ring resonator based photonic integrated transmitter". Master degree thesis. Politecnico di Milano, 2015. URL: http://hdl.handle.net/10589/114565.

[6] M. Carminati et al. "Design Guidelines for Contactless Integrated Photonic Probes in Dense Photonic Circuits". In: *Journal of Lightwave Technology* 35.14 (July 2017), pp. 3042–3049. ISSN: 0733-8724. DOI: 10.1109/JLT.2017.2710268.

[7] *Complete Quad, 16-Bit, High Accuracy, Serial Input, Bipolar Voltage Output DAC*. AD5764R. Analog Devices. URL: https://www.analog.

com/media/en/technical-documentation/data-sheets/ad5764r.pdf.

[8] Sergio Cova. *Lock-In amplifier with a squarewave reference*. 2015. URL: http://home.deib.polimi.it/cova/elet/lezioni/SSN08d_Filters-BPF4.pdf.

[9] *Dual Differential 16-Bit, 1 MSPS PulSAR ADC 12.0 mW in QSOP*. AD7903. Analog Devices. 2014. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/AD7903.pdf.

[10] *Dual, 16-/12-Bit nanoDAC+ with 2 ppm/K Reference, SPI Interface*. AD5687R. Analog Devices. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/AD5689R_5687R.pdf.

[11] Stefano Grillanda et al. "Post-fabrication trimming of athermal silicon waveguides". In: *Optics letters* 38.24 (2013), pp. 5450–5453.

[12] Emanuele Guglielmi. "Electronics boosts photonics: detector and electronic design for non-invasive monitoring and control of silicon photonic systems". PhD dissertation. Politecnico di Milano, 2019.

[13] Hasitha Jayatilleka et al. "Photoconductive heaters enable control of large-scale silicon photonic ring resonator circuits". In: *Optica* 6.1 (Jan. 2019), pp. 84–91. DOI: 10.1364/OPTICA.6.000084. URL: http://www.osapublishing.org/optica/abstract.cfm?URI=optica-6-1-84.

[14] Opal Kelly. *Front panel manual*. URL: https://docs.opalkelly.com/display/FPSDK/FrontPanel+HDL+-+USB+3.0.

[15] Opal Kelly. *XEM6310*. URL: https://opalkelly.com/products/xem6310/.

[16] *LogiCORE IP DDS Compiler v5.0*. Xilinx. 2011. URL: https://www.xilinx.com/support/documentation/ip_documentation/ds794_dds_compiler.pdf.

[17] *Low Capacitance, Triple/Quad SPDT +/-15 V/+12 V iCMOS Switches*. ADG1234. Analog Devices. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/adg1233_1234.pdf.

[18] *Low Power Instrumentation Amplifier*. AD8421. Analog Devices. 2012. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8421.pdf.

[19] Richard Lyons. *Understanding Digital Signal Processing (3rd Edition)*. Aug. 2011, p. 276. ISBN: 013702741-9.

[20] Richard Lyons. *Understanding Digital Signal Processing (3rd Edition)*. Aug. 2011, pp. 531–547. ISBN: 013702741-9.

[21] Francesco Morichetti et al. "Non-invasive on-chip light observation by contactless waveguide conductivity monitoring". In: *IEEE Journal of Selected Topics in Quantum Electronics* 20.4 (2014), pp. 292–301.

[22] *Nonvolatile Memory, Dual 1024-Position Digital Potentiometer*. AD5235. Analog Devices. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/AD5235.pdf.

[23] *Precision, Very Low Noise, Low Input Bias Current, Wide Bandwidth JFET Operational Amplifiers*. AD8513. Analog Devices. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8510_8512_8513.pdf.

[24] *Precision, Zero-Drift, High-Voltage, Programmable Gain Instrumentation Amplifier*. PGA281. Texas Instrument. 2013. URL: http://www.ti.com/lit/ds/symlink/pga281.pdf.

[25] *Surface Mound RF Transformer*. TT1-6-KK81. Mini Circuits. URL: https://ww2.minicircuits.com/pdfs/TT1-6-KK81.pdf.

[26] Fabio Toso. "Sistema elettronico in tecnologia CMOS per il controllo in anello chiuso di circuiti fotonici integrati riconfigurabili". Master degree thesis. Politecnico di Milano, 2018. URL: http://hdl.handle.net/10589/140071.

[27] *Unity-Gain Stable, Ultralow Distortion, Low Voltage Noise, High Speed Op Amp*. ADA4899. Analog Devices. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ADA4899-1.pdf.

[28] *Very Low Noise, Low Distortion Active RC Quad Universal Filter*. LTC1562. Linear Technology. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/1562fa.pdf.