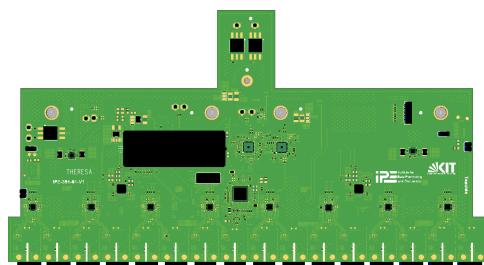


A Terabit Sampling System with a Photonics Time-Stretch ADC

Master Thesis
of

Olena Manzhura

at the Institute for Data Processing and Electronics (IPE)



Reviewer: Prof. Dr. Anke-Susanne Müller (LAS)
Second Reviewer: Dr. Michele Caselle (IPE)

15.11.2020 – 13.08.2021

Declaration

I hereby declare that I wrote my master thesis on my own and that I have followed the regulations relating to good scientific practice of the Karlsruhe Institute of Technology (KIT) in its latest form. I did not use any unacknowledged sources or means, and I marked all references I used literally or by content.

Karlsruhe, 13.08.2021, _____
Olena Manzhura

Approved as an exam copy by

Karlsruhe, 13.08.2021, _____
Prof. Dr. Anke-Susanne Müller (LAS)

Abstract

Analysis of events occurring in the range of femtoseconds is desired in many scientific experiments. The high temporal resolution needed for measuring such events imposes a great technological challenge for Data Acquisition Systems (DAQs) and Analog-To-Digital-Converters (ADCs). In order to relax the requirements on the acquisition systems, the so-called optical time-stretch technique is used to stretch the analog input signal in time. In this way, data converters at relatively moderate sample rate can be used. Measuring the signal with commercial DAQs, such as real-time oscilloscope, still poses another challenge. Due to the limited acquisition time windows of such systems, continuous measurements at high sampling rate and time resolution over a long period of time is not possible. In applications, where measurements of long-term evolution of the ultra-fast events with high temporal resolution is necessary, this is a large limitation. Therefore new concepts of DAQ based on the time-stretch method need to be considered.

In this thesis, a first demonstrator of such a new photonic time-stretch based DAQ system was developed. The system consists of a high bandwidth front-end sampling card, mounted on a back-end readout card integrating a new generation of Radio-Frequency System-On-Chip (RFSoC) for readout of the acquired samples.

The front-end sampling card integrates 16 sampling channels, each containing a Track-And-Hold-Amplifier (THA). The sampling time of these THAs can be delayed individually in steps of 11 ps, covering a range up to 11.2 ns. In this way the so-called time-interleaving method can be implemented to sample the signal at a higher rate than that normally possible due to the Nyquist theorem. The design of the board allows it to be used with the time-stretch method as well as independently from it. Furthermore, the setup allows for different sampling modes. In single-channel mode one detector is connected to one sampling channel, therefore allowing to acquire data from up to 16 detectors at the same time with one sampling point per channel. In the “multi-channel” mode, several channels (up to 16) are connected to one detector via power splitter, therefore allowing multiple sampling points for one detector.

High-speed ADCs, integrated in the RFSoC, with 14-bit resolution and a sample rate of up to 2.5 GS/s allow continuous sampling of the signal with high temporal resolution. Using the time-interleaving technique for all sixteen ADCs results in an overall maximal achievable sample rate of 40 GS/s. When used in combination with the time-stretch technique and considering currently achievable time-stretch factors, a time resolution in the range of hundred of femtoseconds is possible.

The RFSoC on the back-end readout card integrates a processing unit and a Field Programmable Gate Array (FPGA). A firmware running on the FPGA is responsible for programming and controlling the components on the sampling card, as well as collecting the acquired samples and sending it to the following processing system via high-speed connections. The processing unit, hosting e.g. an operating system or a standalone application, allows for the user to control and monitor the overall system via common periphery, e.g. Ethernet.

The name given to the system is THERESA, an acronym for “Terahertz Readout Sampling”.

Zusammenfassung

Résumé

Contents

1. Introduction	1
1.1. Objective	2
2. Motivation	3
2.1. Requirements in THz Science	3
2.1.1. Coherent Synchrotron Radiation	3
2.1.2. Electro-Optic Techniques for Longitudinal Bunch Profile Diagnostics	6
2.2. Photonic Time-Stretch Method	8
2.3. Analog-To-Digital Converter	9
2.3.1. Characteristics of Analog-To-Digital-Converters	11
2.3.1.1. Quantization Noise	12
2.3.1.2. Static parameters	13
2.3.1.3. Frequency-Domain Dynamic Parameters	15
2.3.1.4. Sampling Theory	20
3. Architecture Of The New Readout-System - THERESA	23
3.1. State Of The Art Readout-Systems	23
3.1.1. KAPTURE	23
3.1.2. KAPTURE-2	27
3.2. Proposed Architecture for THERESA	28
3.2.1. Front-End Sampling Card	28
3.2.1.1. Time Interleaving	29
3.2.1.2. Implementation	31
3.2.2. Readout Card	33
4. Design Of The Front-End Sampling Card	35
4.1. Schematics	35
4.1.1. Connectors	38
4.1.2. Sampling-Channel	40
4.1.3. Clock Distribution	47
4.1.4. Digital-To-Analog-Converter Channels	52
4.1.5. Power Supply	53
4.2. Layout	56
4.2.1. PCB Substrate Selection and Metal Layer Stackup	58
4.2.2. Transmission Lines	59
4.2.3. Component Placement and Routing	67
5. Back-End Readout Card and System Integration	71
5.1. Xilinx ZCU216 Evaluation Card	71
5.2. Firmware	75
5.2.1. Programmable Logic - Hardware design	76
5.2.2. Processing Unit - Software Design	77

6. Conclusion and Outlook	79
7. Conclusion and Outlook	81
Acknowledgements	83
Appendix	85
A. Characteristic Impedance Of Coplanar Waveguides	85
B. Code	86

List of Figures

2.1. Basic scheme of an electron storage ring (redrawn from [?])	4
2.2. Electromagnetic spectrum	4
2.3. Incoherent SR and CSR	5
2.4. Electrons interact with their own radiation [?]	5
2.5. Scheme of Scanning-Type Electro-Optical Sampling System [?]	7
2.6. Scheme of Spectrally Encoded Electro-Optical Detection System [?]	7
2.7. Working principle of the electro-optical time-stretch technique [?]	8
2.8. pn-junction with depleted region [?]	9
2.9. Transfer function of ideal, 3-bit ADC	10
2.10. SHA timing example	10
2.11. Track-And-Hold-Amplifier schematic and principle [?]	11
2.12. Measurement setup for quantization error	12
2.13. Quantization noise as function of time (redrawn from [?])	13
2.14. Effects of Offset and Fain error in ADC	14
2.15. ADC Nonlinearities	15
2.16. SFDR definition	17
2.17. Aperture jitter and SJNR	19
2.18. Aliasing	21
3.1. THz measurement with KAPTURE	24
3.2. Photo of the power splitter developed at IPE	25
3.3. General architecture of the KAPTURE system	26
3.4. Signal and sampled points S_1 to S_4	26
3.5. Comparison between KAPTURE v1 and v2	27
3.6. Photo of KAPTURE-2 system	27
3.7. General architecture of the THERESA sampling card	28
3.8. Time-Interleaving Method	29
3.9. Offset-Mismatch in Interleaving [?]	30
3.10. Gain-Mismatch in Interleaving [?]	30
3.11. Timing-Mismatch in Interleaving [?]	30
3.12. Timing-Mismatch in Interleaving [?]	31
3.13. Track-And-Hold Timing diagram	32
3.14. Discrete components vs. IC	33
4.1. Capacitor equivalent circuit	37
4.2. Impedance response of a real capacitor	37
4.3. Rendering of FMC+ connector	38
4.4. Male and female type V connector	39
4.5. LPAM 8 × 50 connector	39
4.6. LPAM 6 × 20 connector	40
4.7. HMC5640 THA schematic	42
4.8. Connection of the analog and digital grounds at the THAs	42

4.9.	Delay chip SDI connections	43
4.10.	NB6L295 delay chip schematic	44
4.11.	Schematic of a resistive voltage divider	44
4.12.	NB6L295 Delay Chip Schematic	45
4.13.	SN74AVC32T245 bus transceiver	46
4.14.	LVPECL driver topology	47
4.15.	Overview of the clocking paths on the sampling board	48
4.16.	PLL block diagram	49
4.17.	PLL block diagram	49
4.18.	PLL loop filter components	50
4.19.	Schematics of the LMX2594	50
4.20.	Schematic of the fanout	52
4.21.	DAC-channel with balun. Signal propagates from right to left.	53
4.22.	EMI-filter used for power supply	54
4.23.	Recommended schematic of the ADP1741 voltage regulator [?]	55
4.24.	Via types	58
4.25.	Metal Layer Stackup	59
4.26.	Screenshot of the Polaris Si9000e	59
4.27.	Coplanar Waveguide with Ground	60
4.28.	CWG, Z_o vs ϵ_r	61
4.29.	CWG, Z_o vs. a	62
4.30.	Z_o vs. lower trace thickness	62
4.31.	Edge-Coupled Coplanar Waveguide	63
4.32.	DCWG, Z_{diff} vs. w	63
4.33.	DCWG, Z_{diff} vs. ϵ_r	64
4.34.	DCWG, Z_{diff} vs. upper trace width	64
4.35.	Offset Differential Coplanar waveguide	65
4.36.	DOWG, Z_{diff} vs. w	65
4.37.	DOWG, Z_{diff} vs. s	65
4.38.	DOWG, Z_{diff} vs. ϵ_r	66
4.39.	DOWG, Z_{diff} vs. upper trace width	66
4.40.	Trace accordions	68
4.41.	Stitching vias	69
5.1.	Topview of ZCU216 evaluation board with labeled components	72
5.2.	Zynq Ultrascale+ RFSoC block diagram	73
5.3.	ZCU216 Evaluation Tool	73
5.4.	GUI of the RF Data Converter Evaluation Tool	74
5.5.	Schematic of the firmware and processing unit on the readout card	75
5.6.	Simple example design with the RF Data Converter	76
5.7.	SDI Timing diagram for the NB6L295 delay chip [?]	77

List of Tables

2.1. Some KARA parameters [?]	5
3.1. Real Time Oscilloscopes Examples	23
4.1. FMC+ Voltages	39
4.2. HMC5640 Characteristics	41
4.3. NB6L295 Characteristics	45
4.4. LMK04808B loop filter characteristics	51
4.5. LMX2594 loop filter characteristics	51
4.6. Power consumption of components on the board	53

1. Introduction

In many scientific applications and experiments the observation of non-repetitive, statistically rare events with very fast occurrences is desired. As these events might occur on a time range of femtoseconds, real-time measurement systems with fine temporal resolution and capable of long acquisition times are necessary. This imposes high technological challenges on Data Acquisition Systems (DAQs) and Analog-To-Digital-Converters (ADCs).

One bottleneck in the acquisition of ultra-fast events is the limited performance of commercially available ADCs. The limitation posed by the converters is a trade-off between the dynamic range (Effective Number Of Bits (ENOB)) and sampling rate of the converters. As the sampling rate increases, ambiguity of the comparators in the ADC and sampling errors due to clock jitter become major limiting factors on the overall performance. [?]

A first demonstration of a concept to overcome these limitations was presented in 1999 by [?]. The idea is to stretch the analog signal in time before digitizing it in the converter and hence relax the demands on the ADCs performance. This time-stretching is accomplished by using chirped optical pulses and dispersion in optical fibers. The concept is therefore called “photonic time-stretch” and was successfully tested in combination with a moderate-speed ADC in [?].

Since then, the time-stretch method has been continuously improved and has found use in many applications. For example, in biomedical diagnostics, a first demonstration of an artificial intelligence based high-speed phase microscope has been developed. It uses time-stretch quantitative phase imaging (TS-QPI), a technique based on the time-stretch concept which enables simultaneous measurement of phase and spatial intensity profiles. This allows label-free classification of cells for cancer diagnostics and drug development. [?]

The time-stretch concept is also very interesting for applications in particle accelerators due to the short timescales involved. Relativistic electron bunches interact with their own radiation which can lead to the formation of spatial microstructures inside the bunches, a phenomenon also called micro-bunching instability. This is a source of intense pulses of terahertz radiation (Coherent Synchrotron Radiation (CSR)) and therefore an important field of study. A first demonstration of direct observation of these instabilities was performed at the synchrotron facility Source optimisée de lumière d'énergie intermédiaire du LURE (SOLEIL) using a time-stretched signal together with a real-time oscilloscope. [?]

The use of the time-stretch method in different applications has demonstrated the advantages to measure events with femtosecond resolution. Still, commercially available real-time diagnostics systems are limited in memory space. The acquisition time of such systems at maximum sampling rate lies in the range of milliseconds at best. It is therefore not possible to measure data continuously over a large period of time. This creates a problem in applications where a longer observation time (up to hours) is required, e.g. in accelerator applications where the turn-by-turn analysis of the electron bunches is desired in order to study the evolution of the bunch profiles.

This challenge was the motivation to design novel ultra-fast acquisition systems based on the photonic time-stretch ADC. Together with the next generation of Field Programmable Gate Array (FPGA)-based systems with integrated high-performance ADCs this gives rise to a new concept of DAQ, the photonic time-stretch DAQ. The photonic time-stretch DAQ consists of a photonic part, which consists of the time-stretching section and the conversion of photons into electrical signal with a photo-detector. Furthermore, such a system has one or multiple ADCs converting the analog values into digital signals. The digital signals are then processed in a computing unit and broadcast to other units as needed if the system is integrated into a cluster of measurement systems.

1.1. Objective

In this thesis, a first demonstrator of a DAQ-system based on the time-stretch concept has been developed. This system, called Terahertz Readout Sampling (THERESA) system, enables high-speed measurements of ultrafast events with a time resolution in the range of femtoseconds.

In order to achieve such high resolution, the time-stretch technique will be used in order to stretch the input signal in the range of pico- to nano-seconds. The input signal will be continuously sampled by high-speed ADCs with a temporal resolution defined by the user as needed. To sample the signal, the ADCs need to have a sampling rate in the order of several GHz. The amplitudes of the signals to be measured are very small and an appropriate resolution of the ADCs has to be chosen in order to guarantee an ENOB of at least 10 bits. [?]

This leads to the next challenge: Sampling at several GHz with high resolution, implies a large amount of data in the range of Terabits per second. In order to enable such a high data-throughput, the system will be based on a new generation of System-On-Chip (SoC), integrating a FPGA and a processing unit together with the high-speed ADCs. The SoC will have high-speed peripherals in order to guarantee the continuous high-speed data-throughput. Combination with the FPGA should allow for flexible system tuning for a user-defined application. The user will be able to control and configure the system via an application or operating system running on the processing unit.

Furthermore, the system should be compatible with already existing high-speed DAQ frameworks (e.g. based on PCI Express (PCIe)) and can be easily integrated into the system for the user application (e.g. through optical fibers to a distributed instrumentation system). However, stand-alone operation should also be possible. Furthermore, the DAQ should be designed in such way that usage independent from the time-stretch method is possible.

The overall thesis is structured in the following way: Chapter 2 gives the necessary theoretical background for the new THERESA system. The subject of Terahertz (THz) science in particular is touched being the main motivation for the design of the novel time-stretch sampling system. Chapter 3 covers the general architecture of THERESA, including also state of the art readout-systems, especially the Karlsruhe Pulse Taking Ultra-fast Readout Electronics (KAPTURE) which is in operation at the Karlsruhe Research Accelerator (KARA). Chapter 4 describes the design steps of the front-end sampling card of THERESA in detail. Chapter 5 covers the description of the back-end readout card, as well as the design of the appropriate firmware. At last, results are concluded and an outlook for the newly developed system is given.

2. Motivation

As the main aspired use case of the newly developed time-stretch Data Acquisition System (DAQ) lies in accelerator physics applications, especially in Terahertz (THz) science e.g. at Karlsruhe Research Accelerator (KARA), an introduction into this topic is given in the following section.

After that the general architecture and basic theory of a photonic time-stretch DAQ is given. First, the basic working principle of the time-stretch concept is explained. Then, a short overview of the basic Analog-To-Digital-Converter (ADC) theory is given, together with the most prominent figures of merit. Knowledge and understanding of ADC characteristics is necessary to evaluate the overall performance of the converter.

2.1. Requirements in THz Science

Recent years have seen an increasing interest in THz radiation, ranging from 3 THz up to 30 THz¹, as it allows non-destructive analysis of organic material. This is possible because unlike e.g. X-Rays, THz radiation is not ionizing. It is therefore of great interest to use THz radiation in fields like biology, medicine or material science. However, until recently the usage of THz radiation was very limited, as generation of such radiation has proven to be difficult.

Electron storage rings, also called synchrotrons, are a potential source of THz-radiation. The emission of THz radiation is closely linked to instabilities of the charged particles which are accelerated in the synchrotron. [?] These instabilities occur in the time range of femtoseconds and cause bursts of THz radiation. The periodicity of these bursts depends on multiple parameters of the synchrotron and therefore imposes a challenge on controlling the emission of THz radiation. Studying the dynamics of these instabilities is an important step towards the application of synchrotrons as source of THz radiation. [?]

2.1.1. Coherent Synchrotron Radiation

In synchrotron radiation facilities synchrotron radiation (SR) is produced by accelerating relativistic electrons. Emission of SR occurs, when electron beams are bent or deflected with dipole magnets or using undulators. The latter are used to make the electrons oscillate by generating a periodic magnetic field. Figure 2.1 shows the general scheme of an electron storage ring.

Electrons, which are grouped to “electron bunches”, are generated with an electron gun and accelerated to relativistic speeds² by a pre-accelerator (often a linear accelerator (LINAC) or a booster ring accelerator). After being brought up to their nominal energy³, the

¹At KARA: 0.1 THz to 1.2 THz

²almost speed of light

³in a booster

bunches are injected into the storage ring. In the ring, the path of the electron bunches is altered by dipole magnets, guiding them on a circular trajectory. Due to emission of SR at each bend, the electrons lose energy, which has to be compensated for. This is done by accelerating them with an electric field inside a Radio Frequency (RF) cavity. Not shown in the drawing are the beamlines, which lead the SR radiation, or rather chosen wavelength ranges, through an optical system to the respective user experiments. [?, ?]

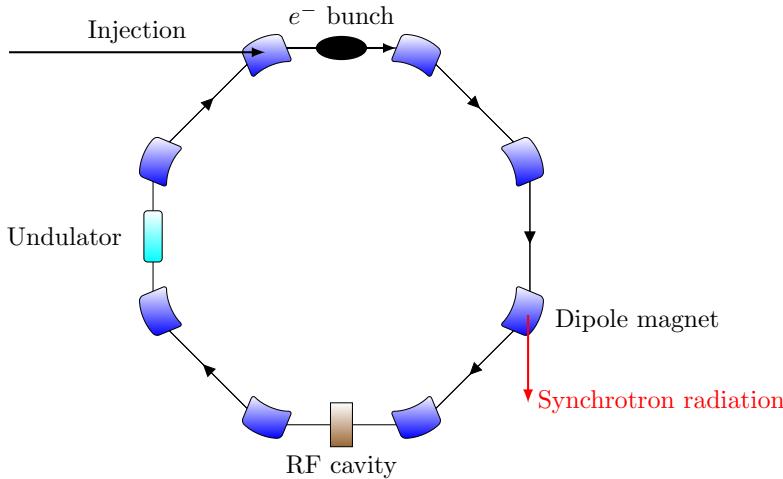


Figure 2.1.: Basic scheme of an electron storage ring (redrawn from [?])

The range of SR reaches from hard X-rays down to the infrared region of the electromagnetic spectrum (see Figure 2.2). SR shows properties like high intensity, high collimation, polarisation and generation in pulses of well-defined time-duration. Due to this properties, synchrotrons are used for microscopy, spectroscopy, and time-resolved experiments in such fields like condensed matter physics, biology, material science and many more.

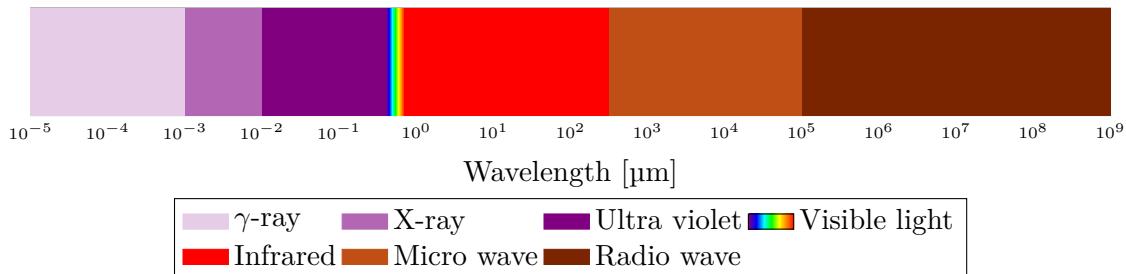


Figure 2.2.: Electromagnetic spectrum

Karlsruhe Research Accelerator

At the synchrotron light source KARA, the possibility to utilize the synchrotron as a source of THz is actively researched. The photonic time-stretch DAQ, which has been developed in this thesis, should also be integrated into the beam diagnostics system at KARA. Therefore, a short overview of some parameters of this facility has been given below.

KARA is located at the Karlsruhe Institute of Technology (KIT) and is operated by the Institute of Beam Physics and Technology (IBPT). The storage ring can be filled up with up to 184 electron bunches with a distance of 2 ns ($\cong 500$ MHz) between two adjacent bunches. The main accelerator parameters are listed in Table 2.1.

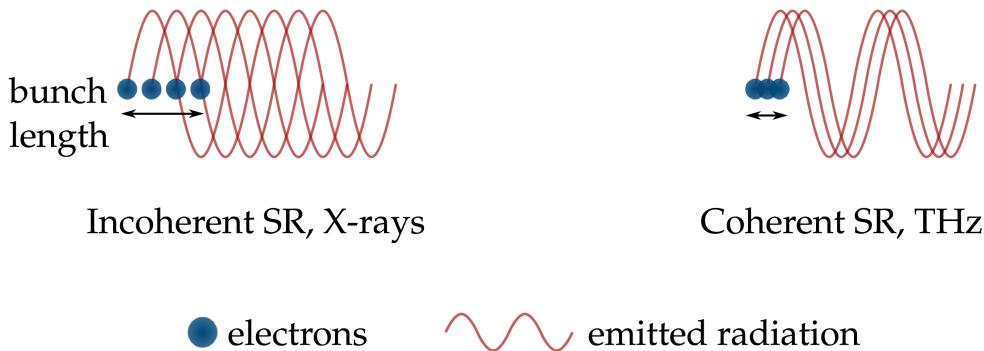
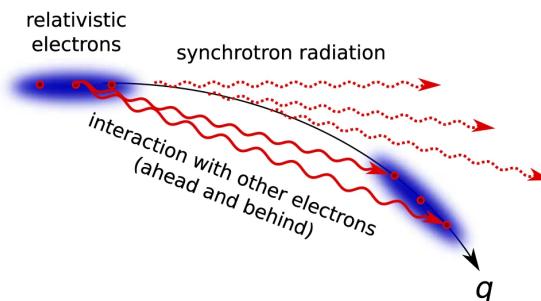
One scientific focus at KARA lies in the study of so-called “micro-bunching instabilities” which are described in the following.

Table 2.1.: Some KARA parameters [?]

Parameter	Value
Beam energy (max.)	2.5 GeV
Circumference	110 m
Revolution Frequency (one electron)	2.7 MHz
Minimum bunch spacing	
multi-bunch	2 ns
single-bunch	368 ns
Bunch length (rms)	
normal operation	45 ps
short bunch	2 ps

Micro-Bunching Instabilities

Increasing demands in current and future accelerators applications call for higher brilliance of the emitted radiation. This is achieved by shortening the electron bunches. As illustrated in Figure 2.3, this results in emission of Coherent Synchrotron Radiation (CSR) the spectrum of which spans from 100 GHz up to THz. Due to this CSR the bunches interact with their own radiation (see Figure 2.4), which introduces complex longitudinal dynamics.

**Figure 2.3.:** Incoherent SR and coherent SR due to shorter electron bunch length [?]**Figure 2.4.:** Electrons interact with their own radiation [?]

These dynamics are the so called micro-bunching instabilities, the formation of micro-structures (in the sub-millimeter range) in the longitudinal density profile of the electron bunches. These instabilities occur in bursts and are hard to control, as they depend on a number of system parameters. This imposes on one side a huge limitation to the stable operation of the overall system at high current density/short bunch length mode. On

the other side, these instabilities themselves emit brilliant THz radiation that could be potentially used in imaging applications. Such applications however require a stable power of the radiation. Therefore, a control of these instability bursts could potentially make them a source of THz radiation for user-applications. A thorough understanding and studying of these beam dynamics is therefore an important step towards providing an applicable THz source. [?, ?] In order to make such investigations possible, appropriate beam diagnostic systems are required, which are capable of both capturing (ultra-)fast and long-term changes in the bunch profile.

Control of Micro-Bunching Instabilities

The Exploration et contrôle ULTRArapide de la dynamique des paquets d'électrons dans les sources de lumière SYNChrotron (ULTRASYNC) project, funded by ANR-DFG⁴, has an objective of ultrafast study and control of electron bunches in synchrotron light sources.

There is the question of control (i.e. suppression) of the bursts of THz radiation occurring during the micro-bunching instability. The goal is to obtain a high power and stable coherent emission. The current experimental setup uses a relatively simple feedback loop:

- A bolometer/Schottky barrier diode detector which produces the input signal for the feedback loop.
- A low-cost Field Programmable Gate Array (FPGA) (Red Pitaya) that controls the accelerating voltage of the synchrotron based on the input

However, there are limitations in the controllable bunch charge in the accelerator this feedback loop can handle, which is around 10 mA. Therefore, an open question is whether measuring each THz pulse using the setup

- Electro-Optical sampling and time-stretching
- Association with the new FPGA-based system, i.e. Terahertz Readout Sampling (THERESA) system
- Finding adequate feedback, programmed in the FPGA

would help in solving the problem and allow the control to succeed also at higher currents (goal: 15 mA) [?].

2.1.2. Electro-Optic Techniques for Longitudinal Bunch Profile Diagnostics

Methods for analyzing the longitudinal profile of electron bunches are based on a similar, if not the same, electro-optical concept as the time-stretch method. Two most prominent methods are briefly described for the sake of completeness.

Scanning-Type Electro-Optic Sampling

The scanning-typeElectro-Optic Sampling (EOS) samples one point at the time of the THz pulse, emitted e.g. from an electron bunch, at each acquisition, hence the naming of this method.

A short laser pulse (duration typically hundreds of femtoseconds) co-propagates with a THz pulse from CSR (range of picoseconds) in an Electro-Optic (EO) crystal. Due to the Pockels effect the THz pulse causes a time dependent birefringence in the crystal. This modulates the polarization state of the laser pulse.

⁴Agence Nationale de la Recherche (ANR), Deutsche Forschungsgemeinschaft (DFG)

To sample the pulse, the delay between the laser and the THz pulse is varied. To detect the changing polarization, the polarization of the laser pulse is transformed into an intensity modulation. This is done by using polarizers, e.g. Quarter-Wave Plates (QWPs) and Wollaston Prism (WP) (as shown in Figure 2.5). A general scheme of the system is shown in Figure 2.5. For this technique a stable emission of the THz pulses is crucial, as they are not measured in one acquisition. [?]

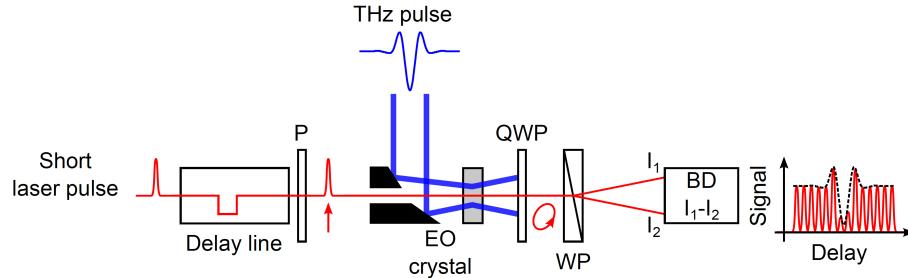


Figure 2.5.: Scheme of Scanning-Type Electro-Optical Sampling System [?]

Spectrally Resolved Electro-Optic Detection

In contrast to the EOS, single-acquisition is possible with the spectrally resolved EO detection technique. The short laser pulse is first stretched to a duration similar to the THz pulse in a dispersive material, called stretcher. In this way the pulse is chirped, meaning the instantaneous frequency of the pulse varies over time. Together with the THz pulse, the laser pulse propagates through an EO crystal. Again, the induced birefringence modulates the laser pulse, not only in time, but also in the spectral domain. The polarization state of the pulse is converted into an amplitude/intensity modulation. This is done with a series of QWP, Half-Wave Plate (HWP) and a polarizer (P) (as shown in Figure 2.6). To retrieve the THz pulse shape in time, the spectrum of the laser pulse is measured with a spectrometer. A general scheme of the system is shown in Figure 2.6. [?]

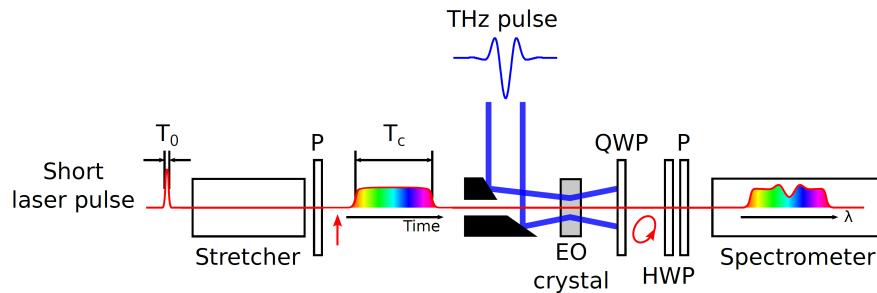


Figure 2.6.: Scheme of Spectrally Encoded Electro-Optical Detection System [?]

The temporal resolution of this method is limited due to the finite chirp rate

$$\text{chirp rate} = \frac{\text{laser bandwidth}}{\text{laser pulse duration after stretcher}}. \quad (2.1)$$

The minimal resolution T_{\min} depends on the bandwidth-limited pulse duration (before stretcher) T_0 and the duration of the chirped laser pulse T_c :

$$T_{\min} = \sqrt{T_0 T_c} \quad (2.2)$$

2.2. Photonic Time-Stretch Method

The operating principle of the optical time-stretch technique can be described in three steps (see Figure 2.7).

First, a short laser pulse (duration typically hundreds of femtoseconds) propagates in a dispersive medium, e.g. an optical fiber of length L_1 (see Figure 2.7). With the optical bandwidth of the laser pulse $\Delta\lambda$ and the dispersion parameter D_1 of the fiber, this results in a chirped laser pulse of the duration

$$T_1 = \Delta\lambda D_1 L_1. \quad (2.3)$$

The next step is the time-to-wavelength-mapping, where a temporal intensity modulation is imprinted on the chirped pulse. This happens when the laser pulse co-propagates with another pulse, e.g. a THz pulse from CSR (duration in the range of picoseconds), in an EO crystal. Due to the Pockels effect the THz pulse causes a time-dependent birefringence in the crystal. The Pockels effect describes the phenomenon of occurring and change of existing birefringence in an electric field, which is linearly proportional to the electric field strength. [?]

After that, the modulated chirped pulse propagates through another dispersive medium, a fiber of the length L_2 . In this way, the temporal modulation of the pulse is further stretched to the duration T_2 , which is long enough for detection with photodetectors and the digitizing with ADCs. [?]

The factor M , by which the pulse is slowed down, is calculated as

$$M = 1 + \frac{L_2}{L_1}. \quad (2.4)$$

As example, assume the length of the dispersive media as $L_1 = 10\text{ m}$ and $L_2 = 2\text{ km}$ and an input signal with the duration $t_{\text{sig}} = T_1 = 1\text{ ps}$. With Equation 2.4 the stretching factor for this set-up is $M \approx 200$. The input pulse is stretched to $T_2 = M \cdot T_1 = 200 \cdot 1\text{ ps} = 200\text{ ps} = 0.2\text{ ns}$. This corresponds to a frequency of 5 GHz which is much easier to handle e.g. for an oscilloscope.

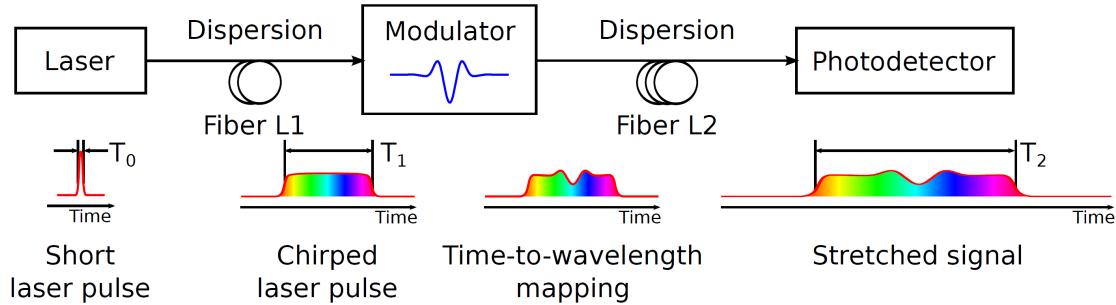


Figure 2.7.: Working principle of the electro-optical time-stretch technique [?]

Photodetector

In order to convert the time-stretched optical signal into an electrical value a photodetector, e.g. a photodiode, is needed. A basic diode photodetector is a photo-diode operated in reverse bias, meaning the p -side is connected to the negative terminal and the n -side to the positive terminal of the power supply. This enlarges the depletion region (see Figure 2.8) of the p/n -junction as the depletion region contains only a very small amount of free charge

carriers. Irradiating diode with photons of sufficient energy generates electron-hole pairs due to the photoelectric effect. If the electron-hole pairs are produced in the depleted region of the *p/n*-junction, they are separated by the electric field applied across it, before they can recombine. This creates a so called photo-current which can be measured. [?]

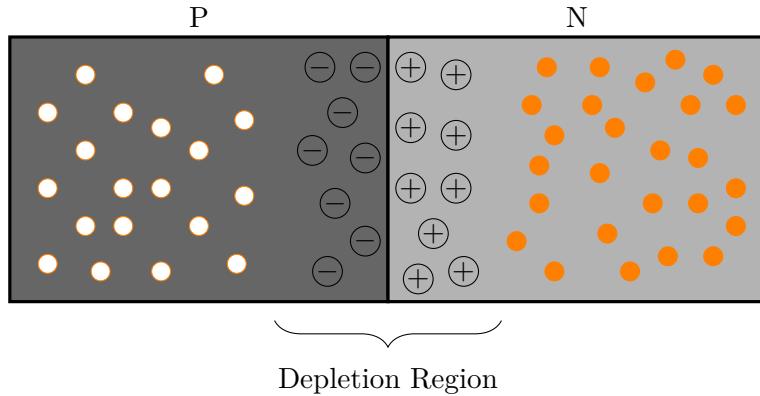


Figure 2.8.: *pn*-junction with depleted region [?]

2.3. Analog-To-Digital Converter

ADCs are used to translate analog signals, like voltages, into the digital representation of these signals. This *digitized* version can then be stored and processed by information processing, computing, data transmission and control systems. This translation, also called “conversion”, can be seen as encoding a continuous-time analog input V_{in} (voltage) into a series of discrete, N -bit words. This process is also called *sampling*. With the full-scale voltage of the V_{FS} , the individual output bits b_k and the quantization error ϵ , the ADC should satisfy the relation

$$V_{\text{in}} = V_{\text{FS}} \sum_{k=0}^{N-1} \frac{b_k}{2^{k+1}} + \epsilon. \quad (2.5)$$

This can also be rewritten in terms of the Least Significant Bit (LSB) or quantum level V_Q

$$1\text{LSB} = \frac{V_{\text{FS}}}{2^N} = V_Q. \quad (2.6)$$

With Equation 2.5 this leads to

$$V_{\text{in}} = V_Q \sum_{k=0}^{N-1} b_k 2^k + \epsilon. \quad (2.7)$$

Figure 2.9 shows the ideal transfer function of a 3-bit ADC. Each digital N -bit word corresponds to a range of input voltage values (*code width*), which is centered around a *code center*. The input voltage is resolved to the code of the nearest code center.

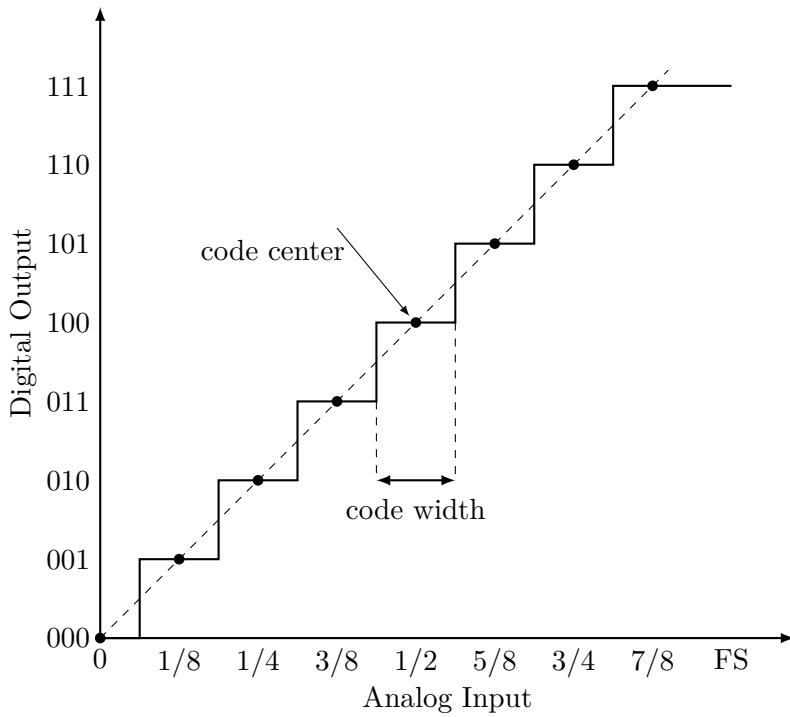


Figure 2.9.: Transfer function of an ideal, 3-bit ADC (redrawn from [?])

Sample-And-Hold-Amplifier

ADCs need a certain amount of time to sample the input signal. If the level of the analog signal changes by more than one LSB during this period, this can result in large errors in the output signal. Therefore so called Sample-And-Hold-Amplifier (SHA) are used in front of the ADC to hold the input level constant for the needed amount of time.

A general block diagram of a SHA is shown in Figure 2.11. It consists of an input and output buffer, a switch controlled by the sampling clock and a capacitor. The analog input is buffered in an input buffer which leads to a switch that is controlled by a sampling clock. During the sample mode, i.e. during the negative sampling clock cycle, the switch is open. At the transition from negative to positive clock cycle, the switch closes, connecting the input signal with the capacitor which is charged in this way.

The ADC sampling time needs to be timed in such way, that the whole duration of an analog-to-digital conversion falls into the hold period of the SHA and does not exceed into the sample period. Figure 2.10 shows a qualitative example for proper sample timing. As conclusion, the upper frequency limitation is not determined by the ADC itself, but rather by the aperture jitter, bandwidth, distortion, etc. of the SHA. [?]

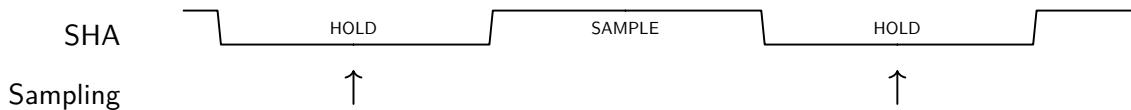


Figure 2.10.: Example for appropriate sampling timing when using Sample-And-Hold-Amplifier. The sample points of the ADC should be inside the period, where the SHA holds the input value.

Track-And-Hold Amplifier

Apart from the SHAs there also exists the so called Track-And-Hold-Amplifier (THA). Though the names are often used interchangeably, there exists one fundamental difference

between a SHA and a THA. Strictly speaking, the output of a SHA is not defined during the sample period. Only when switching to the hold mode, the output is assigned to a defined value: the voltage level at the input in that moment. Contrary to that, the THA acts as a unity gain amplifier during the sample period, meaning the output is just a replication of the input. The THA “tracks” the input signal (see also Figure 2.10). Therefore, instead of speaking of a “sample” period, the term used here is the “track” period. When switching to hold mode, the instantaneous input level is held over the course of the hold period. This principle allows to improve the sampling rate, as the settling time of an THA is in general smaller than one of a SHA. Settling time denotes the amount of time needed for the output voltage to be at a stable level, after the transition from track/sample to hold mode. This process is quicker, when the output voltage is already in the range of the sampled input at the moment, instead of when the hold capacitor first has to be charged to the input voltage. [?]

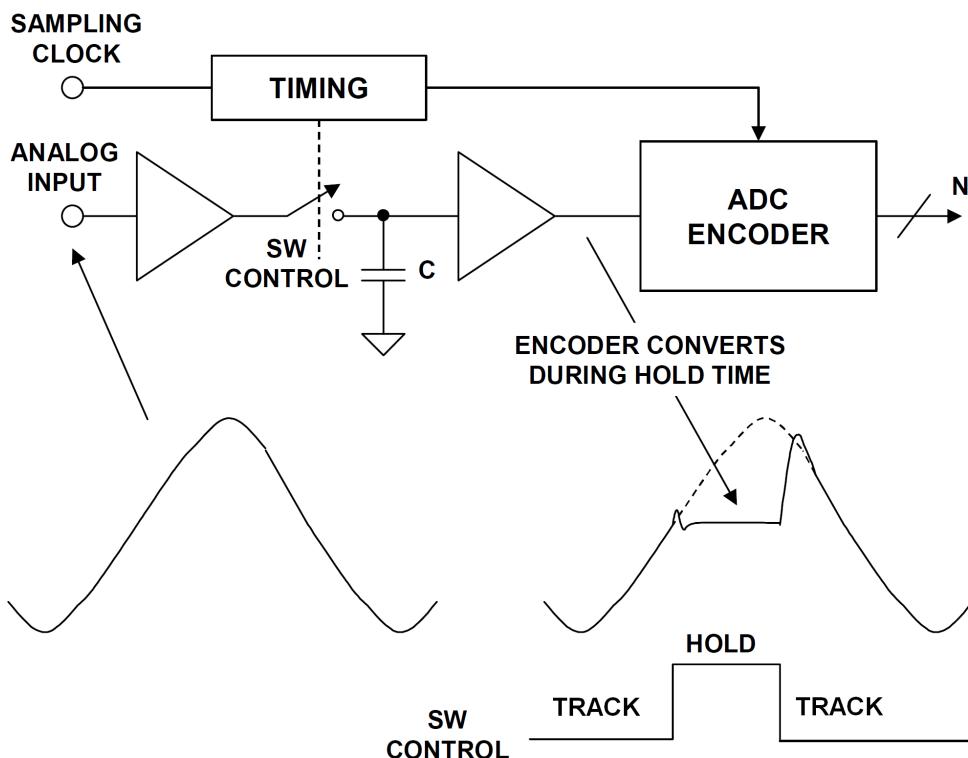


Figure 2.11.: Track-And-Hold-Amplifier schematic and principle [?]

2.3.1. Characteristics of Analog-To-Digital-Converters

For an ideal converter, the number of bits and the sampling rate would be sufficient to fully characterize its performance. Real ADCs however differ from the ideal behavior by introducing static and dynamic imperfections. Different applications have different requirements, which leads to a number of specifications. These can be divided into the categories according to [?]:

- Quantization Noise
- Static parameters
- Frequency-domain dynamic parameters
- Time-domain dynamic parameters

This section provides an overview of these figures of merit. Which of them are needed

to specify the necessary performance of the ADC has to be chosen for each application accordingly.

2.3.1.1. Quantization Noise

Even an ideal N -bit converter has errors resulting from the quantization process which behave like noise. The reason is that each N -bit word represents a certain range of analog input values, which is 1 LSB wide and centered around a code center (see Figure 2.9). [?] The input voltage is assigned to the word of the nearest code center. This means that there will always be a difference between the corresponding voltage of the respective digital code $x_q(t)$ and the actual analog input voltage $x(t)$. This difference is called the *quantization error*. For an equidistant quantization, the quantization error for a code width q is (see [?])

$$|e_q(t)| = |x(t) - x_q(t)| \leq \frac{q}{2}. \quad (2.8)$$

A setup in order to measure this quantization error is shown in Figure 2.12.

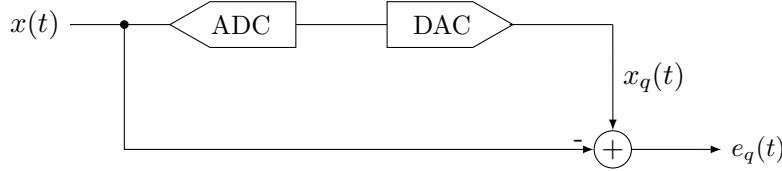


Figure 2.12.: Setup for measuring the quantization error of an (ideal) ADC with input signal $x(t)$

The output of the ADC, the N -bit code corresponding to the voltage level of the input signal $x(t)$, is fed to a Digital-To-Analog-Converter (DAC), which converts this code into a corresponding voltage level $x_q(t)$. The difference between $x(t)$ and $x_q(t)$ is the quantization error $e_q(t)$.

In order to analyze the quantization noise and the resulting theoretical (maximum) Signal-To-Noise-Ratio (SNR) of the ideal ADC, assume a ramp with the slope s as an input signal. Then, the quantization error $e_q(t)$ can be approximated with a sawtooth signal in the time domain (see [?]):

$$e_q(t) = st, \quad -\frac{q}{2s} < t < \frac{q}{2s} \quad (2.9)$$

The function in Equation 2.9 is plotted in Figure 2.13.

The power of this quantization noise can be calculated as the mean-square e_{rms}^2 of $e(t)$ (see [?]):

$$P_{QN} = e_{\text{rms}}^2 = \overline{e^2(t)} = \frac{s}{q} \int_{-q/2s}^{+q/2s} (st)^2 dt = \frac{s^3}{q} \left[\frac{t^3}{3} \right]_{-\frac{q}{2s}}^{+\frac{q}{2s}} = \frac{q^2}{12} \quad (2.10)$$

In order to calculate the maximal SNR of an ideal converter, a full-scale input sine wave is applied to the input:

$$u(t) = u_s \sin(2\pi ft) = \frac{2^N q}{2} \sin(2\pi ft) = 2^{N-1} q \sin(2\pi ft) \quad (2.11)$$

With the effective value of the signal amplitude

$$u_{\text{eff}} = \frac{u_s}{\sqrt{2}} = \frac{2^{N-1} q}{\sqrt{2}} \quad (2.12)$$

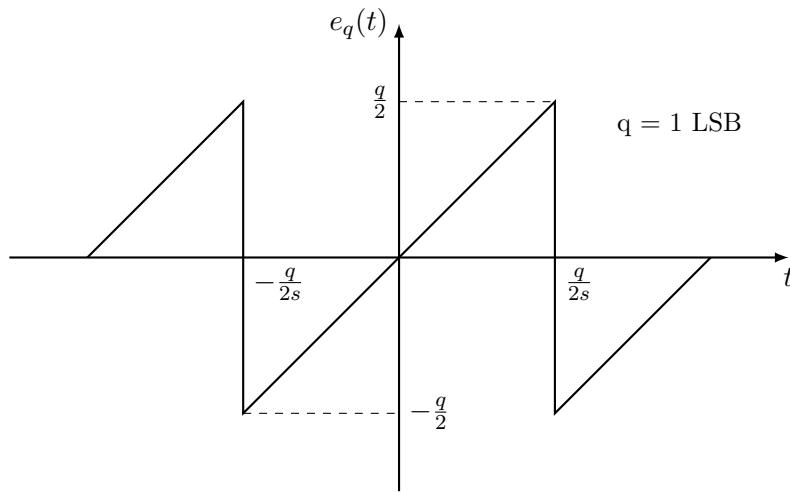


Figure 2.13.: Quantization noise as function of time (redrawn from [?])

the SNR can be calculated as

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}} = \frac{u_{\text{eff}}^2}{e_{\text{rms}}^2} = \frac{2^{2N-2}q^2/2}{q^2/12} = 2^{2N} \cdot 1.5. \quad (2.13)$$

In decibel, the SNR is calculated as (see [?, ?]):

$$\text{SNR|}_{\text{dB}} = 10 \log (2^{2N} \cdot 1.5) = 6.02N + 1.76 \quad (2.14)$$

2.3.1.2. Static parameters

Static parameters are specifications, which can be measured at low speed/DC.

Accuracy

Accuracy is the total error with which an ADC can convert a known voltage, which includes the effects of (see [?]):

- Quantization error
- Gain error
- Offset error
- Non-linearities

Resolution

Resolution is the number of bits N of the ADC. Depending from the resolution are the size of the LSB, which in its turn determines the dynamic range, code widths and quantization error.

Dynamic Range

The *dynamic range* represents the ratio between smallest possible output (LSB voltage) and the largest possible output (full-scale voltage). It can be calculated as

$$20 \log 2^N \approx 6N. \quad (2.15)$$

Offset and Gain Error

The *offset error* is defined as the deviation of the actual ADC transfer function from the ideal ADC transfer function in the point of zero. It is measured in LSB.

Gain Error defines the deviation of the slope of the line going through the zero and full-scale point of the transfer function. Figure 2.14 visualizes the effects of both offset and gain error.

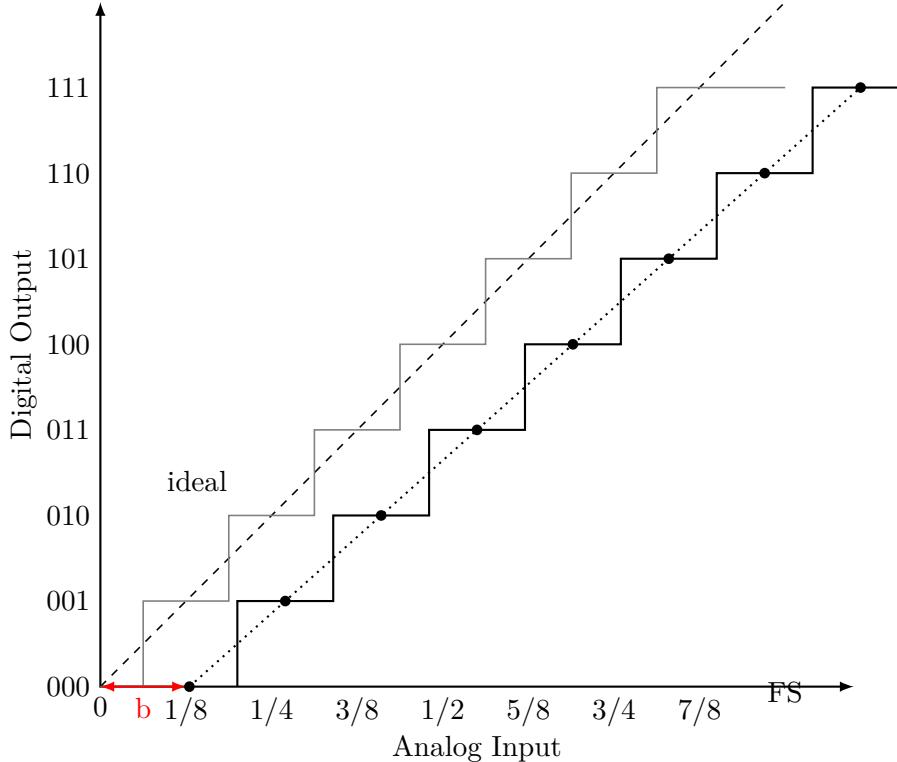


Figure 2.14.: Offset and Gain Error in the ADC characteristic transfer function. The offset error is indicated with the red arrow. The gain error expresses itself via different slope of the real ADC (dotted) compared to the ideal ADC (dashed)

These errors can easily be corrected by calibration. In order to measure the offset and gain error, two different voltage levels V_1 and V_2 are applied at the ADC input. This results in corresponding bit codes b_1 and b_2 . The slope s of the transfer function can then be calculated by

$$s = \frac{b_2 - b_1}{V_2 - V_1}. \quad (2.16)$$

From this, the gain error can be determined. In order to obtain the offset error b , the linear equation

$$b = b_1 - s \cdot V_1 \quad (2.17)$$

is solved.

Integral and Differential Non-Linearity Distortion

Integral Nonlinearity (INL) is the distance of the code centers on the actual ADC transfer function from the ideal line (dashed line in Figure 2.15). It results from the integral non-linearities of the front-end, SHA and also the ADC itself [?, ?].

Differential Nonlinearity (DNL) is the deviation in actual code width from the ideal width of 1 LSB. This non-linearity stems exclusively from the encoding process in the ADC [?, ?]

.

The effect of these errors is shown in Figure 2.15.

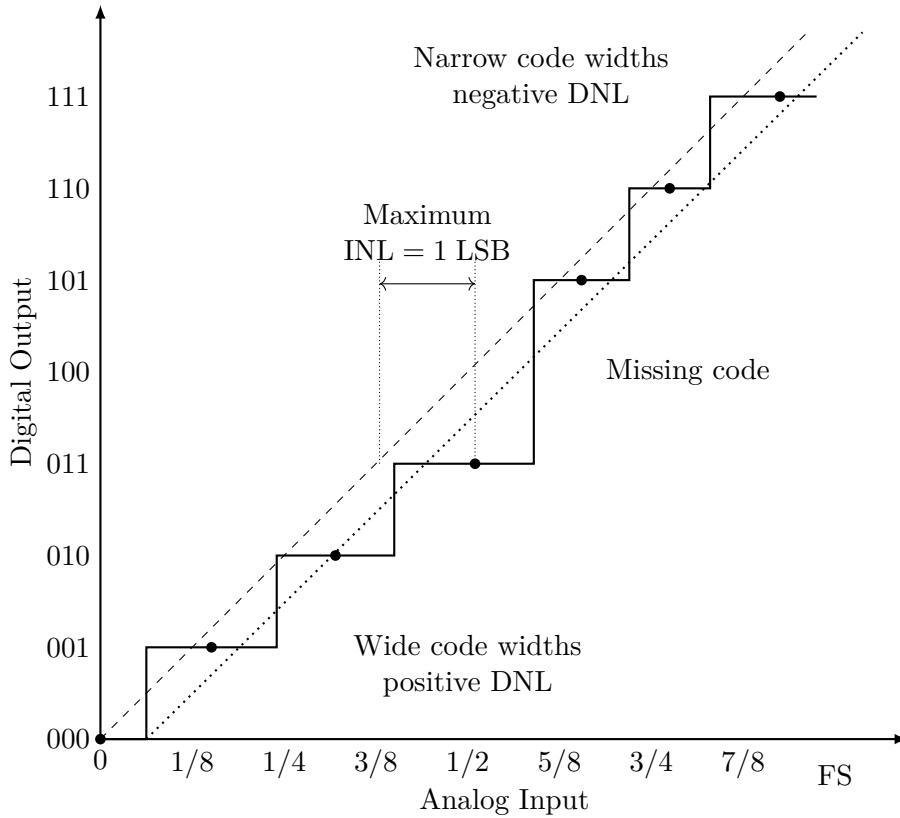


Figure 2.15.: Transfer function of a real ADC showing DNL and INL.[?]

These non-linearities could be measured with a histogram test. A voltage ramp is applied at the input and the number of occurrences of each ADC output code, $n(\text{code})$, is measured. With the ramp slope s an ideal ADC with the sampling frequency f_s would give

$$n(\text{code}) = \frac{\text{LSB}}{s} \cdot f_s = n_{\text{avg}}, \quad (2.18)$$

which ideally would be constant for the whole input range (except for the first and last code). For a real ADC this is not the case and the DNL and INL are calculated as (see [?])

$$\text{DNL}(\text{code}) = \frac{n(\text{code}) - n_{\text{avg}}}{n_{\text{avg}}} \quad (2.19)$$

$$\text{INL}(\text{code}) = \sum_{i=0}^{\text{code}} \text{DNL}(i). \quad (2.20)$$

2.3.1.3. Frequency-Domain Dynamic Parameters

Any real ADC is subject to noise distortion. *Noise* denotes any unwanted random signal, which interferes with the measuring of the desired signal. Examples are quantization noise or random fluctuations due to thermal noise.

Distortion is the term for alteration of the shape of the original signal. As an example, distortion of the amplitude might result due to not equal amplification of the parts of a signal. [?]

In an ADC (with built-in SHA) there are a couple of sources, which introduce noise and distortion:

- **Input Stage:** Wideband noise, non-linearity and bandwidth limitation
- **SHA:** Non-linearity, aperture jitter (see paragraph about Time-Domain Dynamic Performances) and bandwidth limitation
- **ADC:** Quantization noise, non-linearity

For quantification of noise and distortion, frequency-domain metrics are used. Therefore the figures of merit described in the following paragraphs are also called frequency-domain dynamic parameters. These parameters are measured with the help of the Fast-Fourier-Transform (FFT) meaning any modern oscilloscope can be used to quickly assess the frequency-domain dynamic performance for a given input at the ADC. As some parameters, such as Spurious-Free Dynamic Range (SFDR), are only defined for one carrier input frequency, several measurements at different input frequencies need to be made in order to fully characterize the ADC.

In the following paragraphs, an overview of the metrics for quantification of the noise and distortion of an ADC is given.

Signal-to-Noise Ratio

The SNR is defined as the ratio of the input signal power to the power of the noise signal. It is expressed in dB and can be calculated using the Root Mean Square (RMS) value of the signal and noise amplitudes (see [?]):

$$\text{SNR} = \frac{\text{Power}_{\text{Signal}}}{\text{Power}_{\text{Noise}}} \quad (2.21)$$

$$= \left(\frac{\text{Amplitude}_{\text{Signal,rms}}}{\text{Amplitude}_{\text{Noise,rms}}} \right)^2 \quad (2.22)$$

$$= 20 \log \left(\frac{V_{\text{in,rms}}}{V_{\text{Q,rms}}} \right) \quad (2.23)$$

Usually, the SNR degrades at higher frequencies due to sampling jitter. [?]

Signal-to-Noise-and-Distortion Ratio

Signal-to-Noise-and-Distortion Ratio (SINAD) (also called SNDR or S/N+D) denotes the ratio between the RMS of the signal amplitude to the mean value of the Root-Sum-Square (RSS) of all other spectral components, including harmonics, but excluding Direct Current (DC) (0Hz). SINAD is a good indication over the general dynamic performance of the ADC, as it includes all contributions from noise and distortion. The higher the SINAD the stronger the input power is differentiated from noise and spurious components.

SINAD can be calculated from the average power of the input signal P_{signal} , noise P_{noise} and $P_{\text{distortion}}$:

$$\text{SINAD} = 10 \log \left(\frac{P_{\text{signal}}}{P_{\text{noise}} + P_{\text{Distortion}}} \right) \quad (2.24)$$

It is commonly expressed in dB, decibels relative to the carrier (dBc) or decibels relative to full scale (dBFS).

Effective-Number-Of-Bits

The Effective Number Of Bits (ENOB) expresses the SINAD in terms of bits. It can be calculated as

$$\text{ENOB} = \frac{\text{SINAD} - 1.76 \text{ dB}}{6.02 \text{ dB/bit}}. \quad [?] \quad (2.25)$$

This is derived from solving the equation of the “ideal SNR” (Equation 2.14) for the number of bits N and substituting SNR with SINAD. This however means, that this parameter assumes a full-scale input signal. Expressing the ENOB for a smaller signal amplitude requires measuring the SINAD at this level and a correction factor. [?]

Spurious-Free Dynamic Range

SFDR indicates the dynamic range of the converter, which can be used, before there is interference or distortion from spurious components with the fundamental signal. [?] The SFDR is calculated as the RMS value of the fundamental signal to the RMS value of the worst spurious signal, i.e. the highest spur in the spectrum. It is measured over the whole Nyquist bandwidth from DC to $f_s/2$, with f_s being the ADC sampling rate. The spur may or may not be a harmonic of the fundamental signal. [?, ?]

The SFDR is an important characteristic in the sense, that it indicates the smallest signal which can still be distinguished from a strong interfering signal. [?]

The SFDR in dBc can be calculated as (see [?])

$$\text{SFDR}_{\text{dBc}} = 20 \log \left(\frac{\text{Fundamental Amplitude (RMS)}}{\text{Largest Spur Amplitude (RMS)}} \right). \quad (2.26)$$

Figure 2.16 illustrates the SFDR in terms of dBFS and dBc.

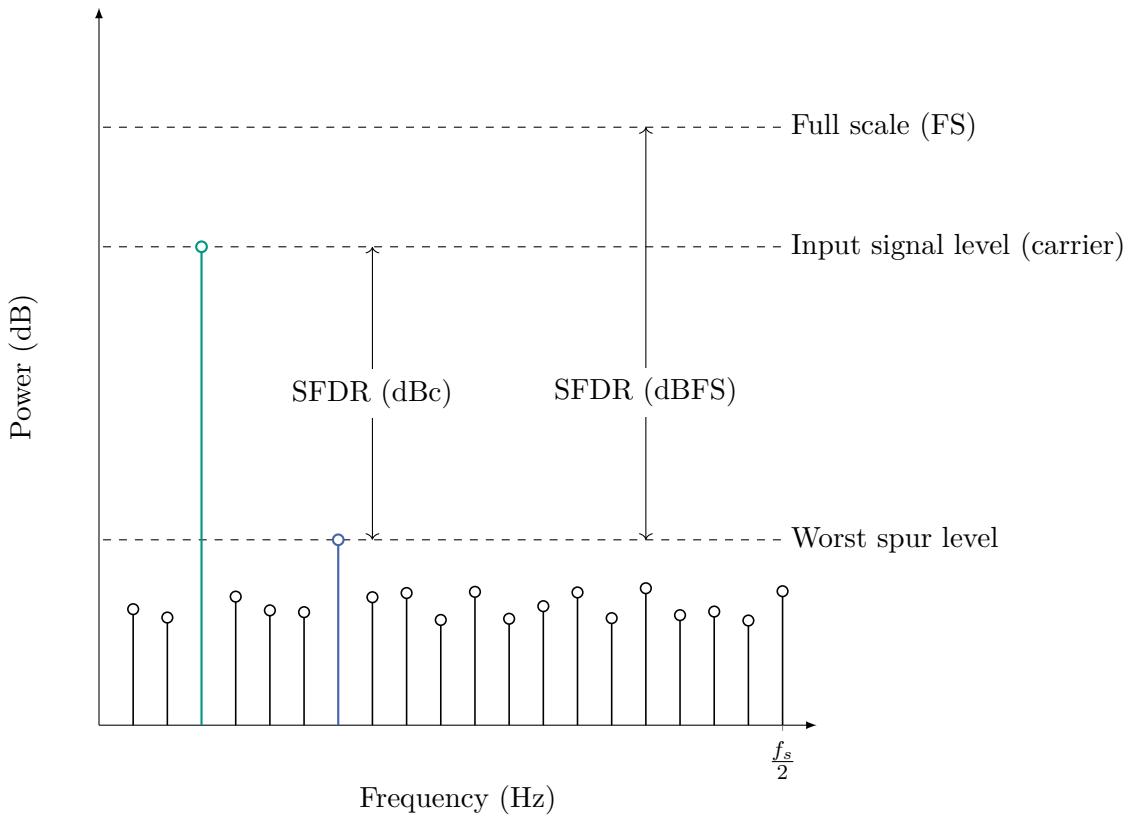


Figure 2.16.: Visualization of the SFDR. It can be indicated either with reference to the carrier frequency in “dBc” or with reference to the Full-Scale Input in “dBFS”. [?]

Total Harmonic Distortion

The *Total Harmonic Distortion* describes the ratio of the RMS sum of the first five harmonic components (or aliased versions of them) to the RMS of the considered fundamental signal. [?]

Effective Resolution Bandwidth

Effective Resolution Bandwidth denotes the frequency of the input signal, at which the SINAD has fallen by 3dB ($\cong 0.5$ bit in terms of ENOB) compared to the SINAD at lower frequency range. [?]

Analog Input Bandwidth

Analog Input Bandwidth is the analog input frequency at which the power of the fundamental is reduced by 3dB with respect to the low-frequency value. [?] It is not to be confused with the maximal analog input frequency which the ADC is able to sample.

Full-Linear Bandwidth

The *Full-Linear Bandwidth* is defined as the frequency at which the SR of the SHA starts to distort the input signal by a specified value. [?] The SR is defined as the rate of how much the voltage v changes over time t :

$$\text{SR} = \frac{dv}{dt} \quad (2.27)$$

A SR of 1 V/ μ s for example means, that the output of the amplifier can not change more than 1 V over the course of 1 μ s. [?]

Time-Domain Dynamic Parameters

Time-Domain Dynamic parameters describe the deviation of the converter's behavior from the ideal one in time domain.

Aperture Delay

Aperture Delay (or *aperture time*) is defined as delay between the triggering of the converter (e.g. rising edge of the sampling clock) and the actual conversion of the input voltage into the digitized value. [?]

Aperture Jitter

Aperture jitter describes the sample-to-sample variation in aperture delay. Jitter can cause significant error in the voltage and decreases the overall SNR of a converter. Especially for high-speed ADCs jitter poses a limit in performance.

Assuming a full-scale sinus-wave V_{in} as input signal with

$$V_{\text{in}} = V_{\text{FS}} \sin(\omega t) \quad (2.28)$$

the maximal slope of this signal is then

$$\left. \frac{dV_{\text{in}}}{dt} \right|_{\max} = \omega V_{\text{FS}} \quad (2.29)$$

Aperture jitter Δt_{rms} occurring during the sampling of this maximal slope produces the RMS voltage error

$$\Delta V_{\text{rms}} = \omega V_{\text{FS}} \Delta t_{\text{rms}} = 2\pi f V_{\text{FS}} \Delta t_{\text{rms}}. \quad (2.30)$$

As variations in aperture time occur randomly, these errors behave like a random noise source. This way, a Signal-to-Jitter-Noise-Ratio (SJNR) can be defined as

$$\text{SJNR} = 20 \log \left(\frac{V_{\text{FS}}}{\Delta V_{\text{rms}}} \right) = 20 \log \left(\frac{1}{2\pi f V_{\text{FS}}} \right) \quad (2.31)$$

The voltage error due to jitter and the SJNR for different aperture jitter values are shown in Figure 2.17.

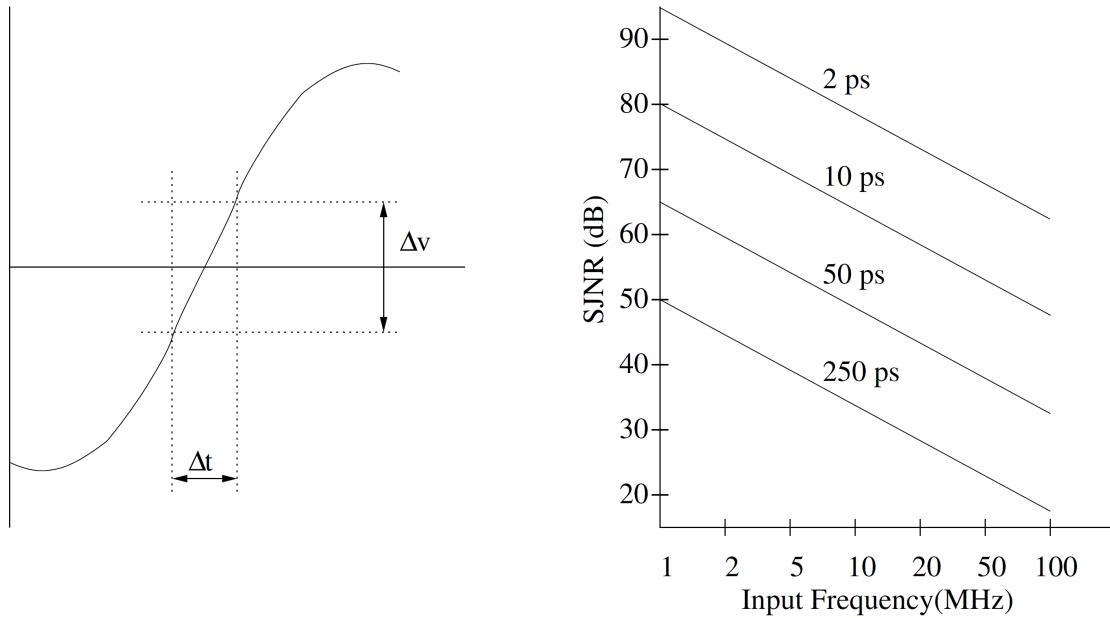


Figure 2.17.: Effects of aperture jitter and SJNR. Left: In time domain, Right: SJNR for different aperture jitter [?]

Transient Response

The *transient response* denotes the settling time of an ADC until full accuracy ($\pm 1/2$ LSB).

2.3.1.4. Sampling Theory

An ADC samples an analog signal with a sample frequency f_s . This frequency has to be chosen in such way, that the original signal can be fully reconstructed. The *Nyquist criteria* states, that in order to accurately reconstruct a band-limited, continuous signal

$$y(t) \circ= Y(f) \quad \text{with} \quad Y(f) = 0|_{f>B/2} \quad (2.32)$$

it has to be sampled with a frequency f_s respecting

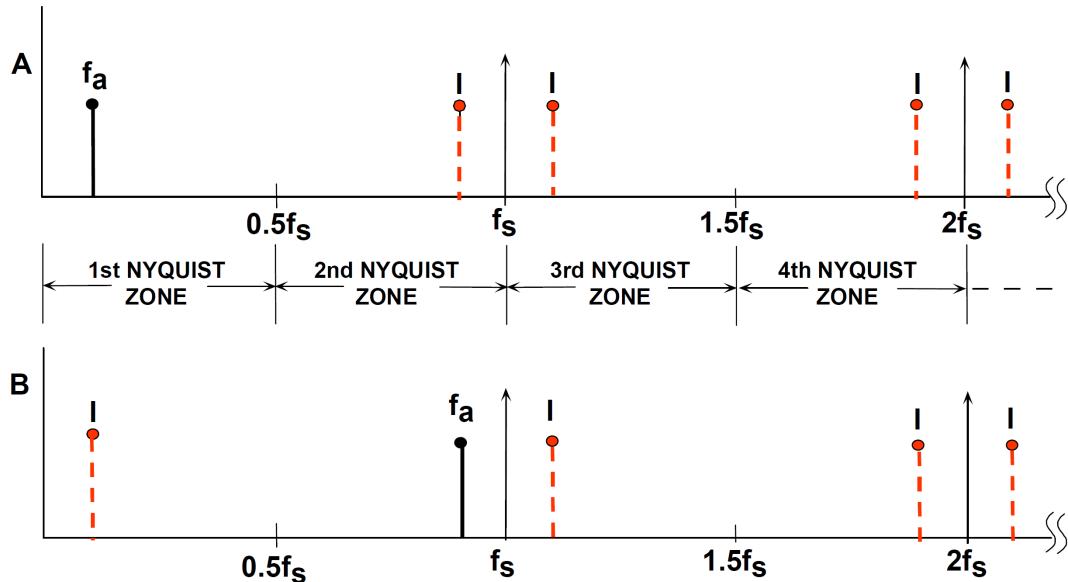
$$f_s > B \quad \text{or} \quad f_s > 2f_a, \quad (2.33)$$

with f_a being the highest frequency contained in the signal. [?, ?] The range from 0 Hz to $f_s/2$ is also called *Nyquist-Zone* (or “1st Nyquist zone”, see Figure 2.18a).

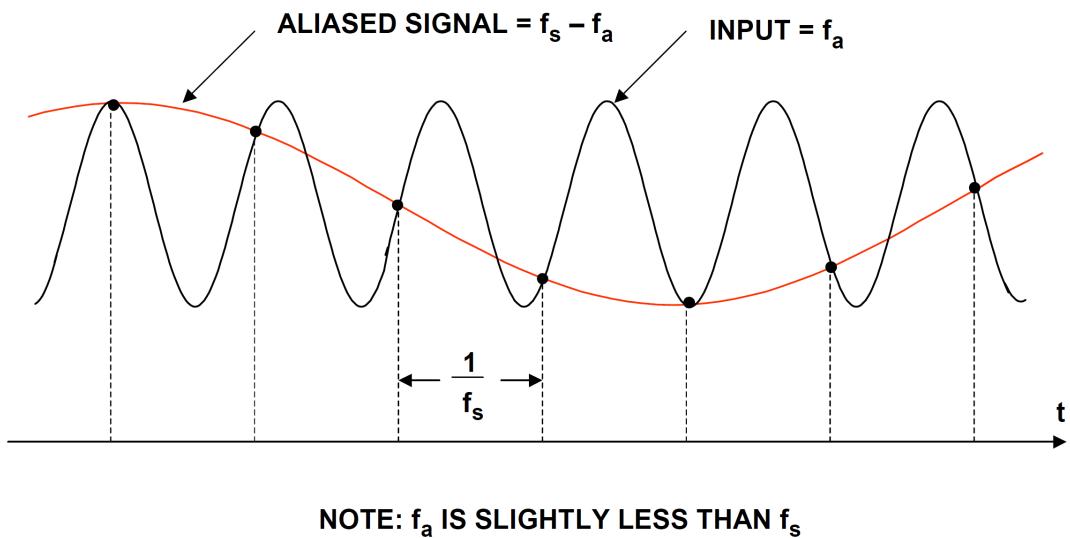
Violation of this rule leads to *aliasing*. The effects of aliasing are shown in Figure 2.18.

When a sine wave of the frequency f_a is sampled with the frequency f_s , this leads to periodic repetition of the signal spectrum in frequency domain in intervals of f_s , or “images” (see dashed, red frequency components in Figure 2.18a). If Equation 2.33 is respected, i.e. f_a lies inside the Nyquist bandwidth, there is no overlap with the images created by the sampling process.

Now assuming a signal frequency $f_a \approx f_s$, the sampling process leads to an image falling inside the Nyquist bandwidth. The reconstructed signal then lies at the frequency of this image which is much lower than the original frequency. The result of this *undersampling* is shown in Figure 2.18b



(a) Sampling process visualized in frequency domain



(b) Effect of aliasing shown in time domain

Figure 2.18.: Analog signal with frequency f_a sampled at f_s respecting (A) and not respecting (B) the Nyquist criteria (see Figure 2.18a). Figure 2.18b shows the effect of case B in time domain. [?]

3. Architecture Of The New Readout-System - THERESA

This section is dedicated to describing the general concept of the new readout-system. The system was given the name THERESA and in the sections to follow this name will be used to denote the new system.

First, a short overview of state of the art systems is given, including commercially available real-time oscilloscopes and the Karlsruhe Pulse Taking Ultra-fast Readout Electronics (KAPTURE). This system was developed at KIT (Institute for Data Processing and Electronics (IPE)) specifically addressing the needs of THz diagnostics at KARA. The working principle of this system is explained in detail, as the new THERESA system is an evolution of the KAPTURE system.

Then, the architecture of the THERESA system itself is described.

3.1. State Of The Art Readout-Systems

Real-Time Oscilloscopes

Real-time oscilloscopes are defined by three key banner specifications: bandwidth, sample rate, and memory depth. Some examples of currently commercially available oscilloscopes are listed in Table 3.1. The acquisition time is given for the case of maximal sample rate. As can be derived from the table, the acquisition time of such oscilloscopes is quite limited, not allowing for continuous sampling of fast input signals.

Table 3.1.: Some example real-time oscilloscopes with (max.) key characteristics

Model	Bandwidth	Sample Rate	Memory Depth	Acquisition time
Keysight MXR608A	6 GHz	16 GS/s	1.6 GS	10 ms
Tektronix DPO70000SX	70 GHz	200 GS/s	1 GS	5 ms
LeCroy LabMaster 10-100Zi	65 GHz	160 GS/s	512 MS	3.2 ms

3.1.1. KAPTURE

KAPTURE (Karlsruhe Pulse Taking Ultra-Fast Readout Electronics) is a fast readout system developed at the IPE for THz diagnostics at KARA. It is designed to digitize the pulses generated by THz detectors at each electron bunch revolution, with a memory-efficient approach to acquire the detector signal on a bunch-by-bunch basis (sampling only

the pulses themselves). The system is able to sample pulses with a Full Width At Half Maximum (FWHM) between a few tens to a hundred picoseconds with a minimal sample time of 3 ps [?].

To showcase the revolution of this DAQ system, the general architecture and concept is explained with the first version of KAPTURE. Then, the improved version KAPTURE-2 is presented. At the end, being a further evolution of these two versions, the architecture of THERESA is explained.

General Concept

The system consists of two parts: the sampling front-end card and a FPGA readout card. In Figure 3.1 the setup for THz radiation measurements with KAPTURE is shown.

The incoming radiation is fed into a detector, which converts the incident photons into an electrical signal. This signal is then amplified in a wide-band Low-Noise-Amplifier (LNA). A wideband lossless power splitter, developed at IPE, splits the detector signal into four identical signals, which are then propagated to the sampling front-end card. The card consists of four parallel sampling channels with adjustable sampling time. Each channel contains a THA and an ADC. This card is connected to a read-out card by a high-speed and high-density connector. The FPGA sets the sampling time for each individual sampling channel and reads, processes and sends all acquired data to a CPU/Graphics Processing Unit (GPU) cluster for further processing [?].

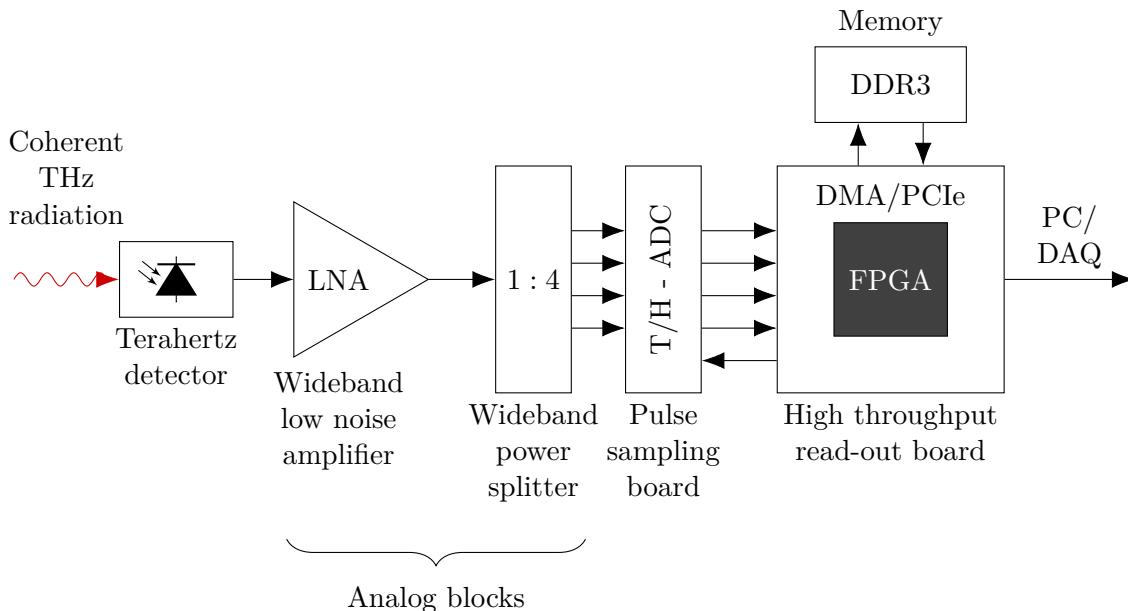


Figure 3.1.: THz radiation measurement setup with KAPTURE (v1) (redrawn from[?])

Analog Front-End

Due to the high bandwidth nature of the detector signal, the analog front-end of the system has to be wideband as well to be able to sample the signal with picosecond resolution.

The used LNA is based on a commercial GaAs Microwave Monolithic Integrated Circuit (MMIC) which operates from DC to 50 GHz. It is needed to compensate the insertion loss¹ of the following power splitter stage. Classical power-splitters are not intrinsically wideband ([?]). For that reason, an wideband power-splitter was developed at IPE which

¹Insertion loss is the loss of signal power which occurs, when a signal passes through a component.

fulfills the bandwidth requirements. The designed power-splitter works up to 100 GHz with an insertion loss of 8 dB (at 100 GHz) and a return loss² of about 20 dB at 50 GHz [?]. A photo of the power splitter is shown in Figure 3.2.

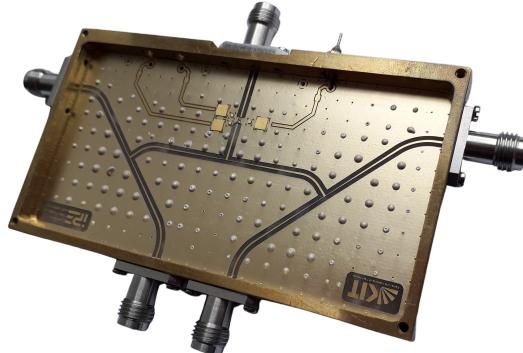


Figure 3.2.: Photo of the power splitter developed at IPE

Sampling Board

The architecture of the front-end board with the power splitter is shown in Figure 3.3.

The power splitter splits the incoming signal into four identical signals, which are then fed into four parallel channels, consisting of a respective THA unit and a 12-bit ADC sampling at 500 MS/s. The sampling time of each unit can be adjusted individually with a delay chip with a resolution of 3 ps (maximal delay range: 100 ps). The delay chips are programmed with the FPGA on the readout card. The clock signal is provided by KARA, which is cleared from jitter by a Phase-Locked-Loop (PLL). This ensures the synchronization of the ADCs with the RF system. The cleaned clock signal is distributed to the delay chips via fan-out buffer [?]. In this way, the pulse can be "locally sampled" by adjusting the different delay with a maximum rate of 330 GS/s possible. A simplified representation of the local sampling of the signal is shown in Figure 3.4.

GPU-DAQ System

The sampling system produces a large amount of data. In order to keep a continuous data acquisition the necessary bandwidth is

$$12\text{bits} \cdot 8 \text{ samples} \cdot 1 \text{ GHz} = 96 \text{Gb/s} \quad (3.1)$$

To ensure high data throughput, a high-speed PCI Express (PCIe) readout card was developed (called "High-Flex") was developed. This card receives the samples and tags them with the respective bunch identification. The data is then sent to a GPU using a PCIe connection based on direct FPGA-GPU direct memory access architecture. The GPU node reconstructs the pulse based on the given sampling points and calculates the amplitude and pulse arrival time. It also performs an online FFT for frequency analysis. To store the data temporary before it is sent to the DAQ system, a large Double Data Rate (DDR)3 memory device is used, as seen in Figure 3.3 [?].

²*Return loss* is the loss of signal power due to reflection by a discontinuity in the transmission line.

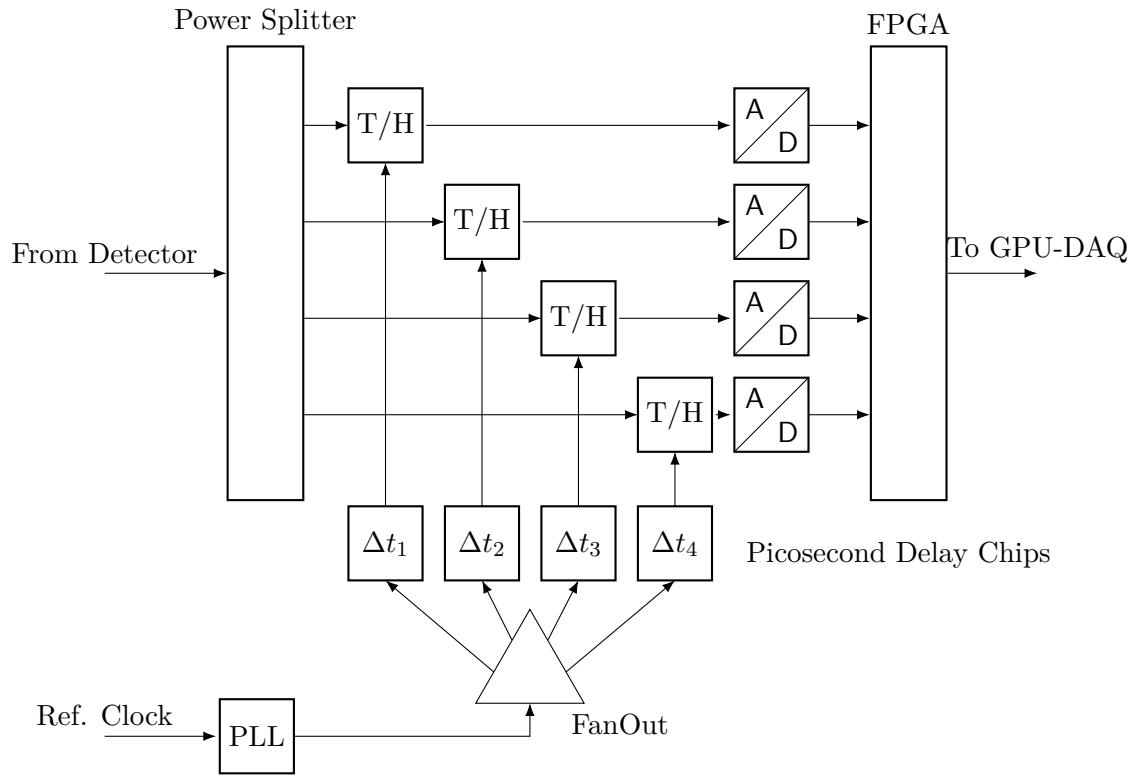


Figure 3.3.: General architecture of the KAPTURE (v1) front-end sampling card (cf. [?, p.2])

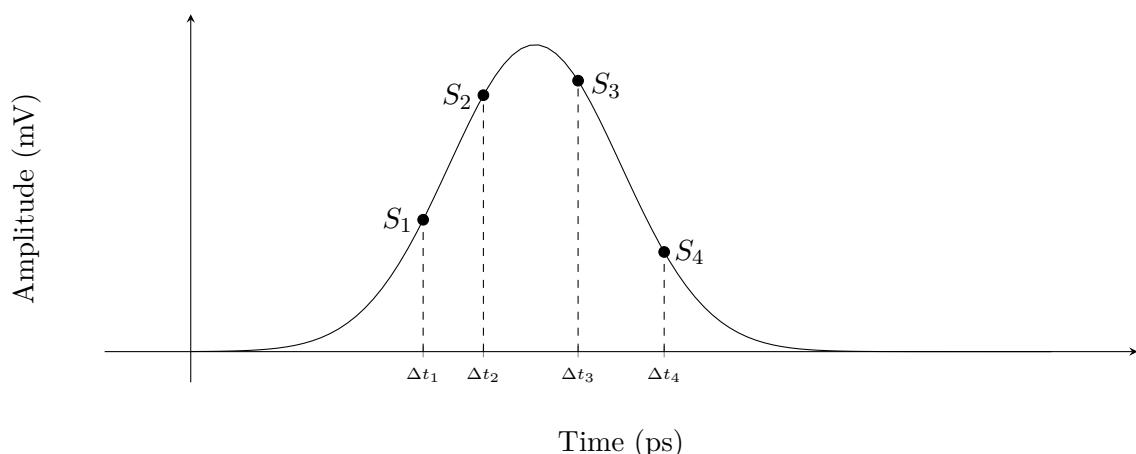


Figure 3.4.: Signal and sampled points S_1 to S_4

3.1.2. KAPTURE-2

The first version of KAPTURE has a limitation concerning the number of sampling points per pulse and does not allow to sample the baseline of the detector. Analyzing the baseline however is very important, as it is changing slightly and affects the pulse amplitude of the bunch. Due to this distortion, calculating the correlation between bunches was limited. For this reason, a second version of KAPTURE was designed in order to overcome these limitations. The PLL on the sampling board allows for synchronization between two or more PLLs located on different boards. With this feature, the sampling time of two boards can be synchronized and in this way extend the number of sampling points beyond four. A comparison of the sampling concepts is shown in Figure 3.5.

In KAPTURE-2, two front-end boards can be connected to directly sample the pulses with up to eight sampling point at the pulse repetition rate 2 GHz. Alternatively, the system can sample the pulse and the baseline between two consecutive pulses with a constant pulse rate up to 1 GHz (see Figure 3.5b). In this way, the read-out card can calculate the correct amplitude of the pulse and send it to the GPU for further processing [?].

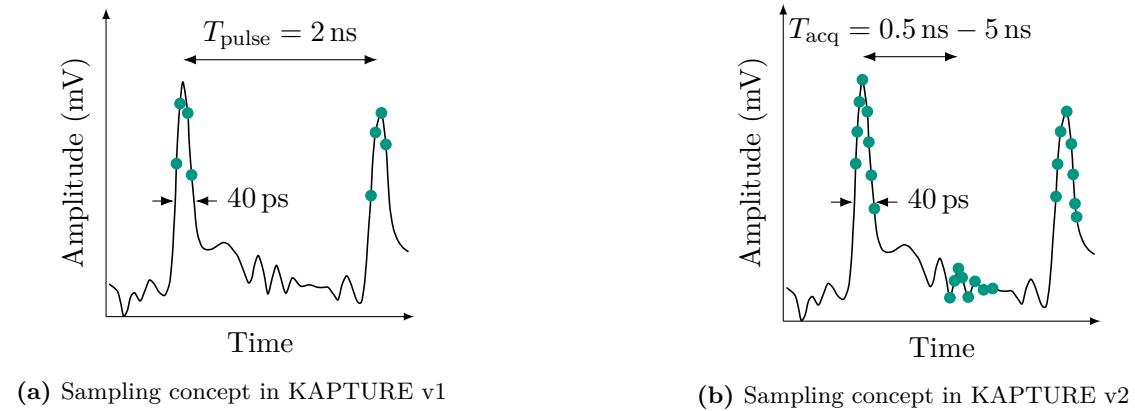


Figure 3.5.: Comparison between the sampling concepts of KAPTURE v1 and KAPTURE v2

Figure 3.6 shows a photo of the system setup of KAPTURE-2.



Figure 3.6.: Photo of the KAPTURE-2 setup

3.2. Proposed Architecture for THERESA

In this section the architecture for the THERESA system is described. The system consists of the optical time-stretch setup, which stretches the analog input signal and the photodetector in order to convert the optical signal into an electrical one. This signal is sampled by a front-end sampling card, which is mounted on a back-end readout card, which processes the acquired samples.

Optical Part

For the optical time-stretch setup, a femtosecond Ytterbium-doped fiber laser from *MENLO GmbH* is used. The emitted pulses have a bandwidth of 50 nm and a total output average power of 40 mW. The photodetector used is an InGaAs photodiode from *Discovery Semiconductors* with a 20 GHz bandwidth.

3.2.1. Front-End Sampling Card

The concept of the front-end sampling card is based on and an evolution of the concept used in the KAPTURE system.

The incoming signal is split into 16 identical signals, each leading to the respective sampling channel on the sampling board. These sampling channels consist of a high bandwidth (18 GHz), low noise THA. The sampling clock to these THAs is provided by respective programmable delay chips. In this way, a time interleaving technique (described below) can be implemented by programming the delay chips accordingly. The main clock is provided by a main PLL, which cleans the incoming reference clock from the system in which the system is integrated. Figure 3.7 shows the general schema of the sampling system, reduced to four channels for presentation purposes.

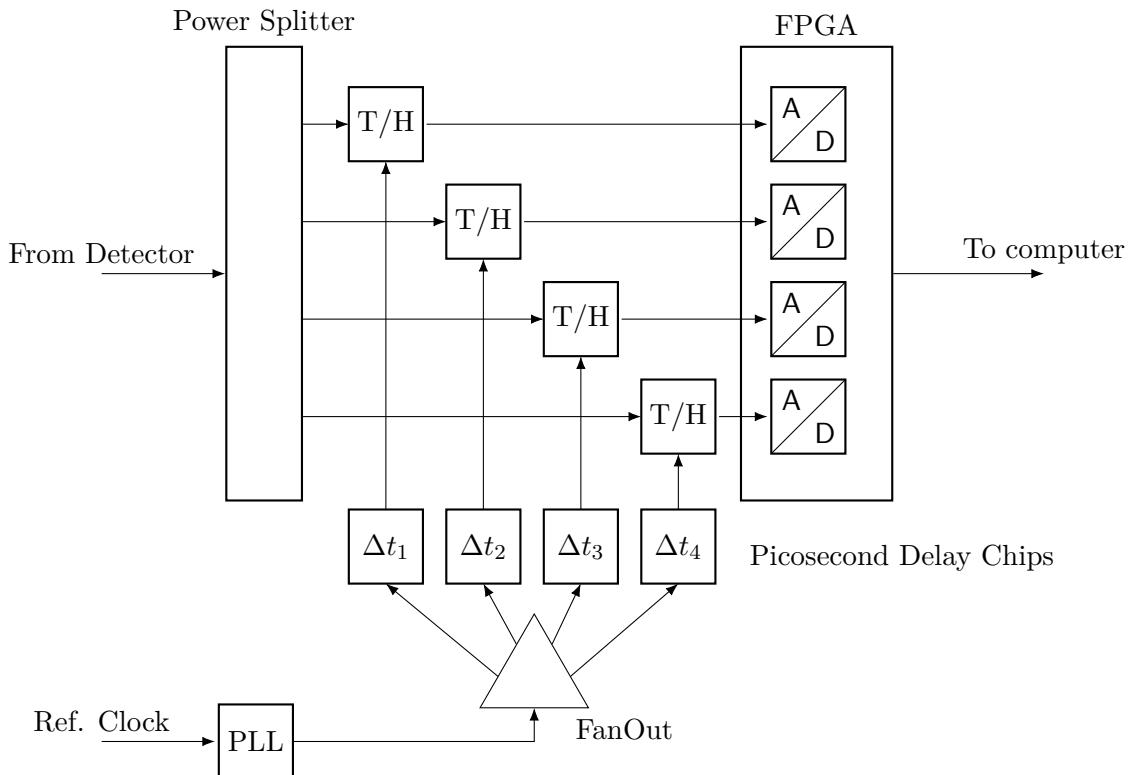


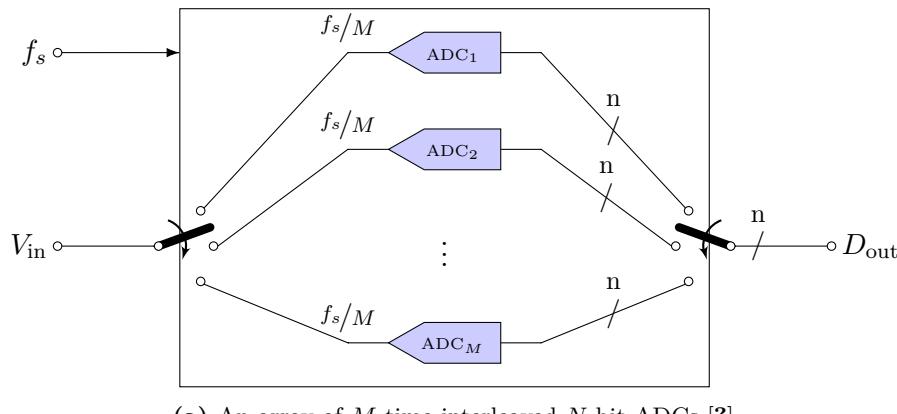
Figure 3.7.: General architecture of the THERESA sampling card with power splitter and ADCs. For presentation purposes only four of the sixteen channels are shown.

3.2.1.1. Time Interleaving

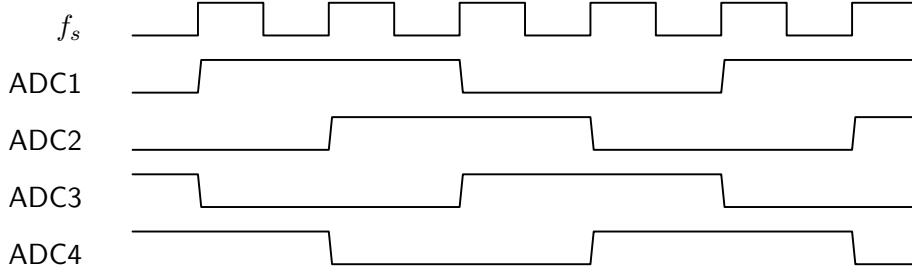
In order to increase the sampling rate, the so called time-interleaving technique is used. In this section, first basic theory about this technique is given. Then, the implementation in the new system is described.

Theory

In the *Time Interleaving* technique multiple ADCs are used in such way, that allows to sample data at a faster rate than the respective sample rate of each individual ADC. The principle is based on time-multiplexing an array of M identical ADCs (see Figure 3.8a), each operating at a sampling rate of $f_c = f_s/M$ individually. The sampling times of the ADCs are shifted in phase as shown in Figure 3.8b with the example of 4 time-interleaved ADCs. At time t_0 the first ADC starts converting the input signal $V_i(t_0)$, after a defined time delay Δt_i the second ADC samples and converts $V_i(t_0 + t_i)$, the third converts $V_i(t_0 + 2t_i)$ and so on. After the M -th ADC has sampled the signal $V_i(t_0 + (M-1)t_i)$, the whole cycle starts anew with the first ADC [?]. An example for such a cycle for 4 ADCs is shown in Figure 3.8b.



(a) An array of M time interleaved N -bit ADCs [?]



(b) Clocking Scheme for interleaving 4 ADCs

Figure 3.8.: Array of M time interleaved ADCs and clocking example for $M = 4$

Challenges

Spurs appear in the spectrum. There are several reasons for this which are described in the following.

First reason is the *offset mismatch* between den ADCs. Each ADC is characterized by a DC offset. Considering as example an interleaving structure with two ADCs and a constant input voltage: when the samples are acquired back and forth between the two ADCs, the resulting output will switch back and forth between two levels due to the different offset levels of the ADCs. This output switches at the frequency $f_s/2$. Therefore this introduces

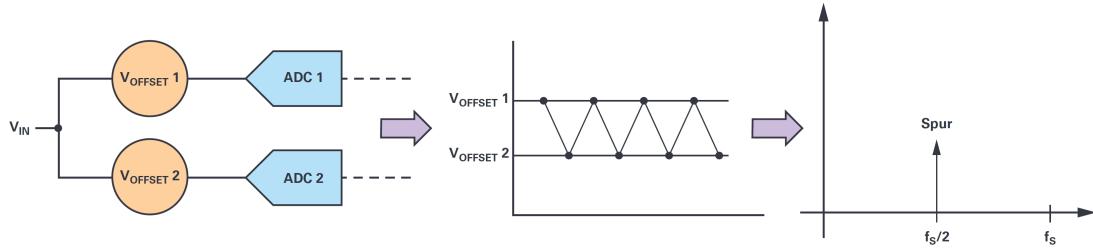


Figure 3.9.: Offset-Mismatch in Interleaving [?]

spurious harmonic components at the frequency $f_s/2$ in the spectrum (see Figure 3.9). The magnitude of the spur depends on the offset difference between the ADCs [?].

Besides of the offset also the gain of the converters can be mismatched. This *gain mismatch* has a frequency component to it, which in case of an input signal of the frequency f_{in} results in a spur at $f_s/2 \pm f_{in}$ (see Figure 3.10) [?].

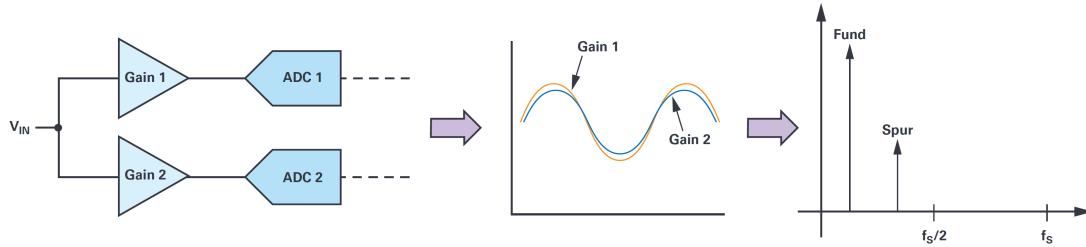


Figure 3.10.: Gain-Mismatch in Interleaving [?]

In the time domain, *timing mismatch* due to group delay in the analog circuitry of the ADC and clock skew³ can occur. The group delay in analog circuitry can vary between the converters. The clock skew has on the one hand an aperture uncertainty component at each of the ADCs. On the other hand it has a component related to the accuracy of the clock phases, which are input to each converter. [?] This mismatch also produces a spurious component at $f_s/2 \pm f_{in}$ (see Figure 3.11).

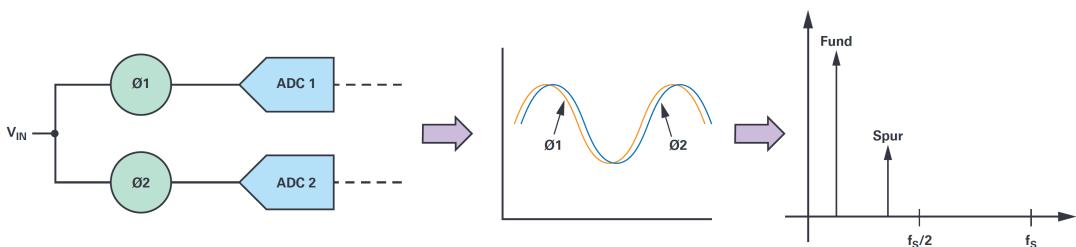


Figure 3.11.: Timing-Mismatch in Interleaving [?]

The last possible mismatch is the *bandwidth mismatch*, which contains both gain and phase/frequency component (see Figure 3.12). Due to bandwidth mismatch, different gain values at different frequencies can be seen. An additional timing component causes different delays for signals at different frequencies through each ADC. Just like gain and timing mismatch, the bandwidth mismatch causes a spur at $f_s/2 \pm f_{in}$.

Due to the presented mismatches, a proper characterization of the ADCs. The characterization is required in order to account for all systematical errors in the ADCs and to reduce

³Difference in arrival time of the clock signal at different components.

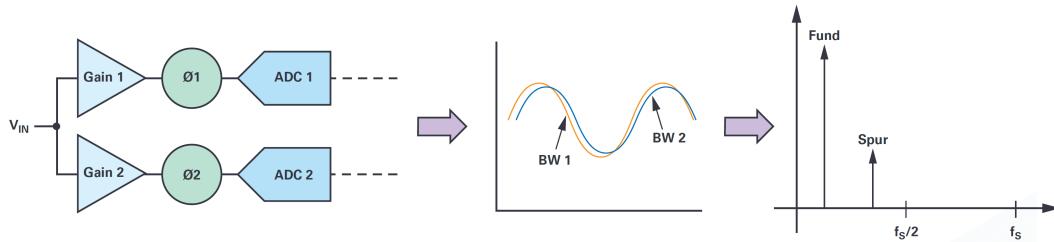


Figure 3.12.: Timing-Mismatch in Interleaving [?]

the spurious components in the spectrum. For this purpose, a circuit on the THERESA sampling board is foreseen, in order to provide the possibility to generate test signals from the readout card.

3.2.1.2. Implementation

On the selected readout card for THERESA, 16 ADCs with a sampling rate up to 2.5 GHz are provided. In order to implement the time-interleaving method, an appropriate delay step size for the sample time has to be calculated. To calculate the maximal step size possible can be calculated as follows: The ADCs on the read-out card sample at a maximal sample rate of 2.5 GS/s, meaning during the time

$$t_s = \frac{1}{2.5 \text{ GS/s}} = 400 \text{ ps} \quad (3.2)$$

all 16 ADCs have to sample the signal one time. This means, a delay step can not be greater than $400 \text{ ps}/16 = 25 \text{ ps}$. With this method, the maximal achievable sampling rate of the card is $16 \cdot 2.5 \text{ GS/s} = 40 \text{ GS/s}$.

On the selected readout card, sampling clock signals are not propagated individually to the respective ADCs. The converters are grouped together into tiles, each tile containing four converters. One single reference clock signal is propagated to all tiles. To implement the optimal time-interleaving method with this card, four individual sampling clocks to all tiles shifted by 90° would be necessary. Analyzing the schematic of the readout board revealed however, that only two individual sampling clocks can be provided to the card. Therefore, another approach needs to be considered. Figure 3.13 shows qualitatively the concept. The main 1 GHz clock is propagated to the THAs, which are in hold-mode when the clock signal is HIGH and in track-mode when the clock signal is LOW. As shown in Figure 3.13 the clock signal to each THA is provided with a respective delay. The maximal delay step size to cover the whole period of the clock is calculated by:

$$\frac{1 \text{ ns}}{16 \text{ channels}} = 62.5 \text{ ps} \quad (3.3)$$

In some way, this implementation can therefore also be regarded as time-interleaving, as each THA holds a different sample point in time, which can then be converted by the ADCs. The two sampling clocks, indicated with “ADC1” and “ADC2”, need to be phase-shifted by 180° . In this way, an alternate clocking of the ADCs is made possible⁴.

⁴As can be derived from the diagram, only the four respective ADC channels should be considered for signal conversion during one sampling point.

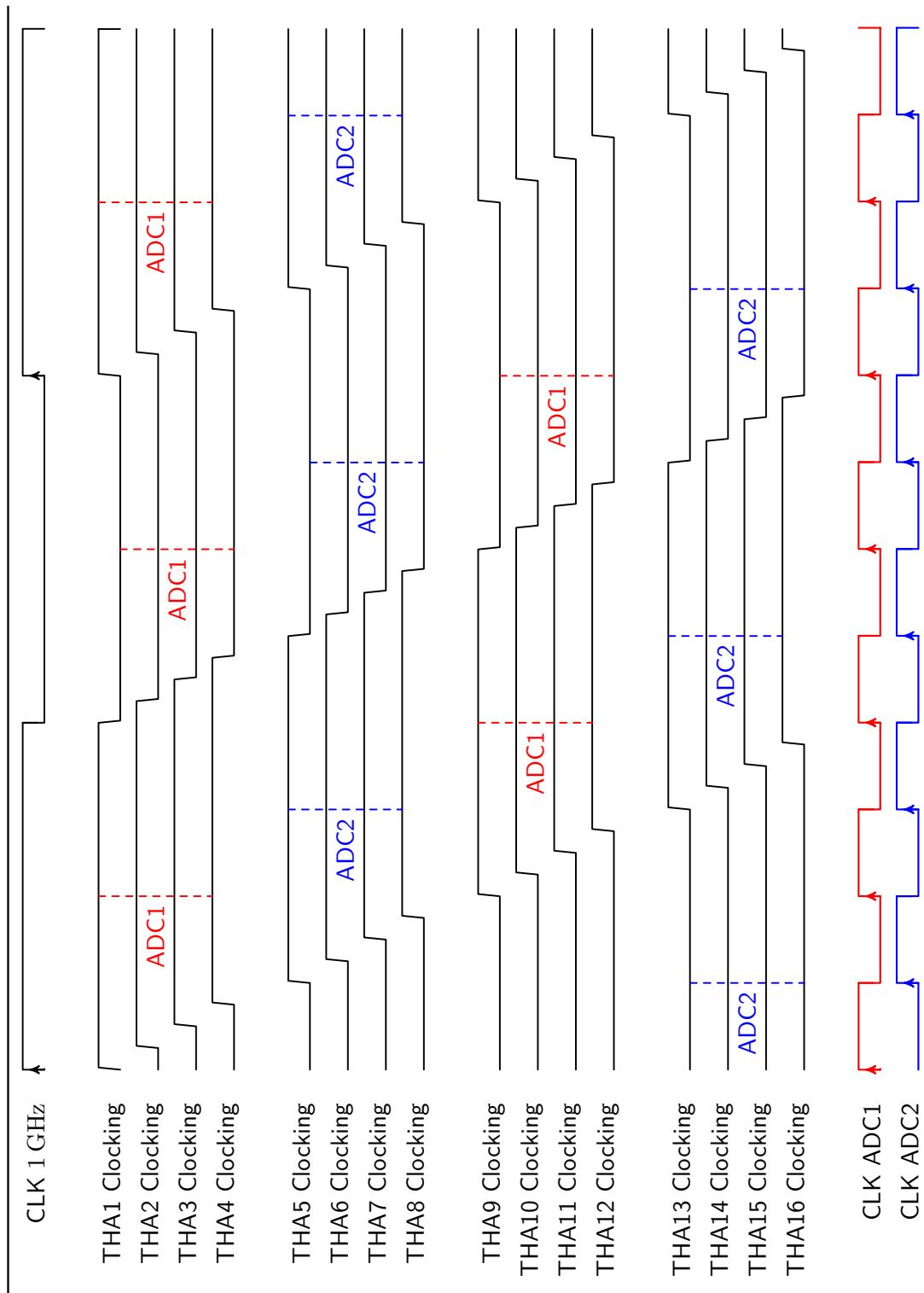


Figure 3.13.: THA Timing diagram. Shows the clocking of the THA (HIGH = hold mode, LOW = track mode). Dashed line represents the sampling of the ADC.

3.2.2. Readout Card

The most important points to consider when choosing the readout card is its capability to handle high data-throughput, provide the possibility for user-defined firmware and control of the system. This flexibility is provided by FPGA-based System-On-Chips (SoCs), which also integrate the required high-speed peripheral connections for data transfer. An important point for THERESA is also to integrate the ADCs inside the SoC. The reason for this is illustrated in Figure 3.14. In order to fulfill the requirements, the system would need a processing unit, an FPGA and a number of data converters (ADC/DAC). Realizing this in discrete components results in a higher footprint, than integrating every component inside one Integrated Circuit (IC).

Integration of the necessary components inside an IC also drastically reduces the complexity of the sampling board. Implementing the data converters in a discrete way would result in a high number of interfaces/connections, especially for a high ADC resolution, making expensive high pin count connectors necessary. Integrating the converters inside the SoC therefore resolves these challenges.

The currently only commercially available system, meeting the mentioned requirements, is the Xilinx ZU49DR Zynq Ultrascale+ RFSoC. This SoC integrates 16 high-speed data converters (ADCs and DACs), Arm processor cores and a programmable logic (FPGA). An evaluation card, containing all necessary peripherals (optical interfaces, USB-interface, ...) and integrating the Radio-Frequency System-On-Chip (RFSoC), was chosen for the implementation of the THERESA system. The card is described in detail in chapter 5.

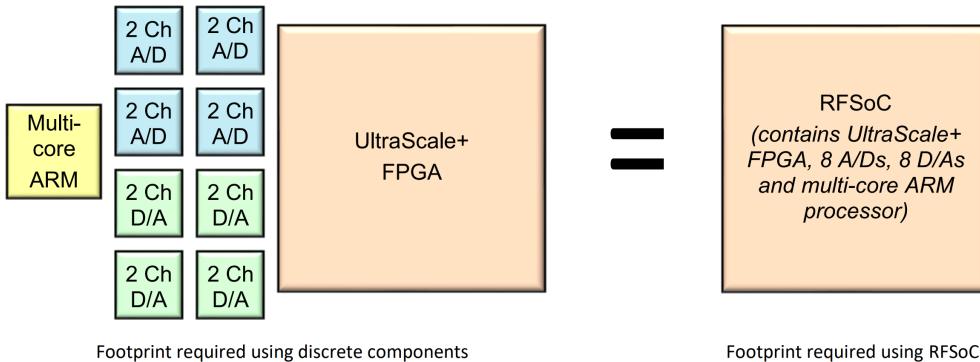


Figure 3.14.: Footprint of discrete components vs. footprint of IC integrating the components

4. Design Of The Front-End Sampling Card

In this chapter, the process of designing the front-end sampling card is described. Designing a Printed Circuit Board (PCB) is a two step process: circuit design and layout design. In this thesis, the software used to cover both of these steps is PADS xDx Designer (for schematic capture) and PADS Layout/Router (for PCB layout design) from Mentor Graphics (subsidiary of Siemens).

4.1. Schematics

Without knowing which components are needed and how they are interconnected, it is impossible to manufacture any board, no matter how high or low the level of complexity is. The schematic is a graphical documentation of an electrical circuit, showing the needed components and their interconnections using standardized symbols. Furthermore, a schematic provides a starting point for automatic placement and routing, i.e. where the components are placed and how they are connected on the physical PCB, which is done with the layout design tool. During the creation of the schematics, the following points have to be considered:

- Deciding which components are needed and what the performance requirements are. Especially for high-speed components carefully considering specifications like signal rise and fall times, jitter, skew, etc. is crucial to achieve the overall expected performance.
- Keeping in mind how many pins are available for high- and low-speed peripheral connections, control signals, etc. Many components have an interface for programming (e.g. Serial Peripheral Interface (SPI)) which requires several pins that need to be connected to the controlling unit. Especially for boards with a lot of components this can quickly become an issue.
- Checking the signaling interfaces of the components. Additional circuitry might be needed for interfacing between two different components. Some signaling interfaces, like Low Voltage Differential Signaling (LVDS), require a specific voltage level, which might result in the need of voltage level translators.
- Keep in mind the different common mode voltages at input/output pins of different components and placing decoupling capacitors if needed.
- Consider placing additional filtering for power supplies in order to reduce noise and PCB, as well as recommended filters from manufacturers of the components.
- Choose suitable type and amount of power supplies/voltage regulators.
- Keep in mind the packaging/Size of the components. The size of the component is important, as space on the board is limited. The package introduces additional capacitive/inductive parasitics, which can be a problem for precise filtering circuits.

- Consider the power dissipation of the components. Components like for example voltage regulators might need coolers or heat sinks. These additional elements might not pose any problems for components which are located on the top side of the board. However, components on the bottom side might create a space issue, if the designed PCB should be mounted on another board.
- For mixed-signal boards, i.e. boards containing digital and analog signal paths, analog and digital ground should be separated. For ICs like THAs or ADCs, where both analog and digital signals are present, connecting the grounds via appropriate components needs to be considered.
- Check if the components are still available and if they can be delivered in the given project time.

This list is certainly not complete, but provides an overview over the most important points which need to be taken into account during design. Decoupling techniques and separation of analog and digital ground are explained a bit more detailed, being very important and crucial steps for design of high-performance PCB.

Decoupling techniques

Probably the most important part in schematics design is proper decoupling of power supplies, as ICs require a stable voltage on the power supply pins for optimal performance. Any ripple¹ or noise can substantially degrade the performance of the ICs, i.e. by decreasing the noise margin. *Noise margin* defines the difference between the useful signal and noise. A sufficient noise margin is necessary to guarantee that the output signal will still be correctly interpreted, even if some noise is added to the signal. Variation on the power supply produces also a variation on the signal and can therefore lead to a smaller difference between signal and noise.

Usually, manufacturers give information about proper power supply decoupling circuits for their component in the data sheet. If this is not the case, there are basic rules of thumb which can be followed to ensure proper decoupling. [?]

Basically, two types of voltage variations on the power supply pin can be distinguished: low frequency and high frequency variation. Low frequency variation occurs for example due to devices (or parts of them) being enabled/disabled or in the event of data traffic or data processing. The current draw during these occurrences can not be compensated immediately by the voltage regulator providing the supply voltage, which leads to drops in the voltage level. Time frames of this variation vary in the range of milliseconds up to days. High frequency variation results from switching events in the device, occurring in the range of the clock frequency and the corresponding harmonics up to about 5 GHz. Spikes due to Electro-Magnetic Interference (EMI) are also a source of high frequency variation and need to be compensated for. [?]

Ideally, one capacitor, which acts as a low-pass filter, should be enough to mitigate these variations. A real capacitor however has parasitics and thus can in general not be modeled by a “pure” capacitive behavior. This reduces the filtering performance. Additional resistances and inductance need to be considered [?]:

- A parallel resistance R_P , which shunts the nominal capacitance (C), representing insulation resistance or leakage.
- A series resistance R_S , or Equivalent-Series-Resistance (ESR), which represents the plates and the leads of the capacitor.

¹ Ripple is additional Alternating Current (AC)-voltage (of small amplitude) superimposed on a the general voltage level.

- A series inductance L_S , or Equivalent-Series-Inductance (ESL), that models the inductance of the plates and leads of the capacitor.
- A parallel resistance and capacitance, R_D and C_D , which model the effect called dielectric absorption. This denotes the phenomenon, that a capacitor which has been charged for a long time, doesn't fully discharge when briefly discharged. Dielectric absorption can be detrimental for high-precision use-cases, for power supply decoupling this effect doesn't have to be considered.

Consideration of all these effects leads to the equivalent circuit shown in Figure 4.1. It can be seen that this forms a RLC circuit, meaning the capacitor will not have the ideal behavior over the whole frequency range. In fact, a real capacitor shows an impedance response as seen in Figure 4.2, which resembles one of a band stop, rather than a low pass. Typical capacitive behavior is seen in region (I). Region (II) shows the influence of the ESR, which is why there is a residual impedance at the lowest point. Region (III) showcases the effect of the ESL. To extend the capacitive behavior over a wider frequency range, at least two capacitors are placed.

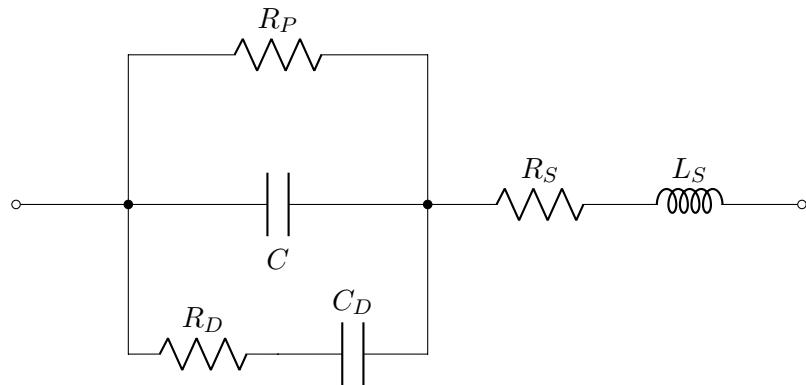


Figure 4.1.: Equivalent circuit of a real capacitance (redrawn from [?])

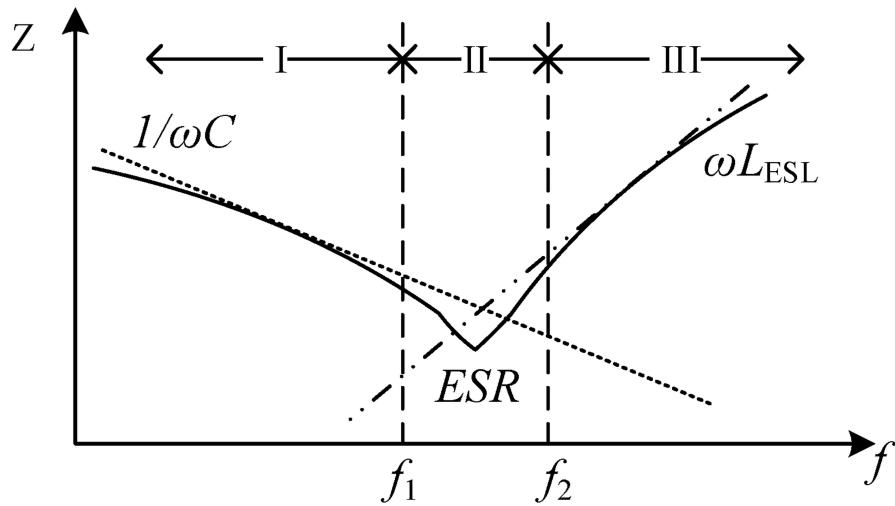


Figure 4.2.: Qualitative impedance response of a real capacitance [?]

To deal with the low frequency variation, a large capacitor (typical values: $10\ \mu\text{F}$ to $100\ \mu\text{F}$) is placed next to the component, not more than 5 cm away. The role of this capacitor is to be a charge supply for the instantaneous needs of the device, i.e. keeping a constant voltage level until the slower control loop of the voltage regulator can compensate for the changed current draw. [?] This capacitor is also called *decoupling capacitor*.

Another, small capacitor (typical values: $0.01 \mu\text{F}$ to $0.1 \mu\text{F}$) is placed as close as possible to the power pins of the component. This capacitor should bypass (therefore also called *bypass capacitor*) the high frequency variation on the power supply line. [?]

To cover a larger frequency range, multiple capacitors can be used.

All capacitors should be connected through vias or short traces to a large area, low impedance ground plane. Vias on a PCB are used to connect different layers, a plane is an uninterrupted area of metal covering the whole (or part) of a PCB layer (basic PCB structures are also explained in section 4.2). Connecting capacitors in this way minimizes the inductance due to connection traces. [?]

An optional ferrite bead in series with the supply pin keeps external high frequency from the device and the noise generated inside the component from the rest of the board. [?]

4.1.1. Connectors

The number and type of connectors is primarily defined by the read-out card, on which the sampling board is mounted. The different connector types serve different purposes, which can be organized into three categories.

Digital Control Signals

For digital control (i.e. SPI, enable signals, ...) and clocking signals a VITA 57.4 FMC+ connector from *SAMTEC* is used (see Figure 4.3).

FPGA Mezzanine Card (FMC) is a standard defined by VMEbus International Trade Association (VITA) to provide a standard mezzanine card² form factor, connectors and modular interface to a FPGA located on a base board (carrier card). [?] The FMC+ standard extends the pin count and throughput of the present high-speed interfaces. An assembly drawing of the FMC+ connector is shown in Figure 4.3.

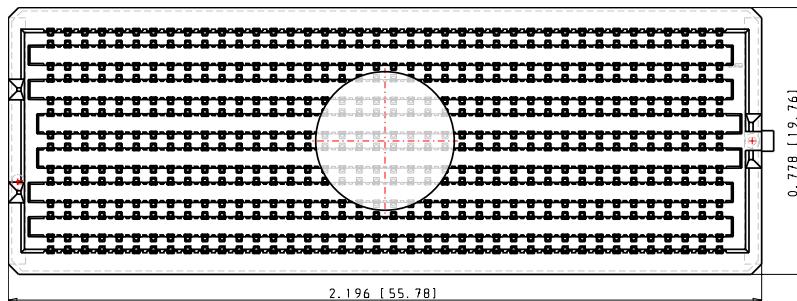


Figure 4.3.: Part drawing of FMC+ connector [?]

The FMC+ connector provides 560 pins arranged in a 14×40 array, 80 of which are additional high-speed interfaces, located on either side of the connector (therefore this connector type is also called High Serial Pin Count Extension (HSPCe) connector, as opposed to the HSPC connector which doesn't have additional rows). For user-defined purpose 160 pins are available. They can be used as single-ended or differential pins. Clocking capable pins can be used to propagate clock signals from the mezzanine to the carrier board.

Furthermore, the connector provides pins for power supply from carrier board to mezzanine card. [?] The voltage levels provided are listed in Table 4.1.

²A PCB which is plugged on a plug-in board. [?]

Table 4.1.: Voltage levels provided by the FMC+

Voltage	Max. current	Max. capacitive load
V_{ADJ} , 0 V to 3.3 V	4 A	1000 μ F
3.3 V	3 A	1000 μ F
12 V	1 A	1000 μ F

Analog Signals

The analog signals coming from the power splitter are propagated to the THAs through 1.85 mm high-frequency connectors. These connectors use an air dielectric filled interface which enables operation up to 65 GHz. This type of connector is also called “V connector”; due to its frequency range it is considered as a mm-wave RF connector. It is therefore used in precision instrumentation and other laboratory applications. The design has been introduced as an open standard under the Institute of Electrical and Electronics Engineers (IEEE) 287 Precision Connector Standards Committee.

?? shows a V connector in male and female type.

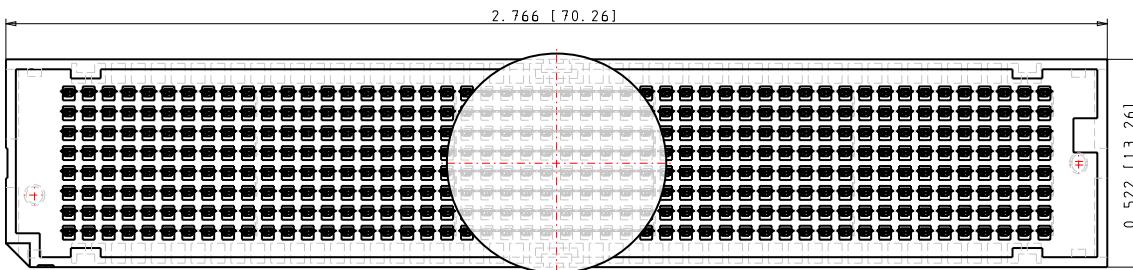


(a) V connector, male type

(b) V connector, female type

Figure 4.4.: Male and female type V connector

On the read-out board two RFMC 2.0 (RF Mezzanine Card) interface connectors are provided. The connectors used are Low Profile Array, Female (LPAF) connectors from SAMTEC with 400 pins arranged in a 8×50 array. One connector is dedicated for transmitting signals from the mezzanine card to the on-board ADCs. The other provides the analog output from the on-board DACs³ to the mezzanine card. On the sampling board, the male counterpart of the connectors, Low Profile Array, Male (LPAM), is used (see Figure 4.5).

**Figure 4.5.:** Part drawing of a LPAM 8×50 connector

³A DAC translates digital values into an analog signal.

Clock Signals

The clock signals from the PLLs on the sampling board are propagated in different ways. The reference clock for the FPGA is propagated through the FMC+ connector. Clocking for the ADCs and the DACs is provided through a 6×20 LPAM connector (see Figure 4.6).

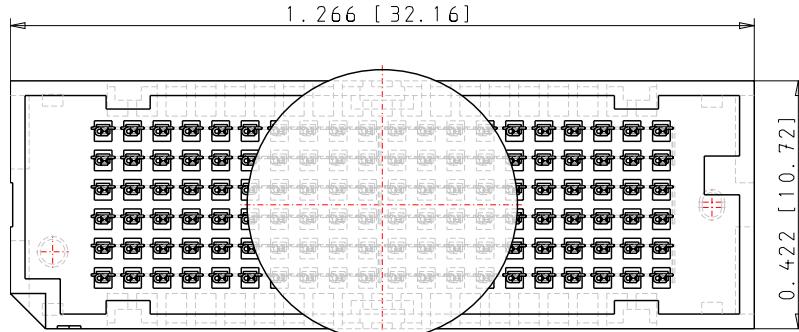


Figure 4.6.: Part drawing of LPAM 6×20 connector

The clock coming from KARA is provided through RF SMA connectors directly to the PLL.

4.1.2. Sampling-Channel

The most important circuit part of the sampling board is the sampling channel. On the board, 16 of such sampling channels are present integrating a wideband THA and delay chip. The sampling time of the THA, derived from the clock coming from the main PLL on the board, can be delayed individually by programming the delay time of each delay chip respectively (via FPGA).

Track-And-Hold-Amplifier

The THA used is the same as in KAPTURE. The component was chosen due to its high bandwidth (> 18 GHz) and low aperture jitter (range of hundreds of femtoseconds) [?]. Therefore it is also a good candidate for the new THERESA sampling board.

The main features of the THA are listed in Table 4.2. These input specifications are important for the later interface with the ADC with the delay chip. Switching characteristics are important for estimation of the maximal sample frequency possible and overall performance of the system.

The input coming from the power splitter is single-ended. However, the analog input of the THA is differential, therefore a 50Ω termination on the unused input pin has been added, as recommended in the data sheet [?].

The differential outputs are connected to the corresponding RFMC LPAM 8x50 connector pins. The schematics of the THA is shown in Figure 4.7.

At the power pins, decoupling capacitors and a ferrite bead were placed. The THA is a crucial component, as it samples the sensor signal, therefore any possible noise should be reduced to a minimum.

Separating Analog and Digital Ground

Digital ground is more noisy than analog ground due to switching of the digital components. Analog components are more sensitive to noise (due to e.g. lower amplitudes) than digital components⁴ and need a clean ground. In a mixed-signal (having both analog and digital

⁴Digital components work with voltage thresholds, rather than concrete voltage levels.

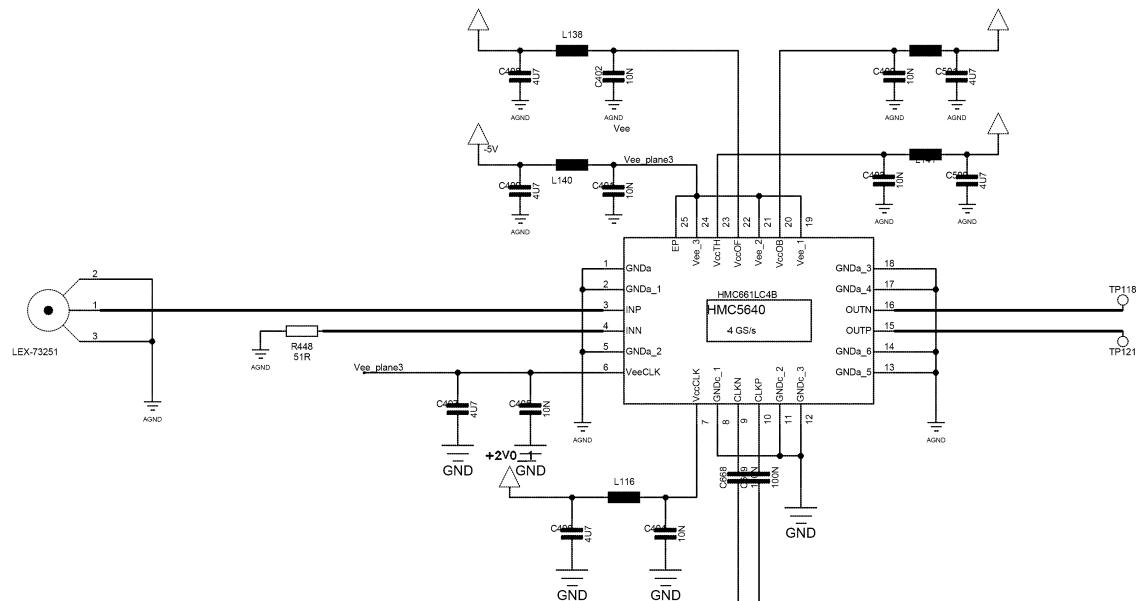
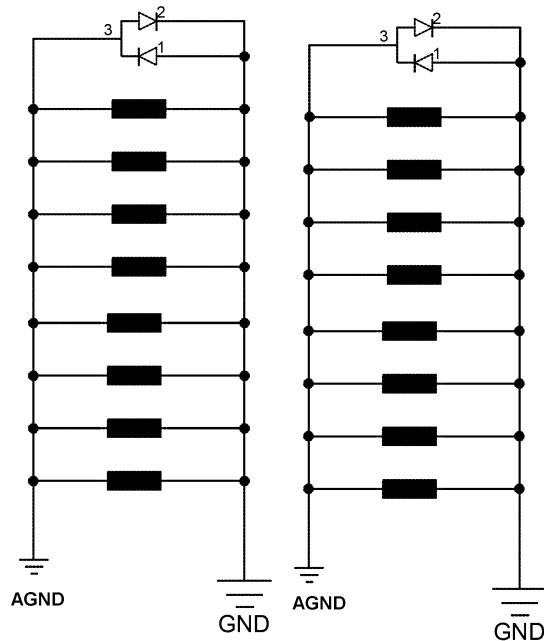
Table 4.2.: Specifications of the HMC5640 THA

Parameter	Min	Typ.	Max	Unit
Analog Inputs				
Differential FS Range		1		V _{pp} ¹
Common mode voltage	-0.1	0	0.1	V
Clock Inputs				
DC Differential High Voltage (Track Mode)	20	40	2000	mV
DC Differential Low Voltage (Hold Mode)	-2000	-40	-20	mV
Common mode voltage	-0.5	0	0.5	V
Analog Outputs				
Differential FS Range		1		V _{pp}
Common mode voltage		0		V
Track-to-Hold/Hold-to-Track Switching				
Aperture Delay		-6		ps
Random Aperture Jitter (FS, 1 GHz)	< 70			fs
Settling time ² (to 1 mV)		116		ps

¹Volt peak-to-peak²*Settling time* is the interval between the internal track-hold transition and the time when the output signal is settled within the specified value.

signals) PCB analog and digital ground should be therefore well separated. For some mixed-signal components, such as THAs, where separate analog and digital ground pins are provided. However, it is recommended to connect both grounds directly at the component. For the THAs in this design, this is done by connecting the ground pins via ferrite bead at each THA (see Figure 4.8). The ferrite bead mitigates any high-frequency components and therefore protects the analog ground from noise. For every THA one ferrite bead is needed, making in total 16 beads (see Figure 4.8).

Protection against a possible high voltage level between analog and digital grounds is implemented by two back-to-back diodes (Figure 4.8). The diodes should limit this voltage around 0.6 V.

**Figure 4.7.:** HMC5640 THA schematic**Figure 4.8.:** Connection of the analog and digital grounds at the THAs

Delay Chip

The delay chips are employed to create a delay in the sampling time of the THA chips. For the selection of the delay chip, the most important characteristics, apart from time jitter, is the delay step size and delay range.

To optimize the performance of THERESA and allowing a sample rate up to 40 GS/s (see subsubsection 3.2.1.2), the step-size of the delay chip must not exceed 25 ps.

With the HMC856 delay chip from *Analog Devices*, which is also used for the KAPTURE sampling board, a minimal step size of 3 ps [?] is possible. This is much less than 25 ps and thus the chip could be potentially used for the intended purpose. However, one drawback is the limited delay range of 100 ps. Considering a signal, which is stretched over several nanoseconds, this range limits the possibility to sample large time-stretched optical pulses. Another challenge, coming from the available number of I/Os, is the programming interface of the chip, which consists of five differential Current Mode Logic (CML) inputs. This means, one chip already takes up 10 pins only for control signals. For in total 16 necessary delay chips, this results in 160 pins used only for control of the delay chips. This uses up all pins of the FMC+ connector (see subsection 4.1.1) available for user-defined purpose.

A better candidate is the dual channel programmable delay chip NB6L295 from *ON Semiconductor*. This chip provides two separately programmable delay channels. This reduces the necessary chip count by half and therefore reduces the overall complexity of the PCB. The minimal delay step size of 11 ps lies under the maximal allowed 25 ps. Therefore the chip is suitable for the targeted interleaving method, covering a total delay range up to 8.8 ns per delay channel.

The chip is programmed via Serial Data Interface (SDI), which only requires 4 pins (enable pin, data pin, clock pin, load pin). Thus, the total number of digital control pins used by the delay chips is $4 \cdot 8 = 32$, which is a significant reduction compared to the 160 control pins needed by the HMC856 chips. This number can be even more reduced, by propagating the same data, clock and load pins to the chips and providing the enable signal on individual lines to the respective chip (see Figure 4.9). In this way, only 11 pins (8 enable pins and 3 pins for data, clock and load) are necessary in total for programming all delay chips.

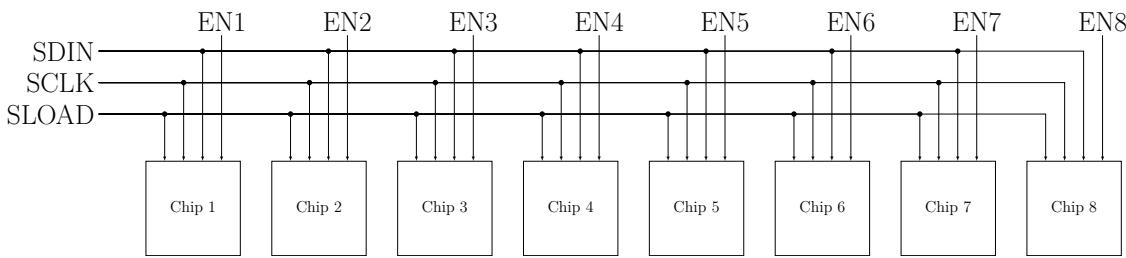


Figure 4.9.: Diagram of the SDI control pins for the NB6L295 delay chip. The data (SDIN), clock (SCLK) and load (SLOAD) pins are shared by all chips. Only the enable (ENx) signals are routed individually.

The schematic of the delay chips is shown in Figure 4.10.

Inputs

The inputs of the delay chip are driven by the preceding low time-skew, low-jitter and high-performance clock distribution , the outputs of which are Low-Voltage Positive Emitter-Coupled Logic (LVPECL) drivers. According to the data sheet, when driving the inputs with a LVPECL driver, the VT_x and $\overline{V}\text{T}_x$ pins of the delay chip need to be connected to

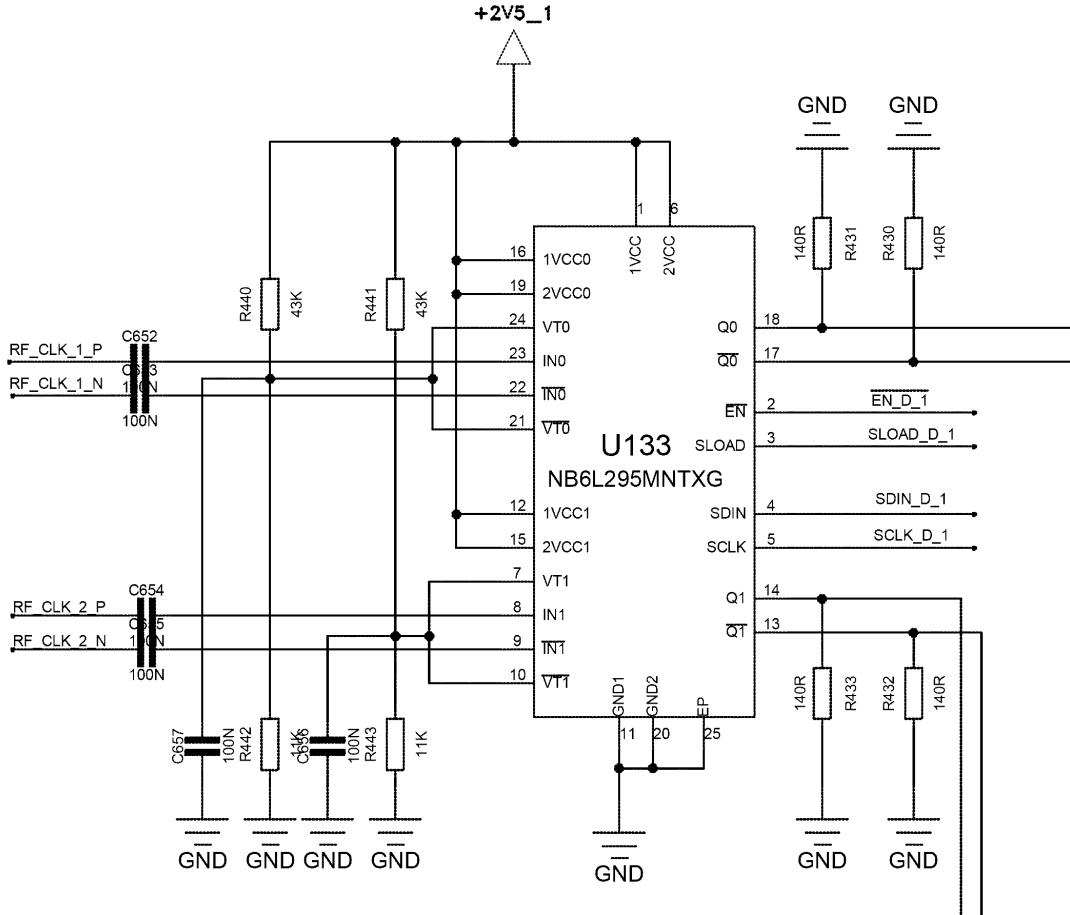


Figure 4.10.: NB6L295 delay chip schematic

$V_{cc} - 2\text{ V}$ (see Figure 4.12). In case of $V_{cc} = 2.5\text{ V}$, this results in a voltage level of $VTx = \overline{VTx} = 0.5\text{ V}$.

This voltage level is achieved by using a resistive voltage divider connected to V_{cc} . A voltage divider with the resistors R_1 and R_2 (see Figure 4.11) produces a voltage V_{out} which is a fraction of the input voltage V_{in} . V_{out} is calculated as

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}. \quad (4.1)$$

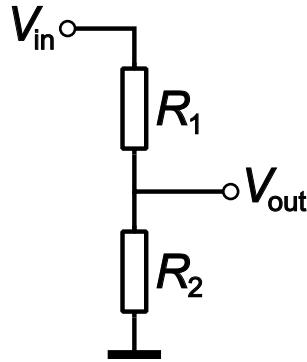


Figure 4.11.: Schematic of a resistive voltage divider

The resistor values are chosen to be $R_1 = 43\text{ k}\Omega$ and $R_2 = 11\text{ k}\Omega$. According to Equation 4.1

this results in a voltage of

$$V_{cc} \frac{11 \text{ k}\Omega}{11 \text{ k}\Omega + 43 \text{ k}\Omega} = 0.5093 \text{ V} \approx 0.5 \text{ V} \quad (4.2)$$

at the V_{Tx} and \bar{V}_{Tx} pins. Resistor values are chosen high to minimize current flow. A 100 nF capacitor is put in parallel to stabilize V_{cc} .

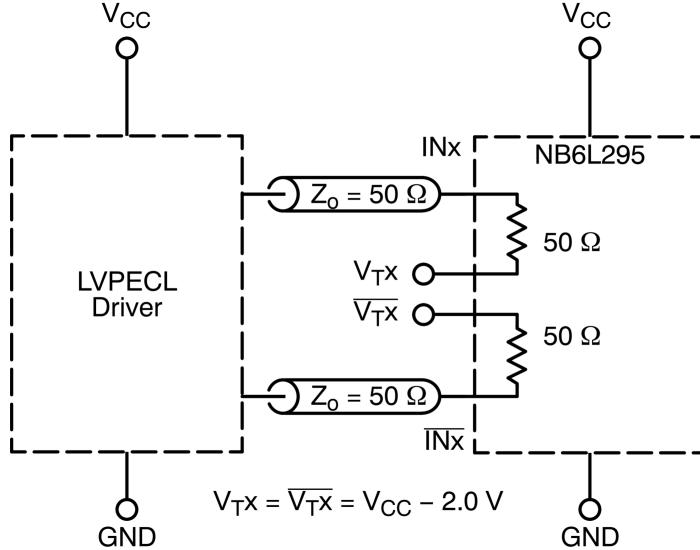


Figure 4.12.: LVPECL recommendations for NB6L295 [?]

Table 4.3.: Specifications of the NB6L295 delay chip [?]

Parameter	Min	Typ.	Max	Unit
Outputs				
Output HIGH Voltage	$V_{cc} - 1075$	$V_{cc} - 950$	$V_{cc} - 825$	mV
Output LOW Voltage	$V_{cc} - 1825$	$V_{cc} - 1725$	$V_{cc} - 1625$	mV
Output HIGH Voltage ($V_{cc} = 3.3 \text{ V}$)	2225	2350	2475	mV
Output LOW Voltage ($V_{cc} = 3.3 \text{ V}$)	1475	1575	1675	mV
Common mode voltage	-0.1	0	0.1	V
AC Characteristics				
Random Clock Jitter RMS		3	10	ps
Output Rise/Fall Times (@50 MHz)	85	120	170	ps
Serial Clock Input Frequency (50% Duty Cycle ¹)			20	MHz
Minimum Pulse width SLOAD	1			ns

¹Percentage of the ratio of pulse width and total period of the waveform.

According to the data sheet [?], the digital control pins need a minimum input HIGH voltage of 2 V. Directly connecting to the FMC+ connector pins is therefore not possible, as the maximal level provided by the readout card is smaller than 2 V. The SN74AVC32T245 bus transceiver from *Texas Instruments* (see Figure 4.13) which allows for level shifting from 3.3 V at the device input to 1.8 V at the device output. In this design, the bus transceiver is configured to propagate signals from the “A” ports (coming from the FMC+ connector) to the “B” ports (going to the delay chips), shifting the signals from the V_{ADJ} (1.8 V) of the FMC+ connector to 2.5 V (see Figure 4.13). Furthermore, resistors are placed at the pins to reduce possible voltage overshootings which result from reflections on the line.

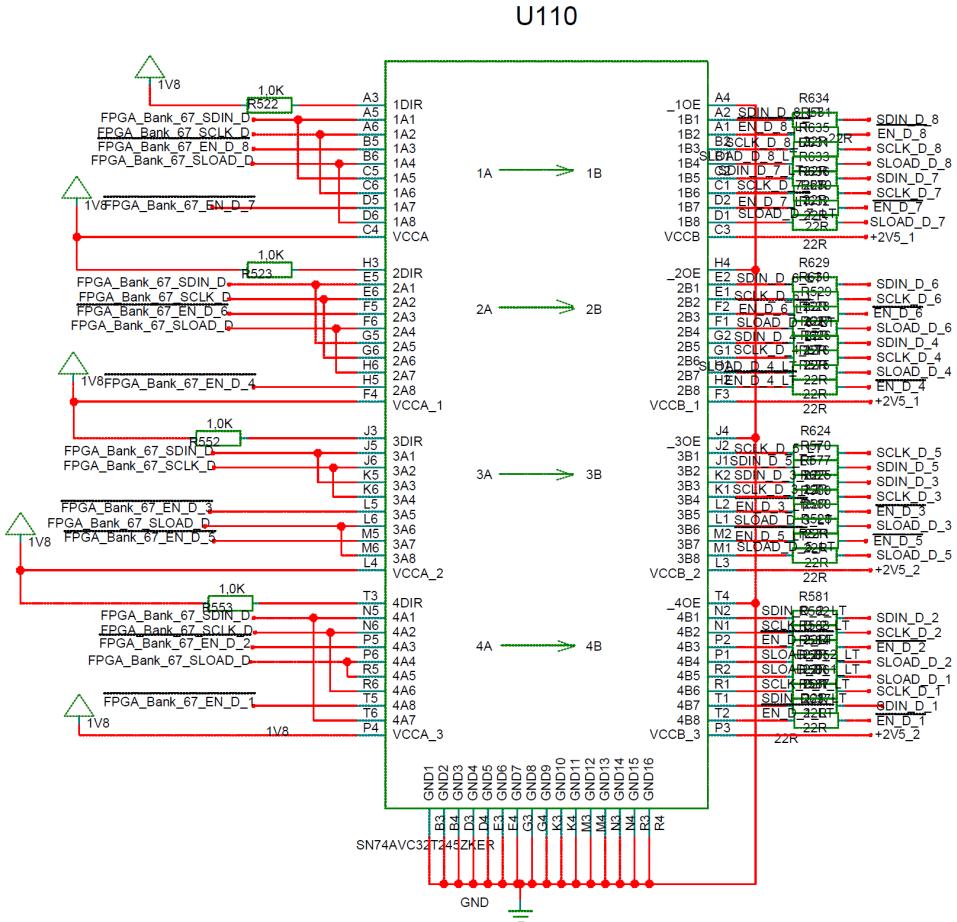


Figure 4.13.: Schematic of the SN74AVC32T245 bus transceiver

Outputs

The output of the delay chip is using a LVPECL signaling interface, which is based on an open-emitter topology (see Figure 4.14). This requires a path to DC, which is achieved by adding $140\ \Omega$ resistors (recommended in the data sheet).

As the output will be connected to the THA, it is necessary to check the compatibility of the maximum amplitude and common-mode of the pins.

According to the data sheet [?], the voltage level of the output can vary between $V_{cc} - 1825\text{ mV}$ and $V_{cc} - 825\text{ mV}$ (see Table 4.3). Maximal voltage amplitude acceptable by the THA inputs is 2000 mV (see Table 4.2). When using a supply voltage of $V_{cc} = 3.3\text{ V}$, provided e.g. by the read-out card through the FMC+ connector, this leads to a maximum output level of 2475 mV . This exceeds the limit given by the THA. Therefore, for V_{cc} a smaller voltage should be considered. In this design a voltage of $V_{cc} = 2.5\text{ V}$ is chosen, which guarantees that the amplitude falls within the range 675 mV to 1675 mV .

The second point to consider is the common mode voltages. According to the data sheet of the THA, the common mode voltage of the input clock pins is 0.1 V (see Table 4.2). The common mode voltage of the delay chip is not explicitly mentioned in the data sheet, thus it has to be calculated.

The common mode voltage V_{CM} is just the mean value between the high level and the low

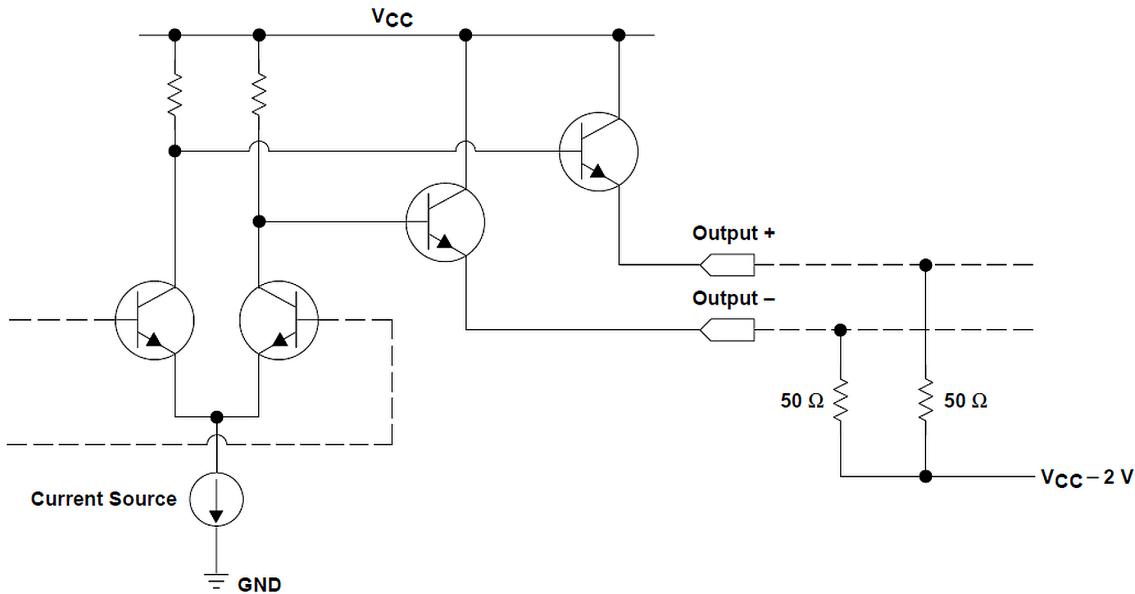


Figure 4.14.: LVPECL driver topology. Left side shows the emitter-follower based driver. On the right, an example biasing with resistors is shown. [?]

level voltage of the output pins:

$$V_{CM} = \frac{V_{out, \text{LOW}} + V_{out, \text{HIGH}}}{2}. \quad (4.3)$$

According to this, the common mode voltage V_{CM} of the delay chip output, when taking the minimum/maximum voltage level values, is

$$V_{CM} = \frac{675 \text{ mV} + 1675 \text{ mV}}{2} = 1175 \text{ mV}. \quad (4.4)$$

This is higher than the maximal input common mode voltage of the THA. AC coupling is therefore necessary in this case, i.e. connecting the pins via capacitors.

4.1.3. Clock Distribution

The clock distribution is designed as shown in Figure 4.15.

The LMK04808B low-noise clock jitter cleaner with dual-loop PLLs from *Texas Instruments* cleans the incoming reference clock provided from the system (e.g. from KARA) for high temporal precision [?]. It is used with an external Voltage-Controlled Crystal Oscillator (VCXO) from *ABRACON*.

The LMK04808B contains two PLLs (therefore called “dual-loop”). The first PLL is used to clean the jitter from the reference clock. The second is then used to generate and distribute the cleaned clock signals to the outputs of the components.

Time skew between the PLL outputs lies in the range of 30 ps [?] which is too high if high temporal accuracy is desired. In order to guarantee low time skew (range of few picoseconds), two fanout buffers are used to distribute the cleaned clock signal to the components on the board. The fanout buffers used are the HMC987LP5E from *Analog Devices*. Time skew of these components between channels typically lies in the range of 1.5 ps, 20 times smaller than the time skew of the PLL. As one fanout buffer has eight outputs, two chips are needed to cover all 16 sampling channels. Each fanout receives the clock reference from one output of the LMK0480B. To ensure exactly identical clocking

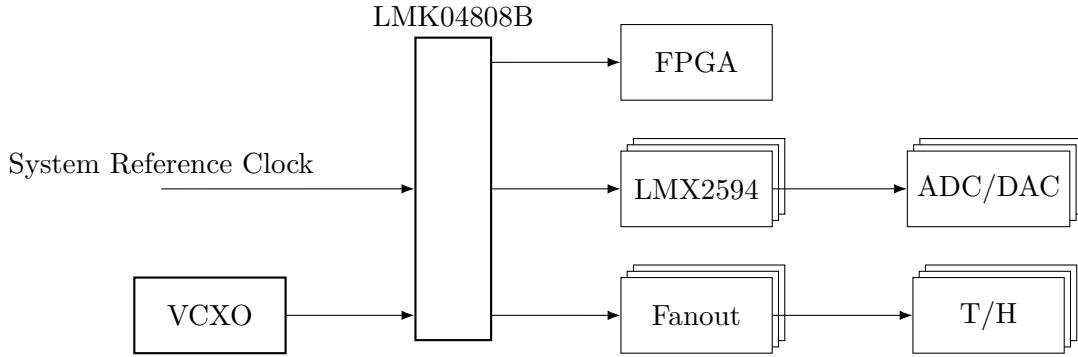


Figure 4.15.: Overview of the clocking paths on the sampling board

signals between the two fanouts they are connected to two outputs of the same output group.

The LMK04808B has only 12 outputs which are divided into six groups à two outputs. Outputs in one group have the same configuration (frequency, phase, ...), which means that effectively only six different outputs are available. This is not enough for the 16 THAs and additional clock signals needed for FPGA, ADCs and DAC.

One output of the PLL is propagated to the FMC+ connector as reference clock for the FPGA.

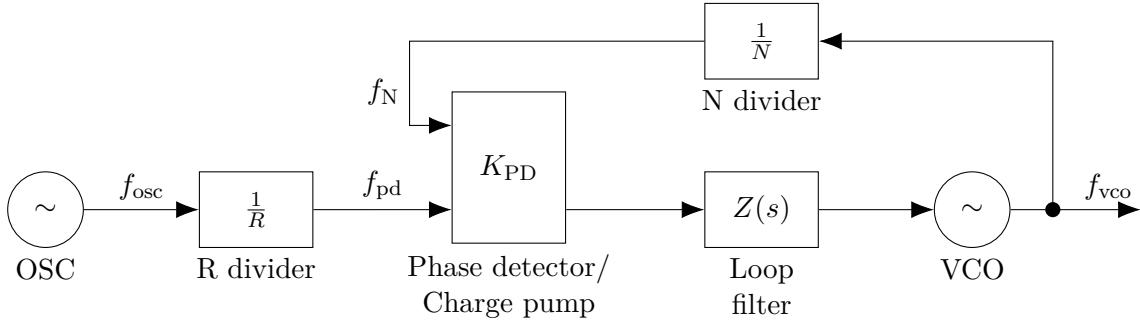
The maximum output frequency of the LMK04808B is 1536 MHz, not enough to clock the ADCs at maximum sampling rate (2.5 GS/s). A second kind of PLL is therefore needed to provide programmable reference clocks to the ADCs and DACs. As Figure 4.15 shows, the LMK04808B also provides a clocking signal to other PLLs, the LMX2594 from *Texas Instruments*. This PLL is able of clocking signal frequencies up to 15 GHz.

Due to the ADC clocking limitations on the read-out card explained in subsubsection 3.2.1.2, two of the PLLs are needed. The reference clock signal is provided by outputs from different output groups of the LMK04808B. This way, the phase of each reference clock can be programmed individually, which allows to implement the ADC clocking technique described in subsubsection 3.2.1.2.

PLLs

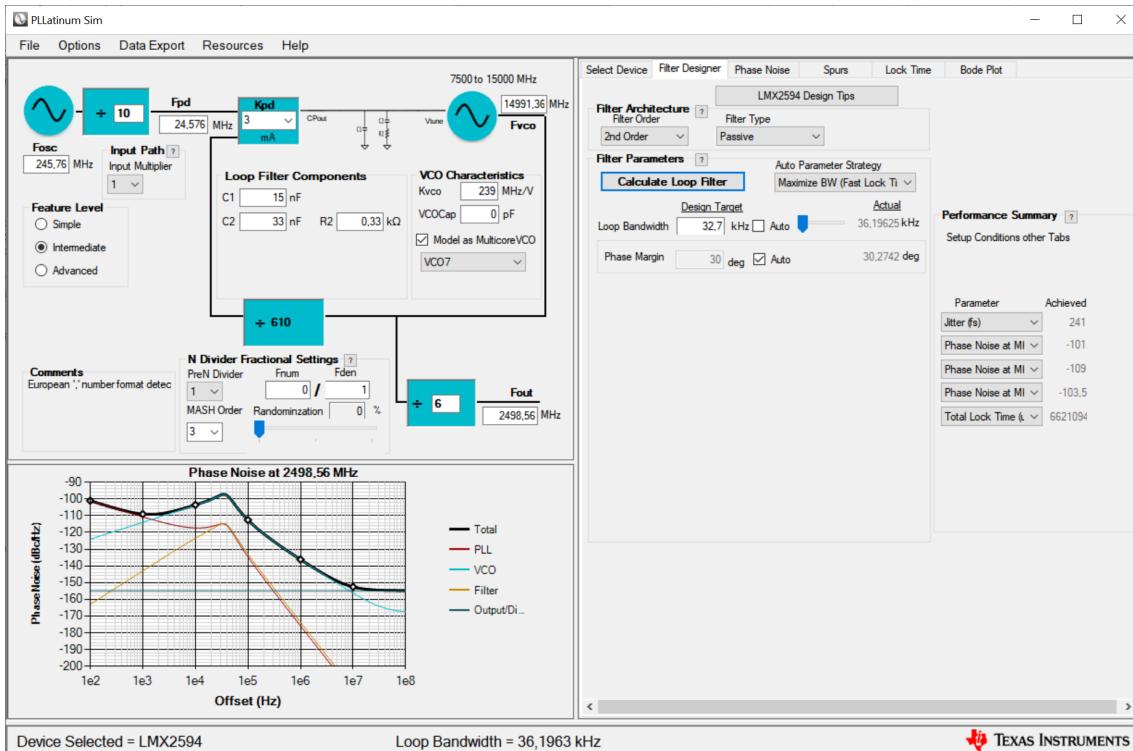
A PLL is a control loop, used to synchronize an output oscillator signal with a reference signal. The principle lies in comparing the phases between the two inputs. When there is a varying phase difference, it means that the signals are at different frequencies. As soon as the phase difference stays constant, it means that both of the frequencies are the same or one is a multiple of the other. In this state the PLL is called to be “locked”. The general architecture of a PLL is shown in Figure 4.16.

For proper noise performance, a properly designed loop filter for both PLLs is needed. The output of the loop filter is the voltage for controlling the Voltage-Controlled Oscillator (VCO). The VCO output of is a frequency f_{VCO} proportional to the applied voltage. f_{VCO} is divided by the N Divider to the frequency f_n and then compared to the phase detector

**Figure 4.16.:** General block diagram of a PLL [?]

frequency f_{PD} in the phase detector. f_{PD} results from dividing the reference frequency f_{OSC} with an R divider. The phase detector produces current correction pulses (with magnitude K_{PD}) with a duty cycle proportional to the phase error between f_{PD} and f_n . These pulses are filtered by the low pass loop filter, which basically converts these pulses into a voltage. [?] The loop filter is one of the key component determining the PLL performance (concerning jitter, noise, ...) and therefore has to be designed carefully.

To calculate the loop filter, the *Texas Instruments PLLatinum Sim* tool is used (see Figure 4.17). This tool provides a convenient way to calculate the necessary loop filter components, given the VCO characteristics, desired filter order, charge pump current and desired performance (e.g. optimize jitter). For the names of the filter components refer to Figure 4.18.

**Figure 4.17.:** Screenshot of the TI PLLatinum Sim tool for loop filter design and PLL performance simulation

The LMK04808B has two PLLs inside (PLL1 and PLL2). For both the loop filter has to be calculated separately. The parameters/values are shown in Table 4.4. The filters are implemented as second order filters, note that PLL2 has already a partially integrated loop

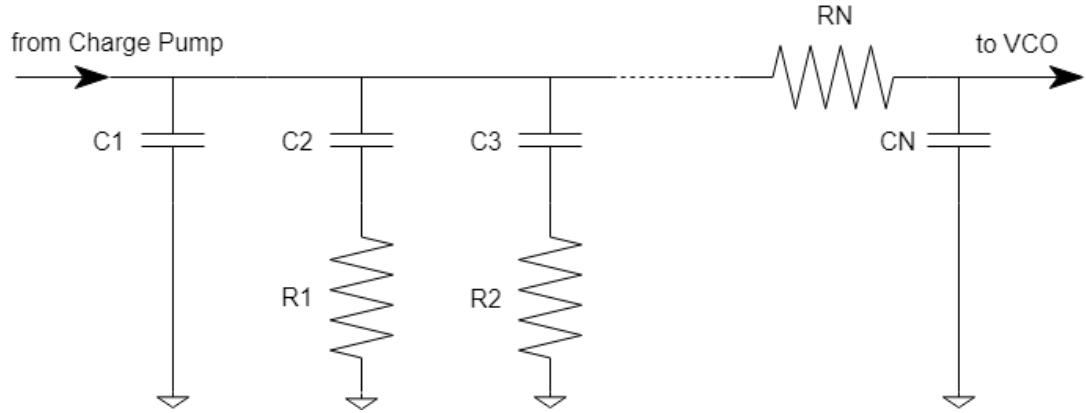


Figure 4.18.: Picture to be replaced. General n-th order passive loop filter for PLL. To reduce the order leave components out.

filter.

The loop filter for the LMX2594 is designed in the using the same *Texas Instruments* software. The calculated values are shown in Table 4.5. The filter is implemented as a third order passive filter. The schematic and layout of the filter have been implemented in a way to enable flexible change of the filter, i.e. changing the order or component values. This gives the possibility to experimentally find the correct order and components for best performance by real measurements with the board. The schematic of the LMX2594 is shown in Figure 4.19.

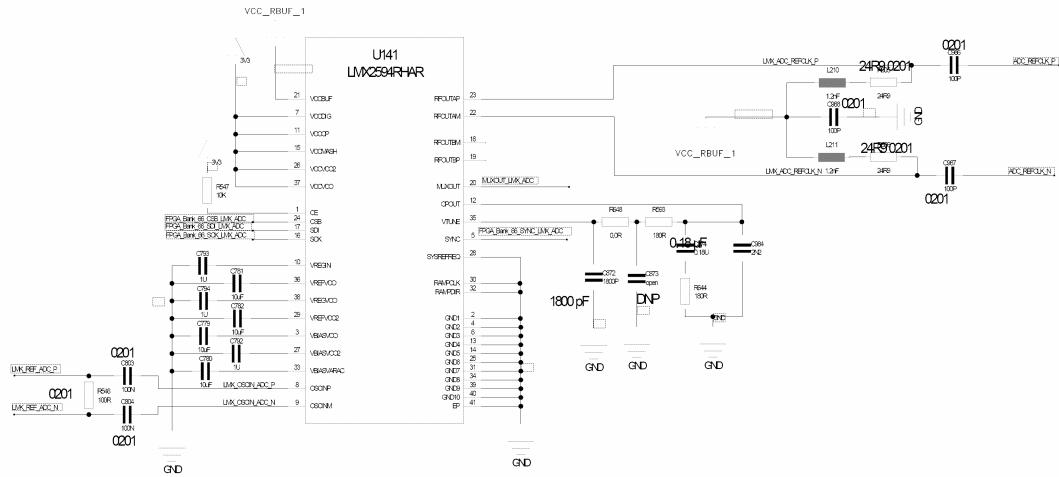


Figure 4.19.: Schematics of the LMX2594

Both PLLs are supplied by the digital 3.3 V coming from the FMC+ connector. For the output buffer supply (VCC_RBUF) of the LMX2594, an additional EMI filter is used to provide effective filtering of the voltage level. The output pins are pulled-up and filtered via ferrite bead to VCC_RBUF as recommended in the data sheet. For the schematic of the LMX2594 see Figure 4.19.

Table 4.4.: Loop filter characteristics of the LMK04808B

Parameter	Value
PLL1 parameters	
VCO Gain	0.15 MHz/V
Loop Bandwidth	0.2578 kHz
Phase Margin	70°
Effective Charge Pump Gain	0.4 mA
Phase Detector Frequency	25 MHz
VCO Frequency	200 MHz
Loop filter components for PLL1	
C_1	39 nF
C_2	1800 nF
R_2	2.2 kΩ
PLL2 parameters	
VCO Gain	30 MHz/V
Loop Bandwidth	390.9624 kHz
Phase Margin	70°
Effective Charge Pump Gain	1.6 mA
Phase Detector Frequency	50 MHz
VCO Frequency	3000 MHz
Loop filter components for PLL2	
C_1	22 pF
C_2	2.2 nF
R_2	3.3 kΩ

Table 4.5.: Loop filter characteristics of the LMX3594

Parameter	Value
PLL parameters	
VCO Gain	239 MHz/V
Loop Bandwidth	32.7 kHz
Phase Margin	69°
Effective Charge Pump Gain	3 mA
Phase Detector Frequency	24.576 MHz
VCXO Frequency	Designed for 15 GHz
Loop filter components	
C_1	2200 pF
C_2	180 nF
C_3	1800 pF
R_2	160 Ω
R_3	180 Ω

Fanout Buffer

The fanout buffer receives as input the clock signal from the main PLL and distributes it to the delay chips. In this design, the HMC987LP5E from *Analog Devices* is chosen due to its low jitter and low skew performance. The schematics of the chip is shown in Figure 4.20.

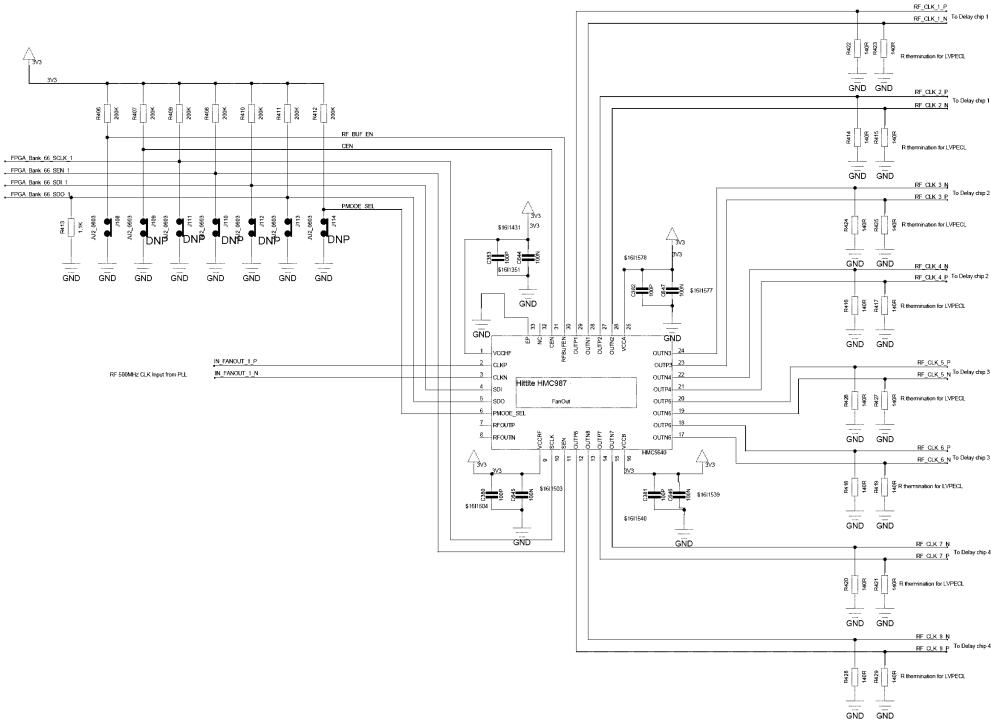


Figure 4.20.: Schematic of the fanout

Decoupling capacitors are placed at every power supply pin in order to guarantee a clean and stable voltage level.

The outputs of the fanout buffer is based on LVPECL signaling interfaces and therefore need to be connected to ground via resistors. Outputs can be enabled or disabled either via SPI (setting pin PMODE_SEL to '0') or by using parallel pin control (setting pin PMODE_SEL to '1').

In parallel pin control the SPI pins **SCLK**, **SDI** and **SEN** are reinterpreted as a 3-bit control bus. In this mode, the pins are either pulled up to V_{CC} or connected to ground via on-board jumpers to represent a logic '1' or '0'. For the design, the parallel pin control mode is chosen, therefore the **PMODE_SEL** pin is pulled up to V_{CC} (\cong logic '1'). In order to have the opportunity to enable the SPI mode in later usage, a jumper is placed at this pin so that it can be connected to ground if necessary (\cong logic '0', enabling SPI mode). The **SCLK**, **SDI** and **SEN** pins are connected to the FMC+ connector. To enable all outputs, the SPI pins need to be set to '111' according to the datasheet, i.e. pulled-up to V_{CC} . Here, jumpers are foreseen as well, to allow enabling/disabling in later usage.

4.1.4. Digital-To-Analog-Converter Channels

For test purposes, two DAC channels from the read-out card are routed on the sampling board. In this way, a programmable analog waveform can be generated by FPGA, without the need for an external signal generator. The differential inputs from the DACs are transformed into single ended outputs with dedicated baluns⁵. the BD3150N50100AHFa

⁵balanced to unbalanced

and the BD4859N50100AHF from *Anaren*. These are used for the signal frequency range 3.1 GHz to 5.0 GHz and 4.8 GHz to 5.9 GHz respectively.

The single-ended output is connected to a miniature RF connector from *Hirose Electric*. The programmable analog waveform, generated by the DAC (operating up to 10 GS/s), can be applied to the input of the sampling board as a test signal and be employed for testing, characterizing and calibrating each sampling channel individually.

The schematic of a DAC channel is shown in Figure 4.21.

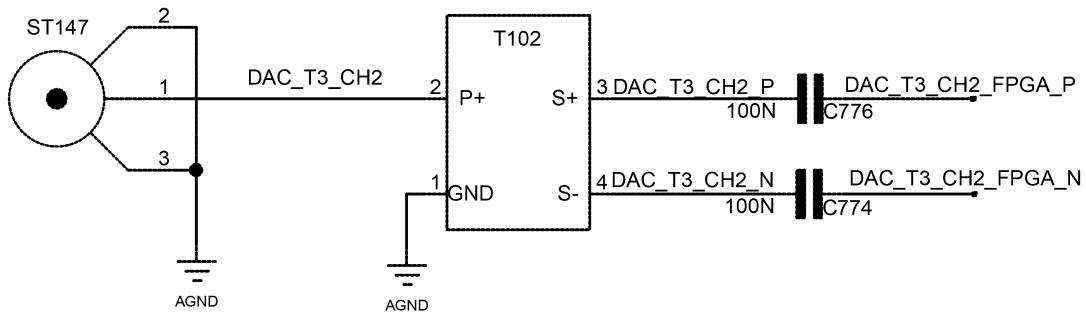


Figure 4.21.: DAC-channel with balun. Signal propagates from right to left.

4.1.5. Power Supply

Low-ripple power supply is an important point key for low noise performance of the board. Especially high-performance ICs, such as THAs, highly rely on a low-ripple voltage level for correct functionality. Therefore, proper power supply design is an important step which needs to be handled with care. This step includes the selection the right type voltage regulators, as well as providing appropriate filtering. Furthermore, in order

Table 4.6 lists the power supply requirements of all the components used on the board.

Table 4.6.: Power consumption of components on the board

Component	V_{cc} (V)	I_{max} (A)	P_{max} (W)	#components	$I_{tot, max}^1$ (A)
HMC5649 (THA)	2	0.221	0.442	16	3.536
NB6L295 (Delay chip)	2.5	0.170	0.425	8	1.36
HMC987LP5E (Fanout buffer)	3.3	0.234 ²	0.772	2	0.468
LMK04808B (PLL)	3.3	0.590 ³	1.947	1	0.590
LMX2594 (PLL)	3.3	0.340	1.122	2	2.244
VCXO	3.3	0.03	0.198	1	0.03

¹for 16 ADCs

²All Outputs and RF-Buffer

³All CLKs

In general, there are three different voltage levels provided by different components:

- 1.8 V for digital components coming from FMC+ connector

- 3.3 V for digital components coming from FMC+ connector
- 3.3 V and -5 V for analog devices from external power supplies

An EMI filter needs to be placed in order to keep noise of the power supply and read-out card from the sampling board (see Figure 4.22).

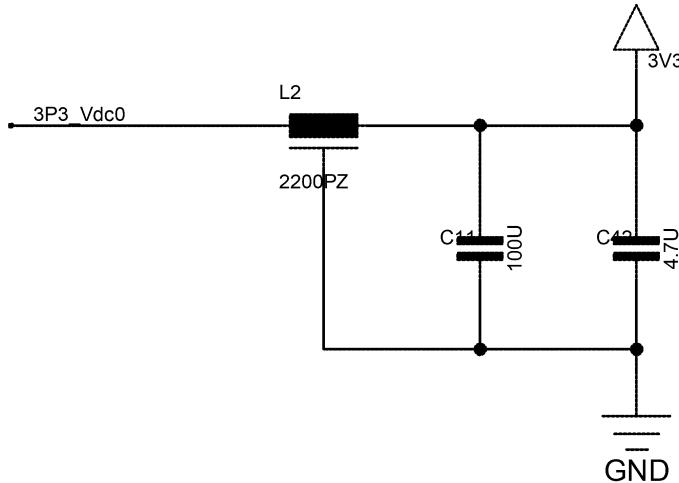


Figure 4.22.: EMI-filter used for power supply

For the sensitive components like THA, voltage regulators are needed, to guarantee a stable voltage level.

Voltage Regulator for Track-and-Hold-Amplifiers

The THAs need a low-ripple voltage level for optimal operation. Linear voltage regulators are capable to maintain a stable output voltage and are therefore to be used with the THAs.

On the KAPTURE sampling board, the Low Dropout Voltage Regulator (LDO) ADP1708 from *Analog Devices* is used to provide a power supply for the THAs. A LDO is able to operate at a low potential difference between the input and output voltage. This low potential difference has also the benefit of low power dissipation, which also reduces the power supply can provide at maximum 1 A to the load. In order to minimize the amount of components needed on the board and to save space, a different LDO which provides higher currents should be used. This way, one single voltage regulator can be used for more components.

For the new board, the ADP1741 low-dropout voltage regulator from *Analog Devices* has been selected. This voltage regulator has adjustable output voltage from 1.6 V to 3.6 V and a maximum output current of 2 A.

It is necessary to think about the number of voltage regulators needed. As a rule of thumb, the power supply should provide at least twice the maximum current (i.e. power) needed by the components it drives. [?] The power consumption/maximum current for the THAs on is listed in Table 4.6.

The maximal output current $I_{\max, \text{LDO}}$ from the ADP1741 is 2 A. With the rule mentioned above and the maximal current draw $I_{m, \text{THA}} = 0.221 \text{ A}$ from the THA, the maximal number N of components which the LDO can handle is calculated as

$$I_{\max, \text{ LDO}} > 2 \cdot N \cdot I_{\text{m, THA}}$$

$$I_{\max, \text{ LDO}} / (2 \cdot I_{\text{m, THA}}) > N$$

$$2 \text{ A} / (2 \cdot 0.221 \text{ A}) > N$$

$$4.52 > N \rightarrow N = 4$$

This means, 4 LDOs are needed to cover 16 THAs.

The output voltage level of the regulator is set by an external divider with the resistors R_1 and R_2 (refer to Figure 4.23). According to the datasheet [?] the voltage V_{OUT} is determined by

$$V_{\text{OUT}} = 0.5 \text{ V} \left(1 + \frac{R_1}{R_2} \right) \quad (4.5)$$

In order to achieve the required 2 V, the values of the resistors are chosen to $R_1 = 30 \text{ k}\Omega$ and $R_2 = 10 \text{ k}\Omega$.

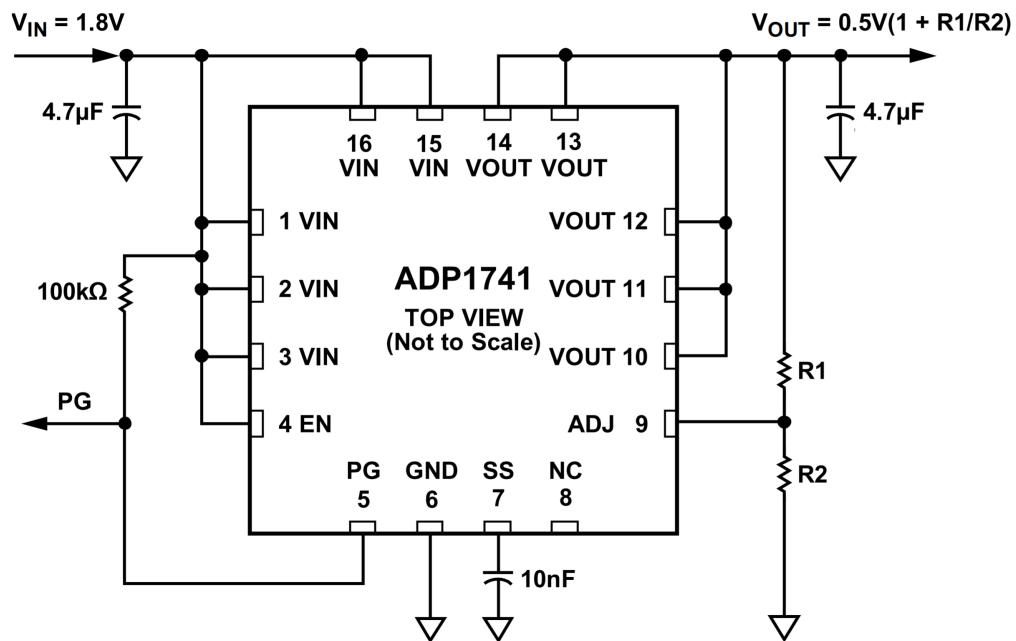


Figure 4.23.: Recommended schematic of the ADP1741 voltage regulator [?]

As input voltage, the 3.3 V from the external power supply is provided.

Capacitors and resistors are placed as recommended in the datasheet[?](see Figure 4.23).

Voltage Regulator for Delay Chips

The delay chips require a voltage level of 2.5 V. As they propagate the sensitive clock signals they also need stable voltage levels. The number N of delay chips with a maximal current draw $I_{m,\text{Delay}}$, which one LDO can handle, can be again calculated as:

$$\begin{aligned}I_{\max, \text{ LDO}} &> 2 \cdot N \cdot I_{\text{m, Delay}} \\I_{\max, \text{ LDO}} / (2 \cdot I_{\text{m, Delay}}) &> N \\2 \text{ A} / (2 \cdot 0.170 \text{ A}) &> N \\5.88 &> N \quad \rightarrow N = 5\end{aligned}$$

Therefore, two regulators are needed to cover the 8 delay chips. In order to keep the current draw evenly distributed among the regulators, 4 chips are assigned to one regulator respectively.

In order to set the output voltage of the regulator to 2.5 V, the resistor values $R_1 = 12 \text{ k}\Omega$ and $R_2 = 3 \text{ k}\Omega$ are chosen (refer to Figure 4.23 and Equation 4.5).

The 2.5 V are also used as input for the bus transceiver which acts as a level translator. The current draw from this component lies in the range of μA and can thus be neglected.

As input voltage these regulators receive the digital 3.3 V from the FMC+ connector. The ground pins are connected to the digital ground of the PCB. This is important, as the regulator for the digital components should be separated from the analog part of the PCB.

Power Dissipation of the Voltage Regulators

According to the data sheet of the ADP1741 [?], the power dissipation P_D of the regulator can be calculated with the input and output voltage V_{IN} and V_{OUT} , load current I_{LOAD} and ground current I_{GND} ⁶:

$$P_D = (V_{IN} - V_{OUT}) \cdot I_{LOAD} + (V_{IN} \cdot I_{GND}) \quad (4.6)$$

I_{GND} is very small (range of μA), thus the power dissipation due to this current can be neglected. Therefore the equation above can be simplified to:

$$P_D = (V_{IN} - V_{OUT}) \cdot I_{LOAD} \quad (4.7)$$

The power dissipation $P_{D, THA}$ of one voltage regulator for the THAs is therefore

$$P_{D, THA} = (3.3 \text{ V} - 2 \text{ V}) \cdot (4 \cdot 0.221 \text{ A}) = 1.149 \text{ W}. \quad (4.8)$$

The power dissipation $P_{D, Delay}$ of one voltage regulator for the delay chips is

$$P_{D, Delay} = (3.3 \text{ V} - 2.5 \text{ V}) \cdot (4 \cdot 0.17 \text{ A}) = 0.544 \text{ W}. \quad (4.9)$$

In order to dissipate the heat, an exposed pad is provided under the component. This pad can be connected through vias to a (ground) plane located on the inner layers, in this way allowing to dissipate the heat through the PCB. Furthermore, a matrix of vias is placed underneath the component area in order to improve the heat flow. This should be enough to handle the calculated power dissipation. Heat sinks could be added later during operation, if the deployed method

4.2. Layout

After completing the schematic capture, the following step is the PCB layout design. During this process, the following points need to be considered:

- An appropriate PCB substrate has to be chosen. The most important parameter of a substrate is its dielectric constant. For high-frequency circuits, a low dielectric constant is necessary.
- Generally, complex PCBs consist of a number of layers. In order to be able to route all the signals, it is necessary to think about the number of layers needed.

⁶difference between input and output current

- Closely linked to the dielectric constant are the transmission lines. The geometry of these lines has to be calculated in order to meet the desired characteristic impedance (single-ended: 50Ω , differential pair: 100Ω). As this impedance also is defined by the dielectric constant, this step is closely linked to the selection of the substrate.
- Components need to be placed in a way that minimizes traces and routing. Sensitive components, like THAs have to be placed first
- Route traces, taking care that traces of the same group (e.g. clock signals distributed to the THAs) have the same length. For sensitive signals take care that these are shielded by ground planes on the layers above and below.
- Places additional structures to reduce cross-talk, EMI, etc. (via fences, stitching vias, ...)
- Create proper power distribution by placing planes at appropriate places, i.e. reducing overlapping with traces carrying signals that could induce noise on the power plane.

For better understanding, first a general overview over PCB structures is given. Then the steps mentioned above are described.

PCB Structures Overview

In this section an overview over the basic structures on a PCB is given.

Traces

A *trace* is a strip of metal, which establishes an electrical connection and carries signals between two (or more) points in the horizontal plane of a PCB. [?]

Planes

Plane denotes an uninterrupted area of metal, which covers the whole PCB layer. If this area only covers a part of the layer, it is called a *planelet*. These areas provide power distribution across the PCB and present an important transmission medium for the return current⁷. [?]

Vias

A via is metal-plated hole, which is used to route a trace in vertical direction, i.e. from the PCB outer layer to the inner layers. They carry signals and power. Three types of vias are [?]:

- Blind via: A blind via connects the surface layers with at most three layers below.
- Buried via: A buried via only connects internal layers.
- Through-via: A through-via goes from one PCB surface to another and is used to connect any layer.

In this design only blind and through vias are used due to manufacturing limitations.

⁷Any current, which is injected into the components/boards, needs a return path, as otherwise there is no closed circuit.

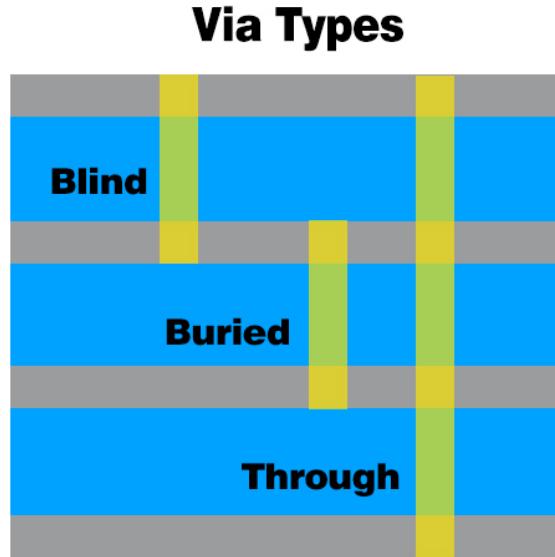


Figure 4.24.: Visualization of via types [?]

4.2.1. PCB Substrate Selection and Metal Layer Stackup

Proper substrate material has to be selected in according to the use-case. The MEGTRON 6 from *Panasonic* is designed for high-speed/high frequency applications. Characteristics of this material are:

- Low dielectric constant: $\epsilon_r = 3.61$ at 10 GHz, 3.71 at 1 GHz
- Low dielectric dissipation factor: 0.002 at 10 GHz, 0.004 at 1 GHz
- Low transmission loss
- High heat resistance: Decomposition temperature $T_d = 410^\circ\text{C}$

Another important step is deciding the number of layers. The complexity of the board implies that a lot of layers are needed. For this design a number of 16 layers is chosen.

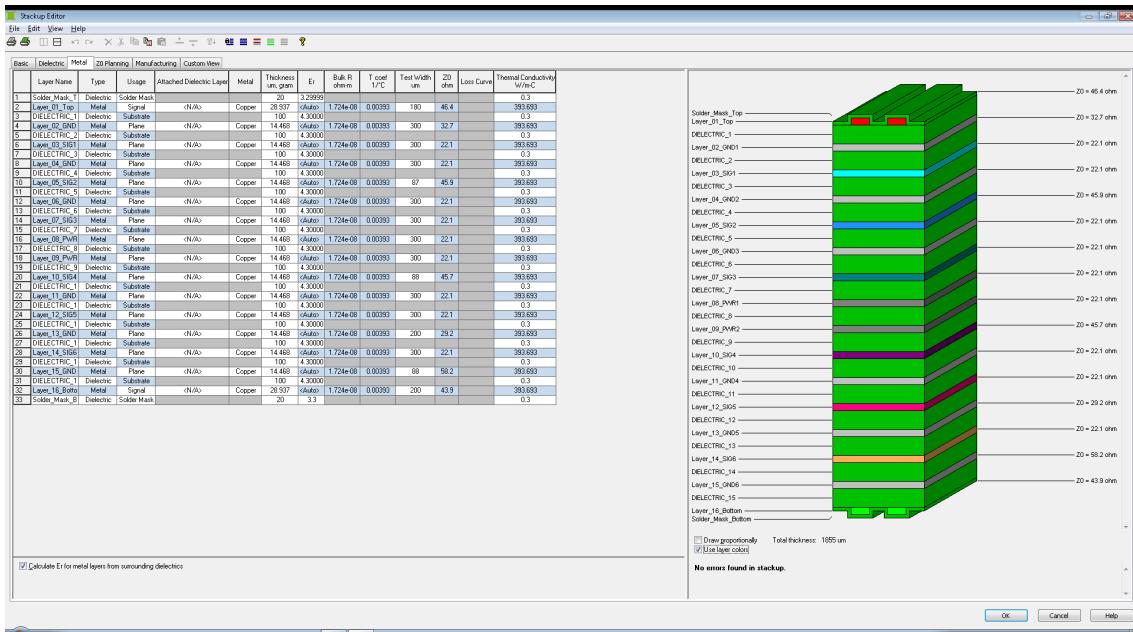


Figure 4.25.: Metal Layer stackup showing 16 layers.

4.2.2. Transmission Lines

Transmission lines guide electromagnetic waves from one point to another. They have a characteristic impedance which is determined by parameters like width of the trace, separation from ground plane, etc. Not matching correctly can lead to reflections and damping. For single-ended signals the waveguide characteristic impedance should be $50\ \Omega$, for differential pairs $100\ \Omega$. The impedance has to be matched especially for sensitive, high-speed signals, e.g. clock signals. Proper calculation of the geometrical parameters is therefore very important to ensure signal integrity and reduce reflection and damping.

Formulas to calculate the characteristic impedance are quite lengthy and not easy to solve. To make the design of transmission lines easier, tools exist to quickly calculate the geometric values needed for appropriate impedance. For this design, the Si9000e tool for modeling PCB transmission lines from *Polar* (see Figure 4.26) is used to calculate the necessary trace widths, trace separations, etc.



Figure 4.26.: Screenshot of the Polaris Si9000e tool for modeling PCB transmission lines, showing calculation of characteristic impedance of a coplanar waveguide

As there are a lot of parameters which can be tuned, as a starting point the geometrical parameters from the KAPTURE system are applied. These were carefully designed for optimal signal transmission. However, the substrate used in the KAPTURE system, has a different dielectric constant than the Megtron6 substrate used for the new design. Therefore,

the impedance has to be recalculated to check whether it is still acceptable. A deviation of 10% from the ideal 50Ω and 100Ω is still regarded as acceptable, as tolerances during manufacturing need to be considered. The change in impedance/parameters is assumed to be negligible, as the difference in dielectric constants between the two boards is not large (KAPTURE: $\epsilon_r = 3.52$, new board: $\epsilon_r = 3.61$)

Three types of waveguides are used in this design:

- Surface coplanar waveguide with ground for analog input to the THAs
- Differential surface coplanar waveguide with ground for output from the delay chips to the THAs
- Offset differential coplanar waveguide for clock signals and signals coming from the THAs

These waveguide types are presented and the geometric dimensions calculated with the Si9000e tool are presented.

Surface Coplanar Waveguide with Ground

The surface coplanar waveguide has the geometry shown in Figure 4.27. The single trace of thickness t and width a lies between two ground planes on a dielectric of thickness h and the effective dielectric constant ϵ_r . Another ground plane is located at the bottom of the dielectric. Separation between trace and ground plane is defined as $(b - a)/2 := d$.

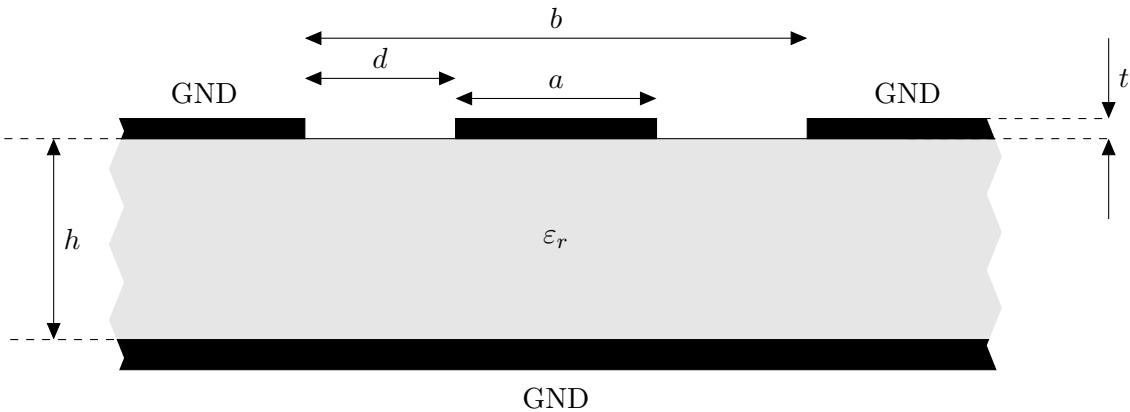


Figure 4.27.: Coplanar Waveguide with Ground

To have a rough starting point of the dimensions of the parameters, the following widths are taken from the KAPTURE board:

- $a = 213\mu\text{m}$
- $d = 250\mu\text{m}$

In the Si9000e tool, an upper and a lower trace width can be specified, therefore taking into account the etching process during manufacturing. As the exact upper trace width is not known, both widths are assumed to be of the same value if not stated otherwise. The thickness t of the trace and the thickness h of the dielectric is defined by the used substrate. For Megtron6 it is

- $t = 30\mu\text{m}$
- $h = 100\mu\text{m}$

With all these parameters, the value for the characteristic impedance is calculated to $Z_o = 47.33 \Omega$. This lies well in the 10% tolerance range of 45Ω to 55Ω .

According to the datasheet of the Megtron6, the dielectric constant ϵ_r changes over frequency (see subsection 4.2.1). As the dielectric constant ϵ_r of the Megtron6 substrate varies between 3.61 and 3.71 depending on the frequency, the effect of the changing ϵ_r should also be studied. The Si9000e tool provides the possibility to simulate the characteristic impedance versus a changing parameter. In Figure 4.28 the characteristic impedance Z_o is plotted against ϵ_r . It can be seen that with higher effective dielectric constant the characteristic impedance decreases. The lowest values lies around 47Ω , a change of 0.7%, which is still inside the 10% tolerance range.

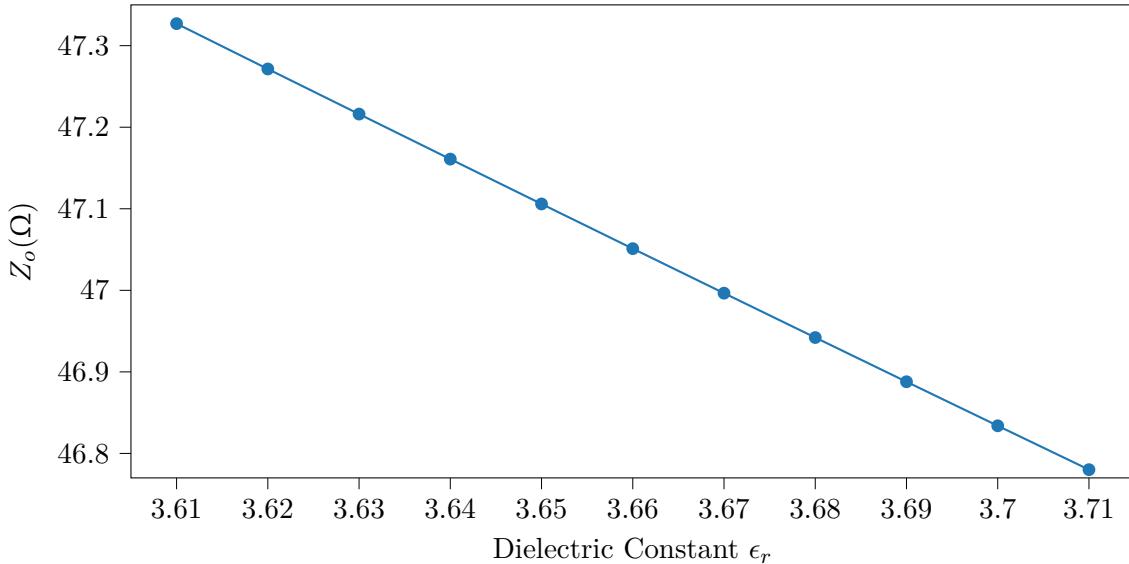


Figure 4.28.: Characteristic impedance Z_o of a coplanar waveguide versus dielectric constant ϵ_r assuming $a = 213 \mu\text{m}$

Furthermore, the effect of changing the trace width on Z_o is studied and shown in Figure 4.29. This plot shows that for best matching of the impedance a trace thickness of around $200 \mu\text{m}$ is the best choice. This result however does not take into account the real upper tracewidth.

For an estimation of the effect of the upper trace width on the impedance, a constant lower trace width of $213 \mu\text{m}$ and $\epsilon_r = 3.61$ is assumed, while varying the upper trace width from $183 \mu\text{m}$ to $213 \mu\text{m}$. The result is shown in Figure 4.30. With decreasing width the characteristic impedance approaches 50Ω , meaning the matching can potentially become better due to manufacturing.

Differential Pairs on Surface

The geometry of the differential surface is similar to the waveguide type before, with the difference of having a pair of traces instead of one single trace (see page 63). The characteristic differential impedance Z_{diff} of this transmission line type is determined by the trace width w , the trace separation s , the trace-to-ground-separation d , the thickness of the trace t and thickness of the dielectric h .

The parameters t and h have the same value, as for the coplanar waveguide described below. For the other parameters first the following values are assumed:

- Trace width $w = 180 \mu\text{m}$

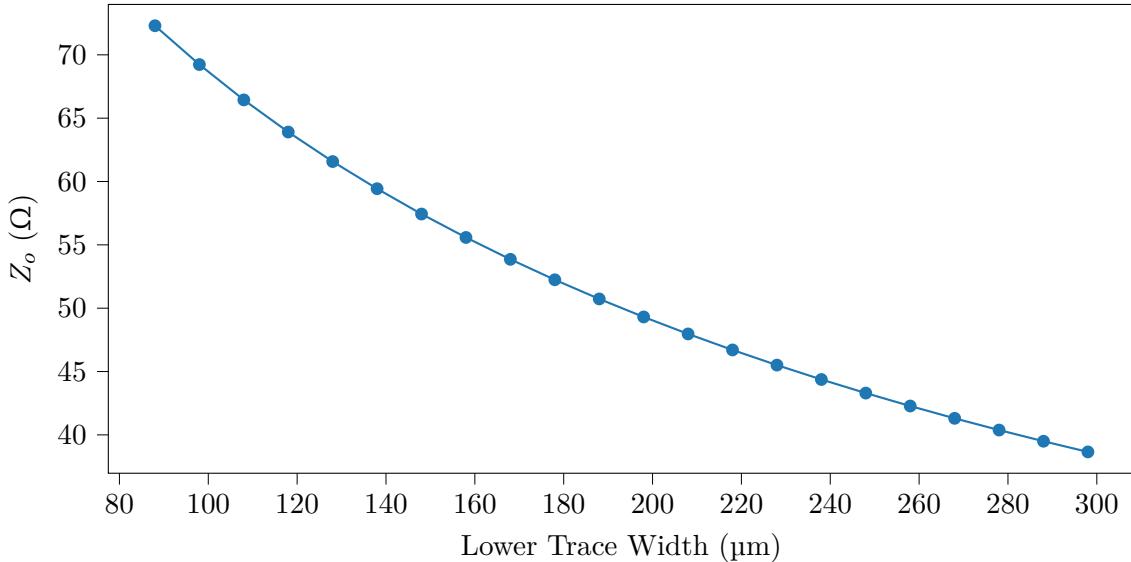


Figure 4.29.: Z_o vs. lower trace thickness a , assuming $\varepsilon_r = 3.61$

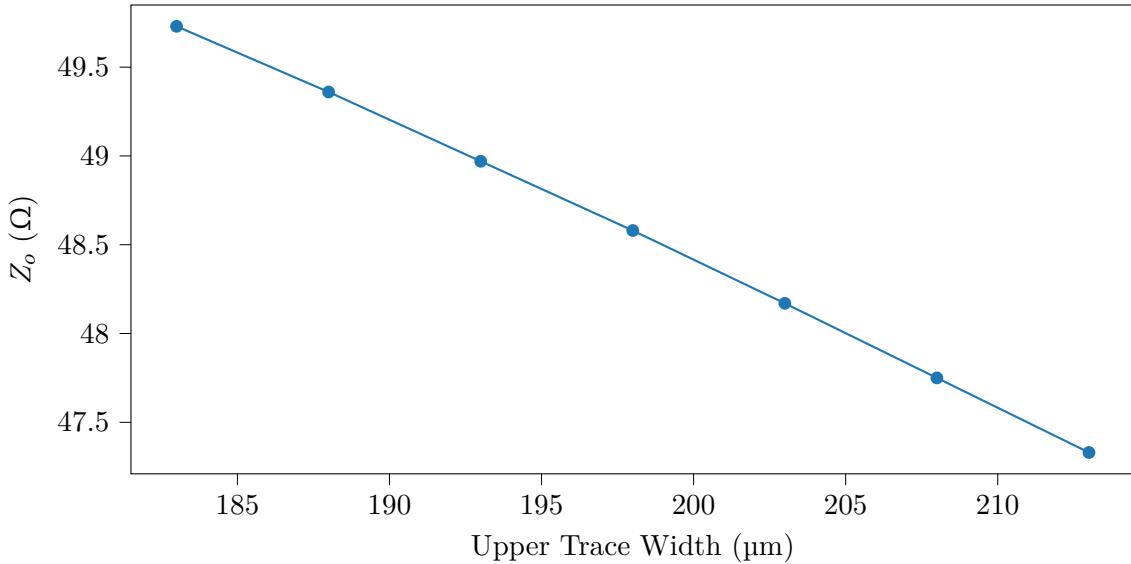


Figure 4.30.: Z_o vs. lower trace thickness

- Trace separation $s = 150 \mu\text{m}$
- Trace-to-ground separation $d = 600 \mu\text{m}$

For these parameters and an $\varepsilon_r = 3.61$ an impedance of 92.35Ω is calculated with the Si9000e tool. This is still inside the tolerance band, but can potentially be improved.

In Figure 4.32 a the characteristic impedance Z_{diff} is plotted against the trace width⁸. The impedance Z_{diff} lies around 100Ω for a trace width $w \approx 155 \mu\text{m}$.

Setting the width to $155 \mu\text{m}$ indeed gives an impedance of $Z_{\text{diff}} = 99.37 \Omega$. The influence of the changing dielectric constant ε_r is studied in this case as well (see Figure 4.33). At the maximal value of $\varepsilon_r = 3.71$, the impedance lies around 98.4Ω corresponding to a change of 0.88 % compared to the value at $\varepsilon_r = 3.61$.

Furthermore, assuming $\varepsilon_r = 3.61$ and a lower trace width $w = 155 \mu\text{m}$, the impedance over a varying upper trace width is plotted in page 64.

⁸Assuming lower and upper trace width are equal.

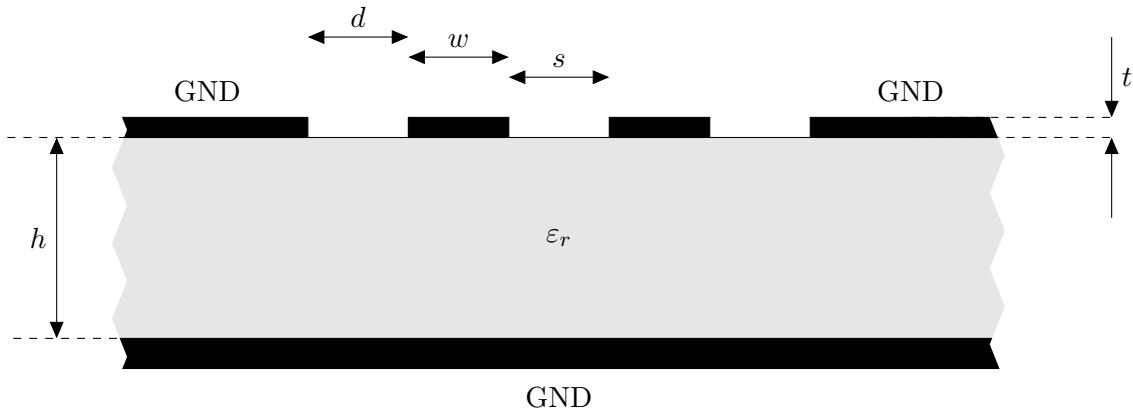


Figure 4.31.: Edge-Coupled Coplanar Waveguide

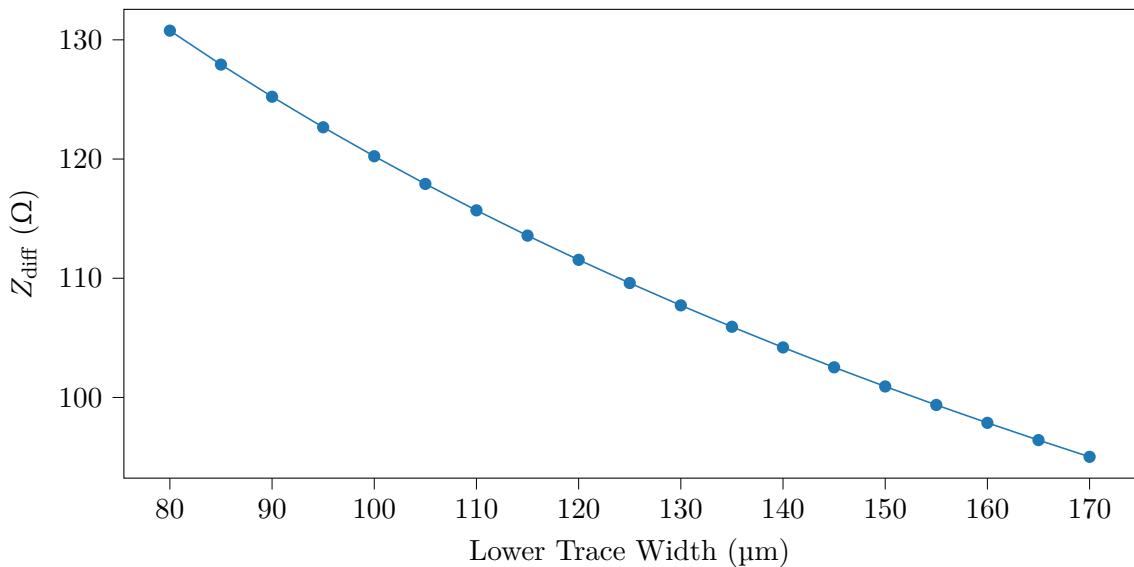


Figure 4.32.: Z_{diff} vs. lower trace width w , assuming $\epsilon_r = 3.61$

Differential Pairs between Layers

The analog signals from the THAs, as well as the clock signals, are propagated through differential pair traces on the inner layers of the PCB. This forms an offset coplanar waveguide as seen in Figure 4.35. The impedance of this waveguide type depends on the trace width w , the trace separation s , the trace-to-ground separation d , the thickness t of the trace, as well as the thickness of the dielectrics h_1 and h_2 and their respective dielectric constant ϵ_1 and ϵ_2 .

The parameters are assumed as

- Trace width $w = 88 \mu\text{m}$
- Trace separation $s = 150 \mu\text{m}$
- Trace-to-ground separation $d = 250 \mu\text{m}$

Thickness of the dielectrics is $h_1 = h_2 = 150 \mu\text{m}$ and the dielectric constant is equal for both ($\epsilon_1 = \epsilon_2 = \epsilon_r = 3.61$.) With these parameters the impedance is calculated as $Z_{\text{diff}} = 90.40 \Omega$. In Figure 4.36 Z_{diff} is plotted against the trace width w (assuming upper

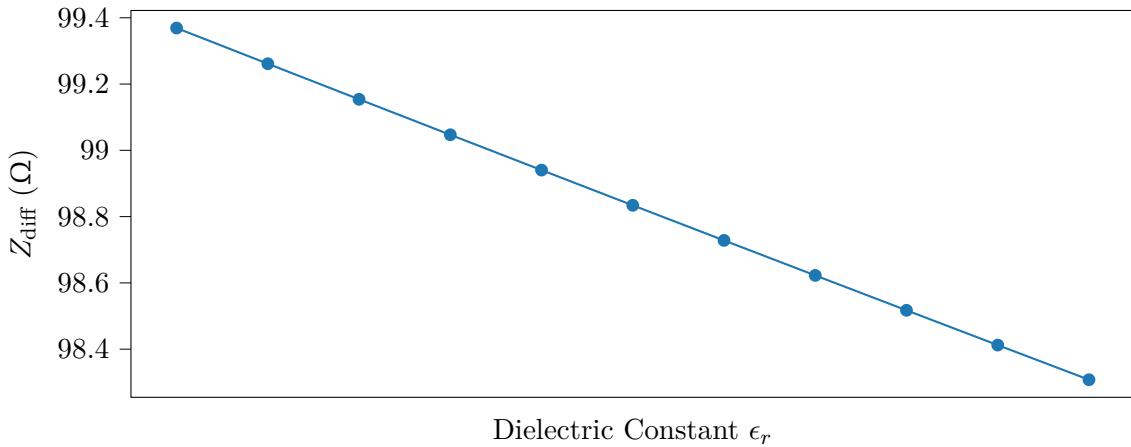


Figure 4.33.: Z_{diff} vs. dielectric constant ϵ_r

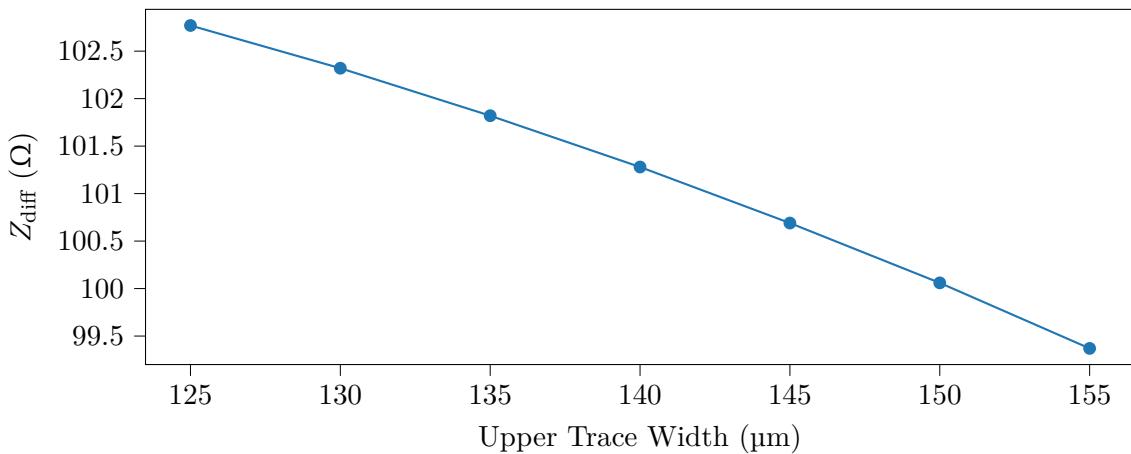


Figure 4.34.: Z_{diff} vs. upper trace width, assuming lower trace width $w = 155 \mu\text{m}$ and $\epsilon_r = 3.61$

trace width equal to w). It can be seen, that in order to improve the impedance, one should decrease the trace width. Due to the manufacturing technology the minimal trace width possible is $88 \mu\text{m}$. Therefore this option is not feasible.

Keeping the trace width constant at $w = 88 \mu\text{m}$ the trace separation could also be changed. Figure 4.37 shows Z_{diff} plotted against the trace separation s . It can be seen that Z_{diff} does not change significantly over a large range of s . For a trace separation of around $300 \mu\text{m}$ (more than 3 times larger than the trace width itself) $Z_{\text{diff}} \approx 94 \Omega$ and not significantly improved. Taking this into consideration, as well as the space on the board, the parameters are left as is.

The influence of the dielectric constant ϵ_r is shown in Figure 4.38. Z_{diff} decreases with higher value of ϵ_r and even get below 90Ω , exceeding the 10 % tolerance. However, the upper trace width has also to be taken into account, which is in any case smaller than the lower trace width due to the etching process during manufacturing. As Figure 4.39 shows, the impedance is potentially higher than calculated by assuming both width equal. Therefor the impedance can still be regarded as falling into the tolerance band.

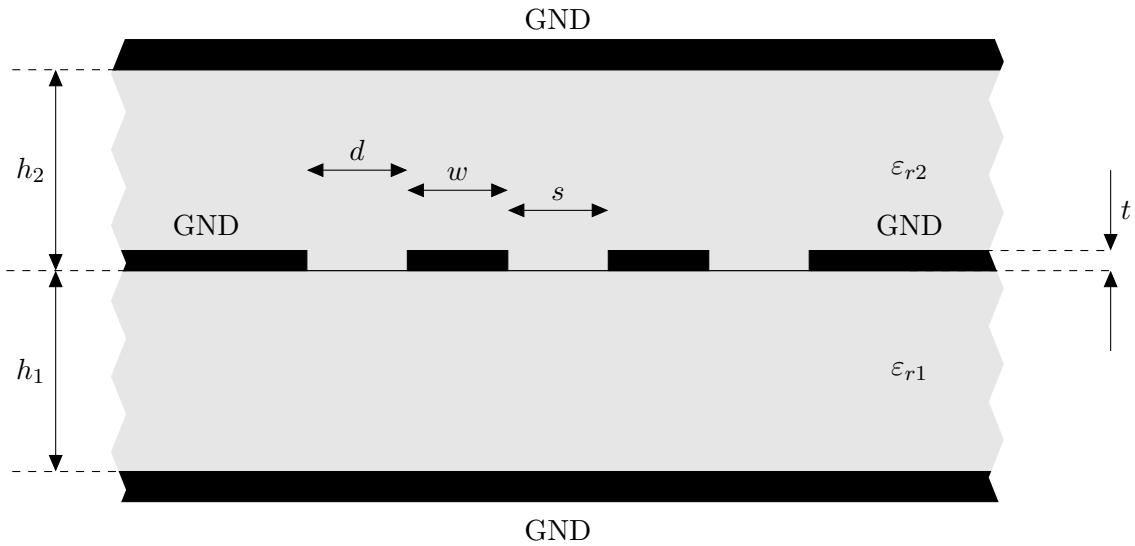


Figure 4.35.: Offset Differential Coplanar waveguide

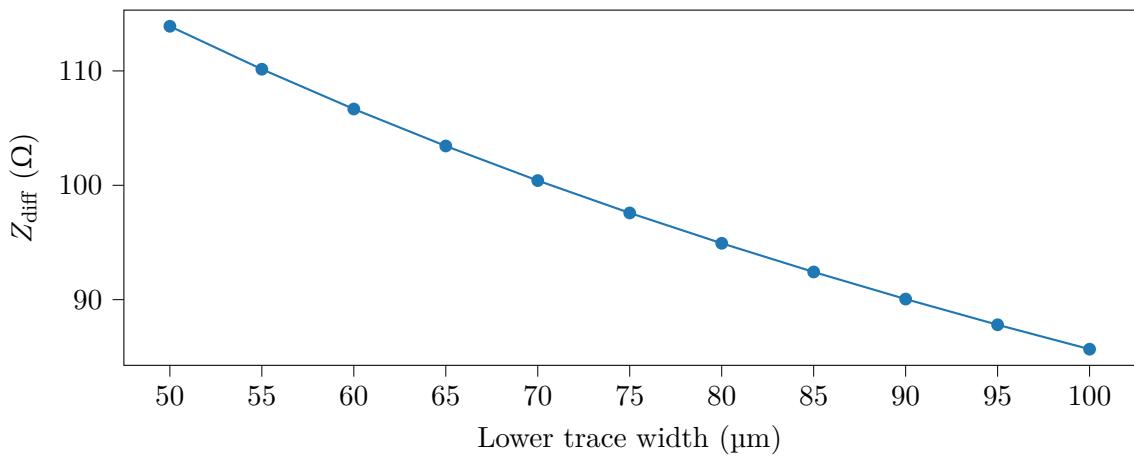


Figure 4.36.: Z_{diff} vs. lower trace width w , assuming upper trace width equals to w

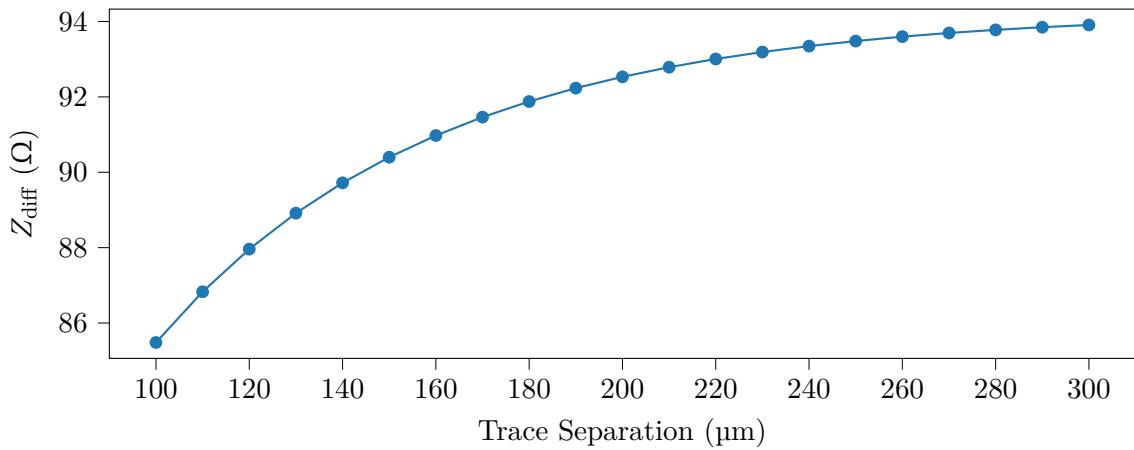


Figure 4.37.: Z_{diff} vs. trace separation s , assuming trace width $w = 88 \mu\text{m}$

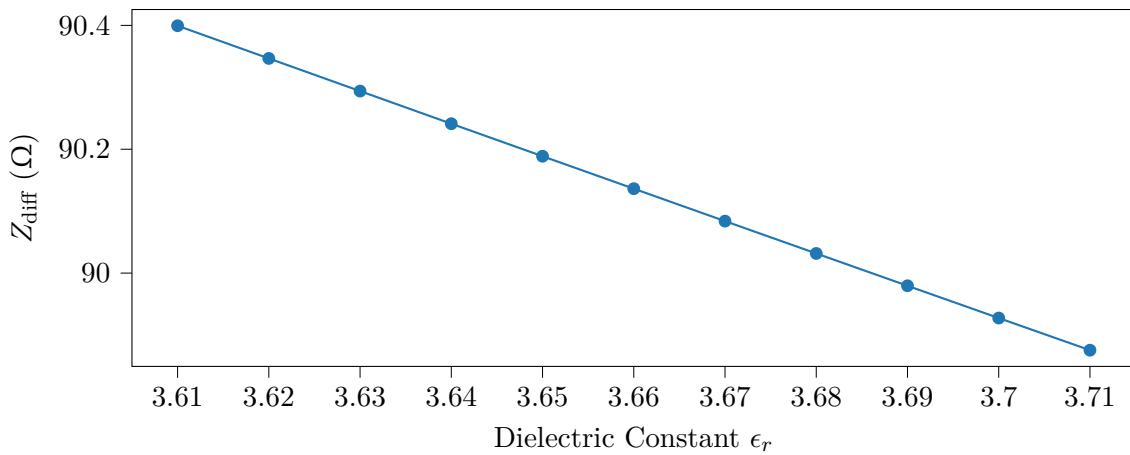


Figure 4.38.: Z_{diff} vs. dielectric constant ϵ_r , assuming lower trace width $w = 88 \mu\text{m}$ and $\epsilon_r = 3.61$

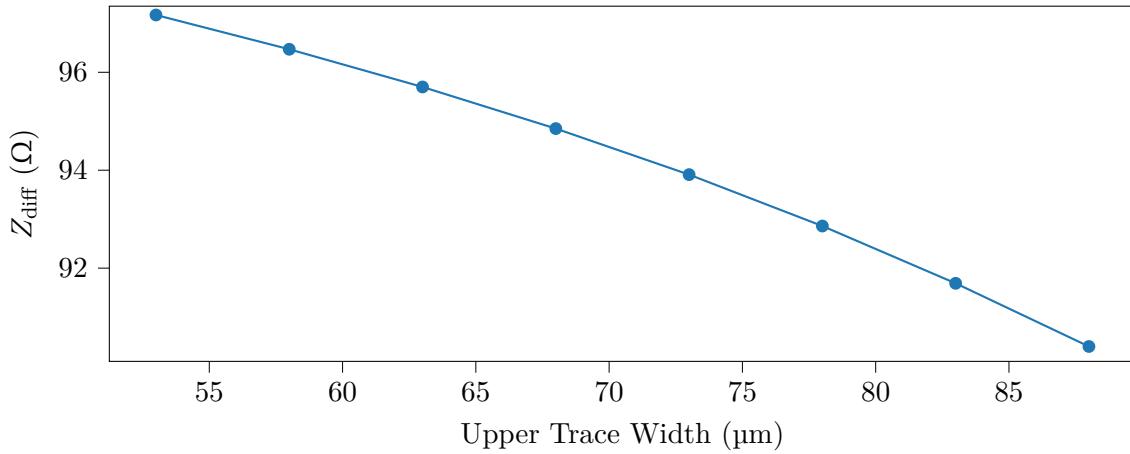


Figure 4.39.: Z_{diff} vs. upper trace width, assuming lower trace width $w = 88 \mu\text{m}$ and $\epsilon_r = 3.61$

4.2.3. Component Placement and Routing

For the placement and routing of the components many steps need to be considered.

At the beginning, the separation of the analog and digital grounds has to be taken care of. Due to the complexity of the board, the respective grounds need to cover the whole plane. Therefore, in order to guarantee a full separation without any interference between the two parts, the PCB is split into two parts. The topside part is dedicated to the digital components and the routing of digital signal. These layers cover the clocking distribution as well as the slow control signal paths leading from the FPGA to the respective components. The bottomside part of the PCB is dedicated to the analog components and integrates the analog signal paths coming from the THAs.

Closely linked to the structuring of the overall PCB is the number of stacked-up metal layers used. In order to integrate all necessary signals and power planes, 16 layers in total are used. The topside 8 layers are therefore used for the digital part, the other 8 for the analog. For shielding purposes and to guarantee a small as possible signal return path for the transmission lines, layers carrying signal paths are “sandwiched” between two ground layers.

Some signals need to be routed from top layer to bottom and vice versa by through vias. In order to maintain the separation between analog and digital parts in such cases, a sufficient isolation between the via and the surrounding (ground) plane has to be ensured.

At some point, the analog and digital grounds need to be connected together. As mentioned in the section about the schematic capture, these connections are deployed by placing ferrite beads at each THA connecting analog and digital grounds. This way, any noise coming from the digital ground is compensated for. RF filtering at the THA is placed in the same manner, in order to mitigate any noise which could interfere with the sensitive analog signal.

Sensitive, i.e. analog components, should be placed first in order to minimize the routing paths, therefore reducing additional inductance and possible interference due to longer traces. Also the transmission lines carrying sensitive analog signals should be routed first, in order to define the further routing of other signal paths, e.g. slow control signals. These transmission lines should be separated from digital signal paths as much as possible in order to reduce cross-talk between the lines.

Routing the transmission lines should be done with time skew control in mind. This is especially necessary for the outputs leading from THAs to the RFMC connector. Due to the asymmetric position of the connector on the board, if the THA component outputs would be connected directly to the connector, the signal paths would vary significantly between components. This would introduce a significant time skew between the lines. Therefore, to account for this problem, signal paths coming from the closest THAs need to be made longer. This is achieved by routing the traces in patterns called “accordions”, which allow for prolonging the trace length in a compact way. An example for such accordions is shown in Figure 4.40.

To achieve high signal integrity, “via-fences” are placed next to the traces carrying analog signals. These consist of via holes connected to analog ground, placed close enough together to form a barrier for electromagnetic wave propagation, in order to shield, or isolate, the traces from surrounding components and from other traces.

Slow control signal paths are routed afterwards, avoiding crossing the area of the analog signal paths.

Component placement:

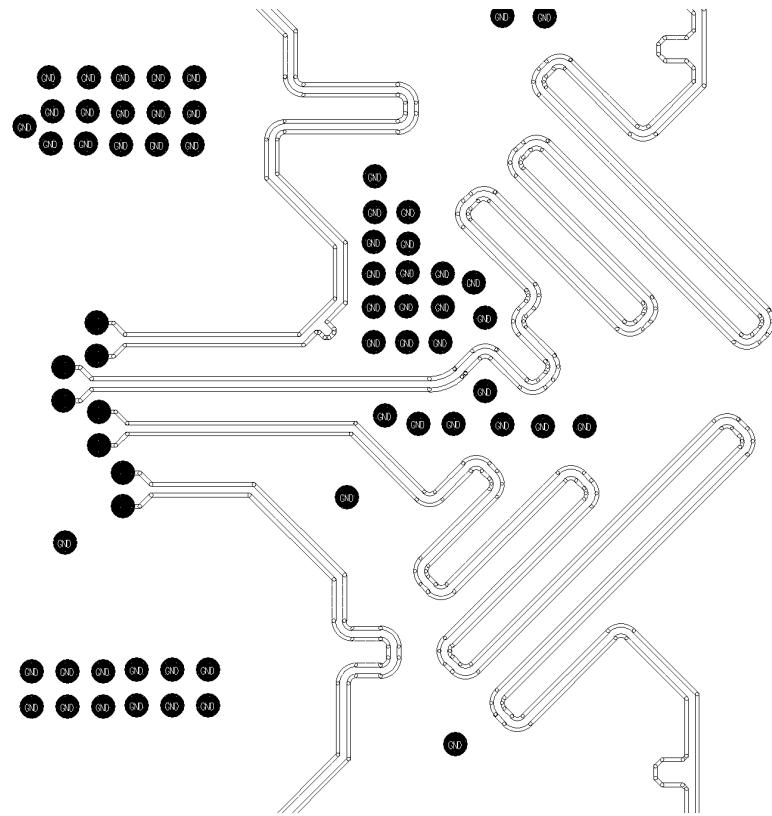


Figure 4.40.: Example for trace accordions which are used to enlarge the trace length when little space is available

- Track-and-Hold -> stitching vias for shielding
- Delay Chips
- Clock distribution
- Power distribution

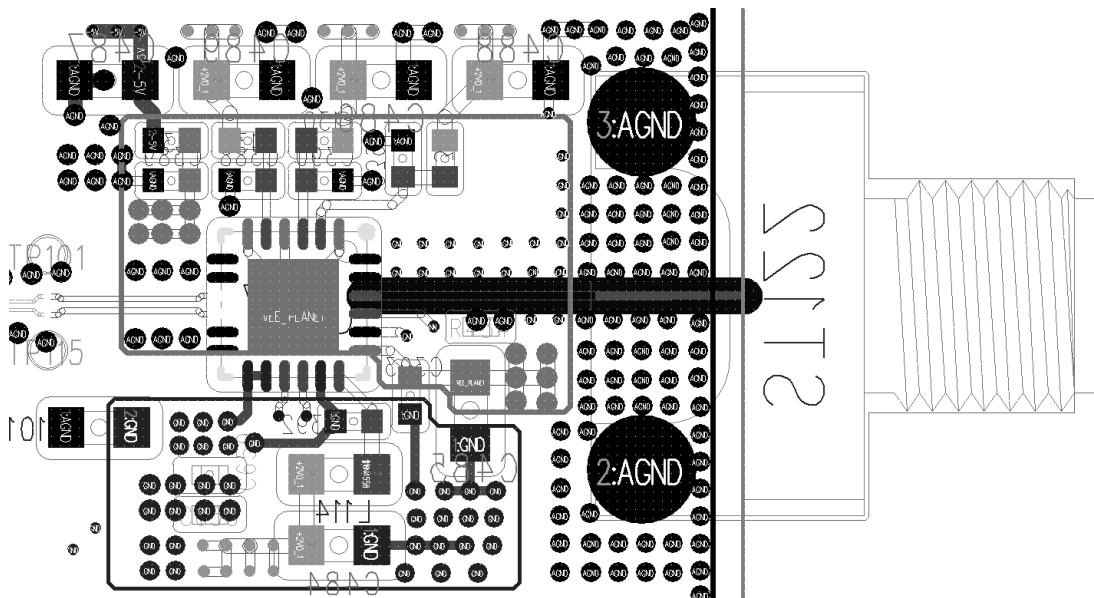


Figure 4.41.: Low noise design shown with the example of a THA

5. Back-End Readout Card and System Integration

The back-end readout card for the system under development, the Zynq UltraScale+ RFSoC ZCU216 Evaluation Card, was chosen taking into consideration the points described in ???. In this section, the overall architecture and features of the card are presented. A possibility for evaluation of the card is also demonstrated. At last, a design for the read-out firmware is proposed.

5.1. Xilinx ZCU216 Evaluation Card

The ZCU216 Evaluation Card is equipped with the ZU49DR Zynq Ultrascale+ RFSoC RFSoC (Gen 3). It allows for quick evaluation of the on-chip RF data converters and quick prototyping of different user-defined systems.

The features which are important for the designed read-out system are listed in the following:

- Sixteen 14-bit, 2.5 GS/s RF-ADC
- Sixteen 14-bit, 10 GS/s RF-DAC
- I/O expansion options: FPGA Mezzanine Card (FMC+) interfaces, RFMC 2.0 interfaces, and Pmod (peripheral module) connections
- DDR4 Dual In-Line Memory Module (DIMM) – 4 GB, 64-bit, 2.666 MT/s, attached to the programmable logic (PL)
- DDR4 Small Outline Dual In-Line Memory Module (SODIMM) – 4 GB, 64-bit, 2.400 MT/s, attached to the processing system (PS)
- High-speed I/Os: 2x2 Small Form-Factor Pluggable (SFP)/SFP+/zSFP+/SFP28 modules
- Breakout cards for evaluation of the ADC and DAC performance, together with a clock add-on card for internal/external reference clocking

Other peripheral connections and features are shown in the topview of the board in Figure 5.1.

ZU49DR RFSoC

Together with an UltraScale+ programmable logic, the ZU49DR RFSoC integrates an Arm®Cortex™-A53 PS. This PS contains a 64-bit quad-core Arm®Cortex™-A53, serving as Application Processing Unit (APU), and a dual core Arm Cortex-R5F, serving as Real-Time Processing Unit (RPU). Furthermore, the system integrates RF data converters, allowing to use the system for RF applications. At the moment of writing, this is the industry's only single-chip, adaptable radio platform [?]. This setup allows to Figure 5.2 shows the general block diagram of the RFSoC.

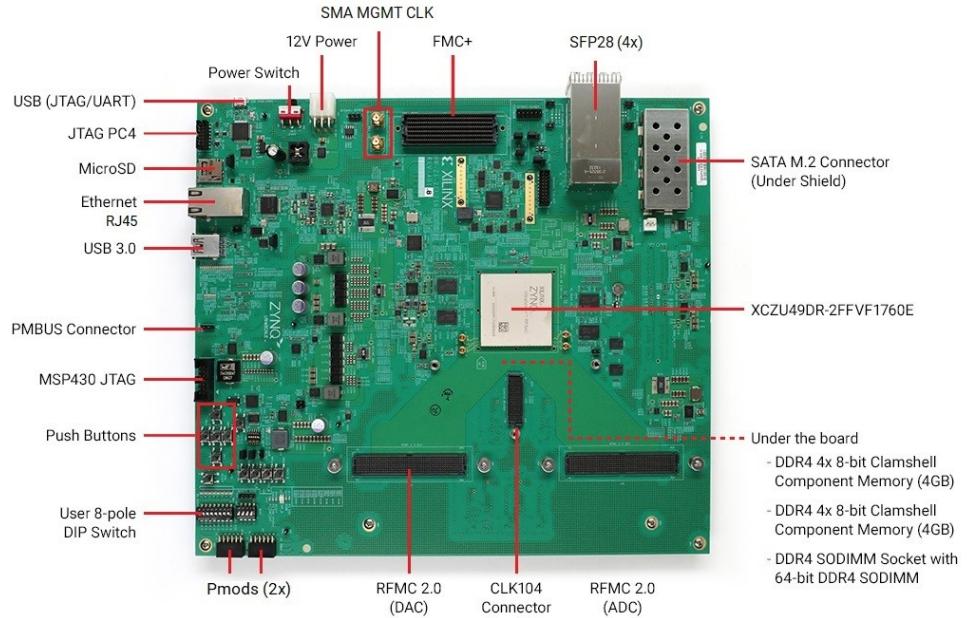


Figure 5.1.: Topview of ZCU216 evaluation board with labeled components

Evaluation Tool

In order to provide a quick evaluation of data converter performance an evaluation framework from Xilinx can be deployed on the card (see Figure 5.3).

It enables control of the ZCU216 Intellectual Property (IPs) and the associated designs from a host PC. With the tool, different RF configurations can be explored, RF data can be generated and captured and different RF metrics (see subsection 2.3.1) can be observed. For this, the breakout card provided with the board, or any other user-designed board, needs to be attached to the evaluation card via the RFMC connectors.

The RF DC (Data Converter) Evaluation Tool consists of two parts:

- Hardware design on the PL/FPGA, in order to implement the configuration and data generation/capture of the data converters. It is built around the RF Data Converter IP Core from Xilinx, described in section 5.2.
- Software design on the PS and host PC, in order to control the hardware implemented on the FPGA. On the PS a Linux application receives commands from the host PC Graphical User Interface (GUI) over Ethernet. Based on the commands, it performs the according action, e.g. setting the sampling clock or enabling/disabling data converter channels. The GUI provides a convenient possibility to configure the data converters. Data generation via DACs can be started, as well as data capture with the ADCs. Furthermore, the tool provides methods to quickly characterize the performance of the data converters (see Figure 5.4) from which the performance of the plugged break-out board can be derived.

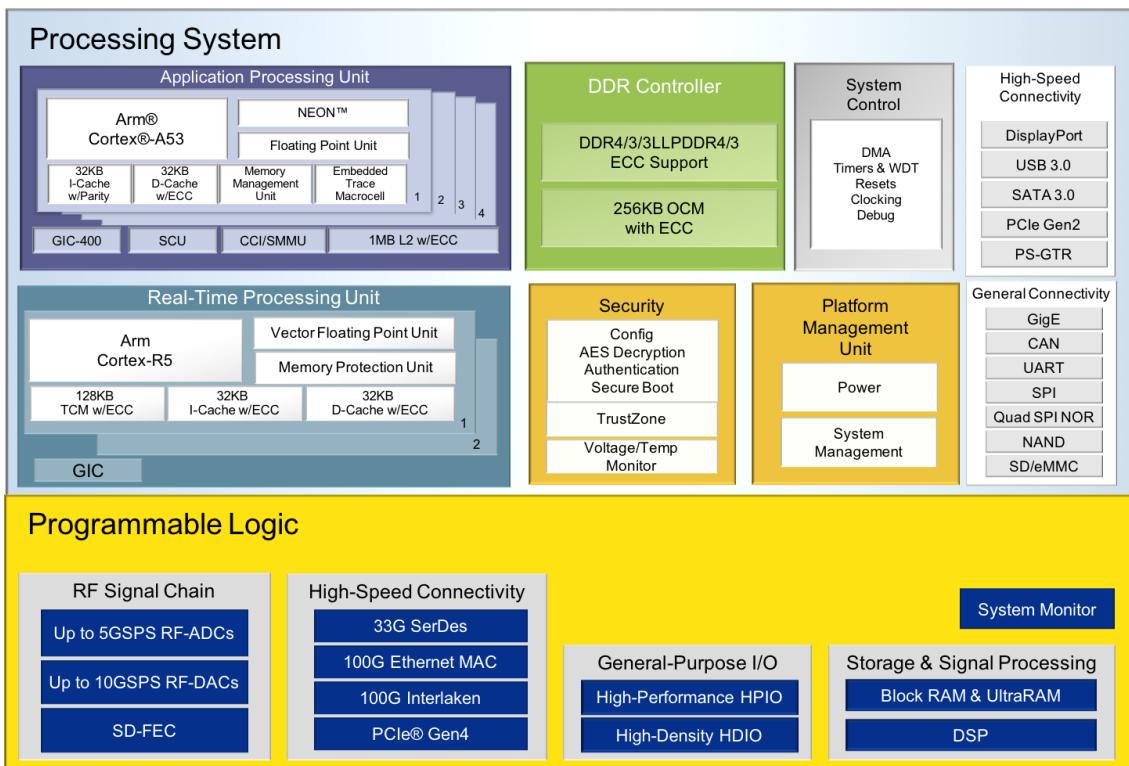


Figure 5.2.: Zynq Ultrascale+ RFSoC block diagram, showing in detail the different components of the PS and PL

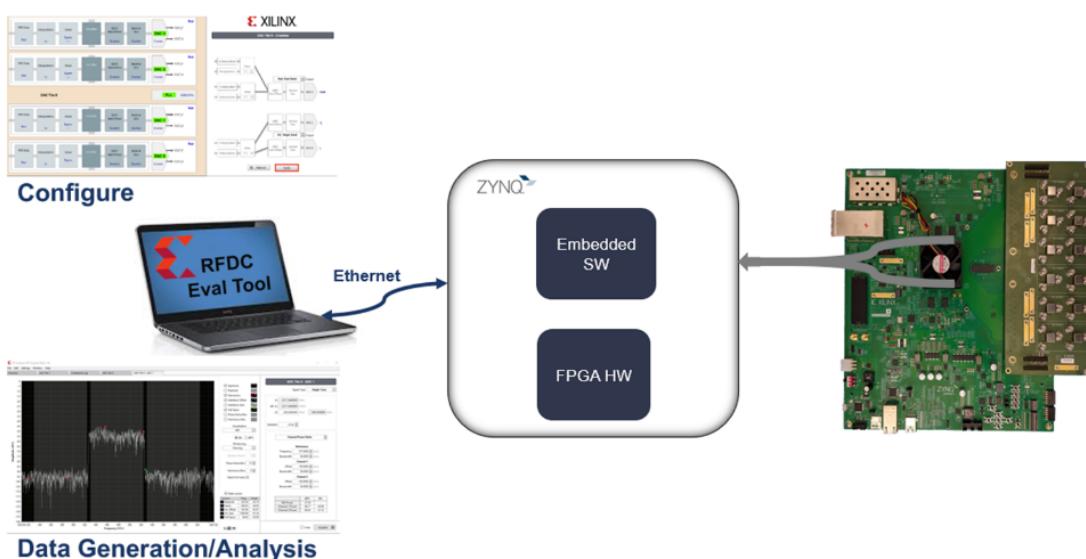


Figure 5.3.: Concept of the Evaluation Tool for the ZCU216, showing the interactions between the host PC and the board

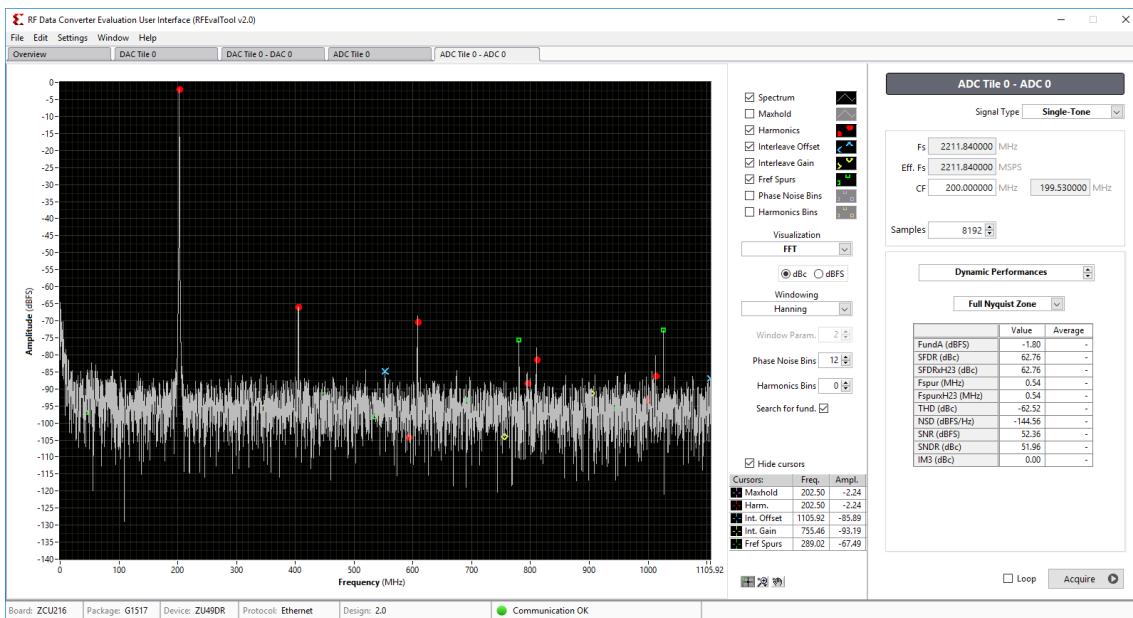


Figure 5.4.: GUI of the RF Data Converter Evaluation Tool

5.2. Firmware

Similar to the evaluation tool, the firmware for the readout system should contain a software part on the PS which allows for control and configuration of the data converters. The hardware design part should implement the interface to the sampling board. This means it should provide the necessary interfaces to configure the data converters on the readout card, as well as slow control to the delay chips and PLLs on the sampling board (i.e. through SDI and SPI). Furthermore, the high-speed data-throughput interface should be implemented.

Figure 5.5 shows the general schematic of the firmware on the readout card.

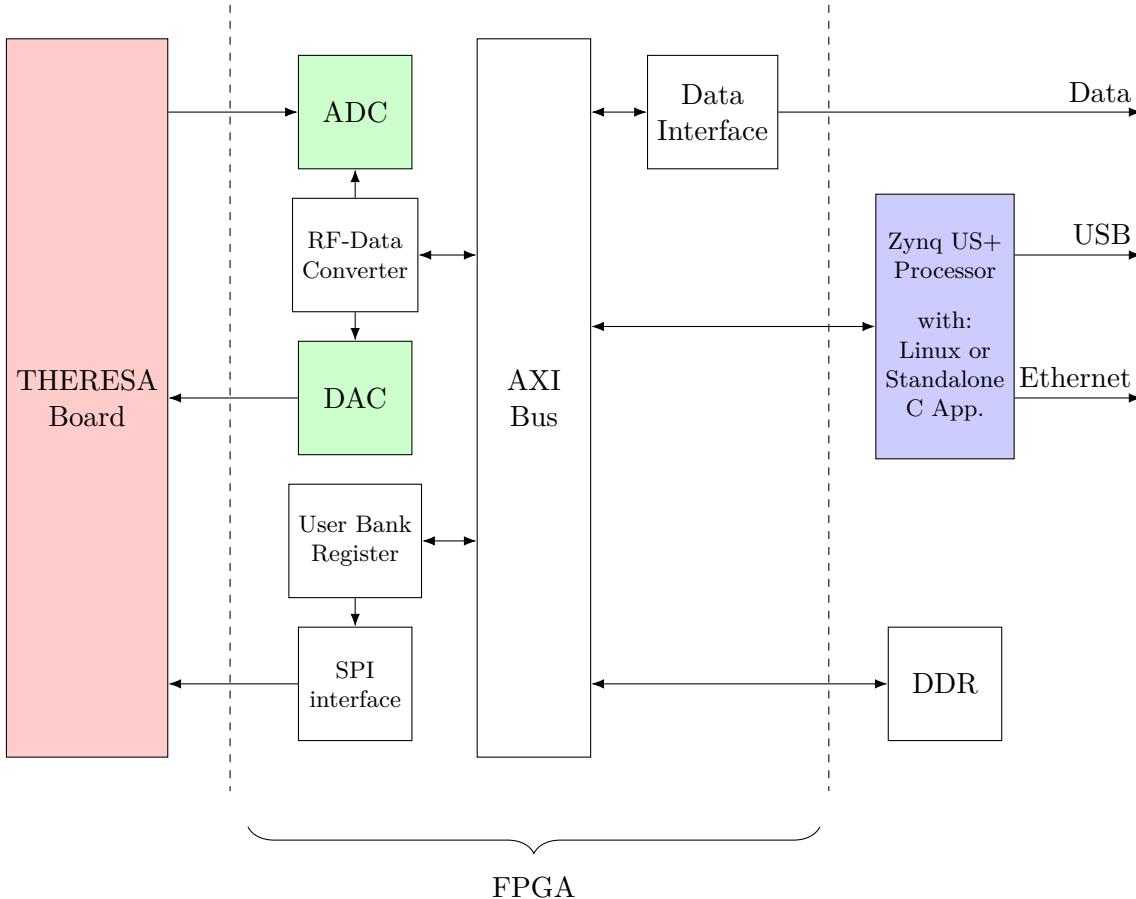


Figure 5.5.: Schematic of the firmware and processing unit on the readout card

The sampled data is propagated from the sampling card to the ADCs inside the FPGA. This data is written in the DDR, from which it is accessible by the data interface, which is responsible for the high-speed data transfer to the following processing node in the DAQ system. A built-in test-loop is implemented with an integrated DAC, in order to produce test signals which are propagated to the sampling board. Configuration of the data converters is done via the Xilinx “RF Data Converter” which is described below.

Components on the sampling board, such as the delay chips, are controlled via SPI interface. This interface is connected to a user bank register, where the user can specify e.g. the desired delay values.

The PS communicates with the PL via Advanced eXtensible Interface (AXI). AXI is a parallel, synchronous, multi-master, multi-slave communication interface. On the PS an operating system, e.g. Linux, or a standalone C application, can run implementing functionalities for the user to be able to control the overall system. Access to the PS is provided by for example via Ethernet or USB.

5.2.1. Programmable Logic - Hardware design

For synthesis and analysis of the Hardware Description Language (HDL) design for the PL, the Xilinx Vivado Integrated Design Environment (IDE) is used. To configure and control the data converters, the RF Data Converter IP Core from Xilinx is used. In order to program the components on the sampling board, appropriate SPI interfaces are implemented, respecting the recommendations in the data sheet.

RF Data Converter IP Core

In order to configure the data converters, the HDL design should be based around the Xilinx RF Data Converter IP Core.

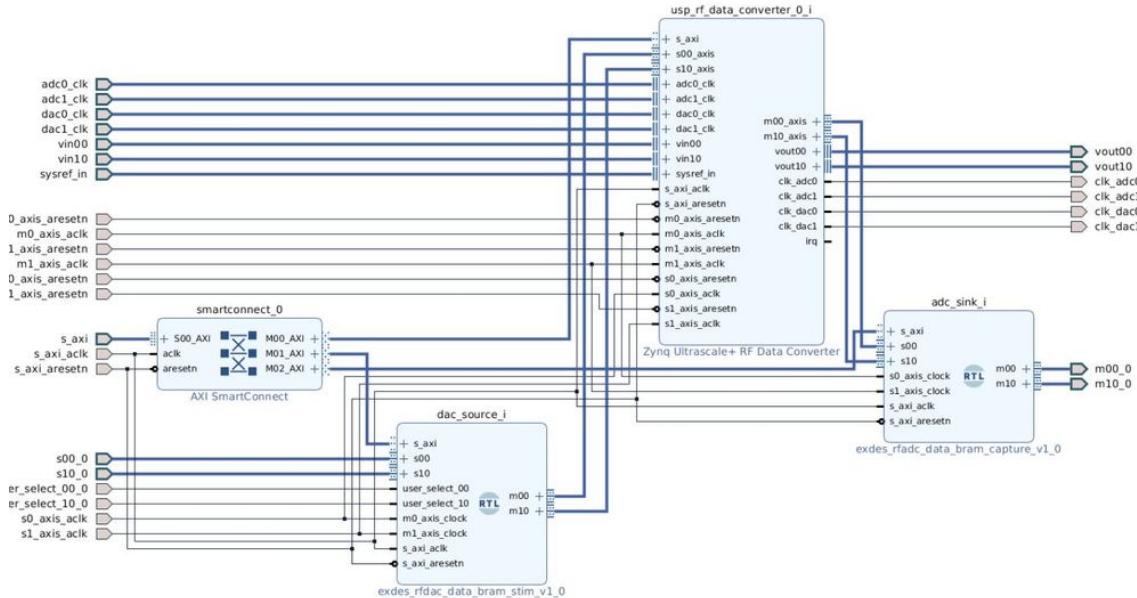


Figure 5.6.: Simple example design with the RF Data Converter

Slow Control

Components on the sampling board are controlled via a slow control interface. As example, the on-board delay chips are controlled via a SDI interface, consisting of four signals: EN, SDIN, SCLK and SLOAD. In order to program a certain delay, the timing diagram shown in Figure 5.7 has to be respected. The delay chip only accepts data when the EN input is HIGH. Therefore, this signal has to remain high in the course of the whole data transaction. The data pin, SDIN, which is clocked in by the SCLK signal, has to respect the following structure: delay channel select bit, mode select bit, followed by 9 data bits, which represent the desired delay value in binary format. At the end, SLOAD has to be asserted HIGH, in order to signal the chip to load the received data into the chip's internal register. The HDL implementation for this interface is listed in the appendix. In a similar way, the interface for other components is to be implemented.

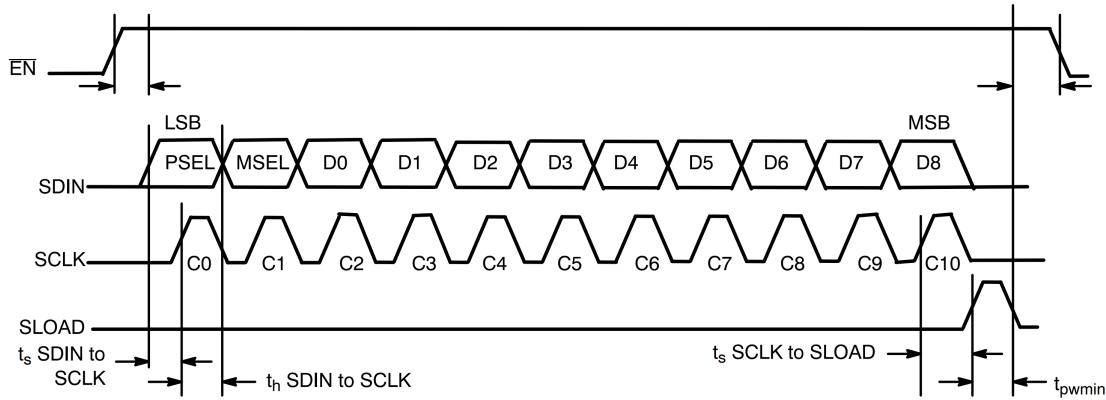


Figure 5.7.: SDI Timing diagram for the NB6L295 delay chip [?]

RDMA over Converged Ethernet (RoCE)

Remote Direct Memory Access (RDMA) is a direct memory access from the memory of one computer into that of another without involving either one's operating system. This permits high-throughput, low-latency networking, which is especially useful in massively parallel computer clusters.

RoCE is a network protocol defined in the InfiniBand Trade Association (IBTA) standard, allowing RDMA over converged Ethernet network. Shortly, it can be regarded as the application of RDMA technology in hyper-converged data centers, cloud, storage, and virtualized environments. It possesses all the benefits of RDMA technology and the familiarity of Ethernet.

The ERNIC (Xilinx Embedded RDMA enabled NIC) IP provides an Initiator and Target implementation of RDMA over Converged Ethernet (RoCE v2) enabled NIC functionality. This IP is specifically designed for embedded applications that require reliable transmission over Ethernet networks.

5.2.2. Processing Unit - Software Design

In order for the user to be able to control the whole system, some kind of application has to be implemented for control of the hardware design. For this the PS side of the RFSoC is used. The Arm Cortex processor allows to implement standalone C application, as well as hosting an operating system like Linux. Providing a number of peripheral connections, like RJ45 or USB, the PS can easily be accessed by the user from another host computer, given the necessary drivers and protocols are implemented. The aforementioned evaluation tool for example uses the Linux application *rftool* in order to receive commands and perform the according action.

6. Conclusion and Outlook

7. Conclusion and Outlook

Analysis of events occurring in the range of femtoseconds is desired in many scientific experiments. The high temporal resolution needed for measuring such events imposes a great technological challenge for Data Acquisition Systems (DAQs) and Analog-To-Digital-Converters (ADCs). In order to relax the requirements on the acquisition systems, the so-called optical time-stretch technique is used to stretch the analog input signal in time. In this way, data converters at relatively moderate sample rate can be used. Measuring the signal with commercial DAQs, such as real-time oscilloscope, still poses another challenge. Due to the limited acquisition time windows of such systems, continuous measurements at high sampling rate over long time is not possible. In applications, where measurements of long-term evolution of the ultra-fast events is desired, this is a major limitation. Therefore new concepts of DAQ based on the time-stretch method need to be considered in order to overcome this limitation.

In this thesis, a first demonstrator of such a new DAQ system based on the photonic time-stretch method was developed. The system consists of a high bandwidth front-end sampling card, mounted on a back-end card integrating a new generation of Radio-Frequency System-On-Chip (RFSoC) for readout of the acquired samples. The name given to the system is Terahertz Readout Sampling (THERESA).

The front-end sampling card integrates 16 sampling channels, each containing a Track-And-Hold-Amplifier (THA) with individually programmable delay in sampling time. The design of the board allows it to be used in two different modes: with and without the time-stretch setup. In single-channel mode one detector is connected to one sampling channel, therefore allowing sampling of up to 16 detectors at the same time with one sampling point per channel. In the second mode, several channels are connected to one detector via power splitter, therefore allowing multiple sampling points for one detector/per channel by setting the delay times accordingly.

High-speed ADCs, integrated in the RFSoC, with 14-bit resolution and a sample rate of up to 2.5 GS/s allow continuous sampling of the signal with high time resolution. Using the time-interleaving technique for all sixteen ADCs results in an overall maximal achievable sample rate of 40 GS/s possible. When using in combination with the time-stretch technique and considering typical stretch-factors, these 11 ps are translated into a range of femtoseconds in the original signal.

The sampling card was furthermore designed to fully exploit all the features of the RFSoC, which integrates a processing unit together with a Field Programmable Gate Array (FPGA). An evaluation tool framework is provided for the selected read-out card, allowing for on-board data generation and capture. This tool was also evaluated; allowing for quick set-up and measurement of key data converter characteristics (Signal-to-Noise-and-Distortion Ratio (SINAD), Spurious-Free Dynamic Range (SFDR), ...) it provides an invaluable tool in order to get a first impression of the performance of the sampling card.

The on-chip FPGA provides the possibility to flexibly adjust the firmware to user needs. Slow-control implemented in the FPGA takes care of programming the components on the sampling card, such as the delay chips. High-speed interfaces, allowing speeds over 100 Gb/s, are a crucial component for the high throughput of the large amount of data generated by the data converters; with the given resolution and max. sample rate this touches the range of TB/s.

The design of the sampling card was approved and the card has been deployed in production. Quick characterization of the card is possible due to the tool provided for the readout-card and can be carried out using the methods described in subsection 2.3.1. THERESA will then be commissioned and taken into operation, improving the research in various scientific fields, especially beam diagnostics at e.g. Karlsruhe Research Accelerator (KARA). There it can be used for studying Coherent Synchrotron Radiation (CSR), in the far-field and near-field electro-optic setup, for study of fast laser dynamics and many other applications. The selected FPGA is suitable for deploying Artificial Intelligence applications (i.e. Reinforcement Learning). Therefore the system can also be used for interfacing with the Bunch-By-Bunch feedback at KARA. In the context of the ULTRASYNC project, funded by ANR-DFG, THERESA can be used in order to study the control of electron bunches in accelerators at KARA and SOLEIL, therefore being an important step towards new usable Terahertz (THz) sources.

Acknowledgments

Appendix

A. Characteristic Impedance Of Coplanar Waveguides

Edge-Coupled Coplanar Waveguide

Characteristic impedance[?, p197-198]:

$$Z_{0,o} = \frac{\eta_0}{\sqrt{\epsilon_{\text{eff},o}}} \left(\frac{1.0}{2.0 \frac{K(k_o)}{K'(k_o)} + \frac{K(\beta_1)}{K'(\beta_1)}} \right) \quad (\text{A.1})$$

$$Z_{0,e} = \frac{\eta_0}{\sqrt{\epsilon_{\text{eff},e}}} \left(\frac{1.0}{2.0 \frac{K(k_e)}{K'(k_e)} + \frac{K(\beta_1 k_1)}{K'(\beta_1 k_1)}} \right) \quad (\text{A.2})$$

$$\epsilon_{\text{eff},o} = \frac{2.0 \epsilon_r \frac{K(k_o)}{K'(k_o)} + \frac{K(\beta_1)}{K'(\beta_1)}}{2.0 \frac{K(k_o)}{K'(k_o)} + \frac{K(\beta_1)}{K'(\beta_1)}} \quad (\text{A.3})$$

$$\epsilon_{\text{eff},e} = \frac{2.0 \epsilon_r \frac{K(k_e)}{K'(k_e)} + \frac{K(\beta_1 k_1)}{K'(\beta_1 k_1)}}{2.0 \frac{K(k_e)}{K'(k_e)} + \frac{K(\beta_1 k_1)}{K'(\beta_1 k_1)}} \quad (\text{A.4})$$

with

$$k_o = \Lambda \frac{-\sqrt{\Lambda^2 - t_c^2} + \sqrt{\Lambda^2 - t_B^2}}{t_B \sqrt{\Lambda^2 - t_c^2} + t_c \sqrt{\Lambda^2 - t_B^2}} \quad (\text{A.5})$$

$$k_e = \Lambda' \frac{-\sqrt{\Lambda'^2 - t_c'^2} + \sqrt{\Lambda'^2 - t_B'^2}}{t_B' \sqrt{\Lambda'^2 - t_c'^2} + t_c' \sqrt{\Lambda'^2 - t_B'^2}} \quad (\text{A.6})$$

$$\Lambda = \frac{\sinh^2 \left(\frac{\pi(s/2.0+w+d)}{2.0h} \right)}{2} \quad (\text{A.7})$$

$$t_c = \sinh^2 \left(\frac{\pi(s/2.0+w)}{2.0h} \right) - \Lambda \quad (\text{A.8})$$

$$t_B = \sinh^2 \left(\frac{\pi s}{4.0h} \right) - \Lambda \quad (\text{A.9})$$

$$\Lambda' = \frac{\cosh^2 \left(\frac{\pi(s/2.0+w+d)}{2.0h} \right)}{2} \quad (\text{A.10})$$

$$t'_c = \sinh^2 \left(\frac{\pi(s/2.0 + w)}{2.0h} \right) - \Lambda' + 1.0 \quad (\text{A.11})$$

$$t'_B = \sinh^2 \left(\frac{\pi s}{4.0h} \right) - \Lambda + 1.0 \quad (\text{A.12})$$

The parameters have to be chosen according to

$$s + 2.0w + 2.0d \leq h \quad (\text{A.13})$$

to guarantee coplanar propagation. [?]

Surface Coplanar Waveguide with Ground

The characteristic impedance of a coplanar waveguide is given as (see [?])

$$Z_0 = \frac{60.0\pi}{\sqrt{\epsilon_{\text{eff}}}} \frac{1.0}{K(k) + K(k_1')} \cdot \quad (\text{A.14})$$

It comprises of the following components, with $K(k)$ being an elliptical integral of the first kind (see also [?, p. 430]):

$$k = a/b \quad (\text{A.15})$$

$$k' = \sqrt{1.0 - k^2} \quad (\text{A.16})$$

$$k_1 = \frac{\tanh(\frac{\pi a}{4.0h})}{\tanh(\frac{\pi b}{4.0h})} \quad (\text{A.17})$$

$$k_1' = \sqrt{1.0 - k_1^2} \quad (\text{A.18})$$

$$\epsilon_{\text{eff}} = \frac{1.0 + \epsilon_r \frac{K(k')}{K(k)} \frac{K(k_1)}{K(k_1')}}{1.0 + \frac{K(k')}{K(k)} \frac{K(k_1)}{K(k_1')}} \quad (\text{A.19})$$

B. Code

```
'timescale 1ns / 1ps

module SDI_Delay_NB6L295(
    input [10:0]           In_1, In_2, In_3, In_4, In_5, In_6, In_7, In_8, // 
                           data for respective delay chips
    input                  Clk,
    input                  Reset,
    output reg [7:0]        EN, // enable signal for delay chips, active LOW
    output reg              SDIN, // configuration data
    output reg              SLOAD, // signals delay chip to load previously sent
                           data
    output                 SCLK // clock for serial communication with delay chips
);

reg                      start_clk;
assign SCLK = start_clk & (!Clk);

reg [21:0]             In_1_reg, In_2_reg, In_3_reg, In_4_reg, In_5_reg,
                       In_6_reg, In_7_reg, In_8_reg; // registers to intermediately store the
                           inputs
```

```

reg [7:0] select; // register used by Priority Encoder to detect
// which input changed

parameter DATA_SHIFT_WIDTH = 11; // number of bits to be shifted
// during transmission, 1 Data word = 11 bits
reg [4:0] clk_cnt;

reg [DATA_SHIFT_WIDTH-1:0] Data_reg; // register for storing data for
state machine

reg data_start; // signal for state machine to start sending
reg is_finished dataSent; // flags if transmission for one delay chip

parameter dly = 1; // delay control

reg delayReady;

always @ (posedge Clk)
begin
  if (select == 'd0) delayReady <= #dly 'b1;
  else delayReady <= #dly 'b0;
end

// Priority Encoder
// Check if any input has changed, select which data should be sent
// accordingly
always @ (posedge Clk)
begin
  if (Reset)
  begin
    In_1_reg <= #dly 'd0;
    In_2_reg <= #dly 'd0;
    In_3_reg <= #dly 'd0;
    In_4_reg <= #dly 'd0;
    In_5_reg <= #dly 'd0;
    In_6_reg <= #dly 'd0;
    In_7_reg <= #dly 'd0;
    In_8_reg <= #dly 'd0;
    Data_reg <= #dly 'd0;

    select <= #dly 'd0;

    start <= #dly 1'b0;;
  end
  else
  begin
    if (~start & delayReady)
    begin
      select[7] <= #dly In_1_reg != In_1;
      select[6] <= #dly In_2_reg != In_2;
      select[5] <= #dly In_3_reg != In_3;
      select[4] <= #dly In_4_reg != In_4;
      select[3] <= #dly In_5_reg != In_5;
      select[2] <= #dly In_6_reg != In_6;
      select[1] <= #dly In_7_reg != In_7;
      select[0] <= #dly In_8_reg != In_8;
    end
    else
    begin

```

```

        if (clk_cnt == 4'd12 & ~start_clk) // = end of
          sequence
            start                      <= #dly 1'b0;
        else
          start                      <= #dly 1'b1;
      end

      casex (select)
        8'b1???????: begin
          if (~dataSent)
            begin
              In_1_reg           <= #dly In_1;
              Data_reg            <= #dly In_1;
              EN                  <= #dly
              8'b01111111;
              start                <= #dly 1'b1;
            end
          else
            begin
              start                <= #dly 1'b0;
              select [7]           <= #dly 1'b0;
            end
          end
        8'b01???????: begin
          if (~dataSent)
            begin
              In_2_reg           <= #dly In_2;
              Data_reg            <= #dly In_2;
              EN                  <= #dly
              8'b10111111;
              start                <= #dly 1'b1;
            end
          else
            begin
              select [6]           <= #dly 1'b0;
              start                <= #dly 1'b0;
            end
          end
        8'b001?????: begin
          if (~dataSent)
            begin
              In_3_reg           <= #dly In_3;
              Data_reg            <= #dly In_3;
              EN                  <= #dly
              8'b11011111;
              start                <= #dly 1'b1;
            end
          else
            begin
              select [5]           <= #dly 1'b0;
              start                <= #dly 1'b0;
            end
          end
        8'b0001?????: begin
          if (~dataSent)
            begin
              In_4_reg           <= #dly In_4;
              Data_reg            <= #dly In_4;
              EN                  <= #dly
              8'b11101111;
            end
          end
        endcase
      end
    end
  end
end

```

```

                start          <= #dly 1'b1;
        end

    else
        begin
            select [4]      <= #dly 1'b0;
            start           <= #dly 1'b0;
        end
    end
8'b00001???: begin
    if (~dataSent)
        begin
            In_5_reg       <= #dly In_5;
            Data_reg        <= #dly In_5;
            EN              <= #dly
            8'b11110111;   start           <= #dly 1'b1;
        end
    end
else
    begin
        select [3]      <= #dly 1'b0;
        start           <= #dly 1'b0;
    end
end
8'b000001??: begin
    if (~dataSent)
        begin
            In_6_reg       <= #dly In_6;
            Data_reg        <= #dly In_6;
            EN              <= #dly
            8'b11111011;   start           <= #dly 1'b1;
        end
    end
else
    begin
        select [2]      <= #dly 1'b0;
        start           <= #dly 1'b0;
    end
end
8'b0000001?: begin
    if (~dataSent)
        begin
            In_7_reg       <= #dly In_7;
            Data_reg        <= #dly In_7;
            EN              <= #dly
            8'b11111101;   start           <= #dly 1'b1;
        end
    end
else
    begin
        select [1]      <= #dly 1'b0;
        start           <= #dly 1'b0;
    end
end
8'b00000001: begin
    if (~dataSent)
        begin
            In_8_reg       <= #dly In_8;
        end
    end

```

```

Data_reg <= #dly In_8;
EN <= #dly
8'b11111110;
start <= #dly 1'b1;
end
else
begin
    select [0] <= #dly 1'b0;
    start <= #dly 1'b0;
end
end
default:
begin
    EN <= #dly
    8'b11111111;
    start <= #dly 1'b0;
end
endcase
end
end

// State Machine for Sending Configuration Data to Delay Chip NB6L295
/*
State Description
-----
RESET Resetting all parameters and registers ->
if (reset): stay; else: to IDLE
IDLE Waiting for start signal from priority
encoder -> if (start): to LOAD_P0; else: stay
LOAD_P0 Load first half of Delay_X - which
corresponds to data for Delay PD0 on delay chip - into
temporary register -> to LOAD_P1
LOAD_P1 Load second half of Delay_X - which
corresponds to data for Delay PD1 on delay chip - into
temporary register -> to SHIFT
SHIFT Shift bits for sending serial bitstream to
SDIN, assert SLOAD -> to END
END End transmission, deassert SLOAD, inform
priority encoder about end of transmission -> to IDLE
*/
parameter RESET = 3'd0;
parameter IDLE = 3'd1;
parameter LOAD = 3'd2;
parameter SHIFT = 3'd3;
parameter END = 3'd4;
reg [2:0] STATE;
reg [DATA_SHIFT_WIDTH-1:0] tmp;

always @ (posedge Clk)
begin
    if (Reset)
        begin
            STATE <= #dly RESET;
            tmp <= #dly 'd0;
            dataSent <= #dly 1'b0;
            start_clk <= #dly 1'b0;
            SLOAD <= #dly 1'b0;
            clk_cnt <= #dly 1'b0;
        end
    else
        begin
            case (STATE)

```

```

RESET:
begin
  if (Reset)
    STATE    <= #dly RESET;
  else
    STATE    <= #dly IDLE;
end // RESET

IDLE:
begin
  SDIN      <= #dly 1'b0;
  clk_cnt   <= #dly 5'd0;
  dataSent  <= #dly 1'b0;
  SLOAD     <= #dly 1'b0;

  if (start & ~dataSent)
    STATE    <= #dly LOAD;
  else
    STATE    <= #dly IDLE;
end // IDLE

LOAD:
begin
  tmp       <= #dly Data_reg;
  STATE    <= #dly SHIFT;
end // LOAD_W1

SHIFT:
begin
  if (clk_cnt < 4'd12) // number of bits to be
  shifted //
  begin
    start_clk      <= #dly 1'b1;
    clk_cnt        <= #dly clk_cnt +1;
    tmp            <= #dly
                  {tmp[DATA_SHIFT_WIDTH-2:0], 1'b0};
    SDIN          <= #dly
                  tmp[DATA_SHIFT_WIDTH-1];
  end
  else
  begin
    SLOAD         <= #dly 1'b1;
    clk_cnt       <= #dly
                  clk_cnt;
    start_clk     <= #dly 1'b0;
    STATE         <= #dly END;
    SDIN          <= #dly 1'b0;
  end
end // SHIFT

END:
begin
  SLOAD         <= #dly 1'b0;
  start_clk     <= #dly 1'b0;
  dataSent      <= #dly 1'b1;
  clk_cnt       <= #dly clk_cnt;
  SDIN          <= #dly 1'b0;
  STATE         <= #dly IDLE;
end // END
default:
  STATE    <= #dly RESET;
endcase
end
endmodule

```