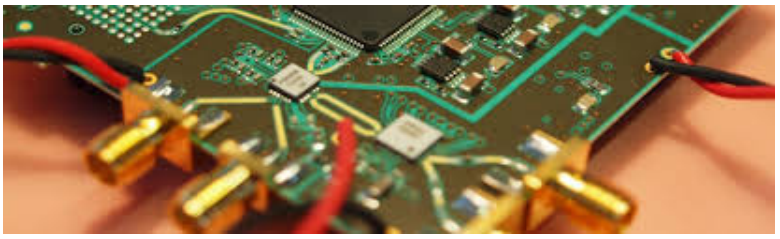
DEPARTMENT OF ELECTRICAL ENGINEERING
AND INFORMATION TECHNOLOGY
Institute for Data Processing and Electronics (IPE)

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

Zusammenfassung

Résumé

Contents

Acronyms **xvii**

1. Introduction **1**
1.1. Time-Stretching techniques in physics 1
1.2. Motivation 1
1.2.1. THz Science and Coherent Synchrotron Radiation 2
1.2.1.1. Micro-Bunching Instabilities 3
1.2.2. Longitudinal Bunch Profile Diagnostics 4
1.2.3. Potential Usage: ANR-DFG ULTRASYNC project? 4
1.3. Objective 5
1.4. General Architecture of a Photonic Time-Stretch Data Acquisition System . 5
1.4.1. Photonic Time-Stretch Front-End 5
1.4.2. Analog-To-Digital Converter 6

2. State Of The Art Read-Out-Systems **21**

3. Architecture Of The New System **23**

4. Design Of The Front-End Sampling Card **25**

5. Back-End Readout Card **27**

6. Results **29**

7. Conclusion and Outlook **31**

Acknowledgements **33**

Appendix **35**
A. Characteristic Impedance Of Coplanar Waveguides 35
B. QuickStart Guide for Evaluation of ZCU216 Board 36
C. 3D model of front-end card 36
D. Code 36

Bibliography **43**

List of Figures

1.1. Use case of the system 2
1.2. Basic scheme of an electron storage ring (redrawn from [Rou14]) 2
1.3. Electromagnetic spectrum 3
1.4. Placeholder [Rot18] 3
1.5. Electrons interact with their own radiation [BBB+19] 4
1.6. Electro-Optical Time-Stretch Technique [SELP+16] 6
1.7. Transfer function of ideal, 3-bit ADC 6
1.8. Aliasing 8
1.9. Track-And-Hold-Amplifier schematic and principle [Kes05] 9
1.10. Quantization noise as function of time (redrawn from [Kes05]) 10
1.11. Effects of Offset and Fain error in ADC 12
1.12. Placeholder: Integral and Differential Nonlinearity Distortion [LV02] 13
1.13. Placeholder: SFDR [Kes09] 14
1.14. Placeholder: Effects of aperture jitter and SJNR [LV02] 16
1.15. Time Interleaving 18
1.16. Placeholder: Offset-Mismatch in Interleaving [Har19] 18
1.17. Placeholder: Gain-Mismatch in Interleaving [Har19] 19
1.18. Placeholder: Timing-Mismatch in Interleaving [Har19] 19
1.19. Placeholder: Timing-Mismatch in Interleaving [Har19] 19

List of Tables

1.1. KARA parameters [Rot18] 4

Acronyms

ADC Analog-To-Digital-Converter. xiii, 3–6, 8

LSB Least Significant Bit. 4, 6, 7

SHA Sample-And-Hold-Amplifier. 4–6, 8

THA Track-And-Hold-Amplifier. 4

THERESA Terahertz Readout Sampling. xi, 13

1. Introduction

Observing non-repetitive and statistically rare signals that occur on short timescales requires fast real-time measurements that exceed the speed, precision and record length of conventional digitizers. Photonic time stretch is a data acquisition method that overcomes the speed limitations of electronic digitizers and enables continuous ultrafast single-shot spectroscopy, imaging, reflectometry, terahertz and other measurements at refresh rates reaching billions of frames per second with non-stop recording spanning trillions of consecutive frames. The technology has opened a new frontier in measurement science unveiling transient phenomena in nonlinear dynamics such as optical rogue waves and soliton molecules, and in relativistic electron bunching. It has also created a new class of instruments that have been integrated with artificial intelligence for sensing and biomedical diagnostics. We review the fundamental principles and applications of this emerging field for continuous phase and amplitude characterization at extremely high repetition rates via time-stretch spectral interferometry.

A new generation of instruments based on photonic time stretch is opening the path to measuring and understanding the behaviour of non-stationary and rare phenomena in ultrafast systems, and harvesting their potential for practical applications such as metrology, machine vision and biomedicine. While these instruments rely on optics for the transformation of a signal's timescale, they are able to operate with either electrical, optical, or terahertz inputs.

1.1. Time-Stretching techniques in physics

1.2. Motivation

Terahertz science and laser applications need to analyse the characteristics of ultrafast events. Commercially available real-time equipment is limited in bandwidth (240 GS/s, LeCroy LabMaster 10-100Zi) and cannot be used to characterize these events in the pico-and femtosecond regime. In this thesis you will evaluate a real-time measurement technique based on the photonic time-stretch analog-to-digital converter (TS-ADC) for the measurement of ultrafast signals with a sampling-rate which exceeds TS/s. The principle is to use a photonic front-end that effectively "slows down" the analog signal in time before it is digitized by fast ADCs.

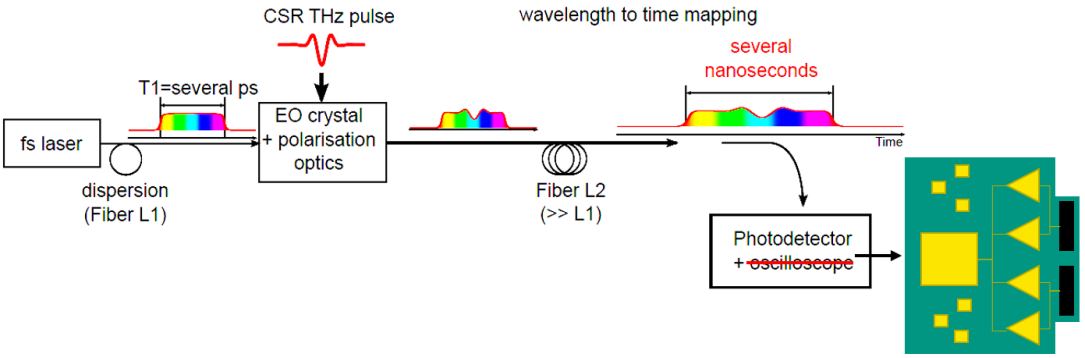


Figure 1.1.: Use case of the system

1.2.1. THz Science and Coherent Synchrotron Radiation

Synchrotron Radiation (SR) is produced in synchrotron radiation facilities (like electron storage rings) by accelerating relativistic electrons. Emission of SR occurs, when electron beams are bent or deflected with dipole magnets or using an undulator[1].

Figure 1.2 shows the general scheme of an electron storage ring. Electrons, or rather electron bunches, are generated with an electron gun and are accelerated to almost the speed of light by a linear accelerator (LINAC). After being broad up to their nominal energy in a booster, the bunches are then injected into the storage ring. In the ring, the path of the electron bunches is altered by bending 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. [Rou14] [Rot18]

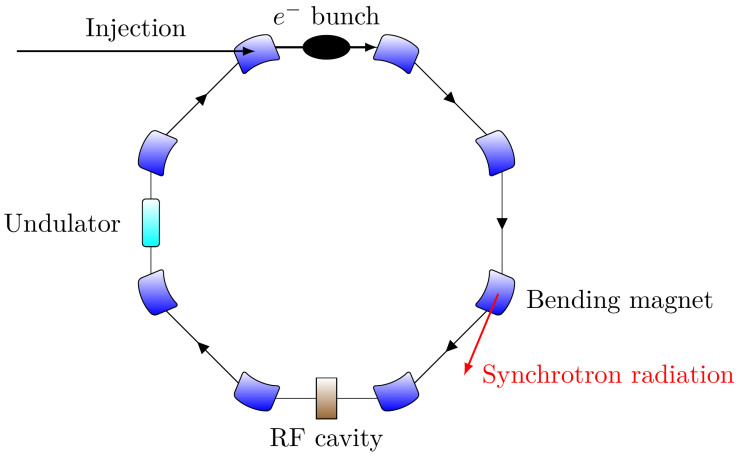


Figure 1.2.: Basic scheme of an electron storage ring (redrawn from [Rou14])

The range of SR reaches from hard X-rays down to the infrared region of the electromagnetic spectrum (see Figure 1.3). In contrast to other sources, it has properties like:

- high intensity

[1]Undulators are used to make the electrons oscillate by generating a periodic magnetic field

- high collimation
- polarisation
- well-defined timing of pulses

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.

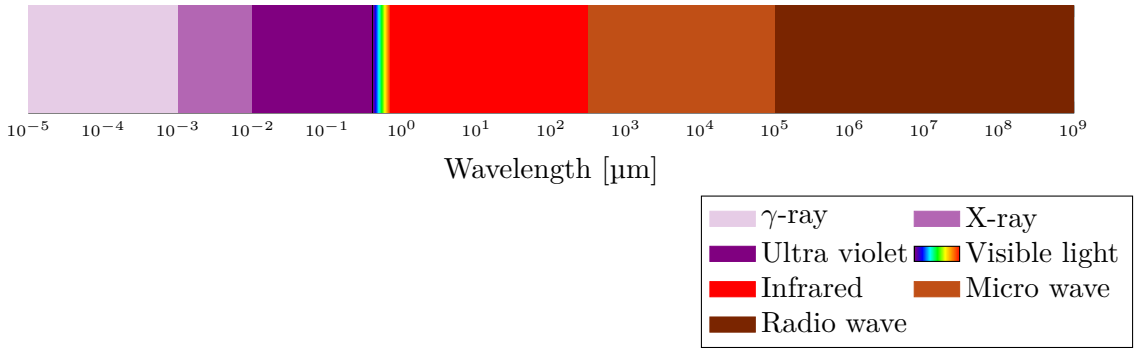


Figure 1.3.: Electromagnetic spectrum

1.2.1.1. Micro-Bunching Instabilities

Increasing demands in current and future accelerators call for higher brilliance of the emitted radiation. This is achieved by increased photon flux and reduction of the transverse emittance. For longitudinal coherence, the electron bunches are shortened, which results in emission of Coherent Synchrotron Radiation (CSR) at frequencies up to the THz range.

However, this introduces complex dynamics, as the electrons interact with their own radiation. This manifests into the so called micro-bunching instability. The formation of micro-structures (in the sub-millimeter to centimeter range) in the longitudinal density profile of the electron bunches. Being on the one side a limitation to the stable operation of the overall system at high current density/short bunch length mode. On the other side, these instabilities can be potential sources of brilliant THz radiation. A thorough understanding of these dynamics is necessary to control the emission in this spectral domain which enables usage in experiments. [Rot18, Bro20]

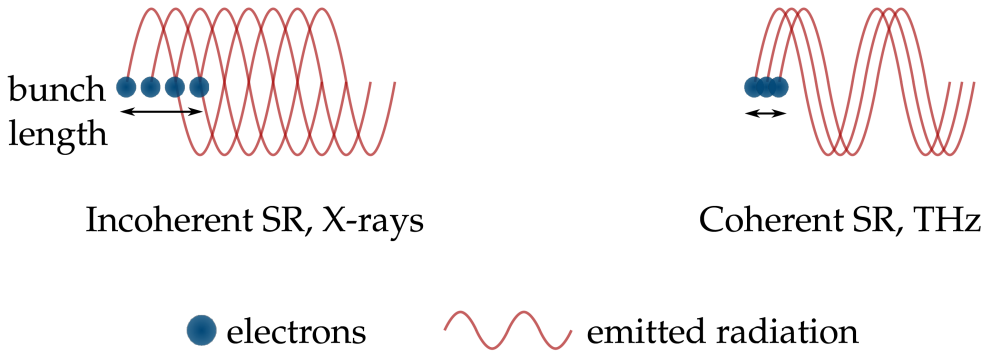


Figure 1.4.: Placeholder [Rot18]

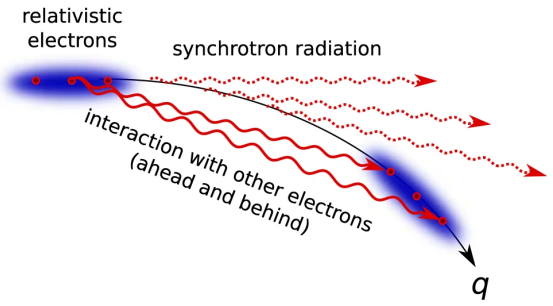


Figure 1.5.: Electrons interact with their own radiation [BBB+19]

1.2.2. Longitudinal Bunch Profile Diagnostics

Karlsruhe Research Accelerator

- Located at the Karlsruhe Institute of Technology (KIT)
- Up to 184 electron bunches can be filled into the storage ring with a distance of 2 ns (500 MHz) between two adjacent bunches
- Operated by the Institute of Beam Physics and Technology (IBPT)
- Microtron, Booster Synchrotron, and Storage Ring

Table 1.1.: KARA parameters [Rot18]

Parameter	Value
Beam energy	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	
normal operation	45 ps
short bunch	2 ps

1.2.3. Potential Usage: ANR-DFG ULTRASYNC project?

From Serge's email:

For accelerator applications, we have also a project (ANR-DFG ULTRASYNC project) with KARA and SOLEIL. There is the question of control (i.e., suppression) of the bursts occuring during the microbunching instability. The objective is to obtain a high power and stable coherent emission (at SOLEIL and KARA). For the moment, the only experimental test has been made using a relatively simple feedback:

- a bolometer signal as the feedback loop input
- a low-cost FPGA (redpitaya) that sends the feedback on the accelerating voltage

There are limitations in the maximal bunch charge in the accelerator. So an open question is whether measuring each THz pulse using EO sampling + time-stretch may help to solve this open problem. In clear:

- EO sampling + time-stretch as in Eléonore's thesis
- association with the new FPGA-based system

- finding adequate feedback that is programmed in the FPGA

may solve the problem, and allow the control to succeed at the highest currents in SOLEIL. Target would be ca. 15 mA for 1 bunch (and feedback control presently works to a little more than ca. 10 mA).

1.3. Objective

Design of a system to be used with photonics time-stretch technique. Fast, continuous sampling of sifnals, integratable in already existing high data-throughput architectures.

1.4. General Architecture of a Photonic Time-Stretch Data Acquisition System

In this section, a general architecture of a Photonic Time-Stretch Data Acquisition System (DAQ) is described. Such a system consists of a photonic front-end, which covers the time-stretching method and conversion of photons into electrical values. The general theory of the time-stretching technique is given, together with some basics about photodetectors.

Furthermore, such a system contains an Analog-To-Digital-Converter (ADC) which converts the analog values into digital signals that can be processed by a computing unit. A short overview of the basic ADC-theory is given, as well as of the most prominent figures of merit. Knowledge and understanding of these is necessary to define and/or evaluate the overall performance of the converter.

Last but not least a DAQ needs an appropriate computing system for collection, processing and visualization of the acquired data. As this part is use-case specific, there doesn't exist a general "theory" which can be described.

1.4.1. Photonic Time-Stretch Front-End

The working principle of the optic time-stretch technique can be described in two steps (see Figure 1.6). First, a short laser pulse (duration typically hundreds of femtoseconds) is propagated in a dispersive medium, e.g. an optical fiber of length L_1 (see Figure 1.6). This results in a chirped laser pulse of the duration

$$T_1 = \Delta\lambda D_1 L_1 \tag{1.1}$$

with the optical bandwidth of the laser pulse $\Delta\lambda$ and the dispersion parameter D_1 of the fiber.

The next step is the time-to-wavelength-mapping, where a temporal intensity modulation is imprinted on the chirped pulse. In this step, the laser pulse co-propagates with another pulse, e.g. tera Hertz (THz) pulse from CSR (duration in the range of picoseconds), in an Electro-Optic (EO) crystal. Due to the Pockels effect[2] the THz pulse causes a time-dependent birefringence in the crystal. This modulates the polarization of the laser pulse.

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/or digitizing with ADCs. [Rou14]

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

$$M = 1 + \frac{L_1}{L_2}. \tag{1.2}$$

[2]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." [Gmb]

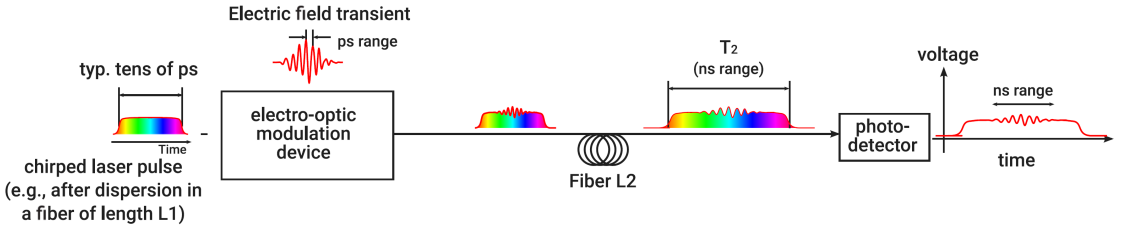


Figure 1.6.: Electro-Optical Time-Stretch Technique [SELP+16]

Detector/Diode

A photodetector converts the incoming optical signal into an electrical signal, known as O/E convertor. The detection and subtraction between the two stretched signals is performed using a amplified balanced photodetector (photoreceiver) from Discovery Semiconductors, with 20 GHz bandwidth and 2800 V/W gain (specified at 1500 nm). The two differential outputs of the detector are sent on a Lecroy LabMaster 10i oscilloscope with 36 GHz bandwidth, 80 GS/s sample rate on each channel and a memory of 256 Mega samples.

1.4.2. Analog-To-Digital Converter

ADCs are used to translate analog voltages into digital signals, which can be processed by information processing, computing, data transmission and control systems. The translation can be seen as encoding a continuous-time analog input (voltage) into a series of discrete, N-bit words. This process, which is also called *sampling*, can be expressed as

$$V_{\text{In}} = V_{\text{FS}} \sum_{k=0}^{N-1} \frac{b_k}{2^{k+1}} + \epsilon \tag{1.3}$$

with V_{In} being the input voltage, V_{FS} the full-scale voltage, b_k the individual output bits and ϵ the quantization error (described in paragraph 1.4.2). Figure 1.7 shows the ideal transfer function of a 3-bit ADC. As one can see, 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.

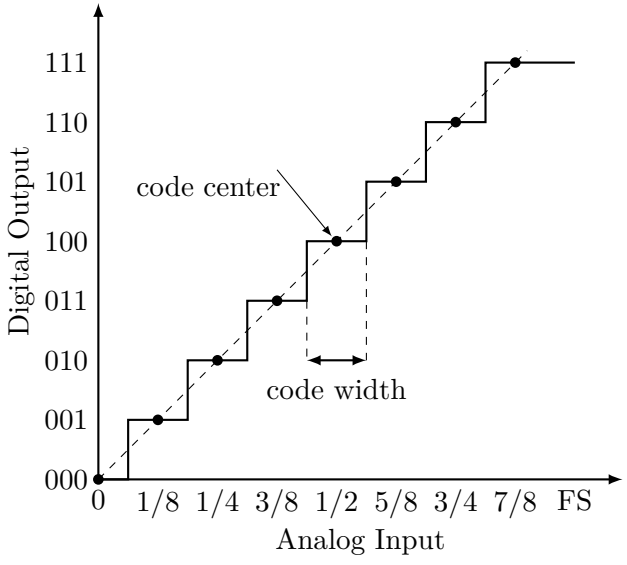


Figure 1.7.: Transfer function of an ideal, 3-bit ADC (redrawn from [LV02])

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 represent a band-limited, continuous signal

$$y(t) \circ\!\!-\!\!- Y(f) \quad \text{with} \quad Y(f) = 0|_{f>B/2} \tag{1.4}$$

it has to be sampled with a frequency f_s respecting

$$f_s > B \quad \text{or} \quad f_s > 2f_a \tag{1.5}$$

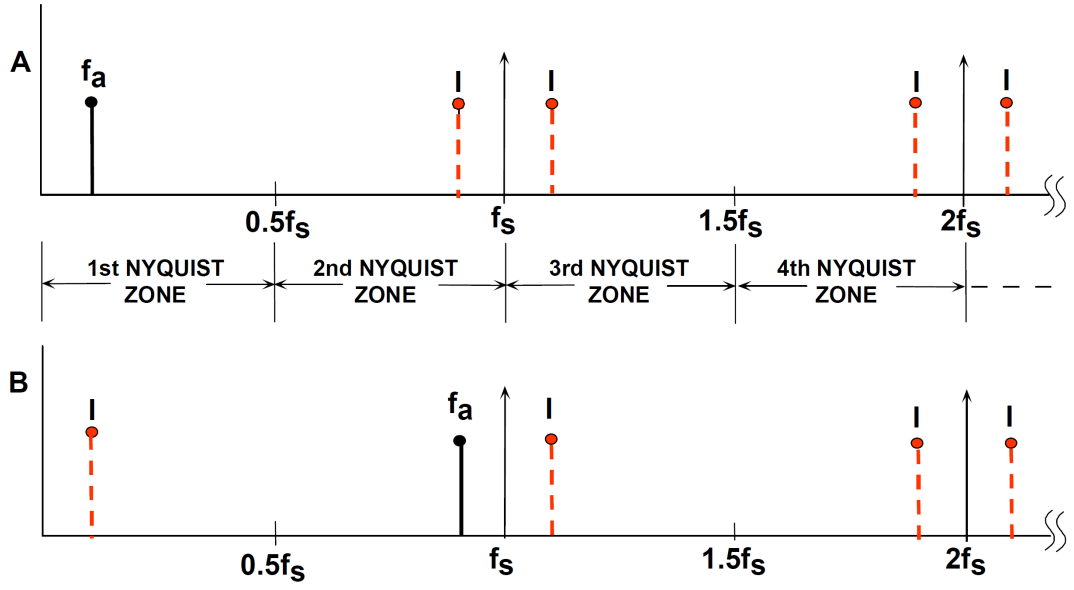
with f_a being the highest frequency contained in the signal. [Kes05, Pue15]

Violation of this rule leads to *aliasing*.

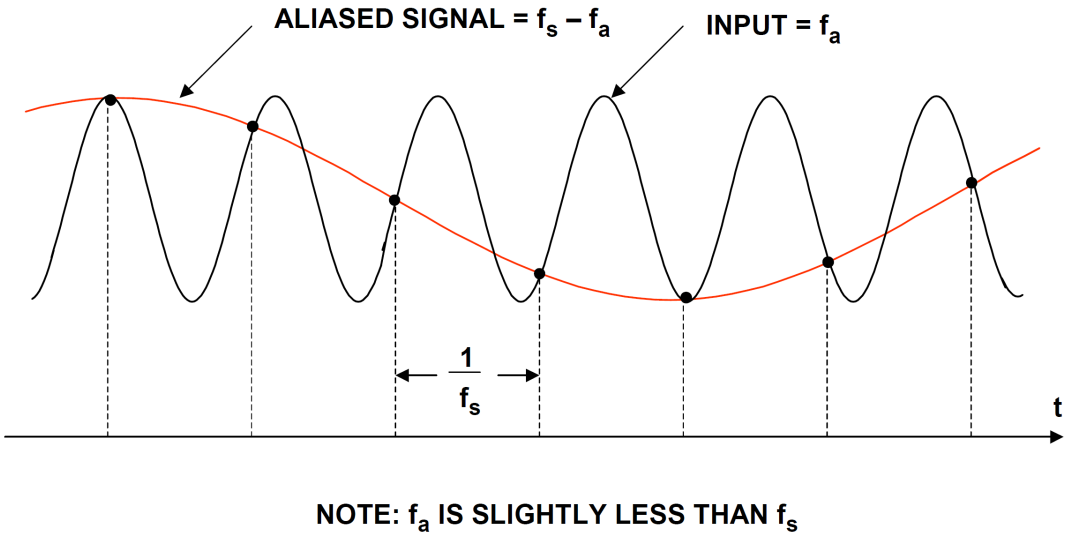
Sample-And-Hold-Amplifier

ADCs need a certain amount of time to sample the input signal. If the level of the analog signal changes by more than one Least Significant Bit (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. The ADC sampling time needs to be timed in such way, that the analog-to-digital conversion falls into the hold period of the SHA and does not exceed into the sample period, for example like shown schematically in the diagram below. Thus, the upper frequency limitation is not determined by the ADC itself, but rather by the aperture jitter, bandwidth, distortion, etc. of the SHA. [Kes05]

In addition to the SHA, there is also the Track-And-Hold-Amplifier (THA). Instead of a sample period, the THA has a track period, where the output of the amplifier tracks the input signal (see also Figure 1.9). When switching to hold mode, the signal at this instant is held. This is opposed to the SHA, where the output during sample mode is actually not defined and is set to the value of the input signal, only when switching into hold mode.



(a) Sampling in frequency domain



(b) Aliasing in time domain

Figure 1.8.: Analog signal with frequency f_a sampled at f_s respecting (A) and not respecting (B) the Nyquist criteria. Figure 1.8b shows the effect of case B in time domain. [Kes05]

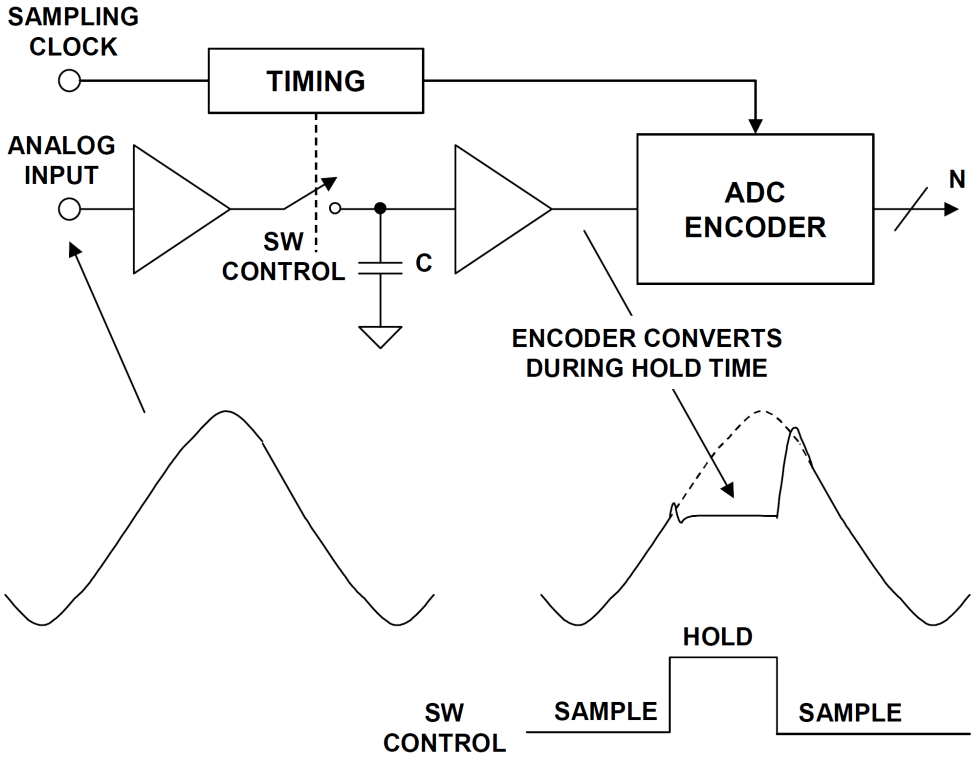


Figure 1.9.: Track-And-Hold-Amplifier schematic and principle [Kes05]

Characteristics of Analog-To-Digital-Converters

For an ideal converter, the number of bits would be sufficient to fully characterize it. 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 following categories [LV02]:

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

This section provides an overview of these figures of merit. Which of these are needed to specify the necessary performance of the ADC has to be chosen for each application accordingly.

Quantization Noise

Even in an ideal N-bit converter there will be errors during the quantization, 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 1.7) [LV02]. 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 with code width q it is

$$|e_q(t)| = |x(t) - x_q(t)| \leq \frac{q}{2}. \tag{1.6}$$

[Pue15]

Assuming the error voltage uncorrelated and uniformly distributed, the theoretical (maximum) Signal-To-Noise-Ratio (SNR) of this *quantization noise* can be calculated. In the time domain, the quantization error $e(t)$ can be approximated with a sawtooth signal:

$$e(t) = st, \quad -\frac{q}{2s} < t < \frac{q}{2s} \tag{1.7}$$

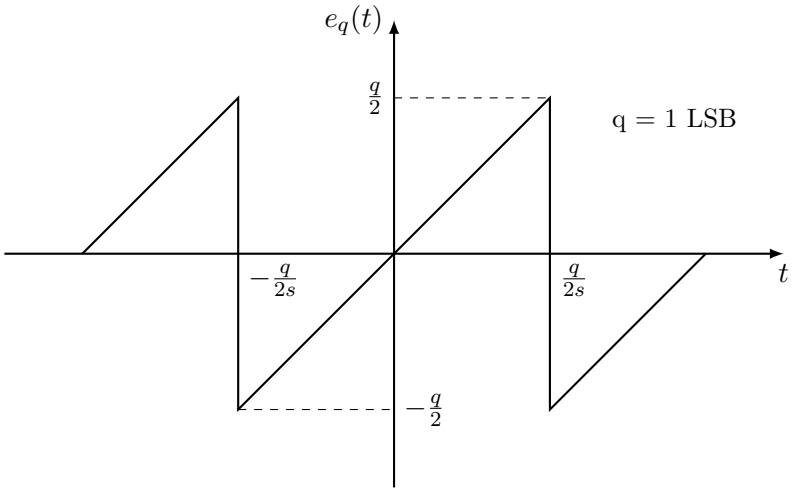


Figure 1.10.: Quantization noise as function of time (redrawn from [Kes05])

The power of the quantization noise, which is assumed to be uncorrelated and broadband, can be calculated as the mean-square $e_{\text{rms}}^2$of $e(t)$ [Kes05]:

$$P_{QN} = e_{\text{rms}}^2 = \overline{e^2(t)} = \frac{s}{q}\int_{-q/2s}^{+q/2s}(st)^2 dt = \frac{s^3}{q}\left[\frac{t^3}{3}\right]_{-\frac{q}{2s}}^{+\frac{q}{2s}} = \frac{q^2}{12} \tag{1.8}$$

To calculate the maximal SNR of an ideal converter, a full-scale input sine wave is assumed:

$$u(t) = u_s \sin(2\pi f t) = \frac{2^N q}{2}\sin(2\pi f t) = 2^{N-1} q \sin(2\pi f t) \tag{1.9}$$

With the effective value of the signal amplitude

$$u_{\text{eff}} = \frac{u_s}{\sqrt{2}} = \frac{2^{N-1}q}{\sqrt{2}} \tag{1.10}$$

SNR

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}} = \frac{u_{\text{eff}}^2}{e_{\text{rms}}^2} = \frac{2^{2N-2}q^2/2}{q^2/12} = 2^{2N}\cdot 1.5. \tag{1.11}$$

In decibel:

$$\text{SNR}|_{\text{dB}} = 10\log\left(2^{2N}\cdot 1.5\right) = 6.02N + 1.76 \tag{1.12}$$

[Pue15] [Kes05]

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:

- Quantization error
- Gain error
- Offset error
- Nonlinearities

[LV02]

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

$$20\log 2^N \approx 6N. \tag{1.13}$$

Offset and Gain Error

The *offset error* is the deviation of the first transition voltage from the ideal 1/2 LSB. *Gain Error* defines the deviation of the slope of the line going through the zero and full-scale point of the transfer function. These errors can easily be corrected by calibration. Figure 1.11 visualizes the effects of both offset and gain error.

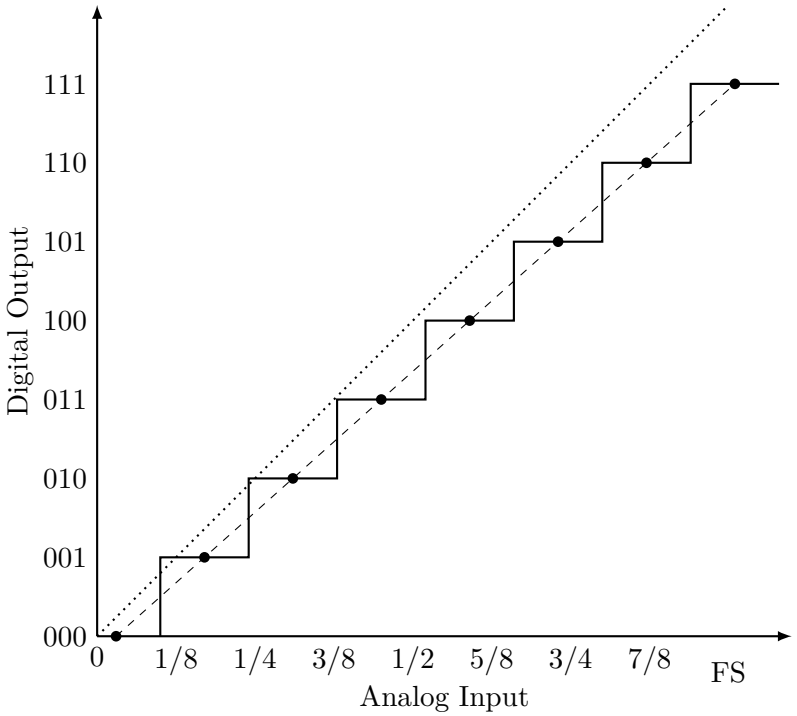


Figure 1.11.: Offset and Gain Error in the ADC characteristic transfer function. Notice the difference between the line going through the code centers (dashed) and the line of an ideal ADC (dotted)

Integral and Differential Nonlinearity Distortion

Integral Nonlinearity (INL) in the transfer function is the distance of the code centers from the ideal line. It results from the integral nonlinearities of the front-end, SHA and also the ADC itself. [Kes05] These nonlinearities depend on the input signal amplitude. [LV02]

Differential Nonlinearity (DNL) is the deviation in code width from the ideal width of 1 LSB. This nonlinearity stems exclusively from the encoding process in the ADC. [Kes05] It not only depends on the input signal amplitude, but also on the positioning along the transfer function. [LV02]

Frequency-Domain Dynamic Parameters

Real ADCs have additional noise sources and distortion processes. *Noise* relates to any unwanted (random or deterministic) signal, which interferes with the desired signal (e.g. additive or quantization noise). *Distortion* is the term for the change of the original signal. These have an impact on the ADC behavior.

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

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

In the following, an overview of the metrics for quantification of these imperfections is given.

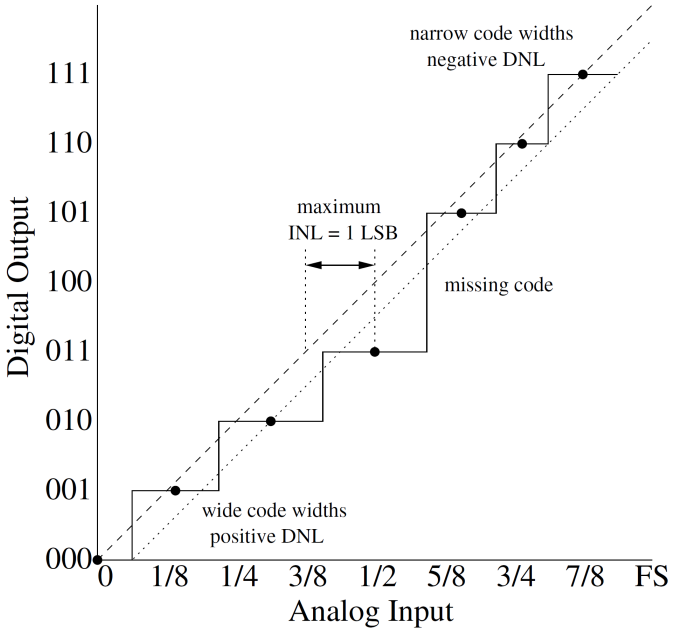


Figure 1.12.: Placeholder: Integral and Differential Nonlinearity Distortion [LV02]

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 Root-Mean-Square (RMS) of the signal amplitude to the mean value of the Root-Sum-Square (RSS) of all other spectral components, including harmonics, but excluding DC (0 Hz). SINAD is a good indication over the general dynamic performance of the ADC, as it includes all contributions from noise and distortion.

Effective-Number-Of-Bits

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

$$\text{ENOB} = \frac{\text{SINAD} - 1.76\,\text{dB}}{6.02\,\text{dB/bit}}. \quad \text{[Kes09]} \tag{1.14}$$

This is derived from solving the equation of the "ideal SNR" Equation 1.12 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. [Kes05]

Spurious-Free Dynamic Range

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. [LV02] 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 (DC (0 Hz) to $f_s/2$, f_s being the ADC sampling rate). The spur may or may not be a harmonic of the fundamental signal. [Kes09] [LV02]

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. [Kes09]

Figure 1.13 illustrates the SFDR in terms of decibels relative to full scale (dBFS) and decibels relative to the carrier (dBc).

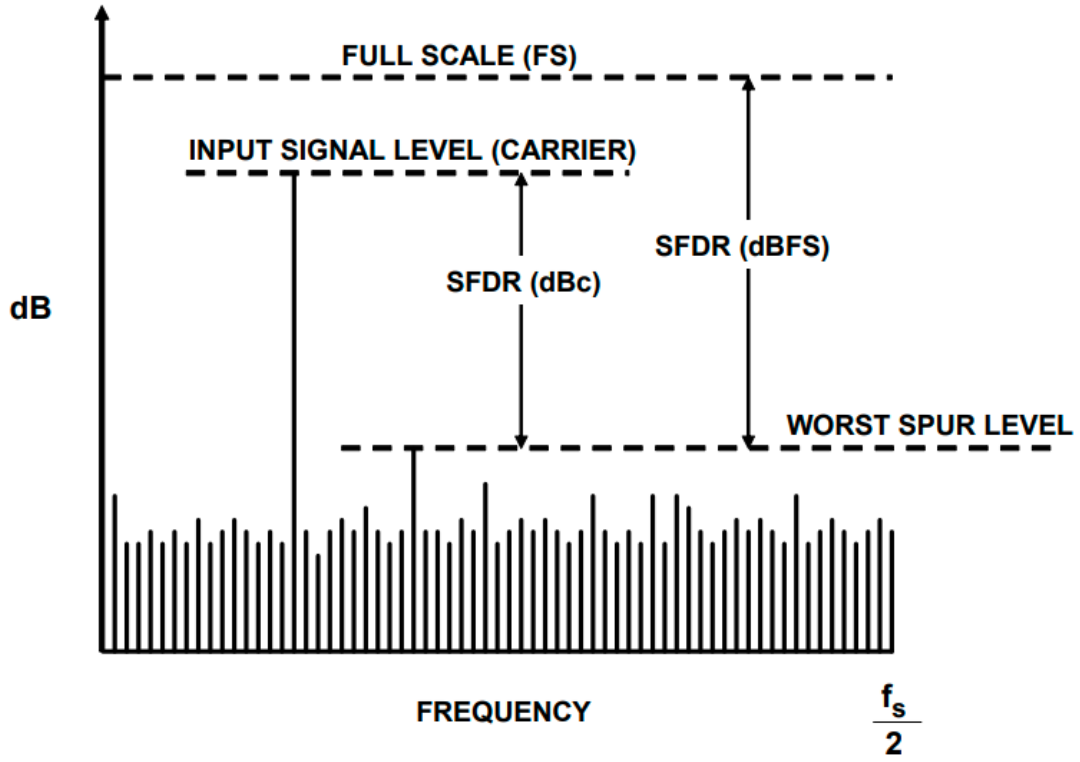


Figure 1.13.: Placeholder: SFDR [Kes09]

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. [LV02]

Effective Resolution Bandwidth

Effective Resolution Bandwidth denotes the frequency of the input signal, at which the SINAD has fallen by 3 dB ($\widehat{=}$ 0.5 bit in terms of ENOB) compared to the SINAD at lower frequency range. [LV02]

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.

Full-Linear Bandwidth

The *Full-Linear Bandwidth* is defined as the frequency at which the slew-rate (SR) of the SHA starts to distort the input signal by a specified value. [LV02] The slew-rate is defined as the rate of how much the voltage v changes against time t:

$$\mathrm{SR} = \frac{dv}{dt} \tag{1.15}$$

A SR of 1 V/µs for example means, that the output of the amplifier can not change more than 1 V over the course of 1 µs.[Col21]

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. [LV02]

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

$$V_{\text{in}} = V_{\text{FS}} \sin(\omega t) \tag{1.16}$$

the maximal slope of this signal is then

$$\frac{dV_{\text{in}}}{dt}\Big|_{\text{max}} = \omega V_{\text{FS}} \tag{1.17}$$

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

$$\Delta V_{\text{rms}} = \omega V_{\text{FS}} \Delta t_{\text{rms}} = 2\pi f V_{\text{FS}} \Delta t_{\text{rms}}. \tag{1.18}$$

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

$$\text{SJNR} = 20 \log \left(\frac{V_{\text{FS}}}{\Delta V_{\text{rms}}} \right) = 20 \log \left(\frac{1}{2\pi f V_{\text{FS}}} \right) \tag{1.19}$$

The voltage error due to jitter and the SJNR for different aperture jitter values are shown in Figure 1.14.

Transient Response

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

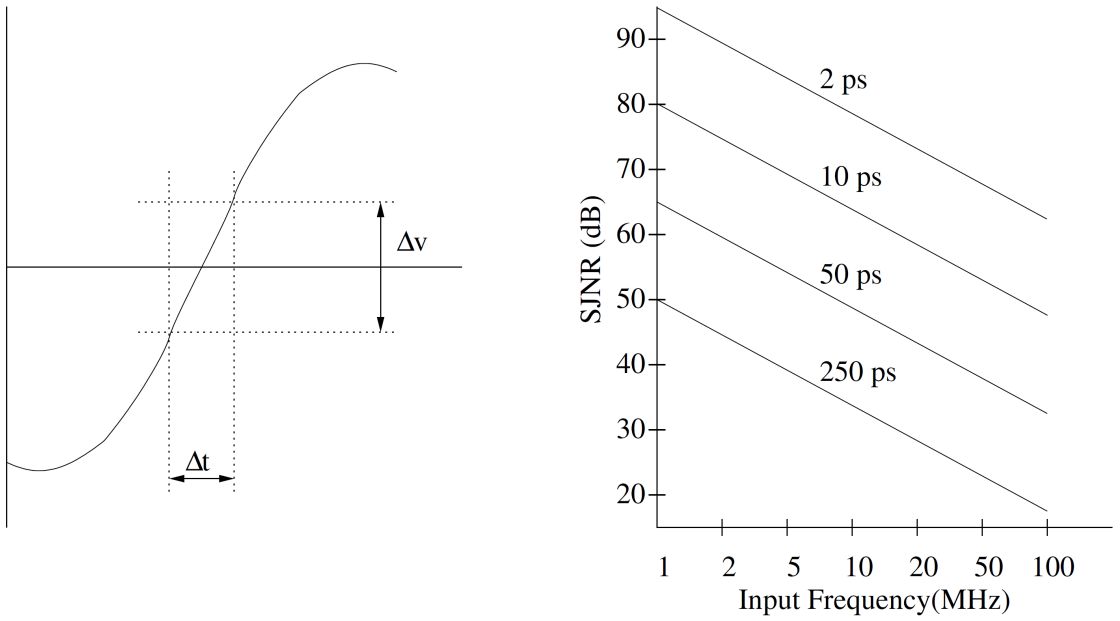


Figure 1.14.: Placeholder: Effects of aperture jitter and SJNR [LV02]

Time Interleaving for Higher Sample Rates

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 1.15), each sampling at $f_c = f_s/M$ individually. This means, the ADCs are clocked in such a way, that they start their respective conversion cycle shortly one after another. At time t_0 the first ADC starts converting the input signal $V_i(t_0)$, after a time delay Δt_i the second starts converting the signal $V_i(t_0 + t_i)$, 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 converter. [MR15]

Challenges

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

First reason is the *offset mismatch* between den ADCs. Each ADC has an DC offset value. 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. This output switches at the frequency $f_s/2$ and therefore introduces an additional frequency component in the spectrum (see Figure 1.16). The magnitude of the spur depends on the offset difference between the ADCs. [Har19]

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 $f_s/2 \pm f_{\text{in}}$ (see Figure 1.17). [Har19]

In the time domain, *timing mismatch* due to group delay in the analog circuitry of the ADC and clock skew[3] can occur. The group delay in analog circuitry can vary between the converters. Furthermore, the clock skew has on the one hand an aperture uncertainty component at each of the ADCs and on the other hand a component related to the accuracy

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

of the clock phases, which are input to each converter. [Har19] This mismatch also produces a spur at $f_s/2 \pm f_{\text{in}}$ (see Figure 1.18).

The last possible mismatch is the *bandwidth mismatch*, which contains both gain and phase/frequency component (see Figure 1.19). 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 $f_s/2 \pm f_{\text{in}}$.

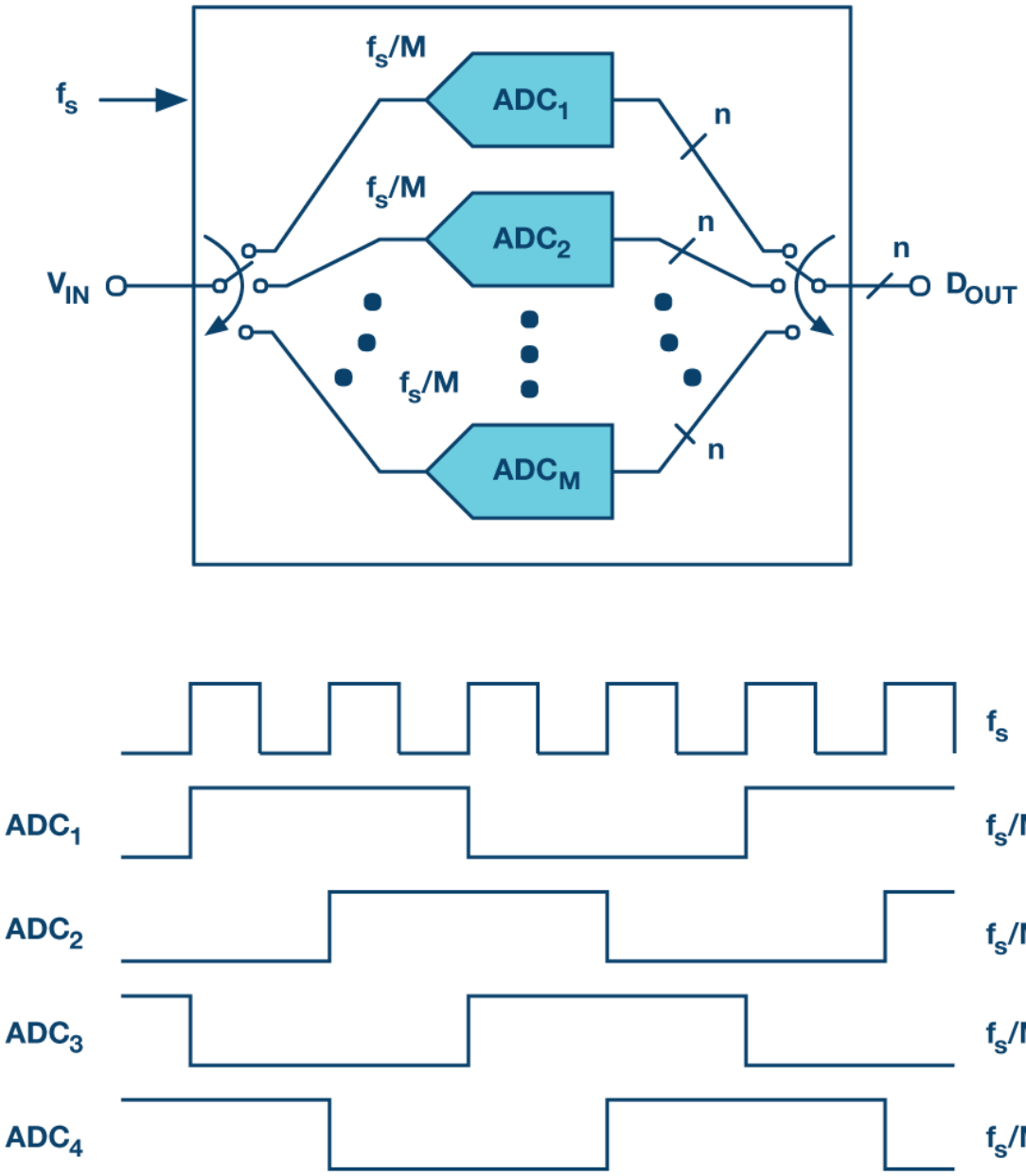


Figure 1.15.: Placeholder: An array of M time interleaved N-bit ADCs with example of clocking scheme for the case of $M = 4$ [MR15]

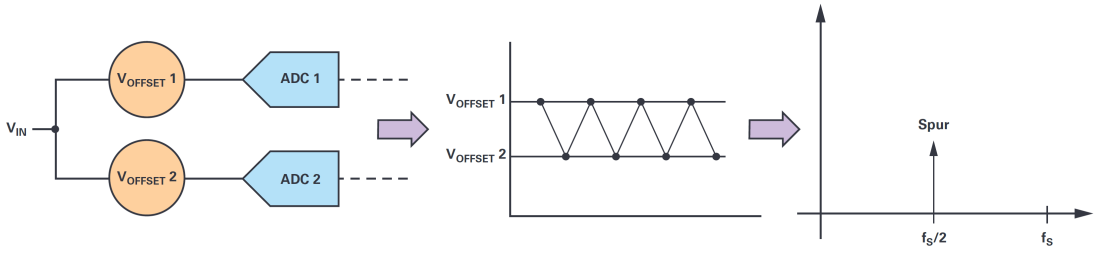


Figure 1.16.: Placeholder: Offset-Mismatch in Interleaving [Har19]

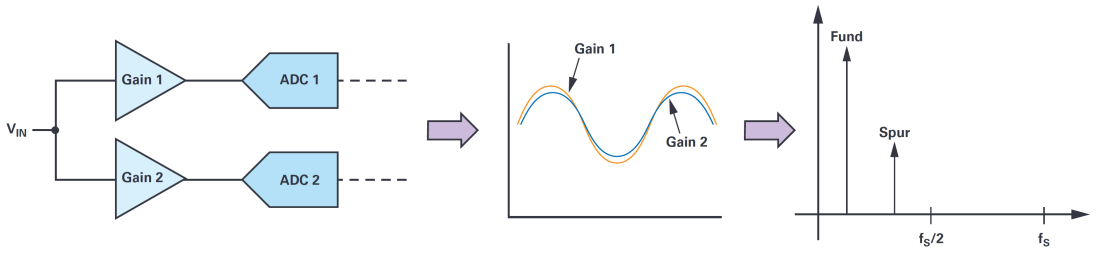


Figure 1.17.: Placeholder: Gain-Mismatch in Interleaving [Har19]

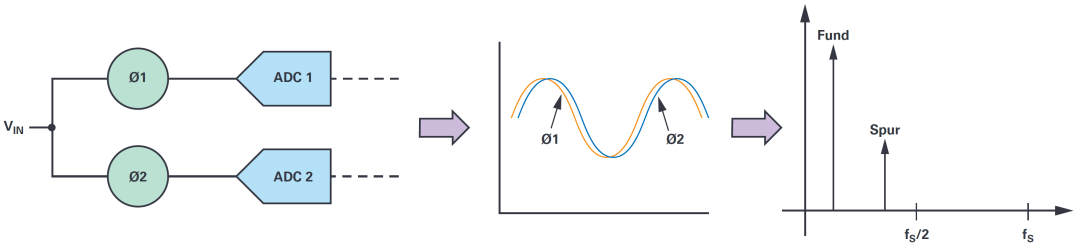


Figure 1.18.: Placeholder: Timing-Mismatch in Interleaving [Har19]

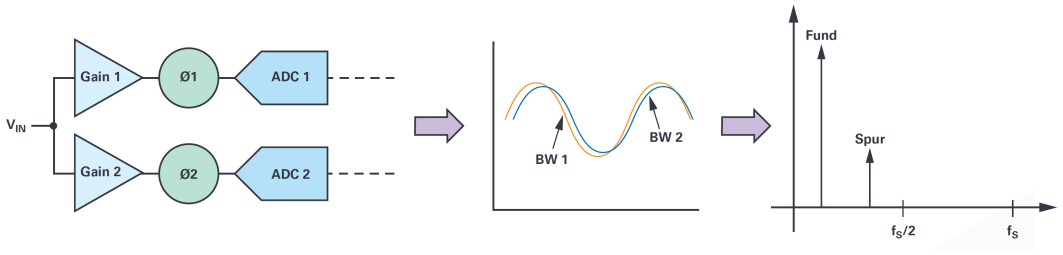


Figure 1.19.: Placeholder: Timing-Mismatch in Interleaving [Har19]

2. State Of The Art Read-Out-Systems

3. Architecture Of The New System

4. Design Of The Front-End Sampling Card

5. Back-End Readout Card

6. Results

7. Conclusion and Outlook

Acknowledgments

Appendix

A. Characteristic Impedance Of Coplanar Waveguides

Surface Coplanar Waveguide with Ground

Characteristic impedance[Wad91, p197-198]:

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

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

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

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

Where

$$k_o = \Lambda \frac{-\sqrt{\Lambda^2 - t_c^2} + \sqrt{\Lambda^2 - t_B^2}}{t_B\sqrt{\Lambda^2 - t_c^2} + t_c\sqrt{\Lambda^2 - t_B^2}} \tag{A.5}$$

$$k_e = \Lambda' \frac{-\sqrt{\Lambda'^2 - t_c'^2} + \sqrt{\Lambda'^2 - t_B'^2}}{t_B'\sqrt{\Lambda'^2 - t_c'^2} + t_c'\sqrt{\Lambda'^2 - t_B'^2}} \tag{A.6}$$

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

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

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

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

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

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

The parameters have to be chosen according to

$$s + 2.0w + 2.0d \leq h \tag{A.13}$$

to guarantee coplanar propagation. [Wad91]

Surface Coplanar Waveguide with Ground

The characteristic impedance of a coplanar waveguide is given as follows [Wad91]:

$$Z_0 = \frac{60.0\pi}{\sqrt{\epsilon_{eff}}} \frac{1.0}{\frac{K(k)}{K(k')} + \frac{K(k_1)}{K(k'_1)}} \tag{A.14}$$

It comprises of the following components, with K(k) being an elliptical integral of the first kind (see also [BSMM99, p. 430]):

$$k = a/b \tag{A.15}$$

$$k' = \sqrt{1.0 - k^2} \tag{A.16}$$

$$k'_1 = \sqrt{1.0 - k_1^2} \tag{A.17}$$

$$k_1 = \frac{tanh(\frac{\pi a}{4.0h})}{tanh(\frac{\pi b}{4.0h})} \tag{A.18}$$

$$\epsilon_{eff} = \frac{1.0 + \epsilon_r \frac{K(k')}{K(k)} \frac{K(k_1)}{K(k'_1)}}{1.0 + \frac{K(k')}{K(k)} \frac{K(k_1)}{K(k'_1)}} \tag{A.19}$$

B. QuickStart Guide for Evaluation of ZCU216 Board

C. 3D model of front-end card

D. 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                 start; // signal for state machine to start sending
    data
reg                 dataSent; // flags if transmission for one delay chip
    is finished

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;
                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

        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

        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
       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;
                                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
    end
```

```
endmodule
```

Bibliography

[BBB+19] Bielawski, Serge, Edmund Blomley, Miriam Brosi, Erik Bründermann, Eva Burkard, Clément Evain, Stefan Funkner, Nicole Hiller, Michael Nasse, Gudrun Niehues und et al.: *From self-organization in relativistic electron bunches to coherent synchrotron light: observation using a photonic time-stretch digitizer*. Scientific Reports, 9(1), July 2019. http://dx.doi.org/10.1038/s41598-019-45024-2.

[Bro20] Brosi, Miriam: *In-Depth Analysis of the Micro-Bunching Characteristics in Single and Multi-Bunch Operation at KARA*. Dissertation, Karlsruher Institut für Technologie (KIT), 2020. 54.01.01; LK 01.

[BSMM99] Bronstein, Semendjajew, Musiol und Mühlig: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 1999.

[Col21] College, Mohawk Valley Community: *Slew Rate and Power Bandwidth*, März 2021. https://eng.libretexts.org/@go/page/3575, [Online; accessed 2021-07-16].

[Gmb] GmbH, LD DIDACTIC: *Demonstrating the Pockels effect in a conoscopic beam path*. https://www.ld-didactic.de/literatur/hb/e/p5/p5451_e.pdf.

[Har19] Harris, Jonathan: *The ABCs of Interleaved ADCs*, 2019. https://www.analog.com/en/technical-articles/the-abcs-of-interleaved-adcs.html.

[Kes05] Kester, Walt: *The Data Conversion Handbook*. Newnes, 2005.

[Kes09] Kester, Walt: *Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor*. Januar 2009.

[LV02] Lundberg, Kent H. und Vin Vq: *Analog-to-Digital Converter Testing*, 2002.

[MR15] Manganaro, Gabriele und Dave Robertson: *Interleaving ADCs: Unraveling the Mysteries*. Analog Dialogue, 2015.

[Pue15] Puente León, Fernando: *Messtechnik - Systemtheorie für Ingenieure und Informatiker*. Vieweg Verlag, 10. aufl. Auflage, 2015, ISBN 978-3-662-44820-5.

[Rot18] Rota, Lorenzo: *KALYPSO, a novel detector system for high-repetition rate and real-time beam diagnostics*. Dissertation, Karlsruher Institut für Technologie (KIT), 2018. 54.02.02; LK 01.

[Rou14] Roussel, Eleonore: *Spatio-temporal dynamics of relativistic electron bunches during the microbunching instability: study of the Synchrotron SOLEIL and UVSOR storage rings*. Dissertation, September 2014.

[SELP+16] Szwaj, C., C. Evain, M. Le Parquier, P. Roy, L. Manceron, J. B. Brubach, M. A. Tordeux und S. Bielawski: *High sensitivity photonic time-stretch electro-optic sampling of terahertz pulses*. Review of Scientific Instruments, 87(10), 2016.

[Wad91] Wadell, Brian C.: *Transmission Line Design Handbook*. Artech House, 1991.