Martin Andreas Pernull, BSc

A Multi-Channel Voltage Level Monitoring Concept for Automotive Environments by Means of Random Sampling

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Telematics

submitted to

Graz University of Technology

Supervisor

Ass.-Prof. Dipl.-Ing. Dr.techn. Peter Söser

Institute for Electronics

External supervisors: Dipl.-Ing. Andreas Kalt
Dipl.-Ing.(FH) Gerhard Pichler

Graz, April 2015

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis dissertation.

Date

Signature

A Multi-Channel Voltage Level Monitoring Concept for Automotive Environments by Means of Random Sampling

The ongoing growth of semiconductor integration density leads to an increasing attractiveness of system on chip solutions for automotive applications. One of their key components is the power supply, which provides various voltage rails for internal as well as for external loads. Safety critical voltage rails must be monitored by an independent observer to detect under- or overvoltage conditions. A high immunity of the voltage monitoring system against electromagnetic interference is crucial to prevent inadvertent warnings.

Established voltage monitoring systems utilize distributed comparators with various fixed threshold voltages and optional hysteresis. They do not permit an independent assessment of the comparator functionality during operation. Due to the distributed system nature the introduction of self-test mechanisms would create a large area effort.

Therefore, this thesis investigates alternative approaches for voltage monitoring systems, which provide better testability while keeping the silicon area usage low. The most feasible concept based on multiplexed analog-to-digital conversion and subsequent digital filtering is examined in detail. The centralized architecture allows complying with functional safety by introduction of a single central self-test feature.

A charge redistribution successive approximation analog-to-digital converter is used to replace comparators with various threshold levels. In the automotive environment voltages up to many tens of volts are encountered, which makes the inevitable active anti-aliasing filters for multiplexed uniform sampling systems consume an unacceptable amount of silicon area. Hence, the use of anti-aliasing filters is completely avoided by means of a random sampling technique.

In the context of functional safety, latent faults affect monitoring systems whose functionality only becomes relevant once a fault condition to be observed actually occurs. The prevention of latent faults in the voltage monitoring system is achieved by introduction of a built-in self-test with high diagnostic coverage. It enables to verify the correct functionality of the analog-to-digital converter during normal operation and mostly reuses already existing resources. Furthermore, it can support the automated test equipment during production testing and consequently saves test costs.

A proof of concept is provided by a digital simulation of a concrete example. However, the feasibility of the new voltage monitoring concept for an application must be evaluated individually based on the precise performance requirements and the available silicon area. Finally, the new concept features high modularity, a potentially low area footprint and enhanced testability.

Keywords: Voltage level monitoring, automotive, random sampling, capacitive charge-redistribution SAR ADC, built-in self-test, system on chip.

Konzept zur mehrkanaligen Spannungspegelüberwachung in automobilen Umgebungen mittels zufälliger Abtastung

Durch die fortschreitende Hochintegration von Halbleitertechnologien, welche im Automobilbereich Anwendung finden, werden System-on-Chip Lösungen immer attraktiver. Einer deren Grundbestandteile ist das Stromversorgungssystem zur Bereitstellung von Spannungen für interne und externe Verbraucher.

Bei sicherheitskritischen Versorgungsspannungen muss das Auftreten von Über- und Unterspannungen von einem unabhängigen Modul überwacht werden. Dabei ist eine hohe Immunität gegenüber elektromagnetisch eingekoppelten Störungen wichtig, um versehentliche Warnungen zu vermeiden.

Etablierte Spannungsüberwachungssysteme bestehen aus dezentralen Komparatoren mit fixen Schwellenspannungen und optionaler Hysterese. Eine unabhängige Überprüfung der Funktionalität während des Betriebs ist jedoch nicht möglich. Durch die verteilte Architektur wäre die Einführung eines Selbsttests für jeden Komparator mit einem hohen Flächenaufwand verbunden.

Deshalb werden in dieser Arbeit alternative Ansätze für Spannungsüberwachungssysteme untersucht, die eine bessere Testbarkeit bei gleichzeitig geringem Flächenbedarf aufweisen. Das brauchbarste Konzept, das auf gemultiplexter Analog-Digital-Wandlung mit anschließender digitaler Filterung basiert, wird im Detail analysiert. Die zentralisierte Architektur erlaubt die Erfüllung von Anforderungen der funktionalen Sicherheit mithilfe eines einzigen Selbsttests.

Ein auf dem Wägeverfahren durch kapazitiven Ladungsausgleich basierender Analog-Digital-Wandler ersetzt die Komparatoren mit verschiedenen Schwellenspannungen. In automobilen Umgebungen treten Spannungen in der Größenordnung von mehreren zehn Volt auf. Deshalb würden die bei regelmäßiger, gemultiplexter Abtastung notwendigen Anti-Aliasing-Filter einen inakzeptabel hohen Flächenbedarf erfordern. Deshalb werden Anti-Aliasing Filter durch die Anwendung von zufälliger Abtastung gänzlich vermieden.

Im Kontext der funktionalen Sicherheit treten latente Fehler in Überwachungssystemen auf, deren korrekte Funktionalität erst relevant wird, sobald der zu detektierende Fehlerfall tatsächlich eintritt. Die Vermeidung von latenten Fehlern im Spannungsüberwachungssystem wird durch einen integrierten Selbsttest mit hoher diagnostischer Abdeckung erreicht. Er erlaubt die Überprüfung der Funktionalität des Analog-Digital-Wandlers während des Betriebs und verwendet dafür größtenteils bereits existierende Komponenten. Zusätzlich kann er die Dauer der Produktionstests reduzieren und somit Kosten sparen.

Anhand eines konkreten Beispiels wird durch eine Simulation ein Machbarkeitsnachweis durchgeführt. Die Umsetzbarkeit des neuen Spannungsüberwachungskonzepts für eine spezifische Anwendung muss jedoch anhand der Anforderungen an die Leistungsfähigkeit und den Flächenbedarf individuell überprüft werden. Schlussendlich bietet das neue Konzept eine hohe Modularität, einen potentiell niedrigen Flächenbedarf sowie eine hohe Testbarkeit.

Stichworte: Spannungspegelüberwachung, Automotive, zufällige Abtastung, SAR ADC, Selbsttest, System on Chip

Acknowledgements

In the course of time many people have contributed to this thesis. At Graz University of Technology I am particularly indebted to:

- Prof. Söser, my supervisor from the Institute for Electronics, for his straightforward feedback, support and helpfulness, and
- Prof. Zangl for providing additional feedback.

I have written this thesis at Infineon Technologies Austria AG in Villach. I want to thank all colleagues for the good working atmosphere and their support. Notably, I want to express my gratitude to:

- Andreas for his guidance and for contributing his expertise in analog design. He provided the link between my theoretical work and the real application,
- Gerhard for his support and system level considerations,
- Franz for offering the thesis topic to me, and
- Peter for taking an interest in my ideas regarding ADCs.

Special thanks go to my family for their everlasting support.

Contents

List of Figures

List of Tables

Part I.

Introduction

1. Initial Situation

Integrated electronic circuits featuring high integration density allow the combination of complete system functionality on one single chip. This system on chip (SoC) architecture is also used in automotive applications where the power supply constitutes a crucial part for the overall functionality as it provides supply voltage rails for internal and external loads. For compliance with the functional safety norm ISO 26262 [18], an independent detection mechanism for under- or overvoltage conditions on safety critical voltage rails is potentially required.

The following section describes the existing power supply unit of an automotive airbag SoC. Specifically, the architecture of the contained voltage monitoring system serves as a motivational example for the development of a new approach with enhanced testability.

1.1. Case Study: Power Supply of a System On Chip for Airbag Applications

The generic architecture of the power supply system of an automotive SoC for airbag applications is depicted in Fig. 1.1. In full operation it provides voltage rails for internal as well as for external loads. The different voltages are in a range from 1.5 V up to 40 V, where the high voltage is specific to the airbag application due to the need for a large energy reserve. In a car accident, despite the possible loss of battery power, the airbag system must be kept running to enable firing of the airbags later on. The necessary energy is stored in a large capacitor and is given by $E = \frac{C \cdot U^2}{2}$. Therefore, to store more energy, increasing the voltage is more efficient than increasing the capacitance value. The capacitor C_{boost} is therefore charged by a boost converter to a higher voltage than supplied by the car battery. To minimize losses in normal operation the boost voltage is converted down to the operating voltage by a buck converter. This converter also ensures that in case of battery power loss the voltage V_{buck} remains constant – even though V_{boost} decays due to the discharging of the energy reserve capacitor C_{boost}.

The system also provides a sleep mode where only minimal circuitry is running to keep the power consumption low. When the system is started, a first power domain gets activated and linear voltage regulators (LVRs) start to supply the internal modules during the start-up phase. After various self-checks the full operation domain is activated. It features DC-DC converter circuits with better power conversion efficiency and higher power capability, which is used to supply external components such as sensors and a microcontroller. Additionally, digital on-chip circuitry operating on even lower voltages is powered by digitally controlled LVRs which are supplied from the down-converted voltage V_{buck}.

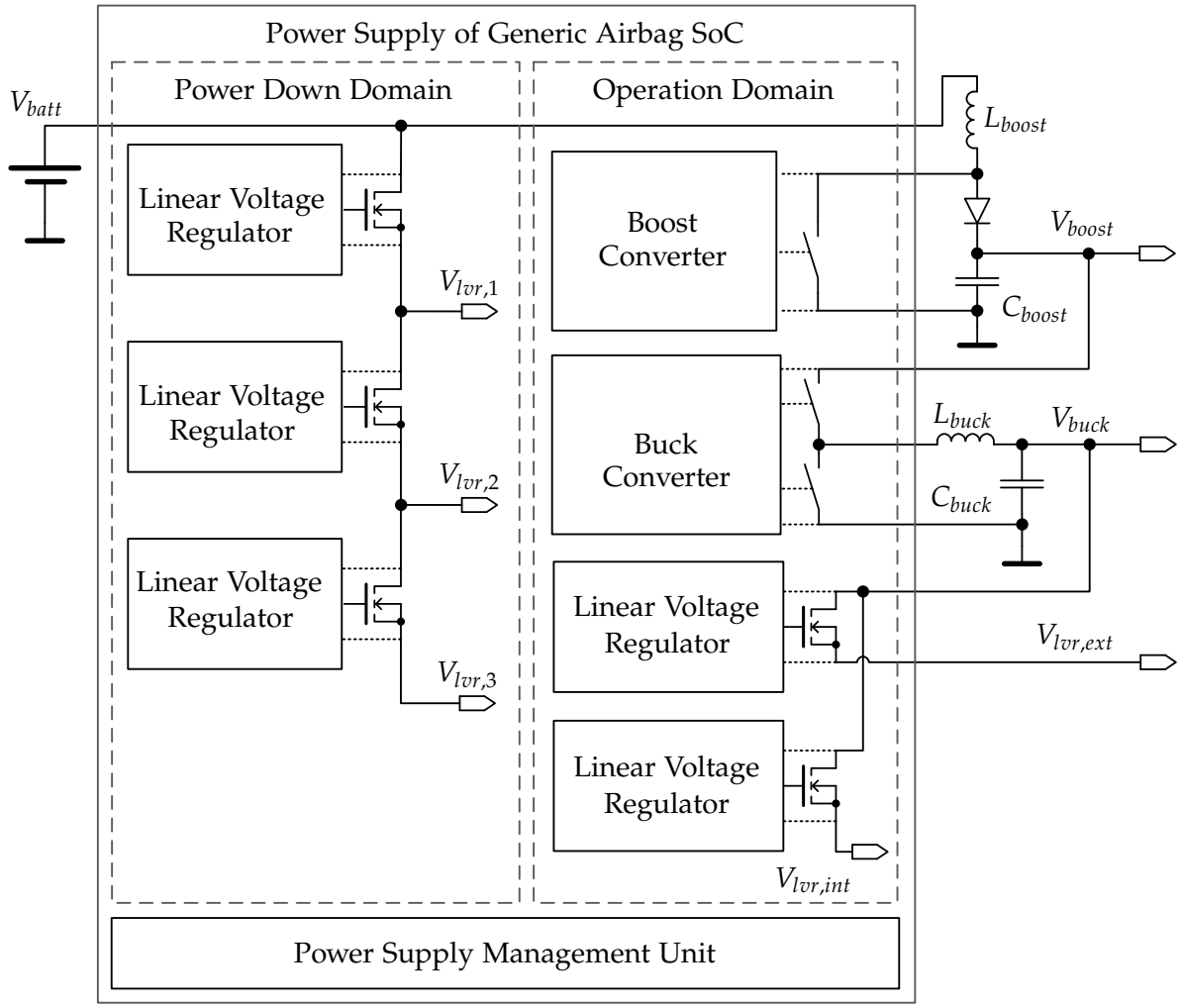

Figure 1.1.: Generic power supply architecture of an airbag SoC with two domains.

1.1.1. Voltage Monitoring System

The voltage monitoring system provides independent monitoring of the power supply rail voltage levels to detect under- or overvoltage conditions. Furthermore, it controls the start-up sequence of the power supply modules by successively enabling of dependent modules once their supply is assured to be stable.

The basic concept consists of independent comparators with various threshold voltages and optional hysteresis. As one of the measures against electromagnetic interference (EMI) the comparator outputs are filtered to prevent disturbances from inadvertently toggling the output signals of the voltage monitoring system. Figure 1.2 shows the basic system architecture with comparators and filters. An additional logic block combines the filtered output signals of multiple comparators to generate the feedback required by the control- and reset logic. The following sections describe the behavior of the components in more detail.

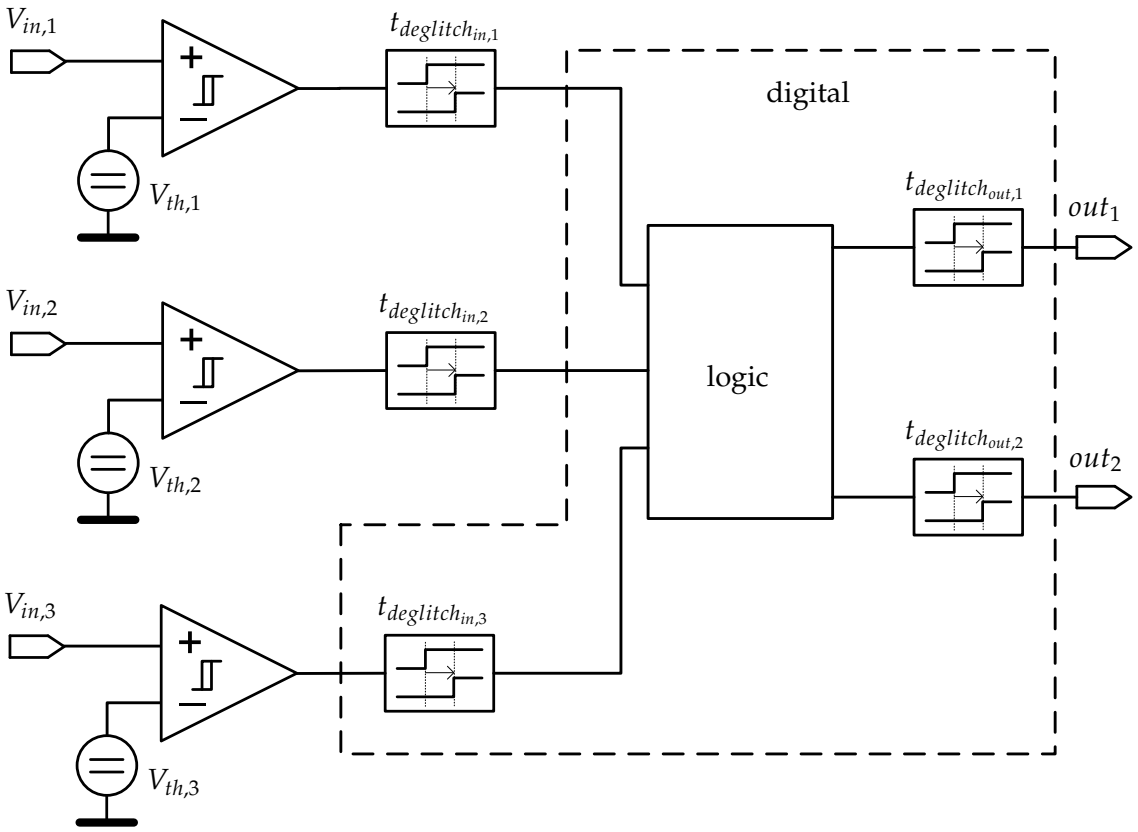

Figure 1.2.: Generic architecture of a distributed voltage monitoring system.

1.1.2. Analog Comparator with Hysteresis

One of many possibilities to build a comparator with hysteresis is shown in Fig. 1.3. The hysteresis is achieved by a feedback from the comparator output to the resistive input divider. This divider converts the potentially high input voltage to a low-voltage signal such that the comparator can be implemented with low-voltage transistors. They have low maximum voltage ratings, but are faster and require less area. To keep the current consumption low, large resistances are necessary which in turn use up more area. The reference voltage is provided by the buffered output of a bandgap reference.

Due to the feedback the threshold voltage depends on the value of the digital comparator output. The following calculations assume an ideal operational amplifier with infinite gain. The threshold voltage V_{th} is defined as the input voltage V_{in} which results in the inputs of the operational amplifier being equal:

$$V_{th} \stackrel{!}{=} V_{in}\Big|_{V_p = V_{ref}} . \tag{1.1}$$

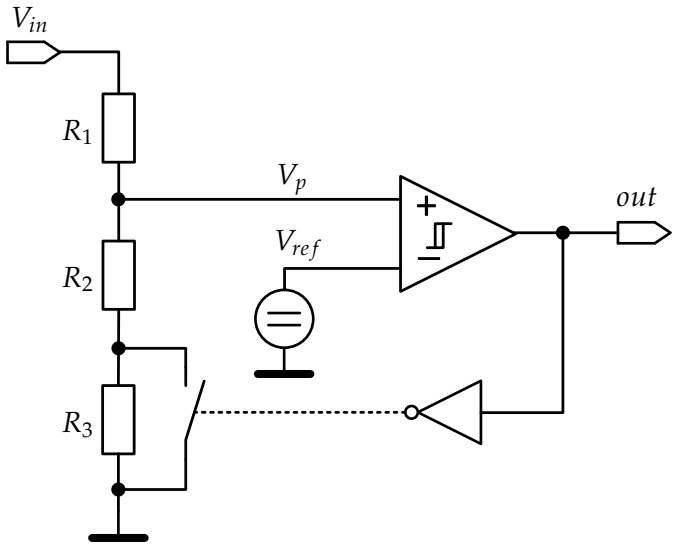

Figure 1.3.: Analog comparator with hysteresis.

If the digital output is low the switch is closed and shorts the resistor R_3, hence the rising threshold is given by:

$$V_{th,rising} \stackrel{!}{=} V_{in}|_{V_p=V_{ref},out=low} \tag{1.2}$$

$$= V_{ref} \cdot \frac{R_1 + R_2}{R_2}\,. \tag{1.3}$$

When the digital output is high the resistor R_3 also contributes to the falling threshold voltage calculations as follows:

$$V_{th,falling} \stackrel{!}{=} V_{in}|_{V_p=V_{ref},out=high} \tag{1.4}$$

$$= V_{ref} \cdot \frac{R_1 + R_2 + R_3}{R_2 + R_3}\,. \tag{1.5}$$

The hysteresis is calculated as the difference between the rising and the falling threshold voltage.

$$V_{hyst} = V_{th,rising} - V_{th,falling} \tag{1.6}$$

$$= V_{ref} \cdot \frac{R_1 R_3}{R_2(R_2 + R_3)} \tag{1.7}$$

It is useful for preventing the comparator output from inadvertent bouncing due to disturbances while the input voltage is close to the threshold voltage. The hysteresis loop behavior resulting from the feedback is visible in Fig. 1.4.

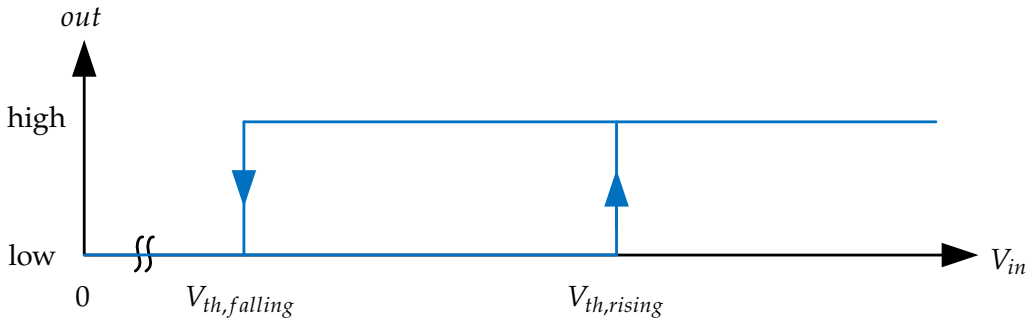

Figure 1.4.: Hysteresis loop of the comparator.

1.1.3. Deglitch Filters

A deglitch filter processes the comparator output signal to increase the immunity against noise and other high-frequency disturbances at the comparator input. The comparator output must be constant for a time period $t_{deglitch}$ to modify the filter output. If a change of the input signal occurs before the $t_{deglitch}$ period is over, the internal state is reset and the process starts anew. This behavior is defined on system level and results in a quasi-low pass behavior of the nonlinear monitoring system.

An analog implementation of such a deglitch filter is presented in the following section. Due to manufacturing deviations, the deglitch time deviates from its nominal value and especially for long deglitch times it is area expensive. A digital implementation of a deglitch filter shown in the subsequent section is preferable, but it can only be used when a clock is available.

1.1.3.1. Analog Implementation

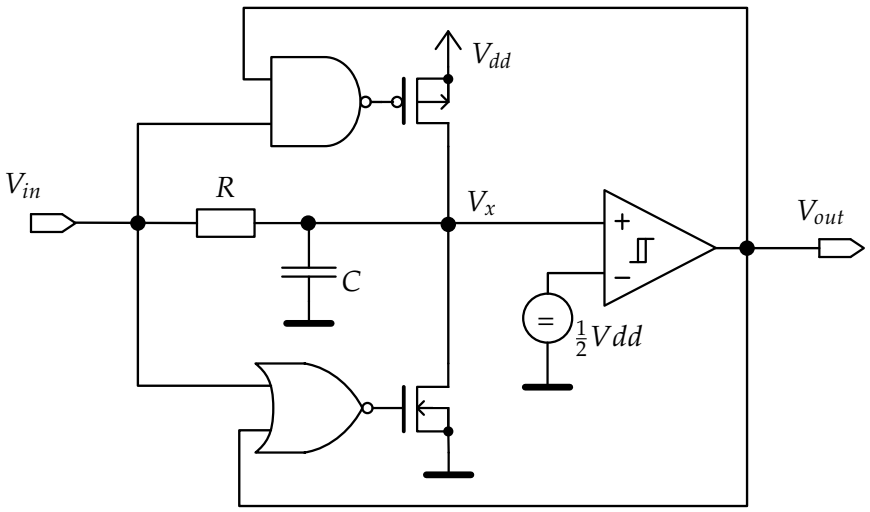

Figure 1.5.: Analog deglitch filter with symmetric rising and falling edge deglitch time.

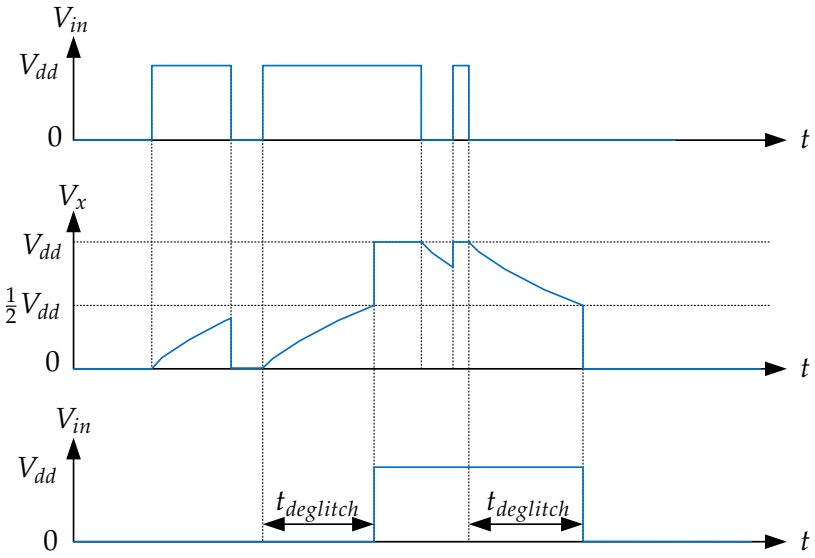

Figure 1.6.: Transient behavior of the analog deglitch filter.

V_{in}	V_{out}	V_x
0	0	0
0	V_{dd}	X
V_{dd}	0	X
V_{dd}	V_{dd}	V_{dd}

Table 1.1.: Truth table of the capacitor control logic.

An example for an analog implementation of a deglitch filter with symmetric rising and falling deglitch times is shown in Fig. 1.5. The NAND and NOR gates together with the respective PMOS and NMOS transistors control the voltage on the transistor according to Table 1.1.

For equivalent input and output voltages the output will keep its state. If they are different, the node V_x is neither forced to zero nor to V_{dd}, but is slowly charged or discharged via the resistor R. The resistor together with the capacitance compose the time constant $\tau = R \cdot C$, which determines the behavior of the filter together with the threshold voltage of the comparator. The behavior of the filter is visualized in Fig. 1.6. If the output $V_{out} = 0$ and the input voltage $V_{in}(t) = V_{dd}$ for all $t \geq 0$ and the initial charge stored on the capacitor is zero, Equation (1.8) describes the voltage on the capacitor.

$$V_x(t) = V_{dd}(t) \cdot \left(1 - \exp\left(-\frac{t}{\tau}\right)\right) \tag{1.8}$$

The comparator which drives the output of the deglitch filter will trip once the voltage $V_x(t)$ reaches the threshold voltage. The elapsed time since $t = 0$ is the rising deglitch time.

$$V_x(t_{deglitch}) \stackrel{!}{=} \frac{1}{2} V_{dd} \tag{1.9}$$

$$\Rightarrow t_{deglitch} = \tau \cdot \ln 2 \tag{1.10}$$

As in the shown configuration the threshold voltage is half of the operating voltage, the rising and falling deglitch times are identical. The falling deglitch time can be calculated in the same fashion – taking into account that the capacitor is discharged starting from V_{dd}. The individual deglitch times can be changed by moving the threshold voltage of the comparator and by adjusting the time constant.

For a realistic deglitch time of around 100 µs the required capacitor and resistor values lead to a large area usage. An implementation using a lower charging current gets by with a smaller capacitor. The necessary small current can be obtained without a resistor by means of a current mirror.

1.1.3.2. Digital Implementation

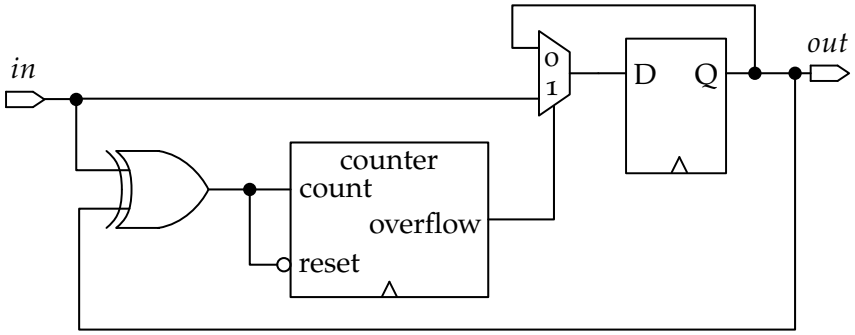

Figure 1.7.: Implementation of a deglitch filter based on a counter.

An example for the digital implementation of a deglitch filter is shown in Fig. 1.7. If the input and output are equivalent the output remains at its value. If the input and output are different the counter starts increasing. In case the input and output become equivalent by either an overflow of the counter or by the input changing its value the counter starts to count from zero if input and output are different once more.

For a synchronous comparator input signal and a counter starting from 0 and overflowing at N, the deglitch time can be calculated as

$$t_{deglitch} = (N+1) \cdot t_{clk}\,. \tag{1.11}$$

To avoid an aliasing effect due to a violation of the sampling theorem, the clock frequency should be more than twice the analog bandwidth of the comparator.

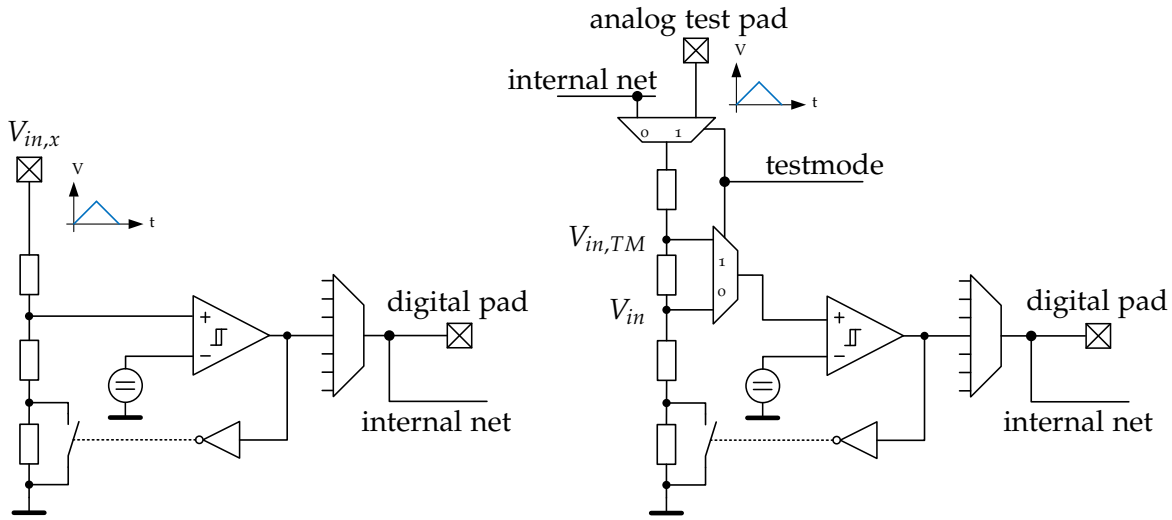

Figure 1.8.: Test concept example for analog comparators: Input net directly available via external pad (left), internal comparator with high voltage input and no direct external connection (right).

1.1.4. Test Concept

Because of possible manufacturing defects, the correct functionality of the voltage monitoring system must be tested after production. For the existing comparator-based voltage monitoring system, an externally applied high-resolution linear rising and falling ramp with low slew rate is used to determine the switching threshold and hysteresis of each comparator. In some situations, the comparator input is not accessible from the outside, which causes additional effort because a unique path solely for testing purposes has to be introduced. This situation is depicted on the right in Fig. 1.8 for a comparator where the threshold voltage is higher than the maximum analog test pad voltage. In this case, an additional tap to the resistive divider is necessary to yield the same threshold voltage for the voltage limited analog test pad input and the operating input voltage. The comparator output must be multiplexed to an external digital test pad because in normal operation the comparator output is directly connected to an internal deglitch filter and hence it is not available externally.

A disadvantage of this test concept is the reduced test coverage, because the signal paths in normal operation and in test mode differ. Consequently, some parts of the system may stay untested. Additionally, because of the slow voltage ramps the long test times create significant costs. The digital deglitch filters can be tested by an automatic scan test which covers all digital circuitry. In contrast, the analog deglitch filters suffer from the same test accessibility problems as internal comparators and hence can only be verified with a high effort.

1.2. Task Description

Based on the system analysis summarized in this chapter, the feasibility of alternative voltage monitoring concepts for system on chip solutions in the automotive environment should

be investigated. The findings should be used to develop a new voltage monitoring concept. Special attention should be paid to the practicability regarding the required silicon area, testability, functional safety and flexibility. Furthermore, a new voltage monitoring concept should reuse as many existing components as possible.

A model of the developed concept should be implemented and simulated to proof the feasibility. Apart from the correct functionality during normal operation, the simulation should also verify that disturbance signals do not cause inadvertent under- or overvoltage warnings. Finally, the results should be discussed and an outlook on future work should be given.

1.3. Structure of This Thesis

This thesis is divided into three parts. Part I begins with the description of the initial situation in chapter 1. The deficiencies of an existing voltage monitoring system are used as a motivation for the development of a new concept. Chapter 2 provides a classification of voltage monitoring systems by their architecture as well as by their time- and amplitude domain properties. Subsequently, the most promising voltage monitoring concept is determined. As it is based on analog-to-digital conversion, the effects of quantization in the amplitude domain and static error measures are explained in chapter 3. Furthermore, the quantization of time and related consequences are explained for uniform and for random sampling.

The concept for a new voltage monitoring system is developed in part II. Chapter 4 starts with an overview of the concept. The architecture based on an analog-to-digital converter and a random sampling scheduler is explained, but also functional safety aspects and the test concept are discussed. The following chapters focus on certain aspects in more detail. Chapter 5 analyses one existing capacitive charge redistribution SAR ADC. Furthermore, the adaptations, which are necessary to use it in the voltage monitoring system, are mentioned. In chapter 6 a new self-test method for this type of ADCs is presented, which provides benefits regarding functional safety and simplified testing. A random sampling scheme for the voltage monitoring system is derived in chapter 7. It also provides the necessary background on pseudo-random number generators based on linear feedback shift registers. The potentially arising problems due to aliasing are discussed at the example of noise and sinusoidal disturbance signals.

Finally, in part III a proof of concept is provided based on a digital simulation. Chapter 8 shows the implementation of all modules required in the digital part of the voltage monitoring system. An automated workflow for the design and analysis of a voltage monitoring system based on a specification is explained in chapter 9. A concrete example for a voltage monitoring system is given in chapter 10. The results of the automatic synthesis and analysis are discussed before a digital simulation serves as a proof of concept.

2. Voltage Monitoring Concepts

This chapter provides a classification of voltage monitoring systems by multiple criteria. Out of all possible criteria, three have been chosen as the most relevant. These are the system architecture, the time domain properties and the amplitude domain properties. After the overview, a feasibility analysis for automotive SoC environments is performed to figure out the best approach for a new voltage monitoring concept.

2.1. Taxonomy for Voltage Monitoring Systems

The voltage monitoring systems are independently classified based on their architecture as well as by their time- and amplitude domain properties. Any combination of one selection in each category results in a valid voltage monitoring system.

2.1.1. Classification by Architecture

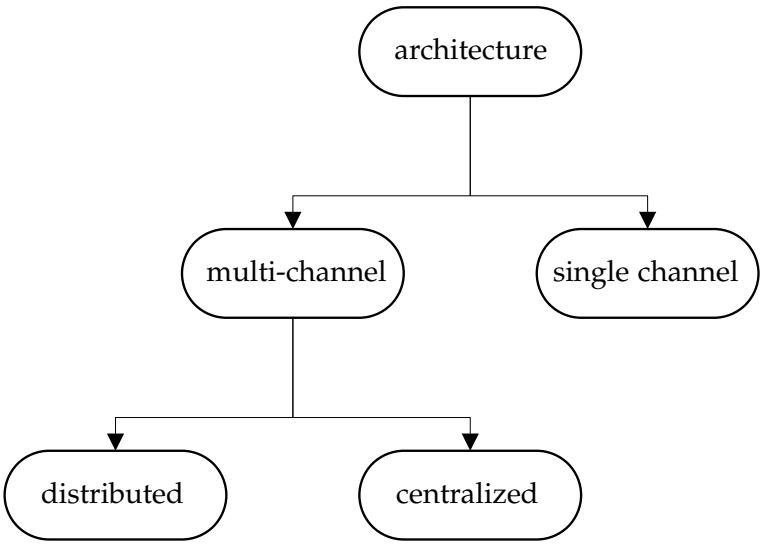

Figure 2.1.: Classification of voltage monitoring systems by system architecture.

The classification by architecture is graphically shown in Fig. 2.1. For a single channel voltage monitoring system there is no choice in the system architecture, while for multi-channel systems there are two possibilities. Either the system consists of distributed entities, which independently monitor one or more channels, or there is one central system with hardwired connections to all necessary voltage rails.

2.1.2. Classification by Time Domain Properties

In the time domain a voltage monitoring system can be either continuous or discrete-time. The continuous time domain requires pure analog circuitry and optional combinatorial logic. In reality all analog signals are in the continuous time domain, hence they must be sampled to gain discrete-time signals. Those signals can be processed by clocked analog circuitry such as switched capacitor circuits, or by combinatorial or sequential digital logic, which requires a further quantization in the amplitude domain.

The sampling process is uniform if the time distance between sampling instants is constant. If, on the other hand, the time distances are not equally spaced but rather randomized in some way, then the sampling method is non-uniform. A summary of the classification in the time domain is provided in Fig. 2.2.

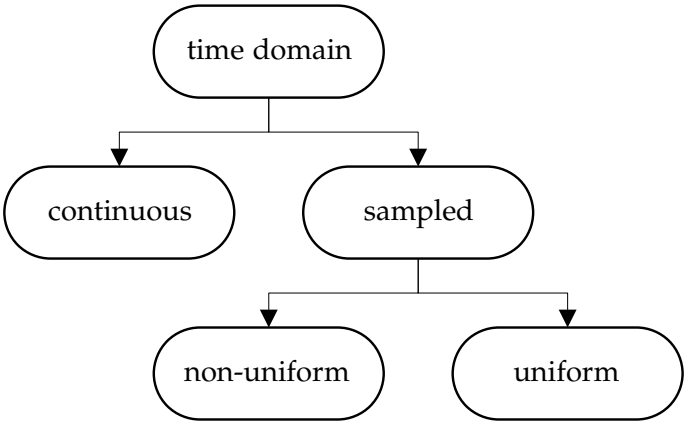

Figure 2.2.: Classification of voltage monitoring systems in the time domain.

2.1.3. Classification by Amplitude Domain Properties

Finally, the voltage monitoring systems can be classified by their amplitude domain properties. If a representative value of the voltage level is required, the amplitude domain must be discrete such that further processing or storing of the information can be done with digital circuitry. If only the crossing of a certain voltage threshold is of importance, the monitoring can be achieved by analog circuitry operating in the continuous amplitude domain as well as by digital circuitry working in the discrete amplitude domain. The case of one threshold can be viewed as quantization with 1-bit resolution. The single threshold detection can also be used in level crossing analog-to-digital converters (ADCs) to perform non-uniform sampling.

All signals in reality are in the continuous amplitude domain. Therefore, to operate in the discrete amplitude domain a conversion has to be performed beforehand. In the amplitude domain, the conversion from continuous to discrete is called quantization. The exact effects of quantization are elaborated in section 3.2. Figure 2.3 shows the classification of the voltage monitoring systems by their amplitude domain properties.

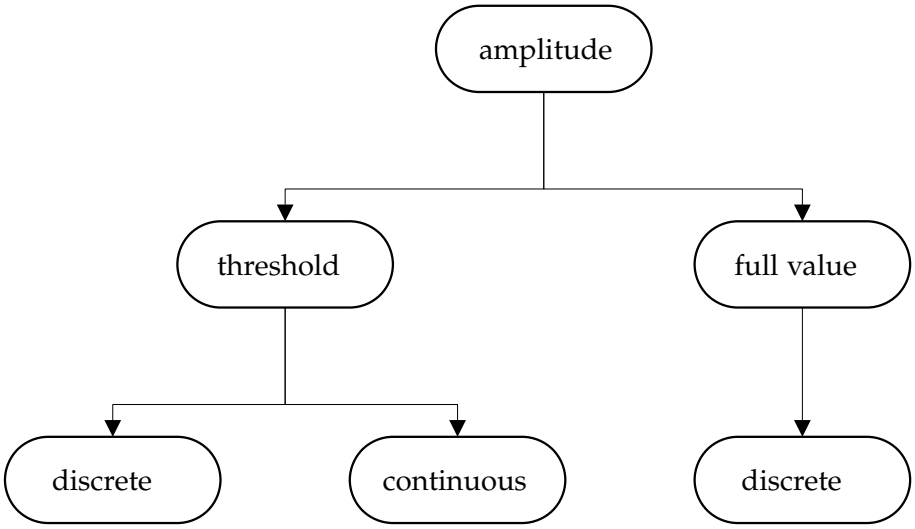

Figure 2.3.: Classification of voltage monitoring systems in the amplitude domain.

2.2. Feasibility for Automotive System On Chip Applications

The feasibility of a voltage monitoring system for an application depends on the requirements in the amplitude and time domain. For usage in integrated circuits, furthermore the silicon area and the testability are important. The conformity with electromagnetic compatibility (EMC) is mandatory, but for applications in automotive environments, the aspects of functional safety must also be considered.

Section 1.1 describes an existing voltage monitoring system for automotive airbag system on chip (SoC) applications. According to the above taxonomy, it can be classified as a continuous-time multi-channel system with a distributed architecture and one continuous-amplitude threshold per channel. In the following sections, the feasibility of selected voltage monitoring concepts is analyzed.

2.2.1. Multiplexed Comparator

Comparator output signals are usually filtered such that electromagnetic disturbances and noise do not cause toggling of the output signal. It seems inefficient to let a comparator monitor one voltage rail continuously even though most gained information is filtered out. Therefore, the idea of using one comparator with a configurable threshold for monitoring of multiple voltage rails comes to mind. Such a concept is depicted in Fig. 2.4. The various input signals are multiplexed to the comparator input and the reference voltage is adapted accordingly such that each monitored signal can have an independent threshold.

The accuracy of the different reference voltages determines the threshold accuracy of the comparator. If they are generated by a resistive or capacitive voltage divider, there is no flexibility to change the threshold without changing the voltage divider or the reference voltage. If a hysteresis is needed, a feedback from the control logic to the reference voltage is required to make the threshold depend on the last comparator output state.

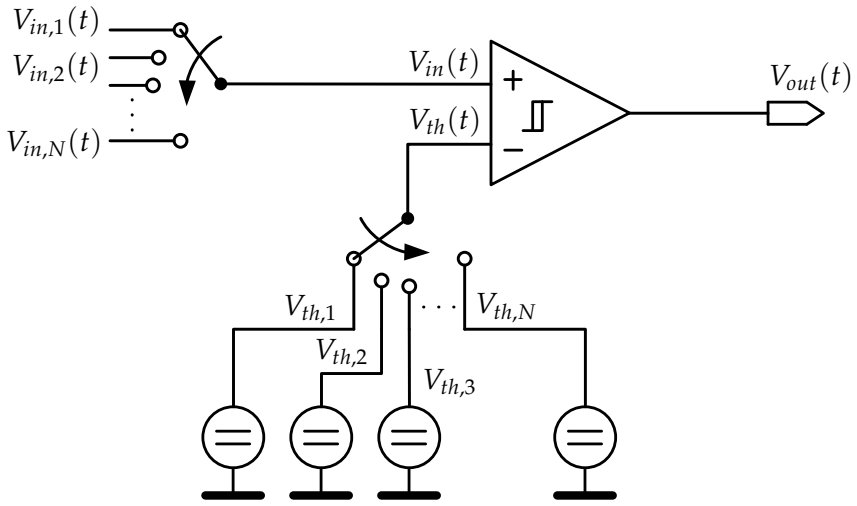

Figure 2.4.: Multiplexed comparator with independent threshold for each channel.

If the multiplexer connects the input channels in a round robin scheme this effectively means that the input signals are uniformly sampled. This could be made more explicit if a sample and hold device was introduced at the input of the comparator. The lower limit for the dwell time at each input is given by the settling time of the comparator t_{settle}. For a total number of N channels this results in an effective sampling frequency of $f_s = \frac{1}{N \cdot t_{settle}}$ per channel. To prevent aliasing, the uniformly sampled input signals must be bandlimited to $\frac{f_s}{2}$ according to the Shannon-Nyquist sampling theorem. The aliasing effect for uniform sampling is explained in more detail in section 3.1.1.

This band limitation can only be achieved by an anti-aliasing filter, because the input bandwidth of the comparator would only suffice as an inherent filter if there was no multiplexer and only one input channel – in this case, there would be no sampling and therefore no need for anti-aliasing filtering anyway. Consequently, additional anti-aliasing filtering is inevitable.

2.2.2. Anti-Aliasing for Multiplexed Systems

To save silicon area it would be tempting to place only one anti-aliasing filter after the multiplexer output in front of the comparator input as depicted on the left in Fig. 2.5. However, the multiplexer performs implicit sampling and hence the aliasing for the individual channels happens already there. A filter after the sampling circuit cannot undo the folding of aliasing components into the baseband by baseband filtering. Furthermore, there is a contradiction regarding the required filter bandwidths. The baseband bandwidth of the anti-aliasing filter would have to be $f_c < \frac{f_s}{2}$ to prevent aliasing for each channel, but the comparator input needs to process all channels in a time multiplex and hence requires a higher input bandwidth.

Therefore, anti-aliasing filtering must be performed on each input channel prior to the multiplexer. This situation is shown on the right in Fig. 2.5. For high-voltage signals in

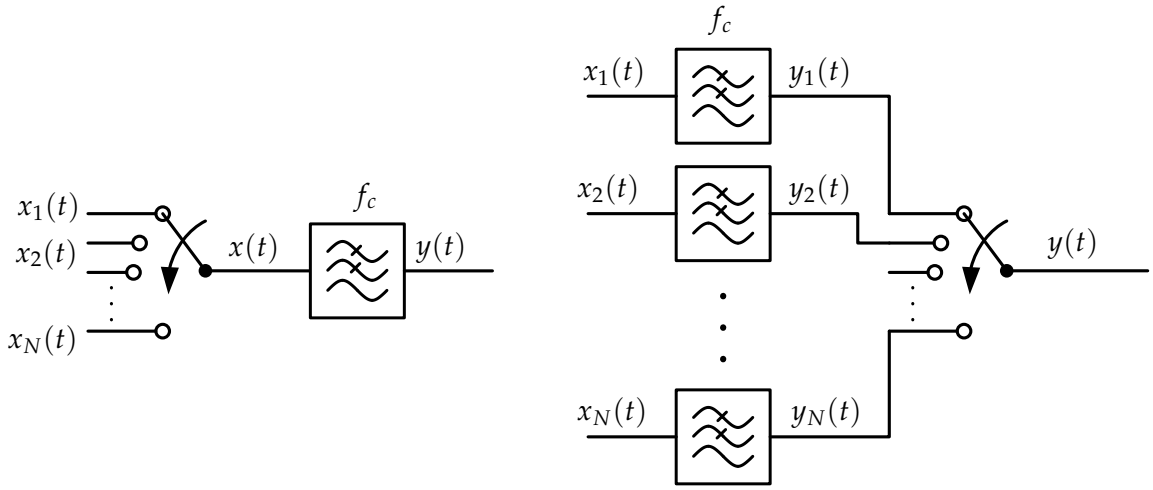

Figure 2.5.: Left: One anti-aliasing filter after the multiplexer. Right: Distinct anti-aliasing filters before the multiplexer.

the range of many tens of volts, the anti-aliasing filter must be composed of high-voltage components. Furthermore, as to not increase the settling or sampling time of the comparator, the anti-aliasing filter must be active. Such an active filter is most likely not smaller than a comparator itself. Therefore, placing a distinct anti-aliasing filter on each channel renders this concept infeasible in terms of silicon area usage. However, multiplexing the channels in a randomized manner avoids anti-aliasing filters completely. The theory behind this non-uniform sampling is treated in section 3.1.2.

2.2.3. Multiplexed Analog-To-Digital Converter

In the previous section a comparator with multiplexed input- and reference voltages is employed. To gain more flexibility, the multiplexed voltage references in Fig. 2.4 could be replaced with a digital-to-analog converter (DAC). The resulting structure resembles the principle of a successive approximation register (SAR) ADC, which consists of a comparator, a feedback DAC and a control loop.

The principle of SAR ADCs is explained in section 5.1. Actually, an ADC based on any architecture could be used to measure a voltage threshold, but an SAR ADC offers the best compromise regarding component count and associated required silicon area, conversion speed and power consumption. While an SAR ADC usually requires N steps to produce an N-bit result, a sophisticated fast compare conversion allows a threshold comparison with a 1-bit result in only one step (see section 5.2.5.2). Hence, for this special application scenario the speed is comparable to a flash ADC, but only one comparator and N reference voltages are required instead of $2^N - 1$ comparators and just as many reference voltages. Hence, the SAR ADC has comparably little power consumption and area usage.

On the left of Fig. 2.6 an ADC with multiplexed inputs and distinct anti-aliasing filters on each channel is depicted. The silicon area can be reduced significantly by saving the anti-aliasing filters at the expense of a more complicated random sampling process. This concept seems

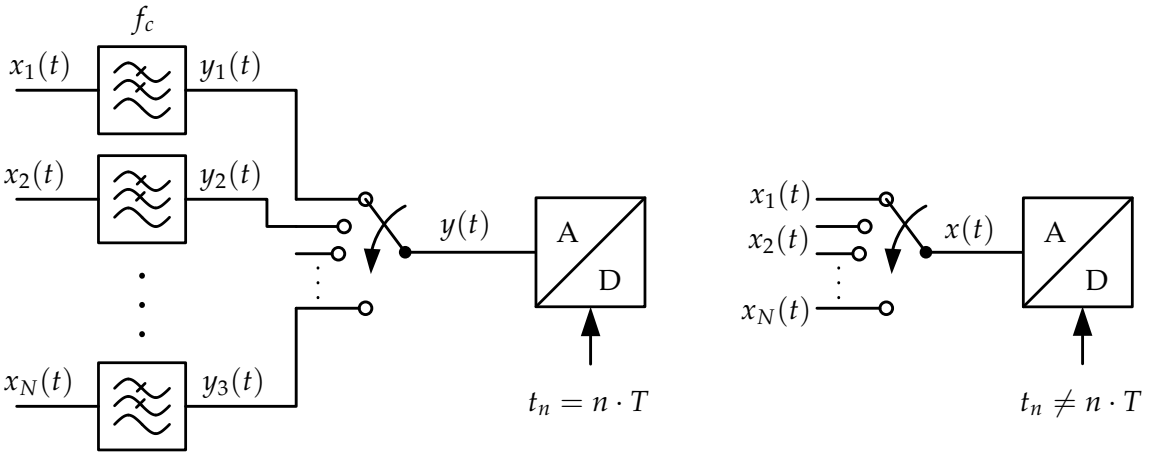

Figure 2.6.: Multiplexed ADC: Uniform sampling (left), nonuniform sampling (right).

the most feasible regarding flexibility and silicon area usage and will be pursued in this thesis. In summary, the chosen concept is a centralized multi-channel system with discrete threshold conversion and non-uniform sampling.

3. Analog-To-Digital Conversion Theory

Analog-to-digital conversion is a fundamental operation in mixed signal systems. It maps signals from the continuous-time continuous-amplitude analog domain to the discrete-time discrete-amplitude digital domain. Figure 3.1 shows a conversion chain for baseband signals. In a first step, a low pass filter restricts the bandwidth of the signal to avoid aliasing. Subsequently, the band-limited signal $V_{lp}(t)$ is sampled at the time points t_n, which means that $V_{sh}(t_n)$ is a discrete-time, continuous-amplitude signal. In the final step, the continuous-amplitude information is quantized to produce the digital output signal $X[n]$. The following two sections will elaborate on quantization in the time- and amplitude domain.

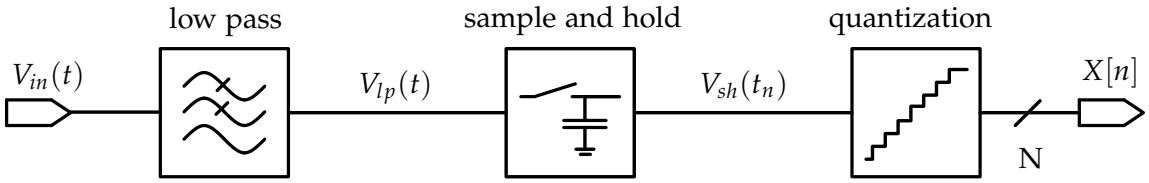

Figure 3.1.: Principle of baseband analog-to-digital conversion (based on [19]).

3.1. Time Quantization

In the processing chain in Fig. 3.1 the first and the second step are related to the quantization of time. Uniform sampling, where the time steps between sampling instants are constant, is the usual case and will be explained in the following section. Section 3.1.2 will then explain the concept behind random sampling and the merit of this approach for certain applications.

3.1.1. Uniform Sampling

The section on uniform sampling is based on [14]. For uniform sampling the distance between sampling instants t_n is constant:

$$t_n = n \cdot T\,, \tag{3.1}$$

with T denoting the constant period.

3.1.1.1. Time Domain

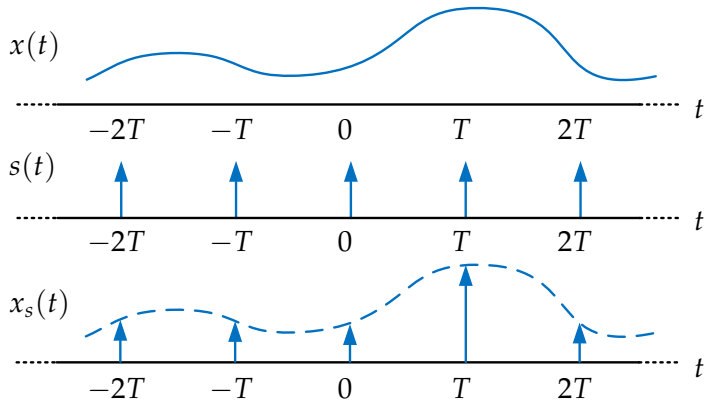

Figure 3.2.: Uniform sampling in the time domain: continuous signal, sampling impulse train and sampled signal (based on [14]).

In the time domain, the process of sampling can be mathematically described by the multiplication of an impulse train with the input signal. Figure 3.2 summarizes the time domain sampling process for an arbitrary signal. The impulse train is given by

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)\,, \tag{3.2}$$

where $\delta(\ldots)$ stands for the Dirac delta function. The multiplication of $s(t)$ with the input signal $x(t)$ leads to the sampled input signal $x_s(t)$. The masking property of the Dirac delta permits to only evaluate the input signal at the time instants $t_n = nT$ because by definition of the Dirac delta $x_s(t)$ is zero for all $t \neq nT$.

$$x_s(t) = x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT) \tag{3.3}$$

3.1.1.2. Frequency Domain

Viewing the uniform sampling process in the time domain paints a very vivid picture, but only the frequency domain considerations give insight on the requirements to be met such that the continuous-time input signal can be reconstructed from the sampled signal. For the analysis, the Fourier transform as given in (3.4) is applied to all time domain signals of the previous section.

$$\begin{aligned} X(j\Omega) &= \mathcal{F}\{x(t)\} \\ &= \int_{-\infty}^{\infty} x(t) e^{-j\Omega t}\,\mathrm{d}t \end{aligned} \tag{3.4}$$

The Fourier transform of the sampling impulse train given in (3.2) with a sampling frequency of $\Omega_s = \frac{2\pi}{T}$ yields

$$S(j\Omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s)\,. \tag{3.5}$$

The multiplication of the sampling impulse train and the input signal in the time domain equals a convolution of the respective spectra in the frequency domain as defined in (3.7). This produces (3.8) for the spectrum of the sampled signal $X_s(j\Omega)$, which clearly shows the periodicity of the signal spectrum due to the convolution with the sampling impulse train.

$$X_s(j\Omega) = \frac{1}{2\pi}(X * S)(j\Omega) \tag{3.6}$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\tau) \cdot S(j\Omega - \tau)\,\mathrm{d}\tau \tag{3.7}$$

$$= \frac{1}{T} \sum_{k=-\infty}^{\infty} X(j(\Omega - k\Omega_s)) \tag{3.8}$$

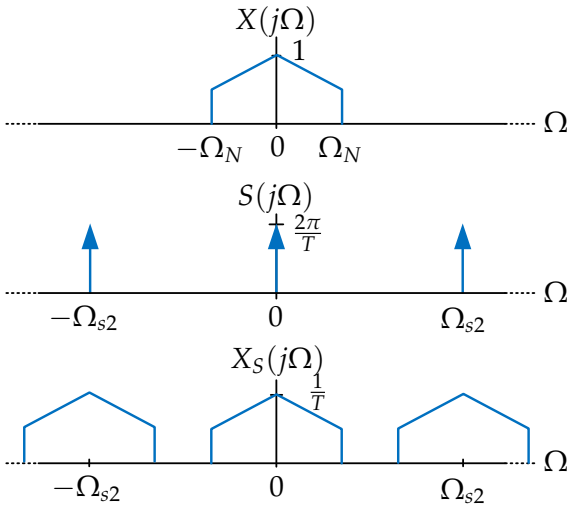

Figure 3.3.: Uniform sampling in the frequency domain: input signal spectrum, impulse train spectrum, and spectrum of the sampled signal (based on [14]).

The process of uniform sampling in the frequency domain is shown for different sampling frequencies in Fig. 3.3 and Figure 3.4. In Fig. 3.3 the sampling frequency Ω_{s2} is sufficiently high in order to prevent overlapping of the convolved spectrum, whereas the lower sampling frequency Ω_s in Fig. 3.4 leads to an overlapping in the convolved spectrum. This effect is called *aliasing* and cannot be reversed, which means that the original input signal cannot be recovered from the samples.

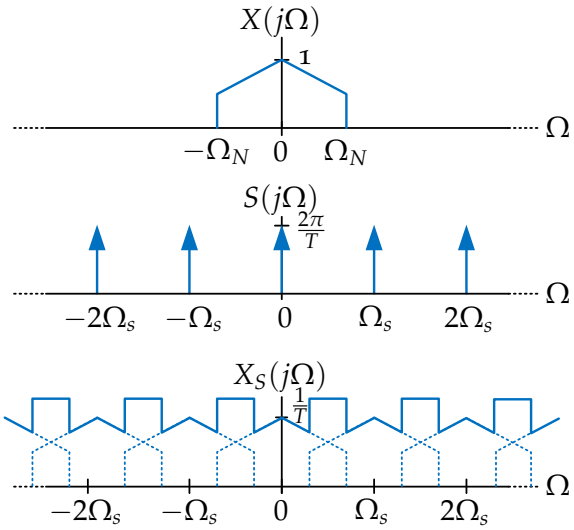

Figure 3.4.: Uniform sampling in the frequency domain with aliasing: input signal spectrum, impulse train spectrum, and spectrum of the sampled signal (based on [14]).

3.1.1.3. Nyquist-Shannon Sampling Theorem

The observations of the aliasing effect with respect to the sampling frequency and bandwidth of the signal can be expressed in a general form as what is called the Nyquist-Shannon sampling theorem. A signal $X(j\Omega)$ which is bandlimited with a bandwidth of Ω_N such that

$$X(j\Omega) = 0 \quad \forall |\Omega| \geq \Omega_N \tag{3.9}$$

can be uniquely reconstructed given the samples $x[t_n] = x(nT)$ which are obtained with a sampling frequency of $\Omega_s = \frac{2\pi}{T}$ if

$$\Omega_s \geq 2\Omega_N \,. \tag{3.10}$$

3.1.1.4. Anti-Aliasing Filter

The prevention of the aliasing effect for sampling of arbitrary input signals requires a forced limitation of the bandwidth. As shown in Fig. 3.1, an anti-aliasing filter is therefore placed in the processing chain before the sampling circuitry. Hence, the bandlimited input signal $V_{lp}(t)$ can be uniquely reconstructed if the sampling frequency is sufficiently high according to (3.10).

Ideally, the transfer function of the anti-aliasing filter would be a box function, but in reality, this noncausal filter cannot be implemented. Accordingly, analog filters with lower roll-off factor have to be used, which means that the signal must be sampled at a higher frequency than theoretically required by the sampling theorem.

3.1.1.5. Jitter

So far the sampling point sequence t_n has been considered as ideal, but in reality all clock signals suffer from deviations in their timing. This so-called jitter has a detrimental influence on the ADC performance. The sampling instants can be mathematically modeled as

$$t_n = nT + \varepsilon(nT)\,, \tag{3.11}$$

where $\varepsilon(\ldots)$ represents the jitter function. For a sine input $u(t) = \hat{u}\sin(\omega t)$ the influence of aperture jitter can be estimated by a linear approximation. The highest slope of the signal is used to estimate the amplitude measurement error Δu: [19]

$$\max\left|\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right| = \left|\frac{\mathrm{d}u(t)}{\mathrm{d}t}\right|_{t=k\cdot\frac{\pi}{\omega},k\in\mathbb{Z}} \tag{3.12}$$

$$= \hat{u}\cdot\omega \tag{3.13}$$

$$\Rightarrow \max|\Delta u| = \hat{u}\cdot\omega\cdot\max_t\{|\varepsilon(t)|\}, \tag{3.14}$$

where the jitter function should only be evaluated at the times $k\cdot\frac{\pi}{\omega}, k\in\mathbb{Z}$ when the slope of the sinusoidal is maximal, but taking into account an unknown constant phase the worst case jitter has to be used. In order to keep the resulting amplitude error smaller than the least significant bit (LSB) voltage V_{lsb}, the maximum jitter must be restricted to

$$\max_t|\varepsilon(t)| < \frac{V_{lsb}}{\hat{u}\omega}\,. \tag{3.15}$$

While for uniform sampling the jitter has a negative influence on the outcome, the next section will show that under some circumstances an intentionally introduced jitter can be advantageous.

3.1.2. Randomized Sampling

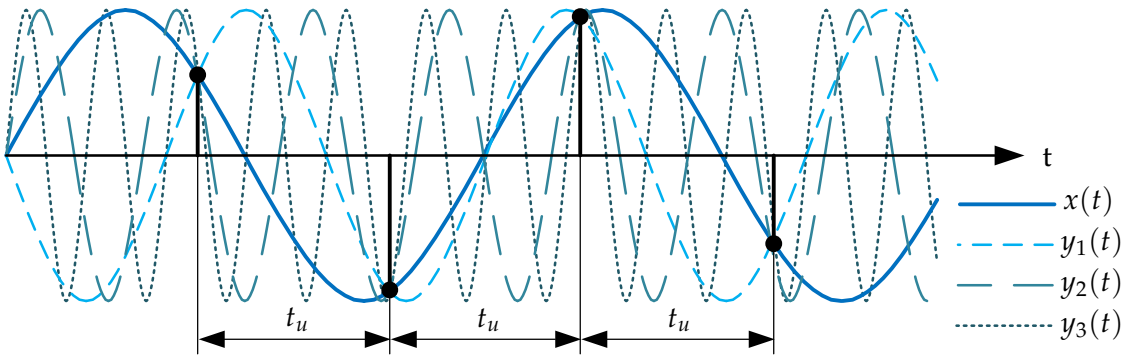

Figure 3.5.: Uniform sampling of $x(t)$ with possible alias components $y_i(t)$ in the time domain. (based on [3]).

As shown in the previous section, the uniform sampling method leads to aliasing if the sampling frequency is not at least twice the signal bandwidth. The question arises if this condition is just sufficient, or if it is a necessary condition for all alias-free sampling processes.

In a situation when the Nyquist-Shannon sampling theorem is potentially violated, the reconstruction of the signal from the observed uniform samples suffers from ambiguity. The situation is shown in Fig. 3.5. The continuous line shows the unique signal that can be reconstructed in case the sampling theorem is adhered to. If the bandwidth of the signal is not guaranteed to be limited to the Nyquist frequency, the reconstruction suffers from ambiguity as any of the shown dashed and dotted signals could have created the sampled sequence. To overcome this ambiguity, the uniform sampling point grid has to be abandoned. This technique is called randomized sampling – the sampling point instants are not predefined but rather determined by a random process. The properties, which the random process has to fulfill in order to guarantee aliasing-free sampling without bandlimitedness of the input signal, are explained in the next sections.

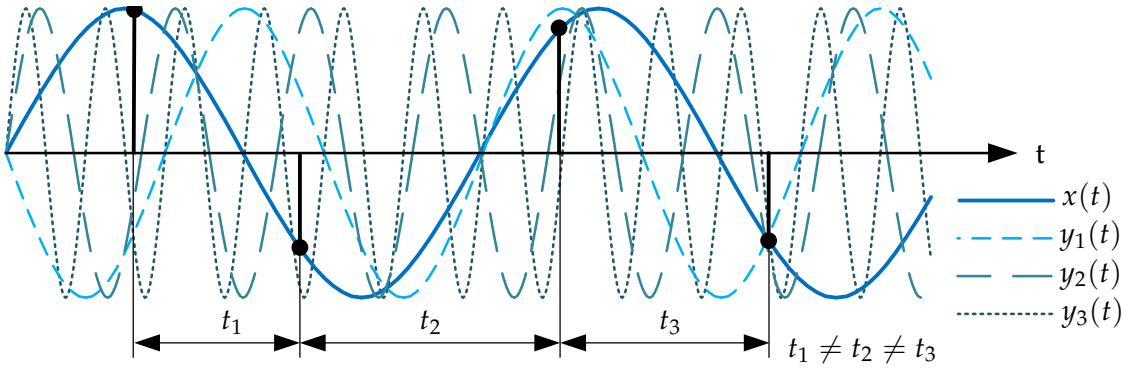

Figure 3.6.: Randomized sampling of $x(t)$ without aliasing components ambiguity in the time domain. (based on [3]).

An example for randomized sampling of the same signal as shown in Fig. 3.5 is depicted in Fig. 3.6. The previously observed ambiguity of potential aliases is resolved – only the unique sine wave $x(t)$ fits through the nonuniform sampling sequence. Consequently, if the random sampling process meets certain criteria (which are explained in sections 3.1.2.1 to 3.1.2.3), there exist no aliases with any phase or frequency up to a certain frequency.

Therefore, the idea behind random sampling is to perform sub-Nyquist rate sampling without aliasing. A complete overview on methods for generating randomized sampling point functions can be found in [3] or [12]. The following sections follow [12] to explain selected random sampling methods and their properties regarding aliasing suppression.

3.1.2.1. Additive Random Sampling

Additive random sampling (ARS) is based on (3.16), where each new sampling time instant is additive to the previous with a random variable τ.

$$t_{n+1} = t_n + \tau \tag{3.16}$$

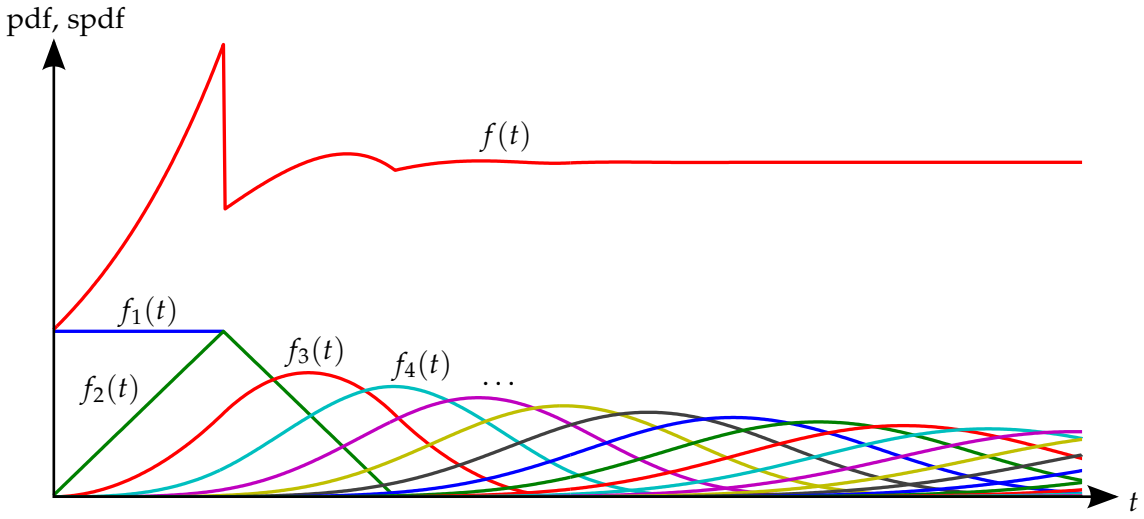

Figure 3.7.: Additive Random Sampling: Transient behavior of the sampling point density function (spdf) $f(t)$ and the probability density functions (pdf) $f_n(t)$ with $\tau \sim \mathcal{U}(0, t_x)$ [12].

Let $f_n(t)$ be the probability density function (PDF) of the sampling time t_n for sample number n and $f_\tau(t)$ be the PDF of the random variable τ. Without loss of generality, for $n \in \mathbb{N}\{0\}$ and $f_0(t) = f_\tau(t)$, the recursive equation (3.16) can be expanded to form a closed expression:

$$t_{n+1} = t_n + \tau = (t_{n-1} + \tau) + \tau = \ldots = (n+1)\tau\,. \tag{3.17}$$

Addition of independent random variables is equivalent to a convolution of the respective PDFs. Accordingly, (3.17) can be expressed in terms of the involved PDFs as

$$f_{n+1}(t) = f_n(t) * f_\tau(t) = f_{n-1}(t) * f_\tau(t) * f_\tau(t) = \ldots = \underbrace{f_\tau(t) * f_\tau(t) * \ldots * f_\tau(t)}_{n \text{ convolutions}}\,. \tag{3.18}$$

According to the common version of the *central limit theorem*, the sum of a sufficiently large number of independent and identically distributed (i.i.d.) random variables will converge towards a normal distribution [15]. This holds independently of the actual distribution of the random variable. Consequently, the probability density $f_n(t)$ tends towards a normal distribution for sufficiently large n:

$$\lim_{n\to\infty} f_n(t) \sim \mathcal{N}(\mu, \sigma^2)\,. \tag{3.19}$$

This effect can be verified by inspecting Fig. 3.7, where the initial probability density function has a uniform distribution $f_1(t) \sim \mathcal{U}(0, 100)$, but the subsequent PDFs tend more and more towards a normal distribution. Even though (3.19) is independent of the distribution of τ, the convergence speed until a normal distribution is approximated depends on the initial PDF.

The sampling point density $f(t)$, which is the probability that a sample will be taken at a certain point in time, is the sum of all the PDFs $f_n(t)$:

$$f(t) = \sum_{n=1}^{\infty} f_n(t)\,. \tag{3.20}$$

According to the *law of large numbers* the average of a sufficiently large number of samples of i.i.d. random variable will converge towards the expected value [15]. Hence, the sampling point density will tend towards a constant factor: [12]

$$\lim_{t\to\infty} f(t) = \frac{1}{E[\tau]}\,. \tag{3.21}$$

The constant sampling point density is a sufficient condition for alias-free sampling [12]. This ensures that the sampling function is uncorrelated with any signal and hence no repetitive patterns can overlap to form aliasing. Even though the sampling is free of aliasing in the classical sense, it can still be observed as an additional noise floor. For the uniformly distributed τ the aliasing noise floor is not flat [12].

The settling time until the constant value is reached depends on the PDF of the random variable τ. While the uniform distribution assumed in Fig. 3.7 takes some time to settle, the exponential distribution immediately starts with the constant sampling point density. Additionally, it produces a flat aliasing noise floor [12].

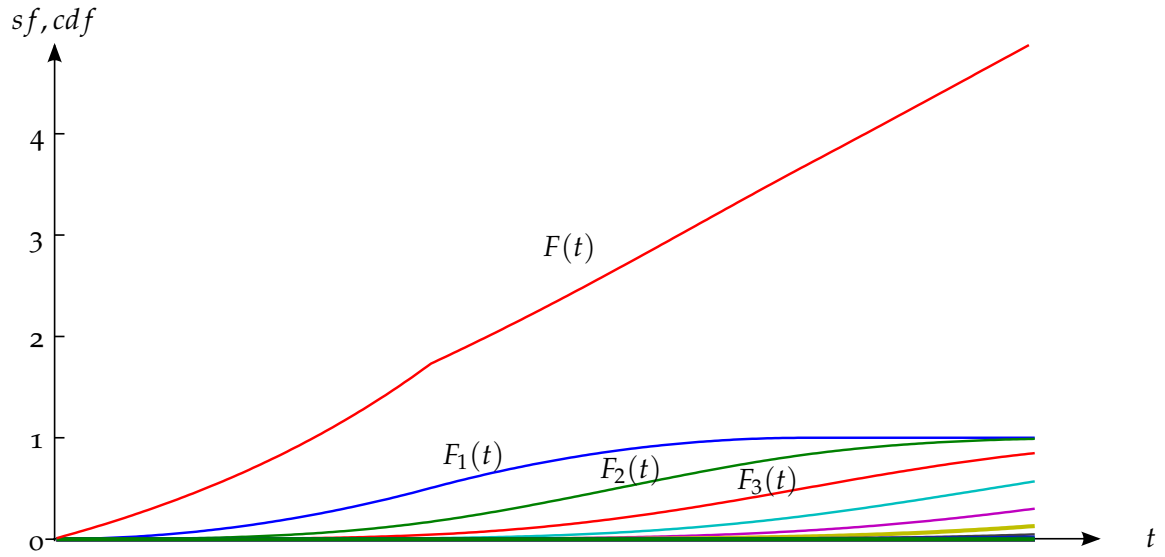

Figure 3.8.: Additive Random Sampling: Transient behavior of the sampling function (sf) $F(t)$ and the cumulative density functions (cdf) $F_n(t)$ [3].

The sampling function, which is the integral over the sampling point density

$$F(t) = \int_{-\infty}^{\infty} f(t)\,\mathrm{d}t, \tag{3.22}$$

is shown in Fig. 3.8. It represents the expected number of samples taken up to a certain point in time.

3.1.2.2. Jittered Random Sampling

In the jittered random sampling (JRS) model the sampling time instants are defined on a regular grid nT with additive jitter τ:

$$t_n = n \cdot T + \tau \,. \tag{3.23}$$

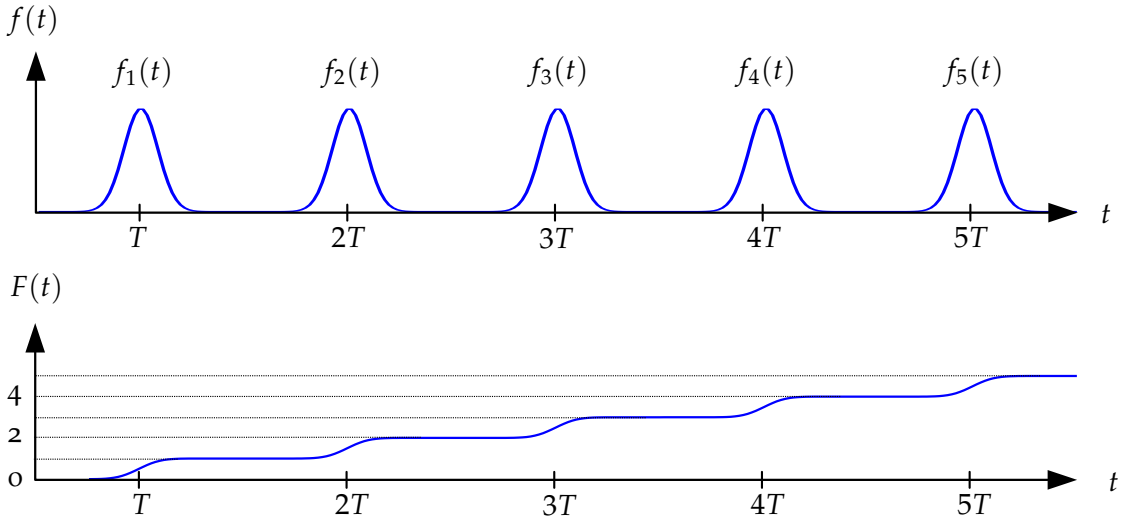

Figure 3.9.: Jittered random sampling scheme with Gaussian jitter.

The jitter τ is a random variable which is i.i.d. with $f_\tau(t)$. Therefore, the distribution of each sampling time t_n can be specified as:

$$f_n(t) = f_\tau(t - nT) \,. \tag{3.24}$$

Again the sampling point density can be defined as the sum of the PDFs of the distributions of the single sampling time instants $f_n(t)$.

$$f_n(t) = \sum_{n=1}^{\infty} f_n(t) = \sum_{n=1}^{\infty} f_\tau(t - nT) \tag{3.25}$$

The random sampling density function $f(t)$ for a Gaussian jitter with $6\sigma = T$ is depicted in Fig. 3.9. The Gaussian PDFs $f_n(t)$ are clipped at $nT \pm 3\sigma$ in order to keep the sampling scheme causal. Apparently, such a scheme does in general not lead to a constant sampling point density – and therefore it cannot completely avoid aliasing [12]. The only exception here is for a uniformly distributed random variable $\tau \sim \mathcal{U}(-\frac{T}{2}, \frac{T}{2})$ [12].

At the bottom of Fig. 3.9, the sampling function as defined in (3.22) is depicted.

3.1.2.3. Random Skip Sampling

Random skip sampling (RSS) represents the bridge between uniform and non-uniform sampling. The following considerations start out with the well-known discrete uniform sampling impulse function $s[n]$ with time granularity T and the associated uniform sampling frequency $f = \frac{1}{T}$:

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{3.26}$$

$$s[n] = s(nT) = \sum_{i=-\infty}^{\infty} \underbrace{\delta[(n-i)T]}_{1 \text{ for } i=n} = 1, \tag{3.27}$$

where $\delta[\ldots]$ is the Kronecker delta function.

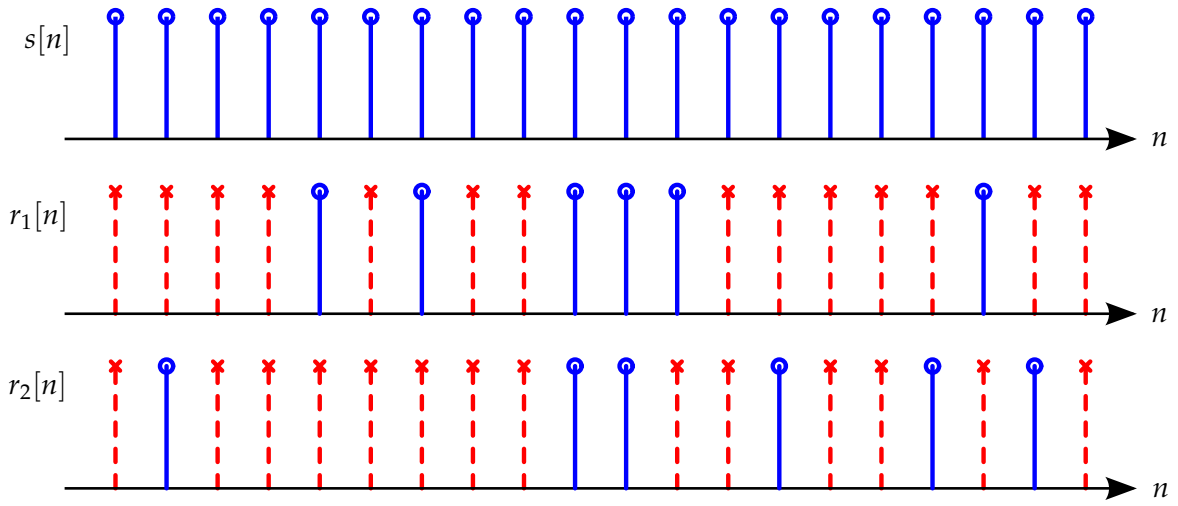

Figure 3.10.: Random skip sampling in the time domain with two realizations of the random process for skip probability $q = 0.7$.

As the name RSS already suggests, the uniform sampling time instants are skipped according to a binary discrete random variable ρ with the probability mass function (PMF)

$$f_\rho[x] = \begin{cases} q & \text{if } x = 0 \\ p & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{3.28}$$

where q is the skipping probability and $p = 1 - q$ is the sampling probability. The random skip sampling process $r[n]$ is then defined as

$$r[n] \sim \rho\,. \tag{3.29}$$

Two realizations of a random skip sampling process $r[n]$ are shown in Fig. 3.10 for a skip probability of $q = 0.7$. The uniform sampling process is aliasing-free for signals which are bandlimited to $[-\frac{1}{2T}, \frac{1}{2T}]$. The same restriction applies to RSS because skipping of samples

does not change the underlying sampling time granularity. The skipping probability q does not change the aliasing frequency, but linked with the sequence length it has an influence on the aliasing noise floor. A formal proof can be found in [12].

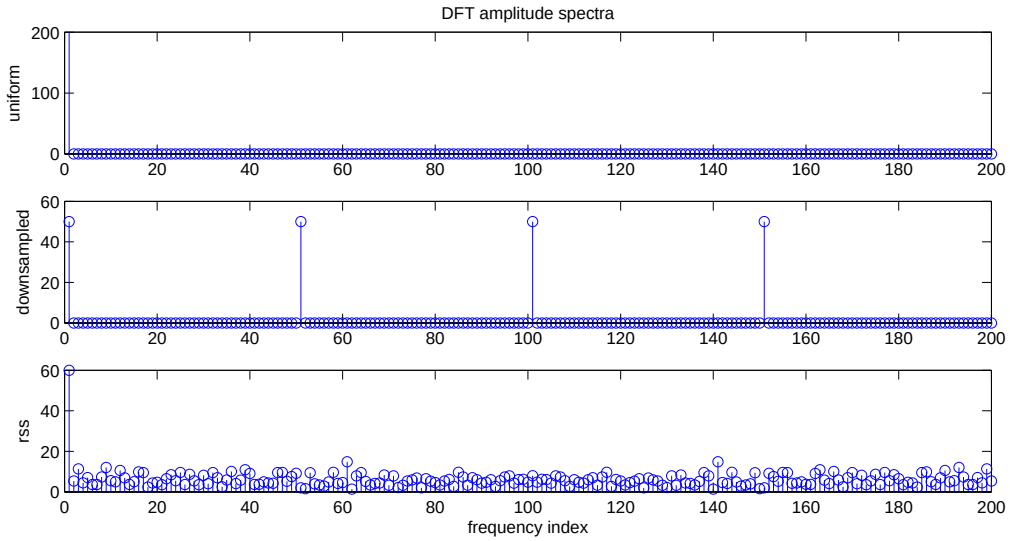

Figure 3.11.: Discrete Fourier transform amplitude spectrum of a uniform sampling sequence, a uniform skip sequence and a random skip sequence with probability $q = 0.75$ and a length of 200 each.

The discrete Fourier transform (DFT) of a uniform sampling sequence is depicted in the top diagram of Fig. 3.11. It has a single peak in the first frequency bin. When a signal is sampled (multiplied) with this function, this is equal to a convolution in the frequency domain. The single peak means that the sampled function would not suffer from any change.

In contrast, the middle diagram shows the spectrum of a uniform skip sequence. Based on the above uniform sequence, only every fourth signal sample is taken in a regular pattern, the rest is forced to zero (downsampling). As a result, in the spectrum three additional peaks appear. The convolution of this spectrum with a signal spectrum would lead to aliasing if the signal is not sufficiently bandlimited to prevent overlapping of components. The effective sampling frequency is reduced to only one fourth of the underlying uniform sequence's sampling frequency.

The spectrum at the bottom is for a realization of an RSS sequence with a length of 200 and a skip probability of $q = 0.75$, which gives the same number of samples as for the middle diagram. However, there is one dominant peak at the first frequency bin and the remaining spectrum is noisy. Therefore, it is very similar to the spectrum of the uniform sampling spectrum in the top diagram. In a convolution, the noise leads to an aliasing noise floor, but there is no aliasing in the classic sense. Hence, the RSS sequence provides the same effective Nyquist frequency as the underlying uniform sequence, but with fewer samples.

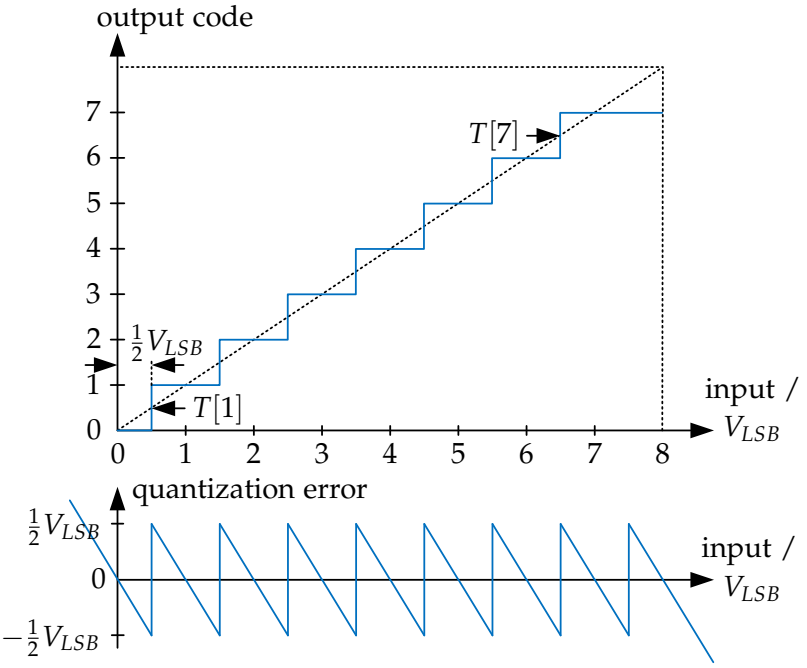

Figure 3.12.: Ideal characteristic of a 3-bit unipolar amplitude quantizer and associated quantization errors (based on [11]).

3.2. Amplitude Quantization

The quantization of the amplitudes is the second step in the conversion process after the time quantization. A continuous-amplitude signal is mapped to one code in a finite set of possible codes. Figure 3.12 shows the quantization characteristic of a 3-bit ADC, which qualitatively applies to ADCs with any resolution. The dotted diagonal line represents the ideal transfer characteristic, which is approximated by the staircase function of the actual transfer characteristic. The ideal minimum voltage resolution of a N-bit ADC is

$$V_{LSB} = \frac{V_{FSR}}{2^N}, \tag{3.30}$$

with V_{FSR} denoting the full scale range (FSR) of the converter, that is the difference between the maximum and minimum operating voltage of the ADC. In the figure not the mid-tread, but the mid-riser convention is used to define the transition levels $T[i]$. The transition levels $T[i]$ are defined as the input voltages that lead to half of the output codes being $\geq i$ and half of them being $< i$.

The quantization error, which is inherent to any amplitude quantization, is shown at the bottom of the graph. It is defined as the difference between the analog input voltage and the equivalent voltage of the output code [11]:

$$error_{quantization} = DAC(ADC(V_{in})) - V_{in}, \tag{3.31}$$

where $ADC(\ldots)$ denotes the ideal analog-to-digital conversion and $DAC(\ldots)$ is the ideal digital-to-analog conversion function.

3.2.1. Quantization Noise

The quantization noise is often modeled as an uncorrelated noise source, but actually, this is only allowed if one of the Quantizing Theorems by Widrow is fulfilled. The characteristic function is the inverse Fourier Transform of a random variable's PDF. The first theorem states that if the characteristic function at the input of the quantizer is bandlimited such that $\Phi_x(u) = 0$ for $|u| > \frac{\pi}{q}$, where q is the quantization step size, the original PDF at the input can be derived from the PDF of the quantized output. If this holds, the quantization noise can be modeled with uniform distribution between $-V_{LSB}/2$ and $V_{LSB}/2$ and a variance of: [20]

$$\sigma^2 = \frac{V_{LSB}^2}{12}. \tag{3.32}$$

The bandlimitedness can be ensured by multiplication of the input's characteristic function with a sufficiently bandlimited function, which in terms of PDFs means a convolution. The convolution of two PDFs is equivalent to the addition of two independent random variables. This means the fulfillment of the Quantizing Theorem condition can be achieved by adding an appropriate noise to the input signal. This concept is known as dithering. [20]

3.2.2. Static Error Measures

Apart from the unavoidable quantization error, real ADCs show further deviations from the ideal transfer curve. The following error measures for ADCs are defined in the IEEE standard 1241 [11]. Dynamic errors are not considered here, because the amplitude measurement accuracy of the voltage monitoring system only matters for slowly changing signals. The most relevant linear static error measures are offset and gain, which are discussed in the following section. The static nonlinear errors are dealt with in the subsequent section.

3.2.2.1. Offset and Gain

The overall equation describing the conversion behavior of the ADC including the influence of gain G and offset V_{OS} is

$$V_{OS} + G \cdot T[k] + \varepsilon[k] = V_{LSB} \cdot (k-1) + T_1, \tag{3.33}$$

where $\varepsilon[k]$ is the residual error at code step k due to nonlinearity and T_1 is the ideal transition voltage of the first code.

Offset and gain errors can be either independently based or terminal based. The first method minimizes the mean squared errors $\varepsilon[k]^2$, while the latter only uses the first and last transition information. The equations for the independently based method may be found in [11], while the simpler equations for the terminal based method are explained now.

The gain G is defined as

$$G = \frac{V_{LSB} \cdot (2^N - 2)}{T[2^N - 1] - T[1]}, \tag{3.34}$$

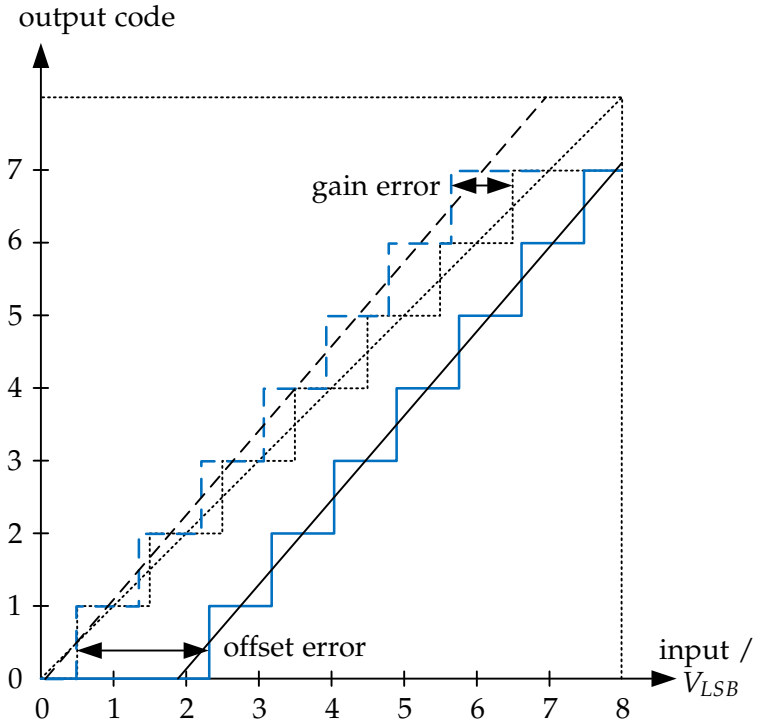

Figure 3.13.: Example for offset- and gain error in an ADC. Dotted: ideal curve, continuous: actual curve, dashed: offset-compensated curve.

and the offset V_{OS} as

$$V_{OS} = T_1 - G \cdot T[1]\,. \tag{3.35}$$

From these definitions the offset and gain error can be derived. They are marked in Fig. 3.13 for an arbitrary transfer curve. In the specification of an ADC they are usually given in units of the LSB voltage or as a percentage of the FSR voltage.

3.2.2.2. Linearity Errors

In the compensated transfer curve, the offset- and gain error are eliminated. The remaining deviations from the ideal transfer curve are due to linearity errors. Figure 3.14 shows a compensated transfer curve of an arbitrary ADC. The ideal transfer characteristic and the ideal gain line are represented by the dashed line. The nonlinearity can be clearly seen by inspection of the interpolated curve through the middle of the treads. The next two sections deal with the integral and differential nonlinearities, which are closely related.

Integral Nonlinearity The integral nonlinearity (INL) in units of LSB is defined for each output code k as

$$INL[k] = \frac{G \cdot T[k] + V_{OS} - T_{nom}[k]}{V_{LSB}}\,. \tag{3.36}$$

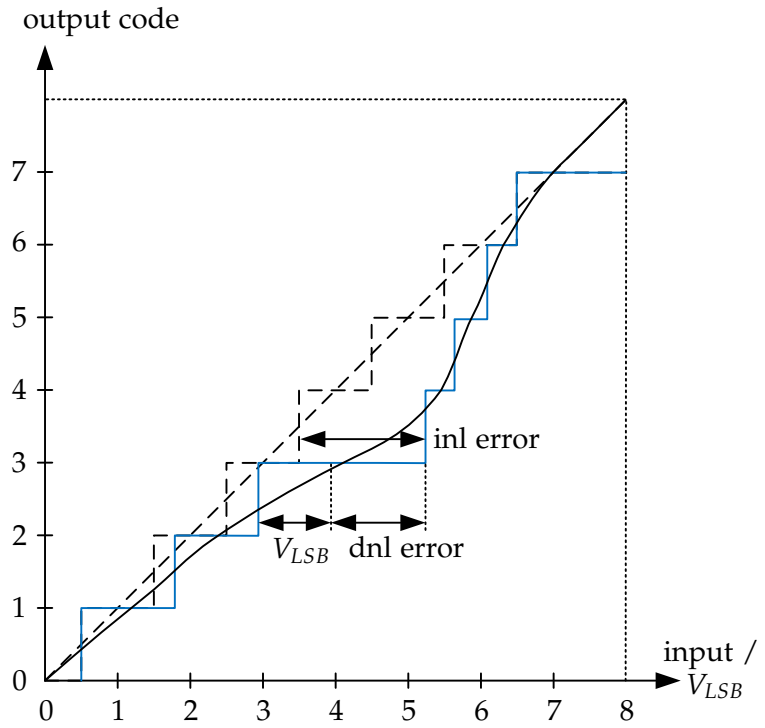

Figure 3.14.: Examples for differential and integral nonlinearity errors in an ADC.

It is not defined for missing codes or non-monotonic codes. Typically, in the datasheet of an ADC only the minimum and maximum INL error is defined. Graphically, in Fig. 3.14 the INL error is given by the deviation from the compensated code step and the ideal code step.

Differential Nonlinearity The differential nonlinearity (DNL) in units of LSB is defined for each output code k as

$$DNL[k] = \frac{W[k] - V_{LSB}}{V_{LSB}} = \frac{G \cdot (T[k+1] - T[k]) - V_{LSB}}{V_{LSB}}, \tag{3.37}$$

where $W[k]$ is the width of the respective tread. The widths of the top and bottom steps are not defined. A code is called a *missing code* if the stair width is less than or equal to 10 % of the nominal value:

$$DNL[k] \leq -0.9\,. \tag{3.38}$$

In Fig. 3.14 the DNL error is clearly marked as the difference between the actual width of a tread and the ideal tread width. Per definition the compensated transfer curve spans the whole operating range and therefore the sum of all DNL values is zero:

$$\sum_k DNL[k] \stackrel{!}{=} 0\,. \tag{3.39}$$

Finally, the INL can be given as a sum over the DNL values:

$$INL[k] = \sum_{i=1}^{k-1} DNL[k] \,. \tag{3.40}$$

Part II.

Random Sampling Voltage Monitoring Concept

4. Concept Overview

A feasibility analysis of various voltage monitoring concepts was conducted in chapter 2 and resulted in the selection of a system based on a central ADC with multiplexed inputs and random sampling (see section 2.2.3). This chapter provides an abstract overview over all required components of the system. Furthermore, the functional safety aspects and the test concept are discussed. The subsequent chapters go into the details of the ADC, the test concept and the random sampling scheme in connection with timing constraints.

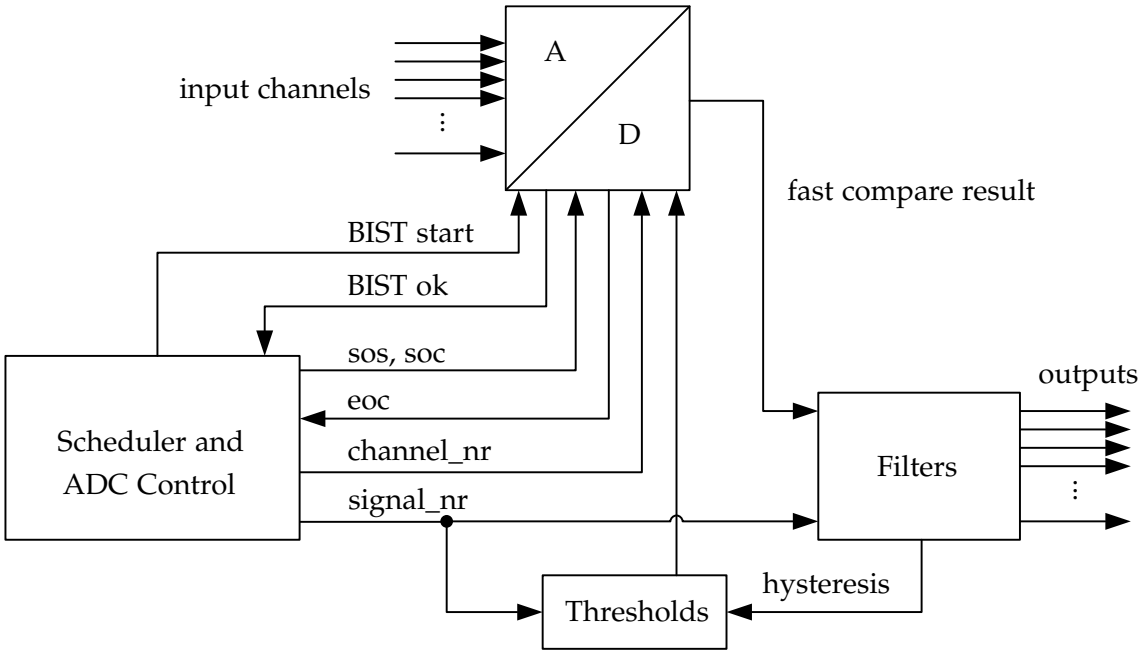

Figure 4.1.: Overall concept for the voltage monitoring system.

Figure 4.1 shows the top-level, where the core is a charge redistribution SAR ADC with M multiplexed input channels. The ADC works in the capacity of a multiplexed comparator with selectable threshold voltage. The scheduler and ADC control block operates the ADC by randomly selecting an input channel of the ADC and successively starting the sampling and the conversion processes. The threshold code for the voltage comparison in the ADC is derived from a lookup table (LUT). The code depends on the signal number and, for the purpose of providing a hysteresis, also on the current output state of a signal. Each output signal consists only of one bit, which is a filtered result based on multiple ADC threshold comparisons. It signifies whether the input signal is above or below the threshold, but it also depends on the history of measurement results. Furthermore, the ADC provides a self-test feature which can be enabled by the scheduler.

4.1. Channels and Output Signals

The ADC has multiple input channels for analog signals. Depending on the total number of input channels to the ADC, each channel is represented by a unique binary code word with the required number of bits. As shown in Fig. 4.2, each channel has at least one, but possibly more associated output signals.

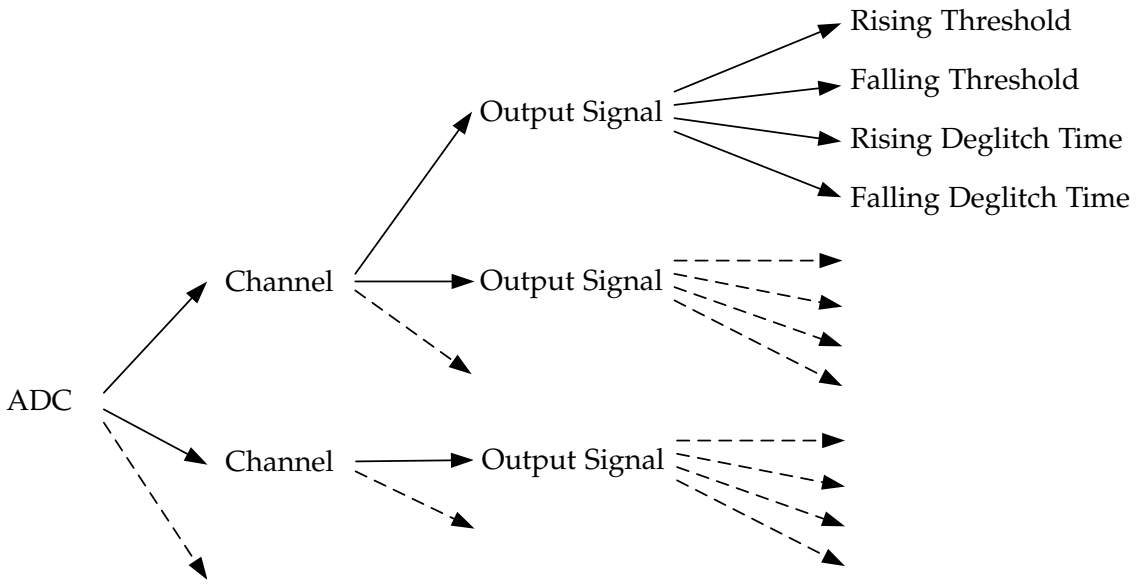

Figure 4.2.: Hierarchy of channels, signals and thresholds.

An output signal is a virtual representation of a discrete comparator in the already existing distributed voltage monitoring system introduced in section 1.1.1. Hence, similar to a comparator, it has independent falling and rising threshold codes – their difference is the hysteresis. The hysteresis increases the immunity against noise when the input voltage is close to the threshold voltage.

The filter function of an output signal is defined by a pair of rising and falling deglitch times. They specify for how long the output of the comparator must remain stable for the filter output to change. Different rising and falling times can be used to adjust to the severity of a condition, but they also influence the immunity against disturbances in terms of not making wrong decisions. In the context of the voltage monitoring system each output signal gets a unique binary code assigned to identify the signal in the inter-module communication.

The total number of channels in the system is not only limited by the ADC's input multiplexer, but also by the timing constraints of the system. For given sampling and conversion speed of an ADC, the total number of channels and output signals is bounded depending on the required aliasing immunity and minimum deglitch times. These relations and their consequences are analyzed and discussed in section 7.

4.2. Analog-To-Digital Converter

The analog-to-digital converter (ADC) with a resolution of N bits allows converting analog input voltages into a digital representation. The digital representation can either be an N-bit value for a full conversion, but it can also be a 1-bit value resulting from a fast compare conversion. In this special case a threshold code is delivered to the ADC which then announces whether the analog input signal is greater than the voltage equivalent of the threshold code or not. Apart from the different outputs, the two methods mainly differ in the time required for the conversion. For a successive approximation register ADC, the full conversion ideally takes N times longer than the fast compare conversion.

The ADC can be equipped with a combination of different input channel types. Each type permits another maximum operating voltage, but as a rule of thumb a higher voltage capability comes at the cost of higher silicon area and a longer minimum sampling time. The interface to the ADC consists of a signal to start the sampling process on a selected channel. A further signal allows starting of the conversion process on the sampled channel with a specified threshold. In return, the ADC provides a signal signifying the end of conversion and at the same time the conversion result is provided. Chapter 5 goes into more detail regarding the architecture and properties of the ADC.

4.3. Random Scheduler

The motivation of using a random scheduler instead of a round robin scheduler lies in the prevention of aliasing without using anti-aliasing filters. The random scheduler has the task of selecting which signal on which output should be converted at which time. In the usual configuration of the voltage monitoring system, all conversions are fast compare conversions.

If so many fast compare conversions with different thresholds must be performed on a channel such that they take longer than one full conversion, then and only then the latter should be used. In that case the threshold comparisons can be simultaneously performed in the digital domain on the basis of the full resolution conversion result. For an ideal N-bit successive approximation register ADC, more than N outputs on one channel would be required to make the full conversion faster for this channel.

Based on the channel to output signal relationship elucidated in Fig. 4.2, there are two possibilities for performing the sampling sequence. In the first method, the scheduler selects signals directly. It then samples the corresponding channel and performs one conversion with the signal's threshold, after which it selects another signal and starts anew. This method is advantageous when output signals on the same channel require significantly different average sampling frequencies. Otherwise, the next method, which is used later on, provides better time efficiency.

In the second method, the scheduler selects a channel, samples it and then performs conversions for all output signals that are attached to the channel. Hence, the sampling time must only be spent once per channel and not once per output signal. The order in which the fast comparison conversion is carried out for the output signals is irrelevant because the sampled

voltage is on hold internally. The actual point in time when the voltage is stored internally is when the first fast compare conversion is started and consequently the sampling switch is opened.

The question of which source of randomness should be used for selecting the channels randomly is still open. Basically, a real source of randomness must be based on a physical phenomenon such as noise. The demand for guaranteed reaction times rules the use of real random sources out, because they would inherently lead to corner cases where timings are violated – even if only with a very low probability. Therefore, in order to guarantee certain timings, a pseudo-random number generator is used to select the channels. For best anti-aliasing properties, an additional random source is used to delay the point in time when the sampling switch is opened. The details of the random scheduler are elaborated in section 7.

4.4. Functional Safety Concept

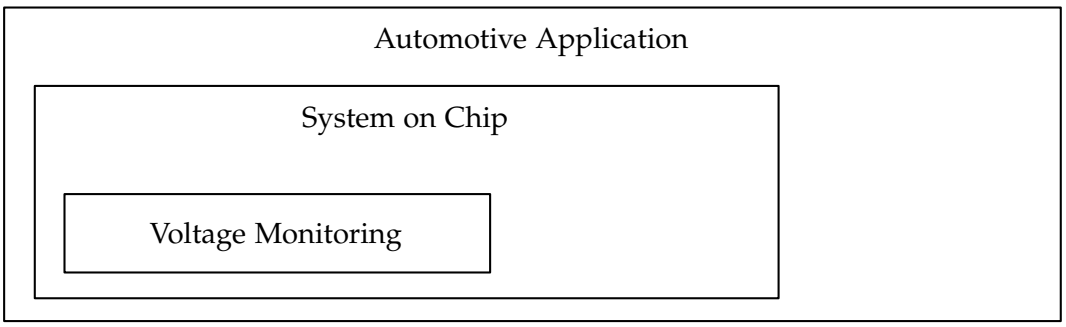

Figure 4.3.: Hierarchical position of the voltage monitoring system in an automotive application.

The voltage monitoring system is designed for automotive environments. In addition to electromagnetic compatibility (EMC), of which some special standards apply to automotive applications, the functional safety norm ISO 26262 [18] is specific for automotive systems. Hence, the voltage monitoring system must comply with the functional safety norm, but it must be kept in mind that the voltage monitoring system is a submodule of a system on chip (SoC). This SoC is again a module of a top-level system such as an airbag system of a car. Therefore, the functional safety requirements on the voltage monitoring system must be specified in the context of the top-level item. Depending on what tasks the SoC is involved in, and what the voltage rails which are monitored by the voltage monitoring system are connected with, the requirements on the voltage monitoring system differ accordingly. Consequently, without a concrete application background the whole functional safety assessment is not possible.

Under the assumption that the voltage monitoring system is only used as an independent observer, which means that no influence on the normal system behavior is based on the output signals, the concept to fulfill the functional safety requirements can be stated more precisely.

4.4.1. ADC and Logic Self-Test

Let it be assumed that any undetected under- or overvoltage condition leads to the violation of a safety goal. Even then a fault in the monitoring system does not have consequences as long as no irregular voltage conditions prevail. Therefore, using the vocabulary of the ISO 26262 standard, a fault in the voltage monitoring system is called a latent fault. This means that the fault stays undetected until the condition to be detected appears. The diagnostic coverage of latent faults can be increased significantly by the implementation of a self-test in the system. It can emulate the occurrence of a condition which is to be detected and then check if it was detected or not.

This test can be performed at power-up of the system and, if required, even during the normal operation. Because many channels and output signals are sampled in a random manner, a cyclic self-test in between some channels does not affect the timing severely if its duration is not longer than the sampling and conversion of a few channels. The details of the built-in self-test for the ADC are discussed in chapter 6.

Apart from the ADC, the voltage monitoring system consists only of digital circuitry. The latter can be equipped with logic built-in self-test (LBIST) functionality to detect failures. The remaining weak spot is the ADC – hence, in chapter 6 a built-in self-test (BIST) is proposed to assess the functionality of a capacitive charge redistribution SAR ADC. These methods in combination with the check introduced in the next section can provide sufficient diagnostic coverage for latent faults in the voltage monitoring system.

4.4.2. Cross-Domain Redundancy and Checks

The voltage monitoring system itself requires a power supply, a reference voltage and a clock source. The detection of problems with the first two can be performed if they are available redundantly in multiple domains. The integrity of the clock signal is not dealt with in the voltage monitoring system because clock monitoring must be performed on system level of an SoC.

If the supply of the voltage monitoring system fails, it cannot operate anymore. If multiple and ideally independent supply domains exist on a SoC, the voltage monitoring system can be supplied redundantly without the need to manually switch between the supplies. This is achieved by an OR-function implemented by one diode per supply as shown at the top of Fig. 4.4. Obviously, a precondition is that the forward voltage drop across the diode can be tolerated. As a result, the voltage monitoring system only fails if all supplies fail. In addition, it can also monitor its own supply voltages and signal the failing of any of them to other modules.

The self-test for the ADC mentioned in the previous section assesses the correct functionality of the ADC core. However, the full scale range (FSR), which depends on the reference voltage of the ADC, cancels out of the measurement. Therefore, a wrong bandgap reference voltage remains undetected. If multiple domains with independent bandgap reference voltages are available in an SoC, then a cross-check between the voltages can be performed. For that purpose the internal structure of the used ADC can be exploited. The reference voltage which can be selectively applied to the bottom plates of the ADC's capacitor array is buffered via

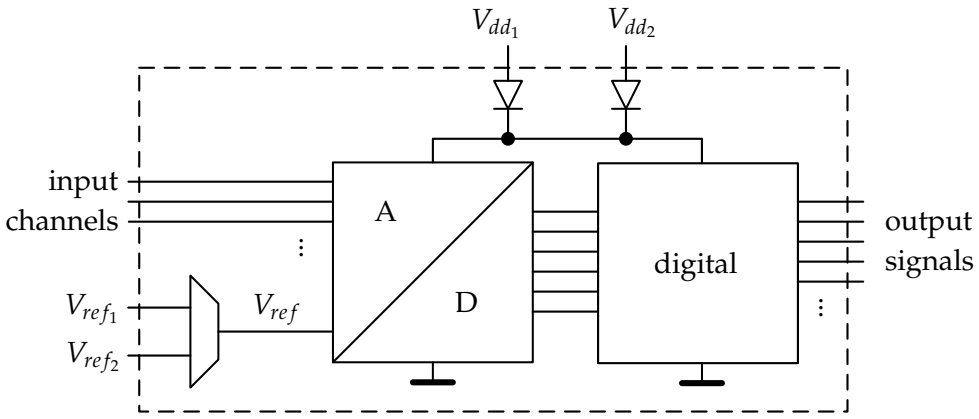

Figure 4.4.: Redundant supply and reference cross-check concept.

an amplifier. Low-voltage input channels are also decoupled by the same buffer, but are sampled to the top-plates of the capacitor array via a different path. Therefore, the role of reference voltage and low-voltage input channels is interchangeable. This allows measuring the bandgap reference voltages in relation to each other. If they differ significantly, the conversion result will be in overflow for one measurement, but a measurement the other way around permits to determine the voltage difference. The lack of an absolute reference prevents a judgment whether one reference voltage is too high or the other too low. Hence, the only decision to be made in case of detected deviations is to signal an error. When more than two reference voltages are available, a two out of three (2003) decision could point a finger on the erroneous reference voltage and continue operation with the remaining references.

4.4.3. Safe State for Output Signals

The deglitch filtering mechanism for the output signals can be described as bistable. Whichever state the output is in; it is only changed if there is a unanimous vote of a certain number of samples in a row. This leads to a inertness in behavior such that when ambiguous threshold comparison results occur, no action is taken. If certain under- or overvoltage signals are critical for safety, it can be considered to assign a safe state for them. For under- or overvoltage signals the safe state is to signal an under- or overvoltage. Whenever the observed threshold comparison data in a time frame is inconclusive, instead of taking no action the affected signals are switched to their safe states. While this can provide benefits in terms of functional safety, such behavior decreases the immunity against disturbances and might lead to a degraded reliability.

4.4.4. Sampling Switches

If one or more of the sampling switches were stuck such that they always connect the input signal to the ADC core, this would be detected by the BIST. The stimulus signal is created internally on the capacitor array, therefore a closed sampling switch would distort the signal and this would be noticed. However, the BIST cannot tell whether the input channel sampling

switches close correctly. Therefore, if a channel is critical for safety, it can be considered to assign one or two additional channels of the ADC to the same input node. Then an additional circuit can process the redundant output signal values to determine if they disagree. When two redundant channels are used, it can only be detected that there is a problem. When three channels are used, a 2003 decision can even determine the erroneous channel and still continue operation based on the data of the working channels.

4.4.5. Retained Comparators

The voltage monitoring system can only start to operate once its power supply and clock signal are stable. If voltage rails must already be observed before the voltage monitoring system is available, the use of normal comparators is inevitable.

4.5. Test Concept

The test concept dictates how the functionality of the system can be tested after production. Test time on automated test equipment (ATE) is very expensive, which means that the test time should be kept as low as possible. For comparison, the test concept for the existing system with distributed comparators was discussed in section 1.1.4. The time-consuming linear ramp testing is completely avoided in the new voltage monitoring system.

4.5.1. Scan Test for Digital Logic

The digital logic constitutes a major part of the system and can be conveniently tested with a scan test. For that purpose, a test mode is introduced where all registers in the digital logic are chained together to form a shift register. Test patterns can then be applied to the registers and monitoring of the combinatorial outputs allows assessing the correct functionality of the logic.

4.5.2. Built-In Self-Test for the ADC

For functional safety reasons the existing ADC must be equipped with a self-test to assess the functionality during power-up and, if needed, even during normal operation. This self-test, as discussed in detail in chapter 6, can be configured to perform an accurate judgment of whether the ADC parameters are in their specification windows or not. To interpret the measurement data provided by the self-test, the automated test equipment (ATE) has to perform some computations. Together with the offset- and gain compensation available in the ADC, this permits to save the otherwise very long test times for an ADC with the standard test process, the histogram test [11]. It requires each output code of the ADC to be hit multiple times when a periodic signal is applied to the input – naturally, this takes a very long time.

4.5.3. Gain Capacitors and Continuity Test

Once the correct functionality of the digital logic and of the ADC core is established, the only remaining untested parts are the multiplexer sampling switches for the channel and the attached channel gain capacitors. If any of the switches was closed all the time, this would be detected by the self-test. The only remaining test is to check if the sampling switches close correctly. This can be done by applying a DC voltage to the externally available channels and via dedicated test pads also to the internal channels. The voltage should be above the threshold of the outputs attached to the channel.

The voltage monitoring system should be operated in a mode where all channels are converted with full resolution in a round-robin scheme; the ATE monitors the output values. It can be checked if the applied voltage only appears on the correct channel. When the voltage appears in the measurement on other channels, or when no voltage can be measured, the sampling switch circuits have a fault. Additionally, the measured code should be in an acceptable range for the applied voltage. As the linearity, offset and gain of the ADC are already verified, one voltage measurement suffices to verify the correct gain by the input gain capacitor of a channel. A few more measurements at different voltages can help to increase the confidence level.

4.6. Voltage Monitoring Specifications

Table 4.1 and 4.2 contain a specification example for two comparators (output signals) which monitor the same 5 V voltage rail. Each parameter has a specification window to allow for some variations due to manufacturing deviations. The voltage monitoring system emulates comparators, which means that the specifications for normal comparators can be directly mapped to the output signals. These include the rising and falling thresholds, the optional hysteresis in between them and constraints on the filtering of the comparator signals.

comparator name	rising threshold		falling threshold		hysteresis	
	min (V)	max (V)	min (V)	max (V)	min (V)	max (V)
undervoltage_5_n	4.8	4.9	4.7	4.8	0.05	0.1
overvoltage_5	5.2	5.3	5.1	5.2	0.05	0.1

Table 4.1.: Example for output signal threshold specifications.

comparator name	rising deglitch		falling deglitch	
	min (µs)	max (µs)	min (µs)	max (µs)
undervoltage_5_n	80	120	40	60
overvoltage_5	40	60	80	120

Table 4.2.: Example for output signal deglitch filter specifications.

5. Charge Redistribution SAR ADC

In the voltage monitoring concept the used analog-to-digital converter (ADC) plays a central role. This chapter explains the successive approximation principle and its application in a single-ended capacitive charge-redistribution successive approximation register ADC. An existing ADC of this type is used in the voltage monitoring system. The timings of the sampling and conversion process are examined in detail. Additionally, accuracy calculations for threshold voltages and hysteresis voltages are presented.

5.1. Successive Approximation Principle

The successive approximation equals a binary search with a runtime of $\mathcal{O}(\log N)$, where N is the number of quantization levels. The principle of a SAR ADC is shown in Fig. 5.1. Notably, the core of the SAR ADC is a DAC, which is located in a feedback path from the output register to the input comparator.

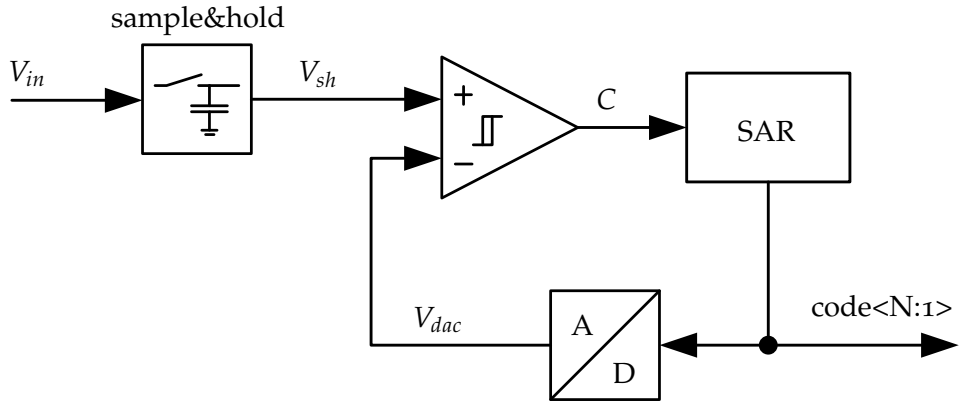

Figure 5.1.: Principle of a successive approximation register ADC.

Let the minimum voltage granularity of the DAC be

$$V_{LSB} = \frac{V_{FSR}}{2^N}, \tag{5.1}$$

where V_{FSR} is the full-scale range voltage – the maximum voltage that can be converted (depending on the transfer curve it might be unreachable). As a full conversion cycle of an N-bit SAR ADC needs at least N clock cycles, the input voltage has to remain constant to prevent conversion errors. This is achieved with the help of a sample&hold circuit at the input. During the conversion the voltage V_{sh} can be assumed to be constant.

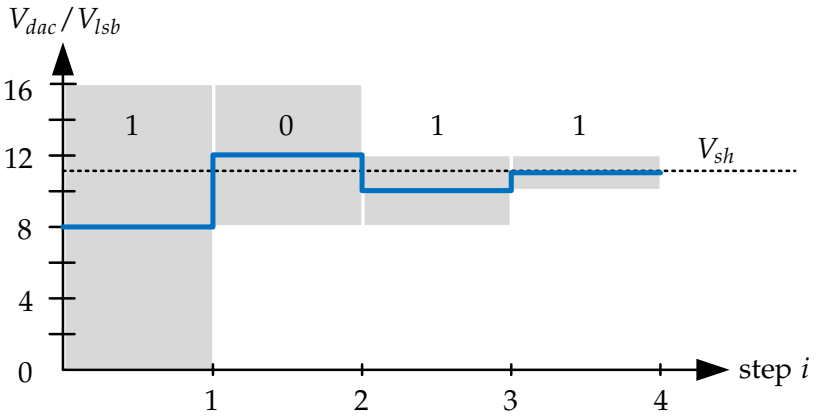

Figure 5.2.: Example for the successive approximation with 4-bit resolution.

The N-bit SAR register controls the whole conversion process. Initially it contains the N-bit unsigned binary code `100...0`, where only the most significant bit (MSB) is set. This code is equivalent to the unsigned decimal value of 2^{N-1}. Hence, in the first step the comparator compares V_{sh} against $V_{dac,1} = V_{LSB} \cdot 2^{N-1}$. If $V_{sh} \geq V_{dac,1}$ the comparator output C is high and the SAR register will keep the MSB and set the next bit such that the code `110...0` is tried in the next step. On the other hand, if $V_{sh} < V_{dac,1}$ and hence C is low, the MSB is cleared and the code `010...0` is tried next. The voltage $V_{dac,2}$ is therefore either $V_{LSB} \cdot 2^{N-2}$ or $V_{LSB} \cdot (2^{N-1} + 2^{N-2})$. According to this algorithm, each bit is determined successively from MSB to LSB in the form of a binary search. An example for the conversion of a 4-bit value is displayed in Fig. 5.2. The gray areas indicate the search window, which is narrowed down by a factor of two in each step.

5.2. Charge Redistribution Successive Approximation Analog-To-Digital Converter

A charge redistribution successive approximation register (SAR) ADC is a sophisticated type of SAR ADC, where the DAC is based on a weighted capacitor array. The good relative matching properties of capacitors in the integrated circuit manufacturing process let this type of ADCs achieve a good resolution combined with medium conversion speeds, relatively low power consumption and a low area footprint. This makes such ADCs very popular.

The next section introduces the principle behind a single-ended charge redistribution SAR ADC and subsequently an existing high-voltage adaptation. Furthermore, the modification for the usage in the voltage monitoring system and error calculations are presented.

5.2.1. Single-Ended Topology

The SAR principle can be implemented with many types of DACs. Because of the inherent direct current (DC) consumption of resistor strings, the usage of a capacitor based DAC is beneficial. A weighted capacitor array provides the DAC functionality by charge redistribution

with the help of switches. Additionally, it also serves as the hold capacitor which saves an extra sample and hold circuit.

The different types of charge redistribution SAR ADCs are distinguished by sampling mode (such as top-plate or bottom-plate) and by comparator reference (single-ended or differential). Redundant capacitor arrays have ratios between successive weights R which are smaller than two to provide inherent error correction capabilities. Now the principle of a simple single-ended charge redistribution top-plate sampling SAR ADC without redundancy is explained.

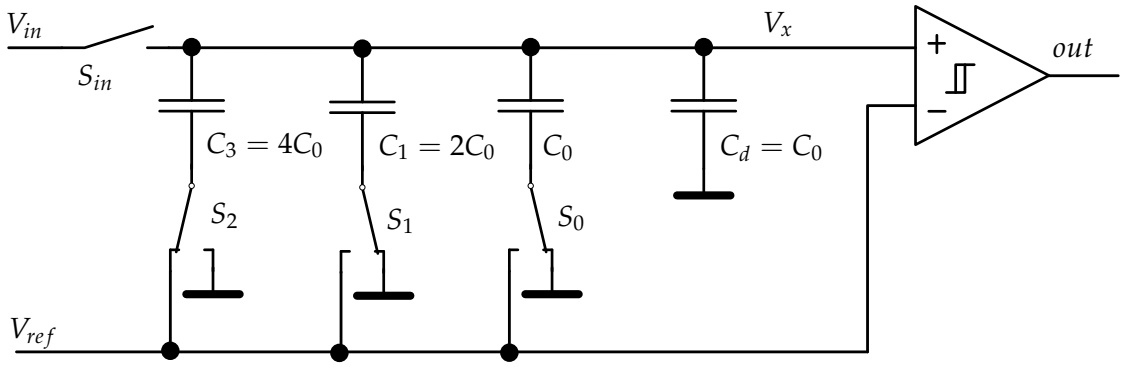

Figure 5.3.: Single-ended top-plate sampling SAR ADC with 3-bit resolution in the conversion mode.

Figure 5.3 shows the principle of such an ADC with 3 bits resolution. The weighted capacitor array consists of three switched capacitors with values $C_i = C_0 \cdot 2^i, i = \{2, 1, 0\}$ and one dummy capacitor with size C_0 such that the total array capacitance equals $C_0 \cdot 2^3 = 8C_0$.

An input signal is sampled via the switch S_{in} while all the other bottom-plate switches S_i are closed to ground. After S_{in} is opened again, the voltage V_x remains the same as the input voltage. This is the implicit sample and hold functionality which was mentioned earlier. Now the conversion process starts by successively adding a DAC value to the voltage at node V_x by switching the bottom plate switches S_i to either ground or to the reference voltage V_{ref}. The voltage V_x in dependence of the position of the switches S_i can be analyzed by superposition of the sampled voltage and the DAC voltage.

The first voltage

$$V_x' = V_{in} \tag{5.2}$$

results from the sampling process. The second super-positioned voltage is the DAC voltage which is created by changing the position of the switches:

$$V_x'' = V_{ref} \cdot \frac{C_x}{\sum C} = V_{ref} \cdot \frac{\sum_{i=0}^{2} S_i \cdot C_i}{8C_0} = V_{ref} \cdot \frac{\sum_{i=0}^{N-1} S_i \cdot C_i}{2^N C_0}\,, \tag{5.3}$$

where $S_i = 0$ for a switch connection to ground and $S_i = 1$ for a connection to the reference voltage V_{ref}. This yields a minimum granularity of the DAC of $V_{lsb} = \frac{V_{ref}}{2^3}$.

The resulting voltage at the input of the comparator is then

$$V_x = V_x' + V_x'' = V_{in} + V_{ref} \frac{\sum_{i=0}^{2} S_i \cdot C_i}{8C_0} = V_{in} + V_{ref} \frac{\sum_{i=0}^{N-1} S_i \cdot C_i}{2^N C_0}\,. \tag{5.4}$$

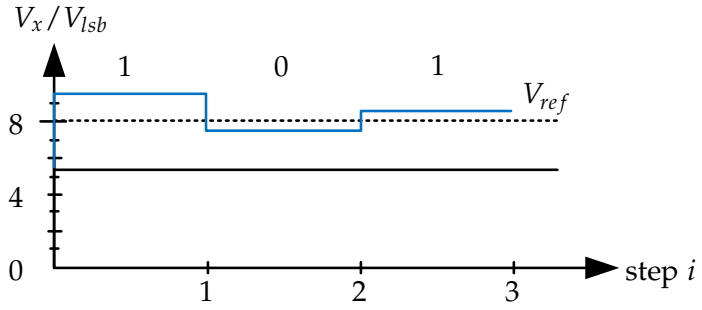

Figure 5.4.: Example for a conversion process with the single-ended top-plate sampling SAR ADC.

The negative input of the comparator represents the threshold voltage and is, in this case, set to the reference voltage V_{ref}. The SAR algorithm is performed as explained in the previous section with the switches mapped to the corresponding bit positions in the SAR register. The comparator output in step i represents the i-th most significant bit of the converted value, but in the feedback path to the DAC the bit has to be inverted. Figure 5.4 depicts this behavior, which is specific for the considered ADC.

The resulting quantization characteristic of the above ADC is shown in Fig. 5.5. The biggest difference lies in the inherent offset of $\frac{1}{2}$ LSB compared to the ideal quantization characteristic.

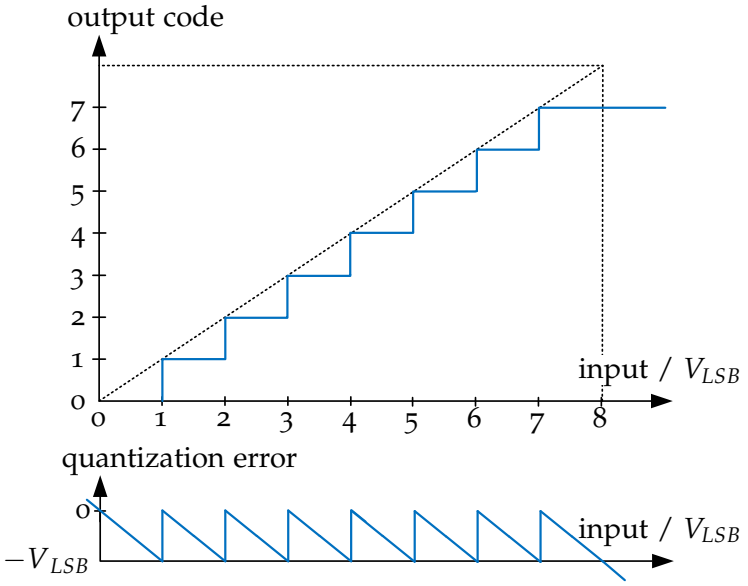

Figure 5.5.: Example of the quantization characteristic of a SAR ADC for 3-bit resolution.

5.2.2. High Voltage Inputs and Multiplexer

This section is based on [4]. With a few additions the single-ended SAR ADC can perform multiplexed conversions of high input voltages in the range of tens of volts. This requires a

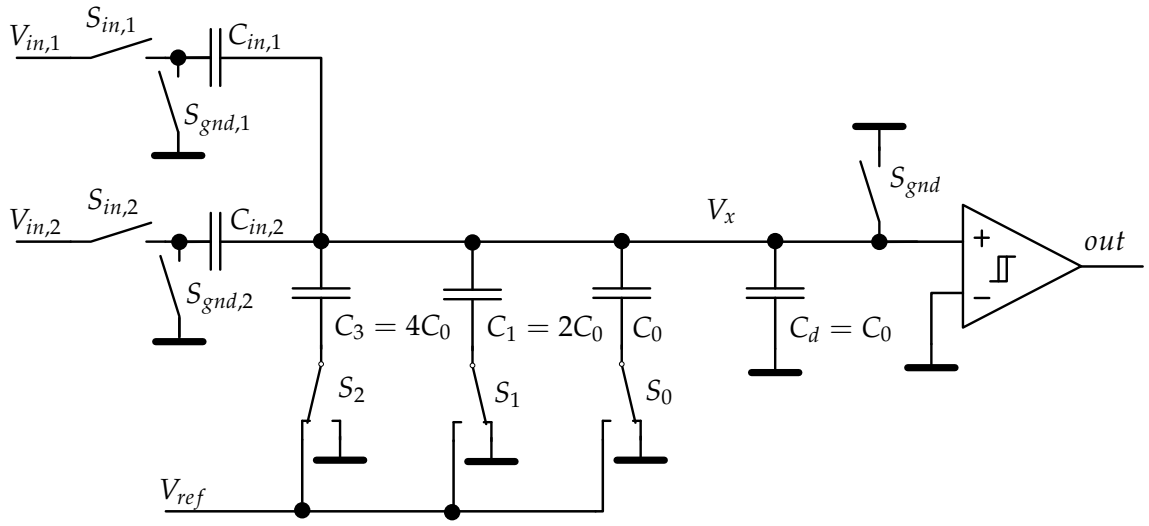

Figure 5.6.: Single-ended top-plate sampling SAR ADC with 3-bit resolution and multiplexed capacitive input voltage divider for high voltages in the conversion state [4].

capacitive input voltage divider and further sampling switches. Fig. 5.6 displays the ADC with two multiplexed input channels and individual input capacitances $C_{in,i}$.

The sampling process takes two steps now. Initially, the switches S_i of the capacitor array are all closed. For input i to be sampled, the switches S_{gnd} and $S_{in,i}$ are closed. This charges the input capacitance $C_{in,i}$ to the voltage $V_{in,i}$. In the next step, the input switches S_{gnd} and $S_{in,i}$ are opened and $S_{gnd,i}$ is closed. This flips around the voltage on the input capacitor and the charge stored on $C_{in,i}$ is redistributed between $C_{in,i}$ and the total array capacitance $C_{array} = 2^N C_0$. The switch $S_{gnd,i}$ is then opened and the negative scaled input voltage is stored on the capacitor array as

$$V_x = -V_{in,i} \cdot \frac{C_{in,i}}{2^N \cdot C_0} \,. \tag{5.5}$$

As the input voltage now appears with a negative sign on the capacitor array, the negative comparator input is set to ground. The DAC voltage is added to the negative input voltage until they compensate each other – the successive approximation logic has to be changed accordingly.

Let the ratio of the input capacitor and the capacitor array be defined as the gain g_i of channel i:

$$g_i = \frac{C_{in,i}}{2^N \cdot C_0} \,. \tag{5.6}$$

Hence, the full scale range (FSR) of the ADC for channel i is externally perceived as

$$V_{FSR} = \frac{V_{ref}}{g_i} \,, \tag{5.7}$$

and the resolution as

$$V_{LSB} = \frac{V_{FSR}}{2^N} = \frac{V_{ref}}{g_i \cdot 2^N} = \frac{2^N \cdot C_0}{C_{in,i}} \cdot \frac{V_{ref}}{2^N} \,. \tag{5.8}$$

Internally the LSB voltage as well as noise parameters of the DAC remain unchanged. Therefore, the scaling of the input voltage decreases the signal-to-noise ratio (SNR). For higher input voltages a lower gain is necessary. This requires lower input capacitances and hence less area. The advantage of this capacitive voltage divider is that the ADC core can work in a low-voltage domain. Low-voltage devices require less silicon area than high-voltage components and, more importantly, are significantly faster. Only the input sampling switches and the input capacitances must be capable of withstanding high voltages. Furthermore, the good matching properties of capacitors allow for little linearity errors in the transfer curve.

5.2.3. Sampling Process

The inputs of the ADC are not buffered; hence, a decaying current is drawn from the input line until the capacitor is fully charged. The equivalent circuit diagram for the sampling process is shown in Fig. 5.7. In a first step, the input voltage V_{in} is sampled on the input capacitor C_{in} by closing of S_{in} and S_2. This exhibits a first order lowpass behavior, where the input switch and its protection circuitry constitute the resistance R.

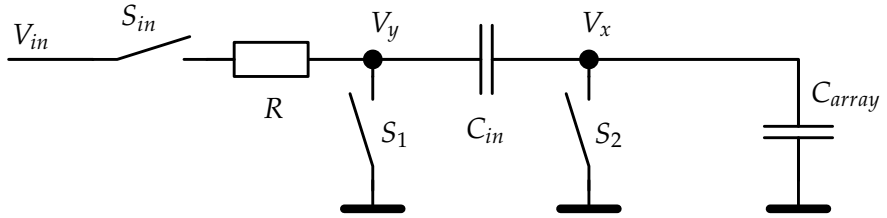

Figure 5.7.: Equivalent circuit diagram for sampling.

The voltage on the input capacitor $V_y(t)$ depends on the time constant τ and the size of the input capacitor C_{in}.

$$\tau = R \cdot C_{in} \tag{5.9}$$

$$V_y(t) = V_{in} \cdot \left(1 - \exp\left(-\frac{t}{\tau}\right)\right) \tag{5.10}$$

Figure 5.8 shows the sampling process normalized with the input voltage. The worst case error occurs with the input voltage being equal to the full scale voltage V_{FSR} of the ADC. For zero error the sampling process would take infinitely long, but the error is neglectable if it is in the range of the quantization error:

$$V_{error} \stackrel{!}{=} \frac{V_{LSB}}{2} = V_{FSR} \cdot 2^{-(N+1)} \,. \tag{5.11}$$

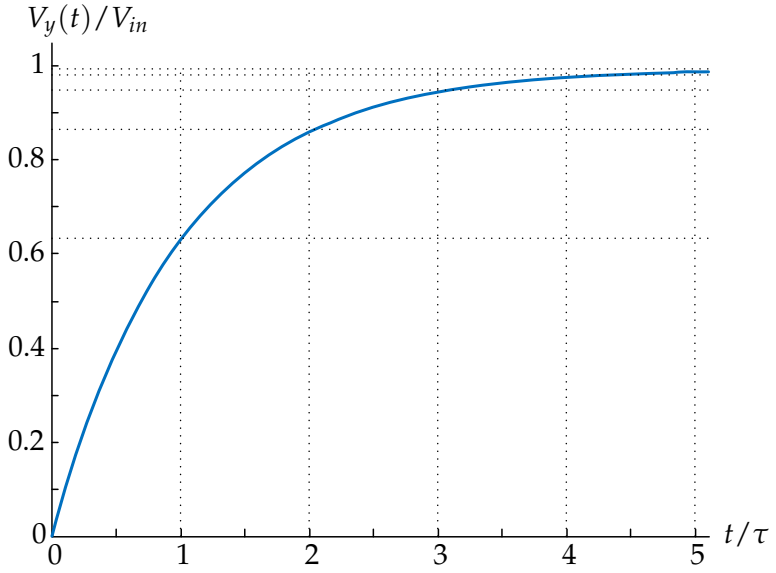

Figure 5.8.: Sampling process with first order lowpass behavior.

The required sampling time for an N-bit ADC is then given by the time it takes to reach the full scale voltage minus the allowed error voltage with a full scale input voltage.

$$V_{FSR} - V_{error} \stackrel{!}{=} V_{FSR} \cdot \left(1 - \exp\left(-\frac{t}{\tau}\right)\right), \tag{5.12}$$

where V_{error} in (5.12) can be substituted by (5.11). After solving for t, the result is

$$t = -\tau \cdot \log 2^{-(N+1)} \tag{5.13}$$

$$= \tau \cdot (N+1) \cdot \log 2\,. \tag{5.14}$$

Hence, the required minimum sampling time for a constant input capacitance increases linearly with the resolution. In the second step of the sampling process, the switches S_{in} and S_2 are opened and S_1 is closed to flip the voltage and redistribute the stored charge. This process is only restricted by the on-resistance of the switches and the series resistance of the capacitors. Therefore, the redistribution time is negligible in comparison to the first sampling time.

5.2.4. Fast Compare Mode

The usual conversion process in a SAR ADC is performed according to Fig. 5.4. For an N-bit ADC this process ideally takes N steps. If a large number of input signals are to be observed in a multiplexed setup, the sampling frequency per channel decreases significantly. Therefore, the ADC should be optimized for the task at hand, where only one threshold crossing information is needed. Instead of performing a binary search over the whole value range, the ADC only needs to set its internal DAC to the one required threshold. This immediately yields the necessary 1-bit information in only one step and allows for a higher sampling

frequency as the conversion takes less time. Additionally, there is no digital comparator required to perform the digital threshold comparison on the full N-bit result from the ADC.

5.2.5. Control Interface

The ADC has an interface with defined ports and protocols, which allow controlling the ADC operation. The ports are connected on the bottom of the ADC block in Fig. 4.1.

5.2.5.1. Full Conversion Mode

The interface of the existing ADC consists of three unidirectional control signals. Two of them are the inputs sos (start of sampling) and soc (start of conversion), the only output signal is eoc (end of conversion). The data interface consists of a channel selection input and an output for the conversion result.

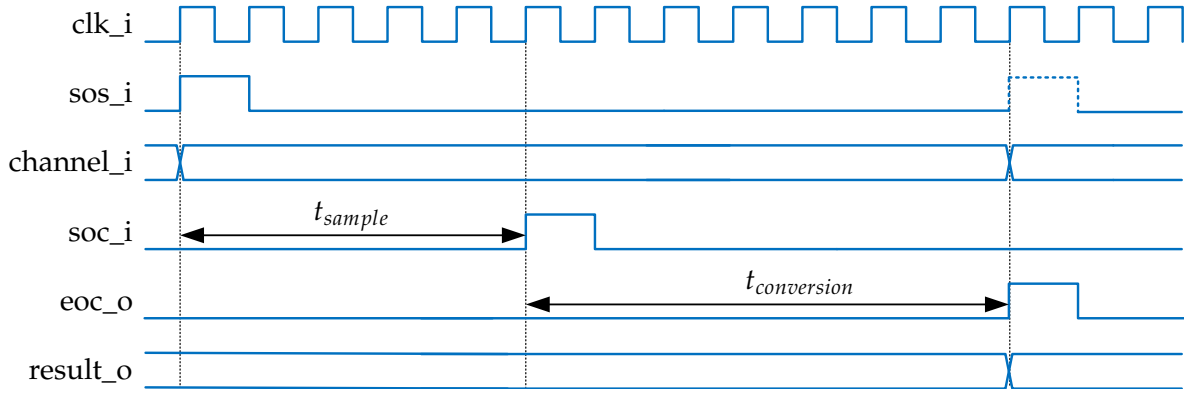

Figure 5.9.: Control interface of the ADC for normal conversions.

A timing diagram of a complete sampling and conversion cycle is depicted in Fig. 5.9. The initial sos pulse starts the sampling process on the selected channel. Waiting for the minimum required sampling time, which depends on the channel type, lies in the responsibility of the block controlling the ADC. Then the soc pulse can be applied, which causes the ADC to open the sampling switch and start the conversion process by successive approximation. For an N-bit ADC, this takes at least N clock cycles. Redundant capacitor arrays increase the time, as do reasons inherent to the internal implementation of the ADC. Once the ADC has finished the conversion, the eoc signal is set and the result is made available at the data output. Now the next cycle can start anew – on the same channel or on another channel.

5.2.5.2. Fast Compare Mode

The fast compare mode requires slightly different behavior. The following interface is a proposal for controlling the ADC in the fast compare mode. An additional port is required to provide the ADC with a code that represents a threshold voltage. Figure 5.10 shows a timing diagram for successive fast compare conversions on one channel. The first fast compare conversion opens the sampling switch. Therefore, all successive fast compare conversions

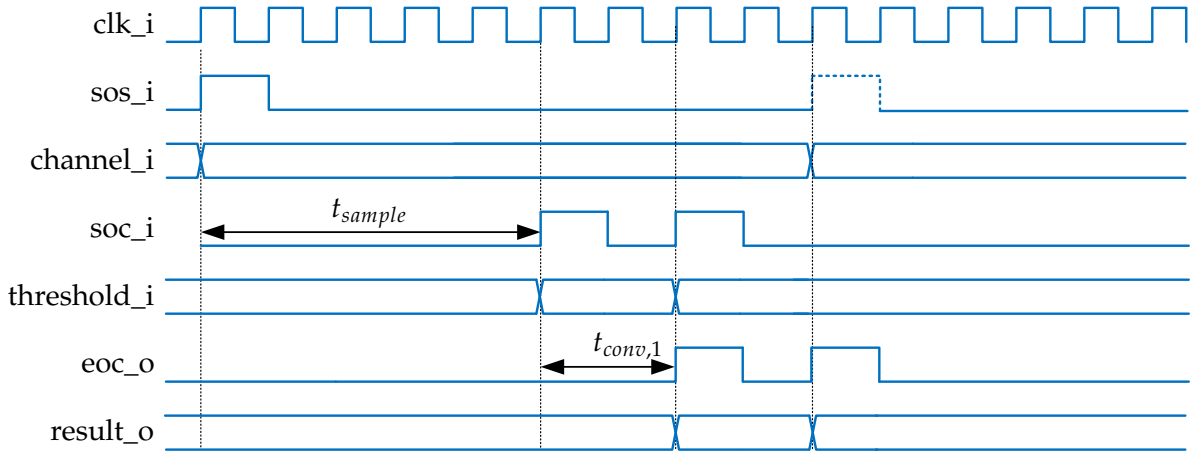

Figure 5.10.: Control interface of the ADC for fast compare conversions.

operate on the same internal voltage which is held by the capacitor array. The number of clock cycles for a fast compare measurement is ideally only one clock cycle, but for implementation reasons more clock cycles might be required. The eoc signal indicates that the 1-bit comparison result is valid. Now either a successive fast compare conversion can be performed or a new sampling process can be started (on the same or on a new channel).

With the same hardware, the ADC can perform fast compare as well as full conversions. Therefore, an additional control line to enable or disable the fast compare mode can be added if both modes are required for an application.

5.2.6. Maximum Sampling Frequency

The previous sections have dealt with the constraints behind sampling and conversion times. The total of both times determines the maximum sampling frequency of the ADC. For the fast compare mode, the conversion time is the sum of all fast compare conversion times.

$$t_{total} = t_{sample} + t_{conversion} \tag{5.15}$$

The conversion time inherently takes a certain number of clock cycles $n_{conversion}$. The sampling time t_{sample} is basically independent of the clock period t_{clk}, but the control logic can only wait for an integer number of clock cycles. Therefore, the sampling time must be rounded to the next highest integer multiple of the clock cycle time.

$$n_{sample} = \left\lceil \frac{t_{sample}}{t_{clk}} \right\rceil \tag{5.16}$$

$$t_{total} = \left(n_{sample} + n_{conversion}\right) \cdot t_{clk} \tag{5.17}$$

5.3. Threshold Accuracy Calculations

The voltage monitoring system makes use of fast compare conversions, which require a digital representation of a threshold. Therefore, the threshold voltages of the comparators must be converted to digital values, but in an ADC there are multiple sources of inaccuracies which are described in section 3.2.2. Additionally, a real ADC is also influenced by the inaccuracy of the reference voltage, which is usually produced by a temperature compensated bandgap circuit.

Per definition an ADC converts a voltage to a digital code, but due to the inaccuracies of the ADC and its voltage reference, the output code can be in a certain range. It is also of interest to find the input voltage range which could produce a certain output code. It is important to ascertain that for a chosen digital threshold code the input voltage range lies within the specification window of the threshold voltage. The LSB voltage of a channel is determined by the bandgap reference voltage V_{ref} and the ratio of the gain and array capacitances:

$$V_{LSB} = \frac{V_{FSR}}{2^N} = \frac{V_{ref}}{g_i \cdot 2^N} = \frac{2^N \cdot C_0}{C_{in,i}} \cdot \frac{V_{ref}}{2^N} \,. \tag{5.18}$$

For the remainder of this section it is assumed that the error parameters e are only available in the form of symmetric specification windows. Hence, if the maximum error is specified as e, the actual error is within $[-e, e]$. All errors can be split into a combined absolute error $e_{absolute}$ and a combined relative error $e_{relative}$. The absolute error consists of the offset error and the integral nonlinearity of the ADC. In the worst case the absolute error can be stated according to section 5.5.1 as

$$e_{absolute} = V_{LSB} \cdot (e_{offset} + e_{INL}) \,, \tag{5.19}$$

where e_{offset} and e_{INL} are given in terms of the LSB voltage. The gain error of an ADC is usually given under the assumption of an ideal reference voltage, but for the real error it must be taken into account. The gain- and reference voltage errors are relative, which means that their impact depends on the value of the input voltage. According to section 5.5.1 the combination of both errors can be given as

$$e_{relative} = (e_{reference} + e_{gain}) \,. \tag{5.20}$$

Let $\varepsilon \in [-1, 1]$ denote the error parameter in the following calculations. A value of $\varepsilon = 0$ yields the ideal (typical) result, $\varepsilon = -1$ gives the minimum and $\varepsilon = 1$ the maximum result in the worst case of completely correlated errors. The calculation of the code Q from the given input voltage V_{in} is carried out as

$$Q_{out}(V_{in}, \varepsilon) = \text{quantize}\left(\frac{V_{in} \cdot (1 + \varepsilon \cdot e_{relative}) + \varepsilon \cdot e_{absolute}}{V_{LSB}}\right) . \tag{5.21}$$

According to the transfer characteristic in Fig. 5.5, the quantization function is the floor operator: $\text{quantize}(x) = \lfloor x \rfloor$. For the standard transfer curve as defined in Fig. 3.12, the quantization function is the normal rounding function $\text{quantize}(x) = \text{round}(x)$. If the rounding function is always used, the floor operation can be achieved by taking into account an additional offset of $-\frac{1}{2}V_{LSB}$. Vice versa the same applies for the ceiling function.

The output code will be in the range $[Q_{out}(V_{in}, -1), Q_{out}(V_{in}, 1)]$. For a given output code, the input voltage which might have caused a certain output value can be calculated as

$$V_{in}(Q_{out}, \varepsilon) = \frac{V_{LSB} \cdot \text{quantize}^{-1}(Q_{out}) + \varepsilon \cdot e_{absolute}}{1 - \varepsilon \cdot e_{relative}}. \tag{5.22}$$

If $\text{quantize}(x) = \text{round}(x)$, then the inverse function is $\text{quantize}^{-1}(y) = y + \xi$ with $\xi \in [-\frac{1}{2}V_{LSB}, \frac{1}{2}V_{LSB})$, and hence in the worst case the input voltage range is

$$V_{in}(Q_{out}, \varepsilon) = \frac{V_{LSB} \cdot (Q_{out} + \varepsilon \cdot \frac{1}{2}) + \varepsilon \cdot e_{absolute}}{1 - \varepsilon \cdot e_{relative}}. \tag{5.23}$$

In the end, without further knowledge about a given output code the input voltage could have been in the range $[V_{in}(Q_{out}, -1), V_{in}(Q_{out}, 1)]$. This permits the back calculation to find out whether a digital threshold code is within the specification window for the threshold voltage.

5.4. Hysteresis Accuracy Calculation

The hysteresis V_{hyst} is defined as the difference between the rising and the falling threshold voltage.

$$V_{hyst} = V_{th,rising} - V_{th,falling} \tag{5.24}$$

To check whether it is inside the specification window, the deviations of the involved threshold voltages must be taken into account. Using the maximum worst-case rising threshold voltage and the minimum worst-case falling threshold voltage for the calculation of the maximum hysteresis voltage overestimates the hysteresis voltage significantly. The reason is the inherent correlation of some error parameters for both the rising and the falling threshold voltages such as the offset error and the gain error. Therefore, the correct hysteresis calculation takes this into account. Substituting (5.19) and (5.20) in (5.23) yields

$$V(Q, \varepsilon) = \frac{V_{LSB} \cdot (Q + \varepsilon \cdot \frac{1}{2}) + \varepsilon \cdot V_{LSB} \cdot (e_{offset} + e_{inl})}{1 - \varepsilon \cdot (e_{reference} + e_{gain})} \tag{5.25}$$

Plugging (5.25) for the rising and falling voltages into (5.24) shows that in (5.26) the offset error cancels out of the hysteresis equation and the bandgap voltage and gain error are only relative to the difference between the rising and falling codes, whereas the INL error and the quantization error double because they are potentially uncorrelated. The typical, minimum and maximum worst-case values for the hysteresis are obtained by choosing an appropriate $\varepsilon \in [-1, 1]$.

$$V_{hyst}(\varepsilon) = \frac{V_{LSB} \cdot (Q_{rising} - Q_{falling}) + \overbrace{\varepsilon \cdot V_{LSB}}^{\text{uncorrelated quantization errors}} + \overbrace{2\varepsilon \cdot V_{LSB} \cdot e_{inl}}^{\text{uncorrelated INL errors}}}{1 - \varepsilon \cdot (e_{reference} + e_{gain})}$$
$$= \frac{V_{LSB} \cdot (Q_{rising} - Q_{falling}) + \varepsilon \cdot (1 + 2e_{inl})}{1 - \varepsilon \cdot (e_{reference} + e_{gain})} \tag{5.26}$$

The assumption of the integral nonlinearity and the quantization error being uncorrelated does not hold for a configuration where $Q_{rising} = Q_{falling}$ and hence there is no hysteresis. Then the above equation yields erroneous results, because only the quantization error remains uncorrelated. A negative worst-case hysteresis value is possible and means that the rising threshold voltage is lower than the falling threshold voltage. In practice this can indeed occur due to a large DNL error and associated missing codes.

5.5. Correlation Between Errors

When multiple errors occur in an accuracy calculation, the correlation between those errors determines the combined error.

5.5.1. Absolute Errors

The variance (5.27) of a sum of two errors $e = e_1 + e_2$ can be expressed as (5.28) by taking into account the correlation coefficient ρ, which is defined as the covariance $\mathrm{Cov}[\ldots]$ normalized with the respective variances [15].

$$\mathrm{Var}[e_1 + e_2] = \mathrm{Var}[e_1] + \mathrm{Var}[e_2] + 2 \cdot \mathrm{Cov}[e_1, e_2] \tag{5.27}$$
$$\mathrm{Var}[e_1 + e_2] = \mathrm{Var}[e_1] + \mathrm{Var}[e_2] + 2\rho \cdot \mathrm{Var}[e_1] \cdot \mathrm{Var}[e_2] \tag{5.28}$$

With the variance denoted as $\mathrm{Var}[e] = \sigma_e^2$ and the standard deviation as $\sigma_e = \sqrt{\sigma_e^2}$, the addition of the errors yields

$$\sigma_{e_1+e_2} = \sqrt{\sigma_{e_1}^2 + \sigma_{e_2}^2 + 2\rho \cdot \sigma_{e_1}\sigma_{e_2}}\,. \tag{5.29}$$

For the special case of completely correlated errors with $\rho = 1$, the result is an addition of the standard deviations. When the correlation is unknown, this is the worst case that must be assumed.

$$\sigma_{e_1+e_2}\big|_{\rho=1} = \sigma_{e_1} + \sigma_{e_2}\,. \tag{5.30}$$

5.5.2. Relative Errors

A relative error e_i is defined such that an error-prone value x' is derived from the ideal value x as

$$x' = x \cdot (1 + e_i)\,. \tag{5.31}$$

If a variable is influenced by two relative errors, the resulting formula can be expanded to

$$\begin{aligned} x' &= x \cdot (1 + e_1) \cdot (1 + e_2) && (5.32)\\ &= x \cdot (1 + e_1 + e_1 + e_1 e_2) && (5.33)\\ &\approx x \cdot (1 + \underbrace{e_1 + e_2}_{e_{12}})\,, && (5.34) \end{aligned}$$

because the term $e_1 e_2$ can be neglected for $e_1, e_2 \ll 1$. This is always the case for realistic ADC errors which are at most in the range of a few percent. The result (5.34) allows the treatment of multiple relative errors by the methods for error addition introduced in section 5.5.1.

6. Built-In Self-Test for Charge Redistribution SAR ADCs

Integrated analog-to-digital converters (ADCs) must be tested after production to ensure that they meet specifications. As these circuits are highly precise, their test requires even more precise stimuli for the measurement data to be of any relevance. These tests are usually performed on automated test equipment (ATE) with externally applied stimuli. Providing these high precision stimuli is expensive. On top of that, in order to reach the required high precision, the stimuli generators need long settling times, resulting in even longer test times, which result in high testing costs.

Once an ADC without self-test features passes the production tests, there are no means to check that the ADC is fully functional during field operation. With regard to some automotive applications with their functional safety requirements this is not acceptable. Hence, it must be possible to verify the correct functionality of the ADC at any given time. Such an on-board verification module is called a built-in self-test (BIST). In this chapter a method is presented which covers all of the mentioned aspects. It allows

- conducting a front-end and back-end test of the ADC to check without the need for external stimuli generators, and
- running an in-system test to check the functionality of the ADC during field operation.

6.1. State of the Art Solutions

This section explains the state of the art for the testing of integrated ADCs in general and sophisticated methods to deal with capacitive charge-redistribution SAR ADCs.

6.1.1. Front-End and Back-End Test with Automated Test Equipment

For the production tests using ATE the code transitions of ADCs are usually measured with architecture independent methods such as the feedback loop, the ramp histogram test or the sine histogram test. They are all part of the IEEE standard 1241-2010 [11].

With the linear ramp method a slowly changing high-resolution high-linearity voltage ramp is applied to the input of the ADC, and the histogram of the conversion results is used to determine the differential nonlinearity and subsequently the integral nonlinearity of the ADC transfer curve. The sine histogram works likewise, only the input signal is a sine wave which changes the expected ideal histogram accordingly. Both methods have in common that they require a minimum number of hits per code to provide usable results. The feedback loop

method changes the input voltage to the ADC until a desired output code is reached. This allows for very accurate results, but also requires long test times.

For all methods, the stimulus signal source must be more accurate than the ADC to be evaluated. As for all methods each code has to be reached at least once, the test time for an N-bit binary ADC grows linearly with the number of possible output codes, which is 2^N. For high resolution ADCs this leads to long test times and to high test costs.

These methods of measurement are only used for production tests. Their use in the field is limited due to the required high precision of the generated test signals.

6.1.2. Built-In Self-Test

The stimuli needed to perform the previously mentioned measurement methods can theoretically be generated on the chip containing the ADC. If evaluation circuitry is available on the chip as well, then the ADC can be tested directly without external equipment. Such a method is called a built-in self-test (BIST).

The stimuli needed for the histogram test can be generated by a precision circuit on the chip. The generation of high precision signals and the implementation of the evaluation circuitry require a silicon area overhead that increases production costs. On top of that, external components might be needed to support the signal generation. An on-chip method proposed in [1] generates a linear ramp with a charge pump which can be used for an ADC test. Easier is the generation of an exponential stimulus by means of a capacitor and a resistor as shown in [7].

Architecture specific approaches can be tailored to the type of ADC under test. Adapting the test method for leveraging the underlying architecture of the ADC can reduce either test time or silicon area overhead, or even both. Capacitive charge redistribution SAR ADCs were introduced in section 5.2. They contain a capacitor array based DAC whose output voltage is compared to the input voltage of the ADC. If the ratios between the capacitors in the capacitor array can be determined, then the transfer curve as well as the static error values, such as INL and DNL, can be computed.

An approach termed major carrier transitions (MCT) helps to find missing codes by using the internal DAC to generate a voltage and subsequently measuring it with the SAR algorithm. An additional integrating DAC with a measurement range of 4 LSBs permits to characterize the DAC capacitor array [10][9]. Another approach for differential top-plate sampling SAR ADCs uses the switches at the bottom plates to create an embedded stimulus which, together with a charge pump, is used for measuring the capacitance ratios [6].

6.2. The Basic Idea

The generation of a stimulus signal with extra components requires additional silicon area. Therefore, the proposed BIST method reuses the existing resources of capacitive charge redistribution SAR ADCs and only requires little additional circuitry with minimal area overhead. An on-chip stimulus signal is generated by multiplexing the existing capacitor

array of the ADC to a charging and a discharging circuit. Depending on how these circuits are implemented, the stimulus shape can be adapted. The generated test signal does not have to be explicitly sampled, because it is directly generated on the internal sampling capacitor array. This enables fast test times. The algorithm compares the changing stimulus signal with an adaptive threshold, which is generated with the internal DAC of the SAR ADC just as during a normal conversion. The measured threshold crossing times allow computing the capacitor array ratios and subsequently the transfer curve and static linearity error values such as INL or DNL.

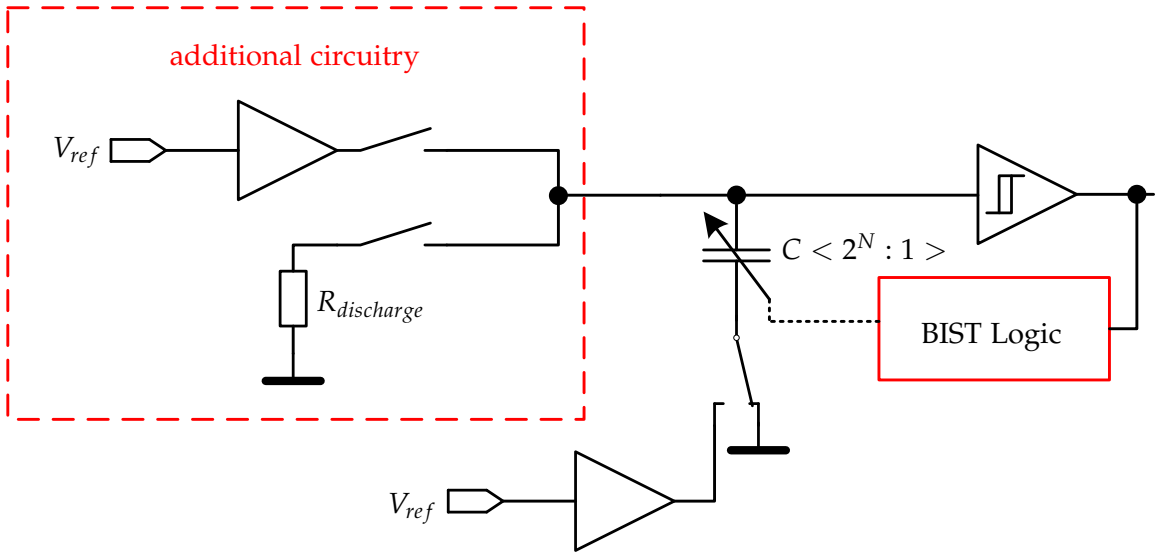

Figure 6.1.: Basic concept drawing of the self-test for a capacitive charge redistribution SAR ADC.

6.3. Safety Considerations

Regarding functional safety (ISO 26262), the BIST enables the use of the ADC in safety critical applications as the tests can be conducted during field use. Hence, latent faults that arise after a time of usage can be detected and the drift of certain parameters as well as defects due to aging can be monitored. As no external components are required for the BIST, it can be run during the front-end test and the load board design for the back-end test can be simplified. The ATE can perform tests in parallel to the internally running BIST, which saves test time and costs. This technique is known as concurrent testing.

In addition to the characterization of the capacitor array, it can also be assessed whether the bottom plate switches and the comparator are working – this increases the test coverage. If the comparator does not work, the threshold crossing times cannot be measured at all. A timeout for the BIST can then signal that the ADC is not working properly. If any of the bottom plate switches is not working, this shows up as a very severe mismatch and is definitely detected. The rest of this chapter shows the application of the BIST method for the capacitive charge redistribution SAR ADC which was introduced in section 5.2.

6.4. Measurement of Threshold Crossing Times

The measurement process consists of the stimulus generation and in parallel of the dynamic threshold adaptation. The result is a set of threshold crossing times, which can be evaluated to characterize the capacitor array.

6.4.1. Stimulus Generation

The underlying principle is to charge the capacitor array to a certain voltage, then to discharge it until an adjustable threshold is reached. The time or number of clock events elapsed until the comparator reacts is measured and used to compute the capacitor array ratios. The ADC shown in Fig. 6.2 allows the initial charging via the feedback path of the comparator.

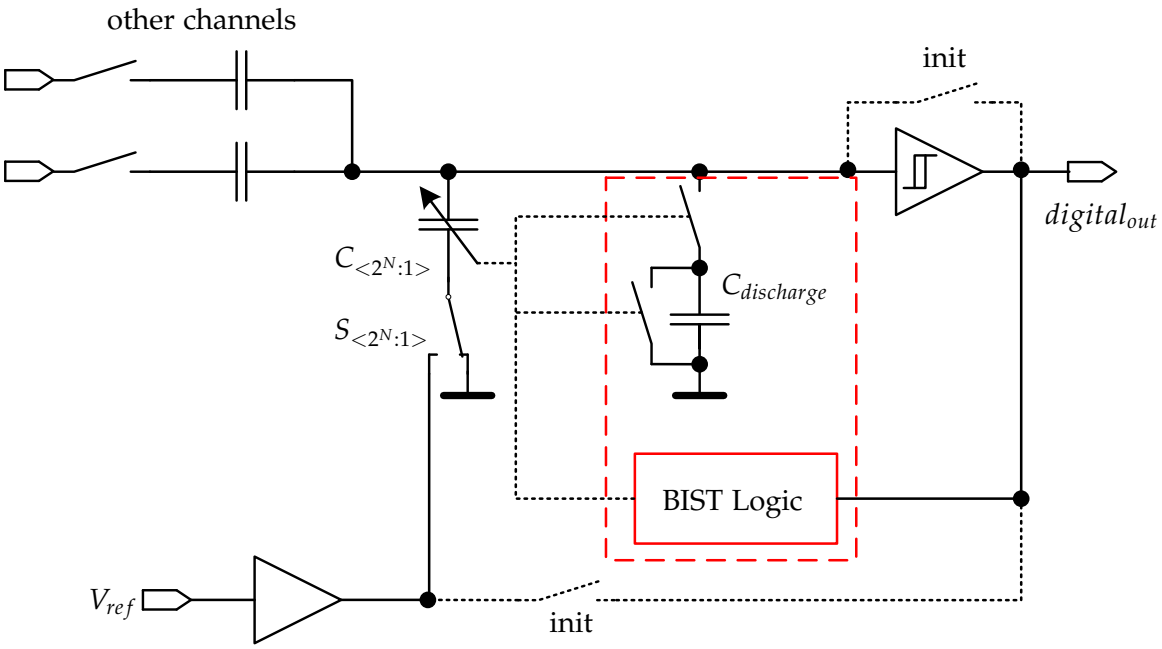

Figure 6.2.: Actual implementation of the BIST for the high voltage ADC.

The discharging can be performed by an ohmic resistor as well as by a switched capacitor; the resulting stimulus signal is exponentially decaying in either case. The choice for one of the methods depends on the needed discharging speed and on the capacitance of the capacitor array. The equivalence of the two approaches is mathematically demonstrated in the next subsection. Apart from the silicon area aspect, the main difference is that the switched capacitor has better matching properties to the capacitor array than an ohmic resistor. Hence, the time constant can be adjusted more precisely. For the used method this is not relevant because the discharging time constant is estimated in the calculation step and hence must not be very accurate.

The stimulus signal is generated as follows:

1. The capacitor array is grounded and charged to the reference voltage over the feedback path of the comparator.
2. The discharging element is connected to the capacitor array for one clock cycle.
3. Repeat 2. until the characterization algorithm is finished.

6.4.2. Discharging By Resistor or By Switched Capacitor

Let the capacitance of the capacitor array be $C_{array} = C_0 \cdot 2^N$, where C_0 is the unit capacitance. If an ohmic resistor with value R is used for the discharging process, the time constant is

$$\tau_r = R \cdot C_{array}\,, \tag{6.1}$$

and the resulting voltage curve is a decaying exponential:

$$V(t) = V_0 \exp\left(-\frac{t}{\tau_r}\right). \tag{6.2}$$

When a switched discharging capacitor $C_{discharge}$ is used, the time constant depends on the ratio r of the array capacitance to the sum of array capacitance and discharge capacitor. Each discharging cycle $i \in \mathbb{N}$ consists of two steps. At first, the empty discharge capacitor is connected in parallel to the capacitor array. The charge on the capacitor array $Q_i = C_{array} \cdot V_i$ is then distributed between them. The parallel capacitors have a total capacitance of $C_{array} + C_{discharge}$, hence the voltage on the capacitor array and the discharging capacitor reduces to

$$V_{i+1} = \frac{Q_i}{C_{array} + C_{discharge}} = V_i \cdot \frac{C_{array}}{C_{array} + C_{discharge}}\,. \tag{6.3}$$

The discharging capacitor is then disconnected and shorted, which removes the charge $\Delta Q_i = V_i \cdot C_{discharge}$. Then the next cycle $i+1$ begins with once again connecting the discharging capacitor in parallel to the capacitor array. The voltage on the capacitor array after the i-th discharging cycle can be obtained by recursively applying (6.3):

$$V_i = V_0 \cdot \left(\frac{C_{array}}{C_{array} + C_{discharge}}\right)^i. \tag{6.4}$$

Now the equivalence between discharging via an ohmic resistor and via a switched capacitor can be proven.

$$V_i = \exp\left(\ln\left(V_0 \cdot \left(\frac{C_{array}}{C_{array} + C_{discharge}}\right)^i\right)\right) \tag{6.5}$$

$$= V_0 \cdot \exp\left(i \cdot \ln\left(\frac{C_{array}}{C_{array} + C_{discharge}}\right)\right) \tag{6.6}$$

Let the cycle time t_{cycle} be the duration of one discharging cycle by means of the discharging capacitor. Then the time can be defined as an integer multiple of the discharge cycle time $t = i \cdot t_{cycle}$. If this is inserted in the above equation and the components are compared with (6.2), the equivalent time constant τ_{sc} can be found in

$$V(t = i \cdot t_{cycle}) = V_0 \cdot \exp\left(-t \cdot \underbrace{\frac{-1}{t_{cycle}} \cdot \ln\left(\frac{C_{array}}{C_{array} + C_{discharge}} \right)}_{\frac{1}{\tau_{sc}}} \right), \tag{6.7}$$

which yields

$$\tau_{sc} = -\frac{t_{cycle}}{\ln\left(\frac{C_{array}}{C_{array}+C_{discharge}} \right)} = \frac{t_{cycle}}{\ln\left(\frac{C_{array}+C_{discharge}}{C_{array}} \right)}. \tag{6.8}$$

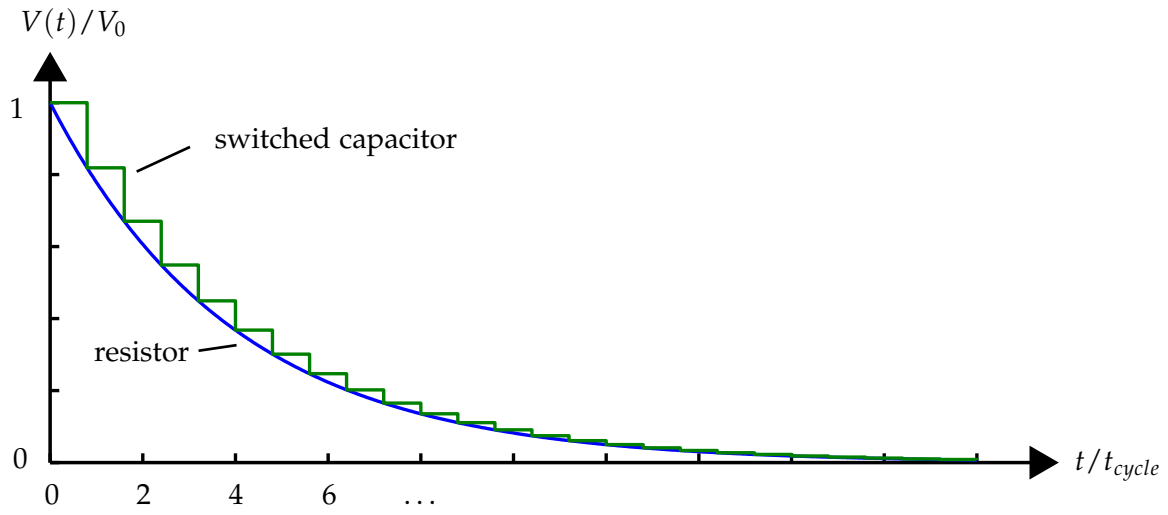

Figure 6.3.: Equivalence of discharging by means of a resistor or a switched capacitor for one exemplary set of device parameters.

As it can be seen in Fig. 6.3, the behavior is only equivalent at multiples of $t = i \cdot t_{cycle}$. In between the resistor discharges continuously, whereas the switched capacitor operates in quasi-discrete steps.

6.4.3. Dynamic Threshold Adaptation

The dynamic threshold adaptation enables the capacitor array characterization within one complete discharging process by measuring of the threshold crossing times. The adaptive threshold for the ADC is determined by the setting of the bottom plate switches.

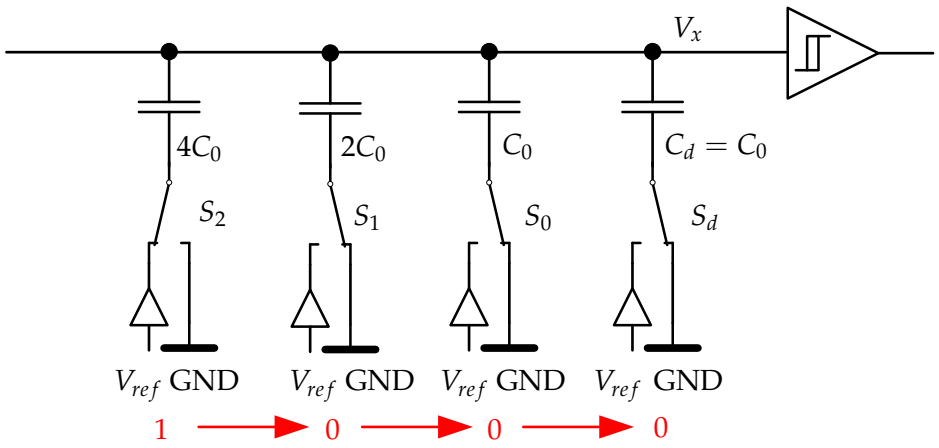

Figure 6.4.: Dynamic threshold adaptation by shifting a one through the bottom switch control register at the example of a 3-bit ADC.

The adaptation process is depicted in Fig. 6.4. The first threshold is determined by the MSB, while the other bits are set to 0. Once the threshold is reached, which is observed by an output change of the comparator, the next highest bit is set to 1, while the others are set to 0. This goes on and is equivalent to a shift register that shifts in zeros from the left side. This results in an exponentially lowered threshold as depicted in Fig. 6.5.

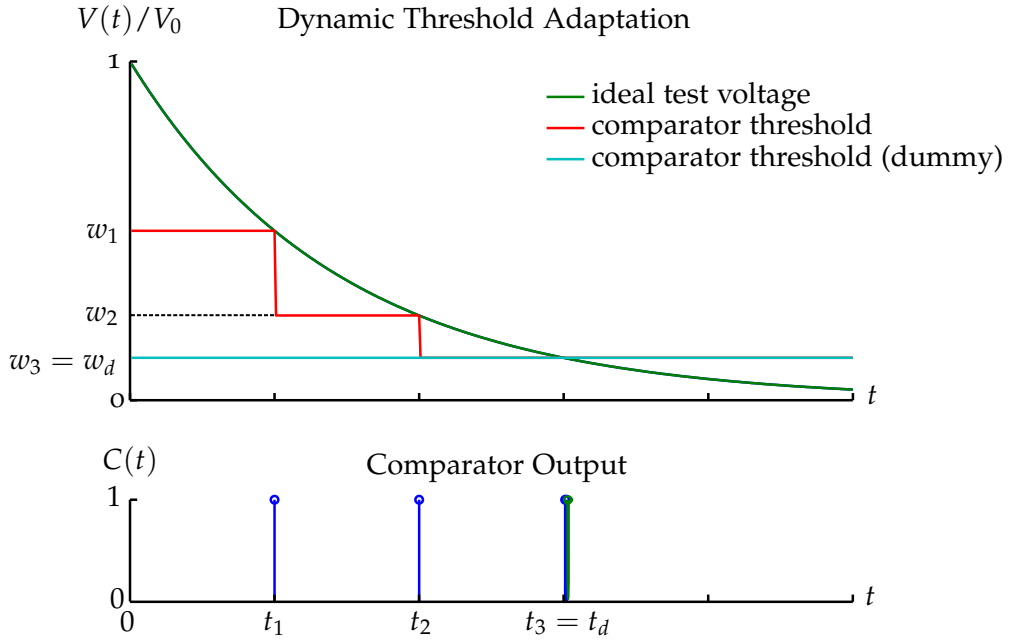

Figure 6.5.: Threshold crossing times for an ideal 3-bit ADC.

The times t_i are recorded when the corresponding thresholds w_i are crossed. The dummy

capacitor's threshold crossing time must be measured in parallel, because ideally its weight w_d is the same as the weight of the LSB w_N, but it can also be mismatched. The dummy threshold does not have to be measured for the highest bits, as it is very unlikely to have such a high weight due to mismatch. For the last few LSBs when the dummy is measured too, there is no need for a second comparator. The only change is that instead of discharging for one cycle and measuring the threshold in the next cycle, there is just an additional cycle when the dummy is also measured before the next discharge cycle.

The capacitor array for a N-bit ADC consists of $N+1$ capacitors, where the bottom plate switches of the capacitors $C_1, C_2, \dots C_N$ are each associated with one bit in the SAR register. The last capacitor $C_{N+1} = C_d$ is the dummy capacitor, which is required for a transfer curve that ends one LSB below the full scale voltage. Usually it is always grounded, but for the purpose of the BIST it is also equipped with a bottom plate switch.

The total capacitance of the capacitor array is defined as $C_{array} = \sum_{i=1}^{N+1} C_i$. When the bottom plate switch of the capacitor C_i connects it to the reference voltage, the comparator threshold is virtually raised by a fraction of the full scale voltage V_{FSR}, that is $\Delta V_t = V_{FSR} \cdot \frac{C_i}{C_{array}}$. The ratio is defined as the relative weight w_i of the bit i

$$w_i = \frac{C_i}{C_{array}} = \frac{C_i}{\sum_{i=1}^{N+1} C_i}\,, \tag{6.9}$$

where the index $i = N+1$ represents the weight of the dummy capacitor, which is not related to a bit in the SAR register. The most important observation is that all weights including the dummy weight inherently sum up to unity:

$$\sum_{i=1}^{N+1} w_i = \sum_{i=1}^{N+1} \frac{C_i}{\sum_{i=1}^{N+1} C_i} = \frac{\sum_{i=1}^{N+1} C_i}{\sum_{i=1}^{N+1} C_i} = 1\,. \tag{6.10}$$

The dummy weight $w_d = w_{N+1}$ is not part of all calculations. If the index for the weights ranges from 1 to $N+1$, then the dummy weight is part of the calculation. If the index only goes up to N, then the dummy weight is excluded from the calculation. To get the threshold voltage that corresponds to a relative weight, it has to be multiplied by the full scale voltage V_{FSR}:

$$V_{t,i} = V_{FSR} \cdot w_i\,. \tag{6.11}$$

The relative weight w_i can also easily be transformed to a weight W_i in relation to the LSB voltage:

$$W_i = 2^N \cdot w_i\,, \tag{6.12}$$

It translates into a threshold voltage by multiplication with the LSB voltage:

$$V_{t,i} = V_{FSR} \cdot w_i = V_{LSB} \cdot 2^N \cdot w_i = V_{LSB} \cdot W_i\,. \tag{6.13}$$

6.4.4. Number of Hits Per Weight

The number of hits per weight H is the number of discharge cycles necessary to reach the next threshold level, which is ideally half of the previous level. A higher number of hits per weight increases the measurement time, but at the same time also increases the measurement accuracy. The time measurement uncertainty stems from the granularity of full clock cycle periods. For a given H, the corresponding discharge time constant τ can be derived.

$$\frac{V_x}{2} \stackrel{!}{=} V_x \cdot \exp\left(-\frac{t}{\tau}\right) = V_x \cdot \exp\left(-\frac{H \cdot t_{cycle}}{\tau}\right) \tag{6.14}$$

$$\Rightarrow \tau = \frac{H \cdot t_{cycle}}{\ln 2} \tag{6.15}$$

The total capacitance of the capacitor array is usually given. As shown in section 6.4.2, the time constant can be implemented by using an adequately sized ohmic resistor or the equivalent capacitor.

6.5. Evaluation of Threshold Crossing Times

Once stored, the threshold crossing times obtained from the previously described algorithm can be processed in two ways. One method permits an accurate linearity evaluation and reconstruction of the ADC transfer curve, while the other method allows a fast judgment of whether the capacitor array matching lies within defined bounds.

Depending on the purpose of BIST, the evaluation can be performed on-chip with simple means or, if an exact linearity evaluation is required, by the ATE. The number of hits per weight should also be adjusted according to the required accuracy and the available measurement time.

6.5.1. Capacitor Array Characterization

This method enables measuring and calculating the bit weights accurately (depending on the chosen number of hits per weight H). The necessary calculations can be carried out either on the ATE or on the chip if a processing unit is available. The basis for the evaluation are the threshold crossing times, which are usually available as a counter value n_i which can be converted into a time by multiplication with the clock period: $t_i = n_i \cdot t_{clk}$.

When the voltage on the capacitor array (6.2) is looked at in relation to the reference voltage, the threshold crossing time t_i is directly related to the weight w_i by the exponential discharging function:

$$V_w(t) = \frac{V(t)}{V_{ref}} = \exp\left(-\frac{t}{\tau}\right) \tag{6.16}$$

$$\Rightarrow w_i = \exp\left(-\frac{t_i}{\tau}\right) \tag{6.17}$$

The nominal value of the time constant τ is known, but due to production process variations, the capacitor array and the discharge capacitor (or resistor) result in a mismatched time constant τ which is different from the nominal value τ_{typ}. Therefore, the weights obtained by plugging the nominal time constant into (6.17) would be potentially incorrect.

There are $N+1$ unknown weights for an N-bit ADC and additionally there is the unknown time constant τ. The measured threshold crossing times provide $N+1$ values, but the equation system is still unsolvable. The necessary additional information that makes the equation solvable is that all weights including the dummy weights must inherently sum up to one.

$$f(\tau) = \sum_{i=1}^{N+1} w_i = \sum_{i=1}^{N+1} \exp\left(-\frac{t_i}{\tau}\right) \stackrel{!}{=} 1 \tag{6.18}$$

In the time domain, there is no closed solution for this equation. Therefore, the straightforward approach is to apply a numeric method to solve the equation. Newton's method can be used to iteratively solve for the unknown τ: [2]

$$\tau_{n+1} = \tau_n - \frac{f(\tau)}{f'(\tau)}\,. \tag{6.19}$$

The required first derivative of $f(\tau)$ can be calculated as

$$f'(\tau) = \frac{\mathrm{d}f(\tau)}{\mathrm{d}\tau} = \frac{1}{\tau^2}\left(\sum_{i=1}^{N+1} t_i \cdot \exp\left(-\frac{t_i}{\tau}\right)\right) \tag{6.20}$$

The numerical solution for τ converges after a few iterations for a good starting point, for which the nominal time constant can be used. If the time constant does not converge, the update step size can be decreased by a downscaling factor. Subsequently, the capacitor array weights can be calculated by evaluating (6.17). Based on the obtained weights, the transfer curve, INL and DNL can be calculated as is illustrated in the next sections, which means that it is sufficiently characterized.

6.5.1.1. Calculation of the ADC Transfer Curve

The ADC transfer curve is completely defined by the knowledge of the single bit weights $w_i,\ i \in \{1,2,\ldots,N\}$. The dummy weight is not required. The following calculations ignore any gain- or offset error, but they could be simply included by multiplying the result with the correct gain and adding of the offset error.

For a N-bit ADC, the possible output codes Q are in the decimal range from 0 to $2^N - 1$. Let the N-bit unsigned binary representation of a code be denoted by the vector $\boldsymbol{b}^T = \begin{bmatrix} b_1 & b_2 & \ldots & b_N \end{bmatrix}, b_i \in \{0,1\}$, where b_1 is the MSB and b_N is the LSB. Using the definition of the weights vector as $\boldsymbol{w}^T = \begin{bmatrix} w_1 & w_2 & \ldots & w_N \end{bmatrix}, w_i \in \mathbb{R}^+$, the transition voltage can be calculated as

$$V_t(Q) = V_t(\boldsymbol{b}, \boldsymbol{w}) = V_{LSB} \cdot \boldsymbol{b}^T \cdot \boldsymbol{w} = V_{LSB} \cdot \left(\sum_{i=1}^{N} b_i \cdot w_i \right) \tag{6.21}$$

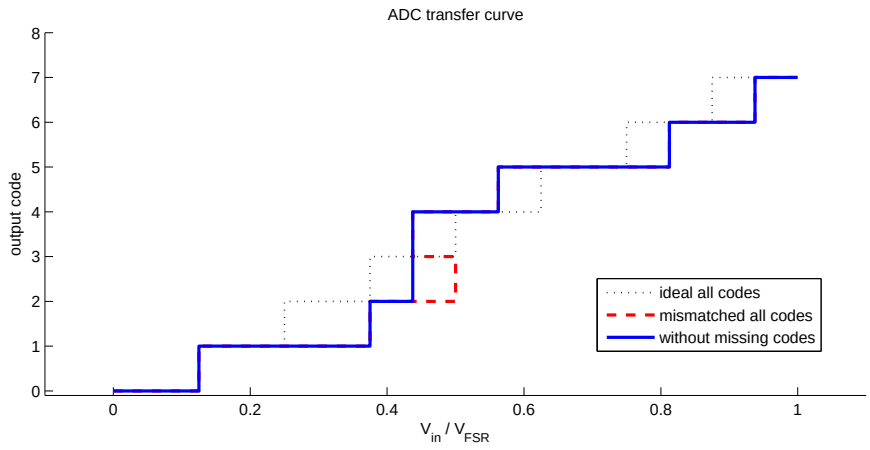

Figure 6.6.: Full transfer curve with and without missing codes for a mismatched 3-bit ADC.

Missing Codes The transfer curve can be plotted by evaluating (6.21) for each code Q, that is for all $\boldsymbol{b} \in \{0,1\}^N$ in the binary representation. An example for such a direct evaluation is shown in Fig. 6.6. It is interesting to note that in the red dashed part of the curve the voltage to code mapping is ambiguous because of the mismatched weights, where the ratio between successive weights is not two. The successive approximation algorithm evaluates the bits from MSB to LSB. If there exists a weight which is smaller than the sum of all lesser weights, then missing codes exist:

$$\exists i \in \{1,2,\ldots N-1\} : \sum_{j=i+1}^{N} w_j > w_i \quad \Leftrightarrow \quad \exists \,\text{missing code} \tag{6.22}$$

The mismatched bit weights used in Fig. 6.6 are contained in Table 6.1. The evaluation of (6.22) is shown in the second row and, as expected, the weight w_1 is smaller than the sum $w_2 + w_3$.

i	1 (MSB)	2	3 (LSB)
w_i	7/16	6/16	2/16
$c_i = \sum_{j=i+1}^{N} w_j$	8/16	2/16	0
$c_i \overset{?}{\geq} w_i$	1	0	0

Table 6.1.: Example for mismatched weights which lead to a missing code in a $N = 3$-bit ADC.

Table 6.2 shows all transition voltages which are the result of evaluating (6.21) for all codes. Once it is known that missing codes exist for a given set of weights, the question arises which specific codes actually are missing. In the approximation process, the weights of the more significant bits take precedence over the weights of less significant bits. If a bit has been determined as being set, then the search window, as for example indicated in Fig.5.2 in the second conversion step, excludes all lower codes. Therefore, codes are missing if there exists a higher code with a lower transition voltage such that it masks all lower codes with higher transition voltages. In Table 6.2 this is the case for the code $Q = 3$, where the transition voltage is higher than the transition voltage for $Q = 4$. If, for example, the voltage $V_{in} = 8/16 \cdot V_{FSR}$ is applied to the ADC, in the first conversion step the MSB with the weight $w_1 = 7/16$ is measured as high. Obviously, all codes lower than $Q = 4$ will never be considered as they are outside the constricted search window. When a lower voltage is applied, the MSB will be detected as low, but the transition voltage $T(3) = 8/16 \cdot V_{FSR}$ for $Q = 3$ is now higher than the applied voltage and by this contradiction the code $Q = 3$ can never be reached.

Q	0	1	2	3	4	5	6	7
$V_t(Q)/V_{LSB}$	0	2/16	6/16	8/16	7/16	9/16	13/16	15/16
$\Delta V_t(Q) = V_t(Q+1) - V_t(Q)$		4/16	2/16	−1/16	2/16	4/16	2/16	
missing		0	0	1	0	0	0	

Table 6.2.: All transition voltages for the mismatched $N = 3$-bit ADC example.

According to the IEEE standard 1241 [11], a code is already missing if its stair width is less than 90% of the nominal stair width. The above considerations regard a code as missing only when its stair width is actually less than zero.

Threshold voltages associated with missing codes, which do not show up in the transfer curve, can still be produced by means of the fast compare conversion. As it does not use the successive approximation algorithm, there is no masking effect involved and every digital code can be used – in the context of fast compare measurements there are no missing codes.

Based on the formulas shown in section 3.2.2.2, the INL and DNL can be calculated for each code. Commonly, in data sheets only the minimum and maximum values for the linearity measures are given. Calculating all code steps and the respective linearity errors is feasible for small values of N, but for high resolution ADCs, the exponential growth of codes with $\mathcal{O}(2^N)$ makes the calculation cumbersome if only minimum and maximum linearity errors are sought after. Therefore, a method to calculate the minimum and maximum linearity errors directly in linear time $\mathcal{O}(N)$ is presented in the next two sections.

6.5.1.2. Simplified Calculation of Worst Case INL

For the purpose of this and the following chapter, the mid-riser convention is used to define the INL and DNL errors. Hence, the integral nonlinearity (INL) errors are defined as the difference between the actual transition voltage for a code and the corresponding ideal transition voltage. The goal is to find expressions for the minimum and maximum INL without having to evaluate all code transition. For an N-bit ADC, let the possibly mismatched real bit weights be $\boldsymbol{w}^T = \begin{bmatrix} w_1 & w_2 & \dots & w_N \end{bmatrix}$ and the ideal weights be $\boldsymbol{\pi}^T = \begin{bmatrix} \pi_1 & \pi_2 & \dots & \pi_N \end{bmatrix}$, where index 1 stands for the MSB and N for the LSB. The dummy weight is not required, as it plays no role for the transition voltages on which the INL is based.

The INL is defined as the difference between the ideal and the actual transition voltages for a code Q [11]. Using the binary vector representation $\boldsymbol{b}(Q)$, the INL calculation can be stated with the help of (6.21) in relation to the LSB voltage as

$$INL_{LSB}(Q) = \frac{V_t(\boldsymbol{b}, \boldsymbol{w}) - V_t(\boldsymbol{b}, \boldsymbol{\pi})}{V_{LSB}} \tag{6.23}$$

$$= \frac{V_{FSR} \cdot \left(\sum_{i=1}^{N} b_i \cdot w_i\right) - V_{FSR} \cdot \left(\sum_{i=1}^{N} b_i \cdot w_i\right)}{V_{LSB}} \tag{6.24}$$

$$= \frac{V_{FSR}}{V_{LSB}} \cdot \left(\sum_{i=1}^{N} b_i \cdot \underbrace{(w_i - \pi_i)}_{\Delta w_i} \right) \tag{6.25}$$

$$= 2^N \cdot \left(\sum_{i=1}^{N} \Delta w_i \right) . \tag{6.26}$$

With this formula, the INL for each code can be calculated directly. For calculating the minimum and maximum INL over all codes, a close look on (6.26) reveals that the highest INL value can be reached with a code $\boldsymbol{b}$ where all $\Delta w_i > 0$ add to the result and all bits where $\Delta w_i < 0$ are masked. This is summarized in (6.27) and the same considerations lead to (6.28) for the minimum INL. All possible INL values reside within the range (6.29) in between the calculated minimum and maximum INL values. This range in units of the LSB voltage V_{LSB} is usually found in data sheets of ADCs. As only the N differences between ideal and mismatched weights must be evaluated, the range can be determined very efficiently – especially for high-resolution ADCs.

$$\mathrm{INL}_{LSB,max} = 2^N \left(\sum_{n \in \eta} \Delta w_n \right), \quad \eta = \{i | \Delta w_i > 0\} \tag{6.27}$$

$$\mathrm{INL}_{LSB,min} = 2^N \left(\sum_{n \in \xi} \Delta w_n \right), \quad \xi = \{i | \Delta w_i < 0\} \tag{6.28}$$

$$\Rightarrow \mathrm{INL}_{LSB}(Q) \in [\mathrm{INL}_{LSB,min}, \mathrm{INL}_{LSB,max}] \tag{6.29}$$

6.5.1.3. Simplified Calculation of Worst Case DNL

Similar to the direct INL calculation, there is also a way to obtain the minimum and maximum differential nonlinearity (DNL) values without resorting to calculating the DNL of every code. The same definitions and conventions as described in the previous section apply. The DNL is defined as the difference between a code step width and the ideal width [11] . With the help of (6.21), it can be stated as

$$\begin{aligned}
\mathrm{DNL}_{LSB}(Q) &= \frac{V_t(\boldsymbol{b}(Q+1),\boldsymbol{w}) - V_t(\boldsymbol{b}(Q),\boldsymbol{w}) - V_{LSB}}{V_{LSB}} && (6.30)\\
&= \frac{V_{FSR}}{V_{LSB}} \cdot \left(\sum_{i=1}^{N} b(Q+1)_i \cdot w_i - \sum_{i=1}^{N} b(Q)_i \cdot w_i \right) - 1 && (6.31)\\
&= 2^N \cdot \left(\sum_{i=1}^{N} \underbrace{(b(Q+1)_i - b(Q)_i)}_{\Delta b(Q)_i} \cdot w_i \right) - 1 && (6.32)\\
&= 2^N \cdot (V_t(\boldsymbol{b}(Q+1) - \boldsymbol{b}(Q), \boldsymbol{w})) - 1 && (6.33)\\
&= 2^N \cdot (V_t(\boldsymbol{\Delta b}(Q), \boldsymbol{w})) - 1 && (6.34)
\end{aligned}$$

The difference between the decimal codes $Q+1$ and Q is defined as the arithmetic difference between two binary vectors $\boldsymbol{b}(Q+1) - \boldsymbol{b}(Q)$. While the elements of the vectors $\boldsymbol{b}$ can only be zero or one, the difference vector $\boldsymbol{\Delta b}(Q)$ between two successive binary codes has elements in $\{-1, 0, 1\}$. So far, the calculation has not been simplified significantly. The analysis of $\boldsymbol{\Delta b}(Q)$ reveals that for N-bit values only N distinct differences occur between successive codes. They are given in (6.35).

$$\boldsymbol{\Delta B}_{distinct} = \begin{bmatrix} \boldsymbol{\Delta b}(2^0)^T \\ \boldsymbol{\Delta b}(2^1)^T \\ \boldsymbol{\Delta b}(2^2)^T \\ \vdots \\ \boldsymbol{\Delta b}(2^{N-1})^T \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & -1 \\ & \ddots & \ddots & \ddots & \vdots \\ 1 & -1 & \cdots & -1 & -1 \end{bmatrix} \tag{6.35}$$

Multiplication of any of the vectors with the base vector reveals that the decimal difference between the vectors is 1:

$$\boldsymbol{\Delta b}(2^i)^T \cdot \begin{bmatrix} 2^{N-1} \\ 2^{N-2} \\ \cdots \\ 2^0 \end{bmatrix} = 1, \quad i \in \{0, 1, \ldots, N-1\} \tag{6.36}$$

The vector $\mathbf{DNL}_{distinct}$ containing all N distinct DNL values can be obtained by a matrix multiplication based on (6.34):

$$\mathbf{DNL}_{distinct} = 2^N \cdot \boldsymbol{\Delta B}_{distinct} \cdot \boldsymbol{w} - 1 \tag{6.37}$$

Hence, due to the binary weight ratios of the capacitor array, only N distinct DNL values occur over the whole code range. The DNL associated with setting the MSB bit occurs only once, the lower bit values occur more frequently depending on the bit position. Consequently, the DNL plot exhibits symmetry. As a result, the minimum and maximum DNL values can be found by simply looking at the N distinct DNL values for a N-bit ADC.

$$\mathrm{DNL}_{max} = \max\left(\mathbf{DNL}_{distinct}\right) \tag{6.38}$$
$$\mathrm{DNL}_{min} = \min\left(\mathbf{DNL}_{distinct}\right) \tag{6.39}$$
$$\Rightarrow \mathrm{DNL}(Q) \in [\mathrm{DNL}_{min}, \mathrm{DNL}_{max}] \tag{6.40}$$

6.5.2. Basic Capacitor Matching Estimation

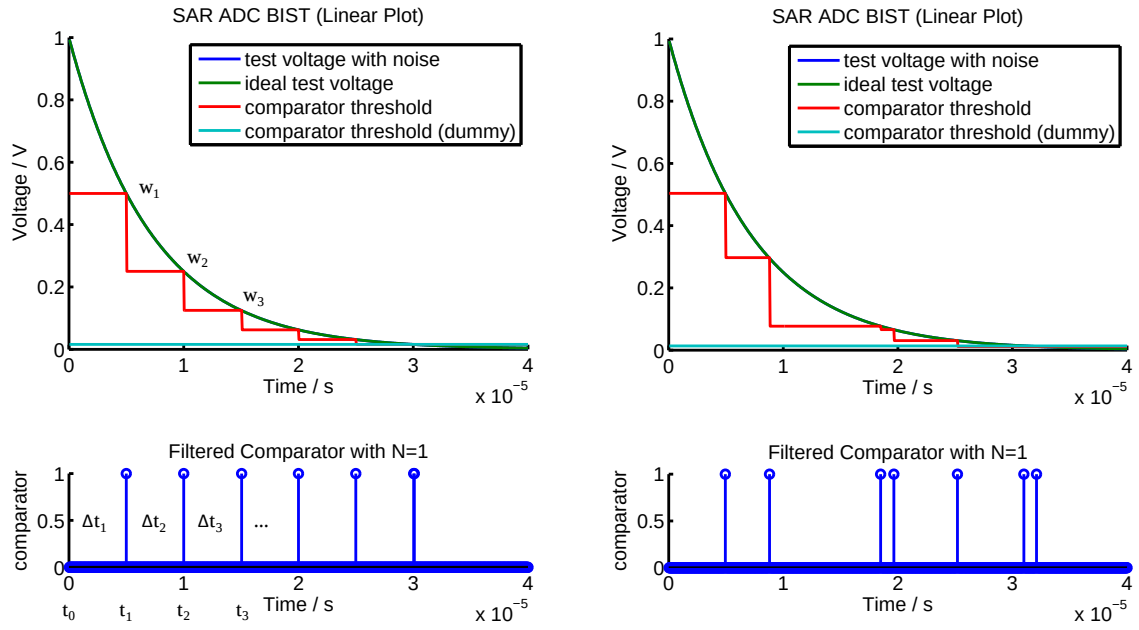

Figure 6.7.: Threshold crossing times for ideal matching and mismatched capacitor array.

The idea behind this method can be grasped by looking at Fig. 6.7, where the ideal capacitor array leads to constant time differences between successive threshold crossing times, but a mismatched capacitor array results in varying time differences. Based on this observation, the goal is to derive an expression that allows checking if all active components of the ADC (switches, comparator, buffer) work and additionally, if the matching of the capacitor array is within acceptable limits.

As the time constant is not available, all calculations are based on the measured threshold crossing times. It should be noted that this method is very inaccurate in comparison with the previous method; hence, it should only be employed when very little computation power is available.

6.5.2.1. Relative Mismatch for Successive Weights

The actual weights are not calculated for the sake of the simplicity of the calculations. However, bounds for the maximum permitted time difference variation are derived which limit the maximum mismatch between successive weights. Therefore, within predefined bounds the method should decide whether the ADC is within specification limits or not. Starting out with (6.17), the ratio of the successive weights w_i and w_{i+1} can be stated as

$$\frac{w_i}{w_{i+1}} = \frac{\exp\left(-\frac{t_i}{\tau}\right)}{\exp\left(-\frac{t_{i+1}}{\tau}\right)} \tag{6.41}$$

$$= \exp\left(-\frac{t_i - t_{i+1}}{\tau}\right), \tag{6.42}$$

which, after taking the natural logarithm on both sides, leads to the expression

$$\ln\frac{w_i}{w_{i+1}} = \frac{t_{i+1} - t_i}{\tau}. \tag{6.43}$$

The unknown time constant τ still appears in the above equation. To get rid of it, the ratio of the weights w_i/w_{i+1} can be put in relation to the next ratio w_{i+1}/w_{i+2}:

$$\frac{\ln\frac{w_{i+1}}{w_{i+2}}}{\ln\frac{w_i}{w_{i+1}}} = \frac{\frac{t_{i+2}-t_{i+1}}{\tau}}{\frac{t_{i+1}-t_i}{\tau}} \tag{6.44}$$

$$= \frac{t_{i+2} - t_{i+1}}{t_{i+1} - t_i}. \tag{6.45}$$

Replacing the time differences by a new variable $\Delta t_i = t_i - t_{i-1}$ and multiplying both sides by the denominator of the left side gives

$$\ln\frac{w_{i+1}}{w_{i+2}} = \ln\frac{w_i}{w_{i+1}} \cdot \frac{\Delta t_{i+2}}{\Delta t_{i+1}}. \tag{6.46}$$

As the logarithm appears in (6.46), it is infeasible to evaluate it on simple computing hardware. As the expected mismatch for the ratio between adjacent integrated capacitors is low for a working ADC, the equation can be linearized. The ideal ratio between successive weights is given by the ideal weights $a = \frac{\pi_{i+1}}{\pi_{i+2}}$. For small deviations from that ideal point, the logarithm can be linearized by a Taylor series. In general, the Taylor series of $\ln(x)$ at point a is: [2]

$$\ln(x) = \sum_{n=0}^{\infty} \frac{\frac{\mathrm{d}^n \ln(a)}{\mathrm{d}x^n}}{n!}(x-a)^n, \tag{6.47}$$

where all terms with $n \geq 2$ can be neglected for small $x - a$, which results in the approximation

$$\ln(x) \approx \ln(a) + \frac{x-a}{a} = \ln(a) - 1 + \frac{x}{a}. \tag{6.48}$$

Substituting with the correct variables leads to the approximation of $\ln\left(\frac{w_{i+1}}{w_{i+2}}\right)$ around the expansion point $\frac{\pi_{i+1}}{\pi_{i+2}}$:

$$\ln\left(\frac{w_{i+1}}{w_{i+2}}\right) \approx \ln\left(\frac{\pi_{i+1}}{\pi_{i+2}}\right) - 1 + \frac{w_{i+1}\pi_{i+2}}{\pi_{i+1}w_{i+2}}. \tag{6.49}$$

When the simplification (6.49) is used to replace the logarithm on the left side in (6.46), this results in

$$\ln\left(\frac{\pi_{i+1}}{\pi_{i+2}}\right) - 1 + \frac{w_{i+1}\pi_{i+2}}{\pi_{i+1}w_{i+2}} = \ln\frac{w_i}{w_{i+1}} \cdot \frac{\Delta t_{i+2}}{\Delta t_{i+1}}. \tag{6.50}$$

Some rearranging leads to an expression for $\frac{w_{i+1}}{w_{i+2}}$:

$$\frac{w_{i+1}\pi_{i+2}}{\pi_{i+1}w_{i+2}} = 1 - \ln\left(\frac{\pi_{i+1}}{\pi_{i+2}}\right) + \ln\frac{w_i}{w_{i+1}} \cdot \frac{\Delta t_{i+2}}{\Delta t_{i+1}} \tag{6.51}$$

$$\Rightarrow \frac{w_{i+1}}{w_{i+2}} = \frac{\pi_{i+1}}{\pi_{i+2}}\left(1 - \ln\left(\frac{\pi_{i+1}}{\pi_{i+2}}\right) + \ln\frac{w_i}{w_{i+1}} \cdot \frac{\Delta t_{i+2}}{\Delta t_{i+1}}\right). \tag{6.52}$$

An expression where only $\frac{w_{i+1}}{w_{i+2}}$ is a function of Δt_{i+1} and Δt_{i+2} is the goal of the derivations. All other variables or ratios must be assumed as ideal. The ideal ratio between successive weights $\frac{\pi_{i+1}}{\pi_{i+2}}$ is defined by a new variable R. For a binary weighted ADC $R = 2$, but this assignment is not performed yet to keep the equations valid for any capacitor ratio. The ratio which represents the expansion point for the linearization of the logarithm is $\frac{\pi_{i+1}}{\pi_{i+2}} \stackrel{!}{=} R$. Simultaneously, it can be assumed that the weight ratio $\frac{w_i}{w_{i+1}}$ is also ideal, consequently $\frac{w_i}{w_{i+1}} \stackrel{!}{=} \frac{\pi_i}{\pi_{i+1}} = R$. Substituting these simplification in (6.52) yields

$$\frac{w_{i+1}}{w_{i+2}} = R \cdot \left(1 - \ln(R) + \ln(R) \cdot \frac{\Delta t_{i+2}}{\Delta t_{i+1}}\right). \tag{6.53}$$

For simplicity, the index is substituted as $j = i + 1$:

$$\frac{w_j}{w_{j+1}} = R \cdot \left(1 - \ln(R) + \ln(R) \cdot \frac{\Delta t_{j+1}}{\Delta t_j}\right). \tag{6.54}$$

The time difference ratio can be defined in a relative way, which allows to cancel the constant term $-\ln(R)$:

$$\frac{w_j}{w_{j+1}} = R \cdot \left(1 - \ln(R) + \ln(R) \cdot \left(1 + \frac{\Delta t_{j+1} - \Delta t_j}{\Delta t_j}\right)\right) \tag{6.55}$$

$$= R \cdot \left(1 + \ln(R) \cdot \frac{\Delta t_{j+1} - \Delta t_j}{\Delta t_j}\right). \tag{6.56}$$

Expanding the right side results in the general equation, which is valid for any ideal weights ration R:

$$\frac{w_j}{w_{j+1}} = R + \ln(R^R) \cdot \frac{\Delta t_{j+1} - \Delta t_j}{\Delta t_j}. \tag{6.57}$$

Equation (6.57) represents a universal expression for the linearized mismatch calculation. Due to many simplifications that assume all unknown variable as ideal, it should not be used to iteratively calculate the weights from the measured times. The intended application is to state a sufficient condition which guarantees that when a certain time deviation between successive threshold crossing times is not exceeded, the mismatch between the weights is within certain limits.

The time limits corresponding to a certain mismatch limit can be precomputed and the algorithm on the chip only has to measure the time differences and check whether they are below the limit. Hence, the weights themselves are never directly calculated and the only sound assumption is therefore to assume all other weights as ideal – as it has been done in the previous calculations. The sufficient condition for small mismatches can be stated as follows:

$$\left|\frac{\Delta t_{j+1} - \Delta t_j}{\Delta t_j}\right| < \varepsilon_j \quad \Leftrightarrow \quad \frac{w_j}{w_{j+1}} \in \left[R - \varepsilon_j \cdot \ln(R^R), R + \varepsilon_j \cdot \ln(R^R)\right] \tag{6.58}$$

6.5.2.2. Relative Mismatch for Successive Weights in a Binary Weighted Capacitor Array

For a binary weighted ADC the ratio $R = 2$, which turns (6.57) and (6.58) to

$$\frac{w_j}{w_{j+1}} = 2 + \ln(4) \cdot \frac{\Delta t_{j+1} - \Delta t_j}{\Delta t_j}, \tag{6.59}$$

and

$$\left|\frac{\Delta t_{j+1} - \Delta t_j}{\Delta t_j}\right| < \varepsilon_j \quad \Leftrightarrow \quad \frac{w_j}{w_{j+1}} \in \left[2 - \varepsilon_j \cdot \ln(4), 2 + \varepsilon_j \cdot \ln(4)\right]. \tag{6.60}$$

A look at Fig. 6.7 permits to verify (6.59) intuitively. For instance, let $j = 1$, which means that $\frac{w_1}{w_2} = 2 + \ln(4) \cdot \frac{\Delta t_2 - \Delta t_1}{\Delta t_1}$. If the time Δt_2 is longer than Δt_1, the voltage associated with the

weight w_2 has discharged for a longer time than w_2 and hence the ratio $\frac{w_1}{w_2}$ is larger than it should be. This is exactly what (6.59) predicts. Please note that no absolute interpretation can be derived from this, because the matching estimation lacks an absolute reference. It is possible that w_1 is ideal and w_2 is smaller than it should be, or that w_1 is larger than it should be and w_2 is ideal. All cases in between are also possible. However, the sum of the weights including the dummy weight inherently sums up to one due to the capacitor array structure – hence, the correct relative matching allows inferring that the absolute weights of the capacitor array match the ideal weights well.

The dummy weight has not been explicitly treated so far. It must be verified, because it is possible that all normal bit weights match perfectly, but due to a very mismatched dummy weight the absolute values of the weights are very far off. The dummy weight $w_{N+1} = w_d$ ideally has the same value as the LSB weight w_N: $\pi_d \stackrel{!}{=} \pi_N$. Therefore, it can be checked by using the same procedure as employed for the LSB weight w_N, that is the dummy weight should be inspected by the ratio $\frac{w_{N-1}}{w_d}$ and the relative time difference to be looked at is $\frac{\Delta t_d - \Delta t_{N-1}}{\Delta t_{N-1}}$. The time difference Δt_d is defined as the time between the threshold crossing before the LSB and the dummy weight threshold crossing, that is $\Delta t_d = t_d - t_{N-1}$.

6.5.2.3. Weight Error Limits in a Binary Weighted Capacitor Array

The methods so far empower to limit the relative mismatch between successive weights, but this calculation lacks an absolute reference. When a capacitor array is severely mismatched or if some part of the ADC is broken, the exact determination of the mismatch is not relevant, as the presence of the mismatch is detected and the part is rejected. When the mismatches are small, it is feasible to infer an absolute deviation of a weight under the assumption that all other weights are ideal. The limitation for small mismatches is apparent because the assumption that one weight is mismatched and all others are ideal is a contradiction as all weights must always inherently sum up to one. Let δ_j be the relative error for a weight:

$$w_j = \pi_j \cdot (1 + \delta_j) = \pi_j + \underbrace{\pi_j \cdot \delta_j}_{\Delta w_j} . \tag{6.61}$$

Then the absolute error Δw_j based on the ideal value π_j of a weight is:

$$\Delta w_j = \pi_j \cdot \delta_j = 2^{-j} \cdot \delta_j . \tag{6.62}$$

If a constant relative error $\delta_j = \delta$ is used, then the permitted absolute deviation for a weight decreases exponentially towards the LSB. Consequently, it is appropriate to scale the relative mismatch boundary δ_j depending on the bit position to get a constant absolute mismatch bound in relation to the LSB:

$$\delta_j \stackrel{!}{=} \delta_0 \cdot 2^j . \tag{6.63}$$

Plugging (6.63) into (6.62) results in

$$\Delta w_j = 2^{-j} \cdot \delta_0 \cdot 2^j = \delta_0 , \tag{6.64}$$

which is indeed a constant value – hence, all weights are allowed the same absolute mismatch. For a binary weighted capacitor array this scaling by 2^j can be efficiently implemented by bit shift operations. The relative error δ_j must be converted into a limit for the time differences ε_j, such that based on the times it can be checked if the relative weight mismatch criterion is fulfilled. Equation (6.59) can be solved for w_j:

$$w_j = \underbrace{w_{j+1}}_{\stackrel{!}{=}\pi_{j+1}} \left(2 + \ln(4) \cdot \underbrace{\frac{\Delta t_{j+1} - \Delta t_j}{\Delta t_j}}_{\varepsilon_j} \right) , \tag{6.65}$$

where the weight w_{j+1} is, as discussed earlier, substituted with the ideal weight π_{j+1}. Then some rearranging is performed to find an expression where the error occurs separately:

$$\begin{aligned} w_j &= 2^{-j-1} \left(2 + \ln(4) \cdot \varepsilon_j \right) && (6.66) \\ &= \underbrace{2^{-j}}_{\pi_j} + \underbrace{2^{-j-1} \cdot \ln(4) \cdot \varepsilon_j}_{\Delta w_j} . && (6.67) \end{aligned}$$

By equating the above absolute error term Δw_j with (6.62), an expression for the allowed (small) relative mismatch between successive weights ϵ_j based on a given absolute weight error is obtained.

$$\begin{aligned} 2^{-j-1} \cdot \ln(4) \cdot \varepsilon_j &\stackrel{!}{=} w_j && (6.68) \\ &\stackrel{!}{=} 2^{-j} \cdot \delta_j && (6.69) \\ \Rightarrow \varepsilon_j &= \frac{2}{\ln 4} \cdot \delta_j && (6.70) \\ & && (6.71) \end{aligned}$$

This conversion allows translating a permitted maximum absolute weight error into a limit for the relative deviation of threshold crossing time differences. For a constant absolute weight error $\delta_j \stackrel{!}{=} \delta_0 \cdot 2^j$ and subsequently

$$\varepsilon_j = \frac{2^{j+1}}{\ln 4} \cdot \delta_0 . \tag{6.72}$$

6.6. Deviations from Ideal Behavior

All calculations so far have assumed that the threshold crossing times t_i can be measured with unlimited accuracy. For the BIST in a real ADC, the sources of inaccuracies which influence the measurement are discussed in this section. The deviation of weights from their ideal value because of mismatched capacitors is not considered an inaccuracy, because they are the errors which are to be quantified by the BIST.

6.6.1. Clock Period

The measurement of the threshold crossing times is based on counters, which operate on the clock frequency. The jitter Δt_{clk} of a clock source makes the i-th triggered edge of a clock fluctuate according to $t_{clk,i} = i \cdot t_{clk,i} \pm \Delta t_{clk}$. Therefore, for a time difference measurement it is obtained that $t_{clk,i} - t_{clk,j} = (i-j) \cdot t_{clk,i} \pm 2\Delta t_{clk}$, hence the inaccuracy is $\pm 2\Delta t_{clk}$.

The time measurement granularity is limited by the clock, because the information of when the threshold crossing occurred during a clock period is lost. For a time difference measurement, the worst case error at the start and end of the measurement add up to $2t_{clk}$. The absolute relative error $|\Delta t_j / t_j|$ of a measured time difference t_j between the BIST start and the threshold crossing is

$$\left| \frac{\Delta t_j}{t_j} \right| = \frac{2t_{clk} + 2\Delta t_{clk}}{t_{clk} \cdot H}\,, \tag{6.73}$$

which means that for a high number of hits per weights H, the relative error tends towards zero:

$$\lim_{H \to \infty} \left| \frac{\Delta t_i}{t_i} \right| = 0\,. \tag{6.74}$$

6.6.2. Offset and Gain Errors

The offset and gain errors of the ADC are systematic errors. The existing ADC already provides methods to calibrate both, but they will never be completely zero. The gain error is determined by the accuracy of the reference voltage source and by the matching between the gain capacitor of a channel and the capacitor array. In the above equations the gain cancels, which means that gain errors do not influence the accuracy of the BIST. However, the offset error does not cancel and hence distorts the threshold crossing times and subsequently the calculated weights. A realistic offset error has little influence on the most significant bits, but a considerable influence on the LSB. The error propagation calculation in section 6.6.4 includes the effects of an offset error.

6.6.3. Noise and Comparator Output Filter

In a real ADC a certain level of noise can be observed at the comparator input. It partially emanates from the reference voltage buffer and has a direct influence on the measurement of the threshold crossing times. Furthermore, the capacitor array itself is a source of kT/C noise, which can only be minimized by increasing the capacitance. It is the result of thermodynamic noise and due to the inherent low-pass behavior independent of the bandwidth: [16]

$$\Delta V^2 = \frac{k \cdot T}{C}\,, \tag{6.75}$$

where k is the Boltzmann constant, T is the absolute temperature in Kelvin and C is the capacitance. In full conversion measurements, the noise has the same influence as during the BIST, but repeating measurements and averaging of the results can reduce the noise influence.

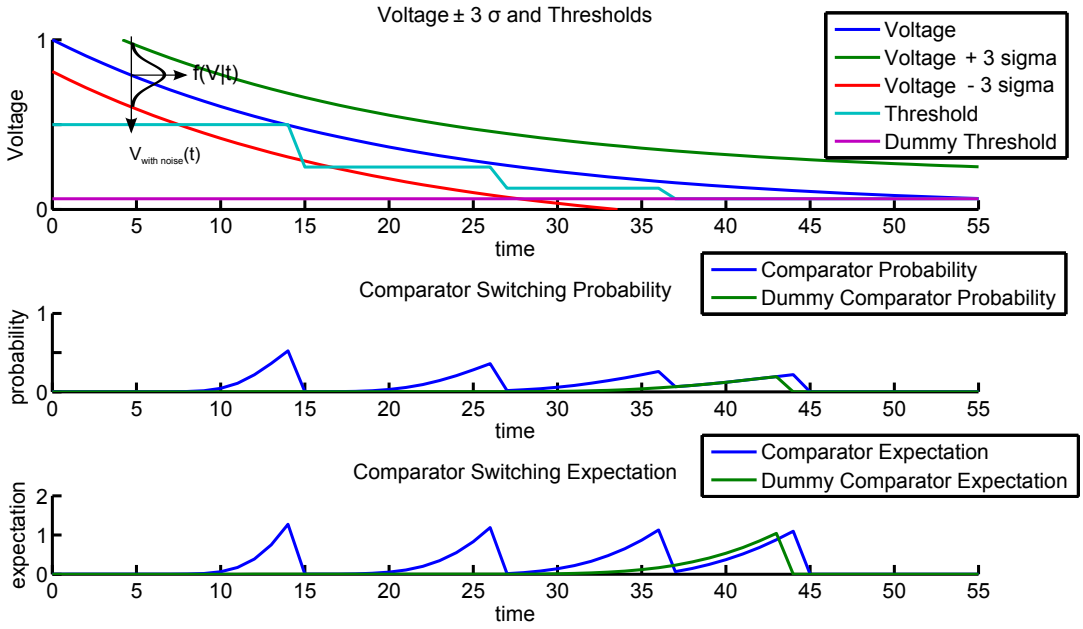

Figure 6.8.: Discharge voltage curve with AWGN and $\sigma = V_{LSB}$ for a 4-bit ADC.

When the time constant is low, the discharging voltage curve is very steep and the noise has little influence. On the other hand, when the time constant is higher because of a high number of hits per weight, the voltage curve stays close to the noise floor for a long time. This is depicted in Fig.6.8 under the assumption of additive white Gaussian noise (AWGN) with $\sigma = V_{\text{noise}} = V_{LSB}$:

$$V_{\text{with noise}}(t) \sim \mathcal{N}(V_{\text{no noise}}(t), \sigma^2)\,. \tag{6.76}$$

The discharging voltage curve is depicted without noise, but the range of $\pm 3\sigma$ is shown in which 99.73% of all noise values reside. In the middle plot the switching probability of the comparator is shown as a function of the time:

$$\mathrm{P}(V_{\text{with noise}}(t) < V_{th}(t)|t) = \Phi\left(\frac{V_{th}(t) - V_{\text{no noise}}(t)}{\sigma}\right), \tag{6.77}$$

where Φ is the cumulative density function (CDF) of the standard normal distribution. The plot on the bottom shows the expectation that the comparator switches. As the time steps are discrete and the comparator output value is either zero or one, the expectation is simply the sum of the probabilities. In the plot the threshold crossing time is defined as the moment

when the comparator expectation is one, that is when the comparator would switch on average. The expectation is then reset for the next threshold. A closer look on the switching probabilities reveals that the switching expectation increases slower for the less significant bits. The expected threshold crossing time for the MSB is rather accurate, but for the LSB the switching happens too early. The reason is that the voltage curve approaches the LSB weight very slowly and hence the noise is very likely to prematurely trigger the comparator. This distorts the measured threshold crossing times.

For low time constants and a noise voltage that is significantly lower than the LSB voltage, the time measurement error is negligible. In other cases, the comparator output can be filtered to deal with the noise.

6.6.3.1. Median Output Filtering and Simple Delay Compensation

A filter on the output of the comparator can diminish the noise influence and enable the BIST method to achieve good results even though measurements are performed immersed in the noise floor. A binary median filter switches its output to high as soon as more than half of the samples in the filter memory are high. This memory requires a median filter of length L to possess L registers, which makes it consume a large area for high L. The longer L is, the better the noise rejection performance of the filter is, but L is also limited by the number of hits per weight H. Otherwise, the holdup due to delayed dynamic lowering of the thresholds would render the threshold crossing times unusable.

The delay introduced by a median filter can be estimated by calculating a moving average of length L over the comparator output switching probability (6.77). The difference between the point in time when the threshold crossing ideally occurs and the time when the expectation of the median filter is $\lceil L/2 \rceil$ is the average filter delay. It depends on the standard variation of the noise and the time constant. It must be subtracted from the measured threshold crossing times to eliminate the filter bias.

For very short filter lengths, the exponential function can be approximated with the linearization at the expansion point when the threshold is crossed. Then the delay can be approximated as half the filter length. This simple approach is used for the implementation of the on-chip BIST where little computation power is available. Only the first threshold crossing time difference is influenced by the filter delay, as the delay cancels in all subsequent time delay differences. Hence, the second complement binary counter which measures the time differences can be initialized to the negative filter delay prior to the measurement and hence the delay is compensated. For the subsequent delay measurements, the counter is reset to zero at each threshold crossing.

6.6.3.2. Delay Compensation for Sum Filter with ATE BIST

When the ADC is characterized in the production tests, a high number of hits per weight is used for increased accuracy. A median filter with a long memory is rather expensive; hence, an alternative would be beneficial. The idea is that a bias introduced by a simple filter can be compensated with the computation power available externally on the ATE. The alternative filter is termed a sum filter and it basically only counts the number of ones. The big difference

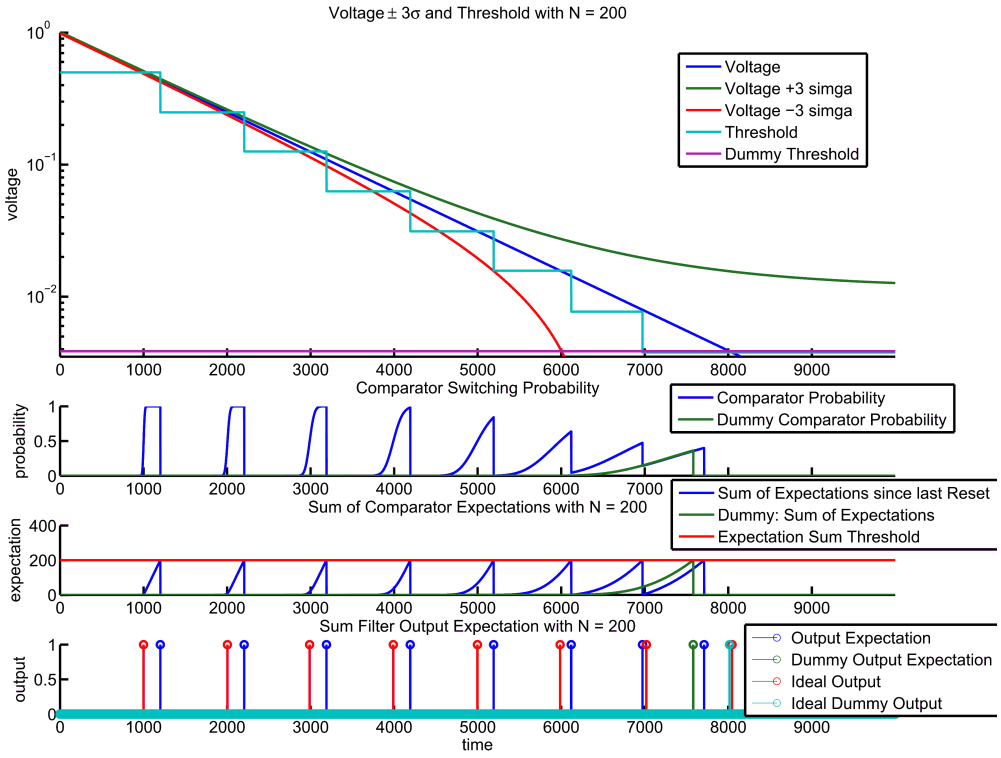

Figure 6.9.: Comparator switching expectation for a sum filter of length 200 in an 8-bit ADC.

is that it does not have a moving evaluation window, but rather infinite memory which is only limited by the maximum counter value. In practice, the filter memory is reset after each dynamical lowering of the threshold voltage.

Figure 6.9 shows the expected comparator switching for an 8-bit ADC during a BIST with $H = 1000$ hits per weight and a filter with length $L = 200$. For the most significant bits, the comparator switching probability jumps to one without a transition phase, which makes the filter delay the output switching for 200 discharging periods. Towards the LSB the delay introduced by the sum filter decreases dramatically and becomes negative, because close to the noise floor the comparator switching probability does not go back to zero anymore after the successive lowering of the threshold. Hence, the threshold is lowered prematurely. The delay times depend on the noise amplitude distribution, but they cannot be precomputed because the mismatched weights influence the time difference between successive thresholds. This directly influences the value at which the comparator's switching probability starts out when the threshold is lowered.

Therefore, the method to compensate for the filter delays must deal with arbitrarily mismatched weights. When the mismatch is guaranteed to be low, it is feasible to use precomputed delay compensation values. If the mismatch can be high an iterative compensation algorithm provides good results. In the first iteration the weights are estimated by the method described in section 6.5.1 based on the measured threshold crossing times. Then the algorithm shown in Fig. 6.9 is used to calculate the filter delays for the estimated set of weights. The original

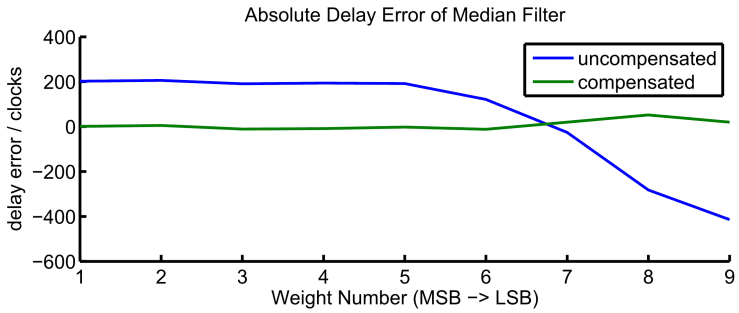

Figure 6.10.: Sum filter delays before and after delay compensation.

threshold crossing times are then compensated with the obtained delay values and the next iteration starts with estimating the weights based on the compensated threshold crossing times. After a few iterations this method leads to very good results. An example for the delay error in comparison with the ideal switching times is shown in Fig. 6.10 before and after the delay compensation for the same BIST as in Fig. 6.9. The threshold crossing times delay for the most significant bits can be eliminated completely, only towards the LSB and the dummy weight a little deviation remains. This remaining error has very little influence on the estimation error for the most significant weights.

6.6.4. Error Propagation for ATE BIST

The previous sections have explained the error sources, which influence the BIST calculation. Here the error propagation for errors introduced by the offset error and by time measurement errors is derived. The error propagation for a function $f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_L)$ can be approximated for small deviations by the total differential:

$$\Delta f = \sum_{k=1}^{L} \left(\frac{\partial f}{\partial x_i} \Delta x_i \right) , \tag{6.78}$$

where Δx_i is the small deviation of which sign and magnitude are known. After the offset voltage is included in (6.17), the weights are calculated as

$$w_i = \exp\left(-\frac{t_i}{\tau}\right) + \underbrace{\frac{V_{offset,absolute}}{V_{ref}}}_{V_{off}} \tag{6.79}$$

The error propagation shows that

$$\Delta w_i = \Delta V_{off} + \exp\left(-\frac{t_i}{\tau}\right) \cdot \left(\frac{\Delta\tau}{\tau^2} - \frac{\Delta t_i}{\tau}\right) \tag{6.80}$$

$$= \Delta V_{off} + (w_i - V_{off}) \cdot \left(\frac{\Delta\tau}{\tau^2} - \frac{\Delta t_i}{\tau}\right) . \tag{6.81}$$

As the time constant is usually $\tau \ll 1$, the error on the time constant $\Delta\tau$ has a large influence on the error of a weight. The offset error directly influences the weight. If the error on the weight is examined in relation to the absolute value, the relative error is

$$\frac{\Delta w_i}{w_i} = \frac{\Delta V_{off}}{w_i} + \frac{w_i - V_{off}}{w_i} \cdot \left(\frac{\Delta\tau}{\tau^2} - \frac{\Delta t_i}{\tau} \right) . \tag{6.82}$$

If the expansion point for the linearization of the offset error is chosen as $V_{off} \stackrel{!}{=} 0$, the result is

$$\frac{\Delta w_i}{w_i} = \frac{\Delta V_{off}}{w_i} + \frac{\Delta\tau}{\tau^2} - \frac{\Delta t_i}{\tau} . \tag{6.83}$$

Hence, the offset error has the highest influence on the lowest weights. The error on the threshold crossing time is known, but the time constant error τ usually not, because the time constant is determined by solving (6.18) which depends on all time constants and the offset error. This feedback makes the analysis cumbersome, but the following example shows what happens when a single threshold crossing time is shorter than it should be.

1. If a threshold crossing time is lower than it should be, (6.17) regards the corresponding weight as higher than it is: $\exists i : \Delta t_i < 0 \Rightarrow \Delta w_i > 0$.
2. Consequently, the normalization in (6.18) yields a $\Delta f > 1$.
3. Therefore, Newton's method results in a smaller time constant such that the sum of all weights is still one: $\Delta\tau < 0$.
4. The evaluation of (6.17) leads to all weights but w_i being smaller than they should be: $\Delta w_j < 0 \quad \forall j \neq i$.

6.7. Application and Simulation

This section shows concept simulation results for practical application scenarios of the BIST.

6.7.1. Front-End and Back-End Test

These tests are performed with a chip after production on the waver and in the package respectively. It is used with a high number of hits per weight for high measurement accuracy, which allows to characterize the ADC. The built-in offset compensation is performed before the BIST is started. The filtered threshold crossing times are sent to the ATE, which then performs the delay compensation and estimates the weights as described in section 6.5.1. Based on the weights, the worst-case INL and DNL values are calculated. In this section also the transfer curve is reconstructed for illustration purposes.

The following test is performed on a 10-bit ADC with an offset error of $\frac{V_{LSB}}{2}$ and additive white Gaussian noise (AWGN) with $\sigma = V_{LSB}$. This configuration is a realistic worst-case setup and

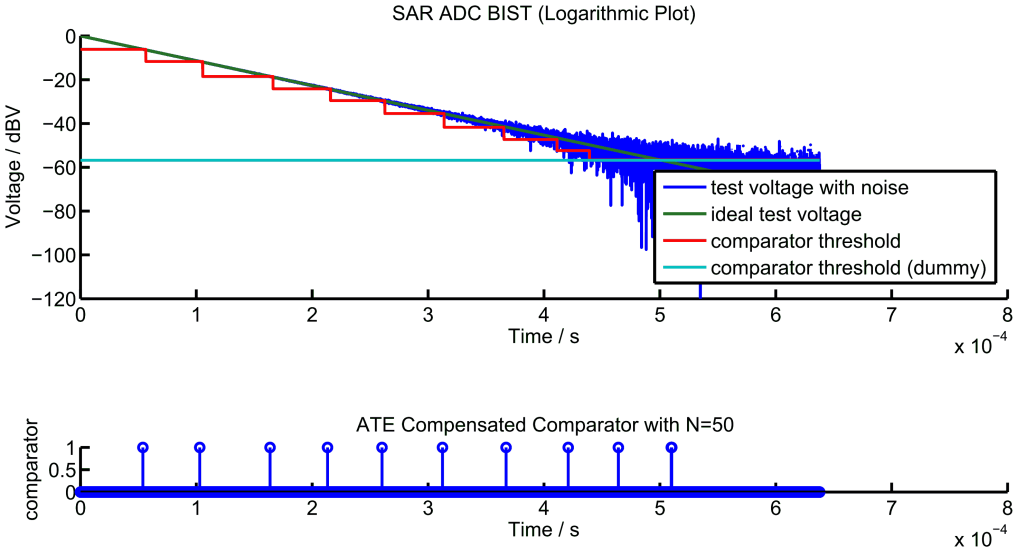

Figure 6.11.: Threshold crossing time measurement process for a 10-bit ADC.

hence shows that even in this case the weight estimation works well. The unit capacitances of the capacitor array are mismatched with $C \sim C_0 \cdot \mathcal{N}(1, 0.05^2)$, the same applies to the discharging capacitor. Such a high mismatch would only occur in bad devices which would definitely have to be discarded, because the expected mismatch due to process variations is lower. In this example the high mismatch is used to highlight that the method also works accurately in this case.

The BIST has $H = 1000$ hits per weight and a sum filter with length $L = 50$ is used. On the ATE the iterative sum filter delay compensation is applied to the measured threshold crossing times. The voltage on the internal capacitor array is depicted in Fig. 6.11 in a logarithmic plot. For the last three bits, the noise floor is higher than the voltage to be measured, but the comparator output filter in combination with the delay compensation permits to achieve good weight estimations. The actual weights w_i and the estimated weights $\hat{w}_i$ are shown in Table 6.3. The relative error increases towards the LSB, but the absolute error in relation to the LSB is smaller than one for all weights.

The deviations are mostly due to the large offset error which has more and more influence on the weights towards the LSB. With a smaller offset error, the estimation is much more accurate. Due to the delay compensation, the noise has little effect on the estimation accuracy.

The methods presented in section 6.5.1.2 and 6.5.1.3 are applied to calculate the worst-case INL and DNL errors, which are shown in Table 6.4 based on the actual weights and on the estimated weights. The estimation errors in every weight accumulate in the calculation of the INL and DNL errors, but the estimated ranges allow judging of whether the ADC is usable or not.

All the relevant information that is needed to discard or accept a device containing the ADC based on its nonlinearity has been determined. For better understanding, the method presented in section 6.5.1.1 is used here to reconstruct the gain- and offset compensated transfer curve. On the right side in Fig. 6.12 the ideal transfer curve of an ADC without

i	w_i	$\hat{w}_i$	error / %	error / LSB
1	253.4672	253.0513	−0.1641	−0.8317
2	133.4668	133.0897	−0.2826	−0.7543
3	60.3200	60.1382	−0.3014	−0.3636
4	31.6140	31.4228	−0.6049	−0.3824
5	16.8354	16.9870	0.9000	0.3030
6	8.4028	8.5789	2.0960	0.3522
7	3.9642	4.1877	5.6365	0.4469
8	1.9506	2.0751	6.3855	0.2491
9	0.9864	1.1775	19.3651	0.3820
10	0.5018	0.6479	29.1119	0.2922
11	0.4908	0.6441	31.2377	0.3066

Table 6.3.: Mismatched actual weights of a 10-bit ADC and weights estimated by the ATE BIST.

parameter	min / LSB	max / LSB
actual DNL	−10.150	16.783
estimated DNL	−11.507	14.749
actual INL	−13.395	13.414
estimated INL	−14.775	14.487

Table 6.4.: Actual and estimated nonlinearity error ranges.

mismatch, the actual transfer curve and the one based on the estimated weights is shown. On the left, the INL and DNL errors of every code are depicted. The symmetry patterns mentioned in section 6.5.1.2 and 6.5.1.3 can be clearly observed. Altogether, the estimated transfer curve and the INL and DNL error estimates match the actual curves sufficiently well.

6.7.2. Power-On Test

The power-on test is performed to assess the functionality of the ADC when it is enabled. This allows for a test with medium accuracy, but because of potentially limited computation power the matching estimation for successive weights is used. In this example a 10-bit ADC with slightly mismatched weights is tested. The number of hits per weights is configured as $H = 100$, the AWGN is distributed with $\sigma = V_{LSB}$ and as a countermeasure a median filter with length $L = 21$ is used. For the filter delay compensation, half of the filter length is subtracted from the first threshold crossing time.

Figure 6.13 shows the measurement procedure, where the noise influence towards the LSB increases, but the median filter prevents premature threshold lowering effectively. Table 6.5 shows all important parameters of the measurement. The starred values indicate that for the

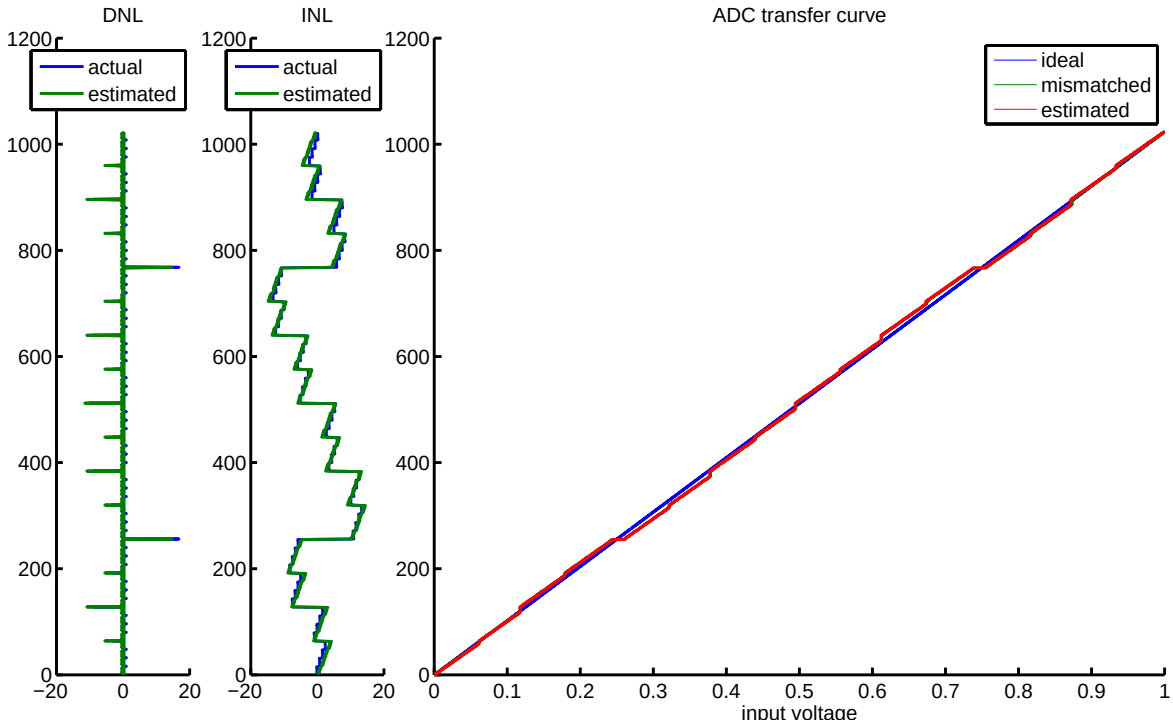

Figure 6.12.: Reconstructed transfer curve of a mismatched 10-bit ADC based on the estimated weights.

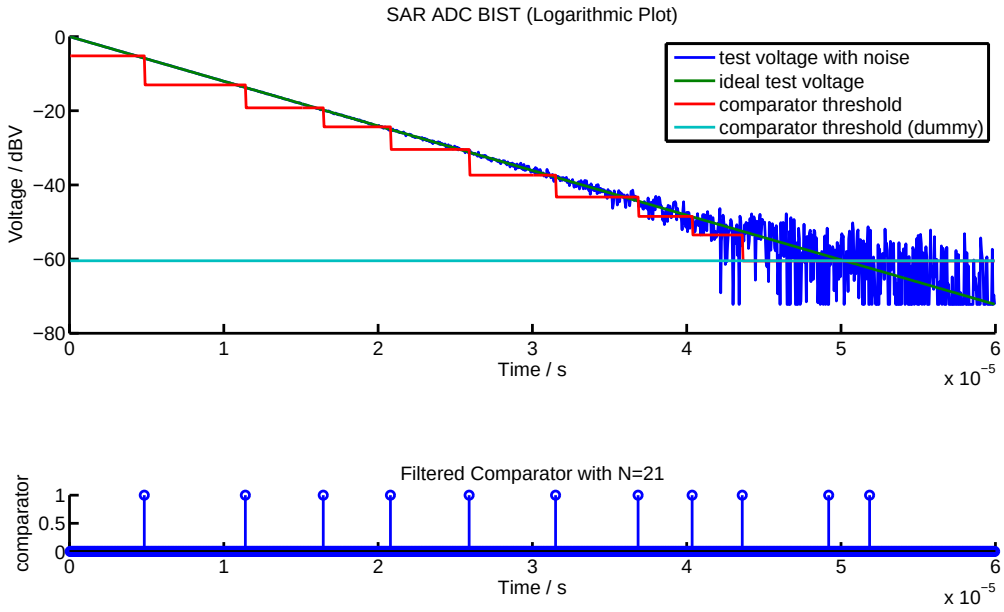

Figure 6.13.: Power-on test for mismatch estimation with $H = 100$ hits per weight.

dummy weight the indexed equations are not valid because it is dealt with in relation to the weight w_{N+1}.

i	Π_i	W_i	ΔW_i	$\frac{\Delta W_i}{\Pi_i} \cdot 100$	$\frac{w_i}{w_{i+1}}$	n_i	Δn_i	$\varepsilon_i \cdot 100$
1	512	562.6791	50.6791	9.8983	2.4680	86	86	52.3256
2	256	227.9935	−28.0065	−10.9400	2.0387	217	131	−22.9008
3	128	111.8344	−16.1656	−12.6294	1.8051	318	101	−13.8614
4	64	61.9562	−2.0438	−3.1935	2.0165	405	87	17.2414
5	32	30.7240	−1.2760	−3.9875	2.2194	507	102	9.8039
6	16	13.8436	−2.1564	−13.4773	1.9697	619	112	−4.4643
7	8	7.0282	−0.9718	−12.1471	1.8269	726	107	−34.5794
8	4	3.8470	−0.1530	−3.8257	1.7798	796	70	−7.1429
9	2	2.1615	0.1615	8.0732	2.2462	861	65	72.3077
10	1	0.9623	−0.0377	−3.7708	2.2279*	973	112	153.8462*
11	1	0.9702	−0.0298	−2.9804		1026	165	

Table 6.5.: Power-on BIST: Weight matching estimation with number of hits per weight $H = 100$.

The ideal weights in units of LSBs are shown as Π_i, and the actual mismatched weights as W_i. The absolute error of each weight in LSBs is shown as ΔW_i, and the relative error in relation to the ideal weight too. The counter values corresponding to measured threshold crossing times are shown as n_i and the successive differences as Δn_i. The value ε_i is the relative deviation of the successive time differences and, according to (6.60), it serves as an estimator for the ratio w_i/w_{i+1}. For reference, this ratio w_i/w_{i+1} is shown in a column, but in practice this value is not available due to lacking knowledge about the weights.

It can be seen that ε_i correlates well with the ratio w_i/w_{i+1}, which it effectively predicts. By applying (6.67), which predicts the error of a weight under the assumption that the next weight is ideal, the correlation between the actual ΔW_i and the value ε_i is also good. Due to the worsening noise influence, the quality of all measured values deteriorates towards the LSB weight.

It can also be observed that the amplitude of ε_i in relation to the error ΔW_i of a weight increases towards the LSB, which is the reason why in (6.63) it is suggested to scale the limit for ε_i exponentially towards the LSB to get constant absolute errors. When this algorithm is performed on the chip, only ε_i has to be calculated and compared to the allowed limit.

6.7.3. Cyclic Test

The cyclic BIST also uses the algorithm for mismatch estimation for successive weights, but it is performed with a very low number of hits per weight. This makes the whole test very fast and hence it can be run in between measurements without blocking the ADC for too long. Such a cyclic check might be necessary to conform to functional safety. The accuracy of the test is very limited, but it checks whether the capacitor array and the bottom plate switches as well as the comparator are basically functional.

i	Π_i	W_i	ΔW_i	$\frac{\Delta W_i}{\Pi_i} \cdot 100$	$\frac{w_i}{w_{i+1}}$	n_i	Δn_i	$\varepsilon_i \cdot 100$
1	512	509.9084	−2.0916	−0.4085	1.9316	10	10	−10
2	256	263.9793	7.9793	3.1169	2.0944	19	9	22.2222
3	128	126.0401	−1.9599	−1.5311	2.0187	30	11	−9.0909
4	64	62.4377	−1.5623	−2.4410	2.0093	40	10	0
5	32	31.0742	−0.9258	−2.8932	1.9840	50	10	0
6	16	15.6622	−0.3378	−2.1112	2.1536	60	10	−10
7	8	7.2727	−0.7273	−9.0911	2.0228	69	9	11.1111
8	4	3.5953	−0.4047	−10.1173	1.7561	79	10	0
9	2	2.0473	0.0473	2.3662	2.0855	89	10	0
10	1	0.9817	−0.0183	−1.8318	2.0452*	99	10	10*
11	1	1.0010	0.0010	0.1046		100	11	

Table 6.6.: Cyclic BIST: Weight matching estimation with number of hits per weight $H = 10$.

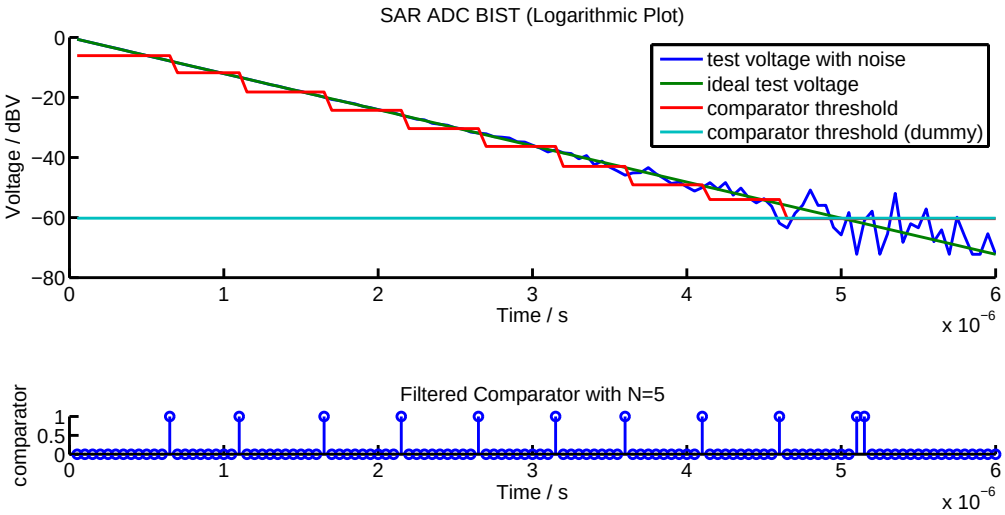

Figure 6.14.: Power-on test for mismatch estimation with $H = 100$ hits per weight.

The same simulation setup as in the previous section is used, only the number of hits per weight is decreased to $H = 10$. Consequently, the median filter length is also reduced to $L = 5$. Figure 6.14 shows the very fast and rough measurement procedure. In Table 6.6 the same parameters as explained in the previous section are displayed. The time differences show that all components of the ADC are basically functional, because the relative difference value ε_i is small.

7. Random Sampling Concept

The need for non-uniform sampling arises if either the Nyquist limit should be overcome or if a band limitation is not possible. For the voltage monitoring system the latter is the case because active anti-aliasing filters require too much area when high voltage rails have to be monitored.

The theory behind random sampling is introduced in section 3.1.2, but all described methods assume that the number of recorded samples is sufficiently large. However, the voltage monitoring system must react within a defined time frame and hence the number of signal samples available for a decision is limited. This chapter describes the considerations about a random scheduler to be used in the voltage monitoring system.

7.1. Deriving a Multi-Channel Random Sampling Scheme for Real-Time Applications

This section discusses the feasibility of the random sampling methods introduced in section 3.1.2 in the context of the multiplexed multi-channel voltage monitoring system. It has been established that the use of anti-aliasing filters in the voltage monitoring system would require too much area. The consequence of performing uniform sampling would be that disturbances could cause wrong voltage level decisions due to aliasing effects. To prevent them, some kind of random sampling scheme must be implemented, but the system must guarantee that the specifications for the minimum and maximum deglitch times are met.

In [5] an approach for estimating the DC voltage level of a signal is presented. It uses additive random sampling (ARS) (see section 3.1.2.1) in combination with a moving average filter to suppress the noise, which is modeled as a sinusoidal wave. However, in the voltage monitoring system only a 1-bit threshold crossing signal is available, which makes the system inherently nonlinear and such estimations inaccurate. Furthermore, the voltage level estimation exhibits different behavior than the state of the art comparators which should be replaced by the new voltage monitoring system.

The random skip sampling (RSS) method (see section 3.1.2.3) can be used for random sampling of a single channel. Based on a uniform time grid with a granularity of Δt, it randomly skips samples and hence converts a signal at a lower average sampling rate while keeping the Nyquist frequency at $\frac{1}{2\Delta t}$. The quality of aliasing suppression depends on the number of samples taken, but an aliasing noise floor always remains.

The following subsections show the step-by-step derivation of a sophisticated multichannel random sampling scheme for the voltage monitoring system. When multiple channels are to be sampled, the single-channel random sampling methods must be adapted. The key

consideration is that as little time as possible should be wasted, such that as many samples as possible can be taken in a given time frame.

7.1.1. Uniform Round-Robin Sampling

Figure 7.1 shows a uniform round-robin sampling scheme, where each channel is sampled after the other in a periodic manner. If an ADC sampling and conversion takes the time Δt, then the sampling frequency f_s^* for each of the M channels is

$$f_s^* = \frac{1}{M \cdot \Delta t}. \tag{7.1}$$

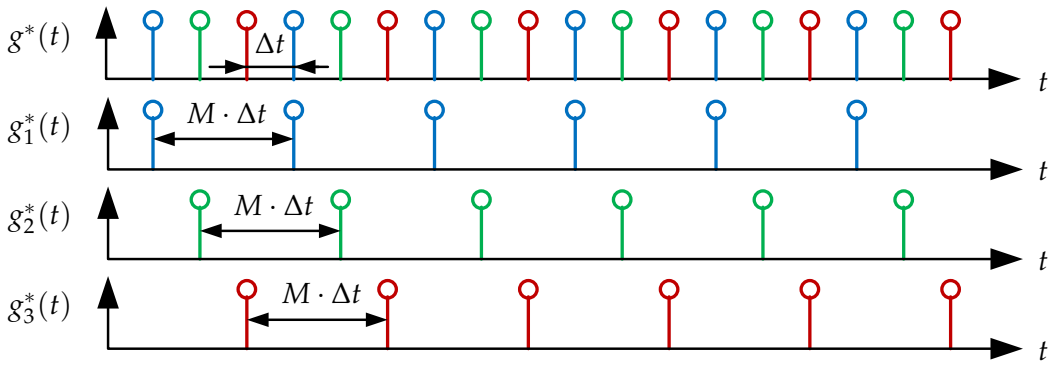

Figure 7.1.: Uniform round-robin sampling scheme for $M = 3$ channels.

7.1.2. Permuting the Channel Sampling Sequence

Based on the round-robin scheme, a random sampling scheme can be obtained by permuting the sampling sequence [3]. An example for such a permutation is depicted in Fig. 7.2 for 3 channels.

By inspecting the resulting sampling sequences on the individual channels it is confirmed that they are non-uniform. In fact, the random permutation for M channels is equivalent to random skip sampling (RSS) for each individual single channel where the skip probability $q = \frac{M-1}{M}$. Therefore, at each sampling time instant of the ADC it is not determined whether to skip the sample or not, but which channel should be sampled. Obviously, the sampling sequences of the individual channels are correlated because by the exclusion principle it is impossible for multiple channels to be sampled at the same time, but as the channel sampling sequences are processed individually, this does not cause problems.

The equivalence with RSS justifies that the effective sampling frequency for each of M channels in the random permutation scheme is

$$f_s = \frac{1}{\Delta t}, \tag{7.2}$$

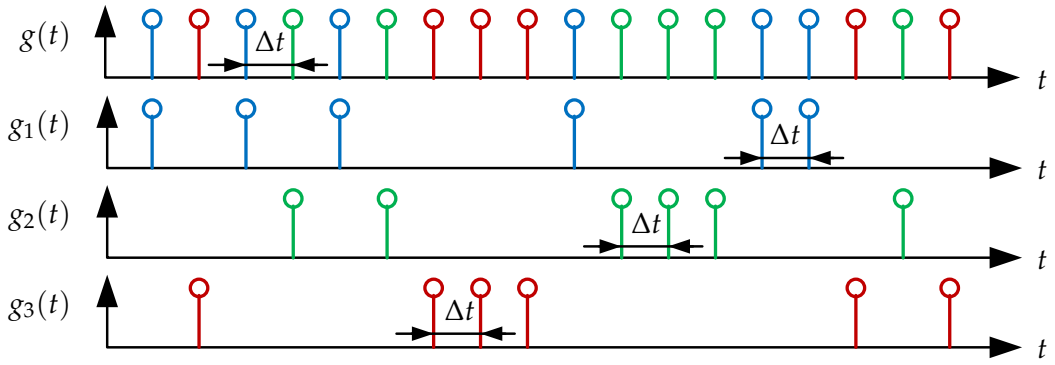

Figure 7.2.: Non-uniform randomly permuted sampling scheme for $M = 3$ channels.

which is M times higher than for round-robin sampling. This means that the achievable effective sampling frequency is independent of the number of channels, but this is only true for sufficiently high numbers of samples. In the real-time voltage monitoring system, the achievable average number of samples that can be obtained within the deglitch time limits the number of channels. The maximum number of channels depends on the targeted resistance against aliasing and on the combined sampling and conversion time Δt.

Performing the random channel selection based on a real random source means that a channel might not be sampled for a long time. The probability for a channel not to be sampled over n periods is q^n. The limit goes exponentially towards zero, but even though it is rare, it cannot be guaranteed that a channel is sampled within a limited number of periods. Therefore, due to the timing constraints the voltage monitoring system cannot use a real random source.

A pseudo-random noise generator (PRNG) generates a deterministic sequence which approximates a random sequence. If the channel selection is based on such a generator, the deterministic nature empowers to guarantee timings and anti-aliasing properties not only on average, but also for the worst case. Because of the deterministic nature also a lookup table (LUT) with the permuted sampling sequence could be used to achieve the same effect, but its size would be much larger than if the sequence was produced by a pseudo-random noise generator (PRNG).

7.1.3. Increasing Time Granularity

The sampling and subsequent conversion process of an ADC usually takes more than one clock cycle, such that the total time is an integer multiple of the clock period:

$$\Delta t = k \cdot t_{clk}, \quad k \in \mathbb{N}\,. \tag{7.3}$$

It seems worthwhile to decouple the sampling time instants from the time grid based on Δt and make use of the finer clock time granularity t_{clk}. For that purpose, additional delays, which are multiples of full clock periods, must be randomly inserted in between sampling instants of the permuted random sampling scheme. As a result, the sampling instants are

placed on a time grid with smaller time granularity, which increases the effective sampling frequency to

$$f_s = \frac{1}{t_{clk}}. \tag{7.4}$$

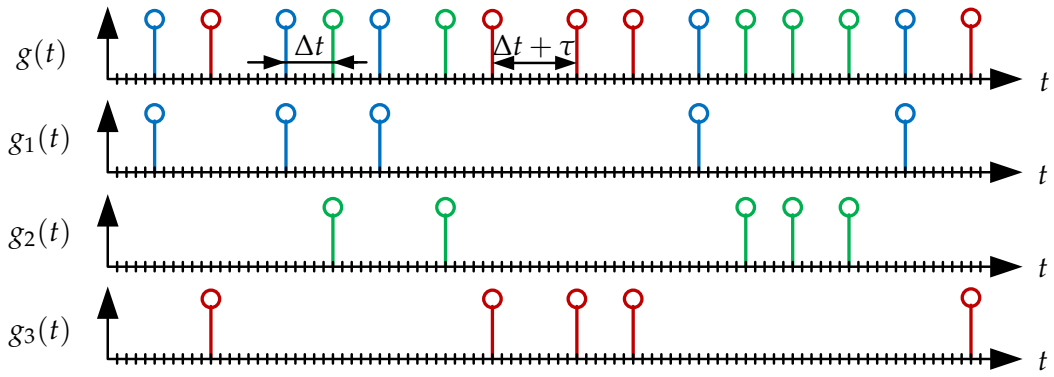

Figure 7.3.: Randomly permuted sampling scheme with clock period time granularity by random delays $\tau \sim t_{clk} \cdot \mathcal{U}\{0, M-1\}$ for $M = 3$ channels.

The important property is that the greatest common divisor (GCD) of the sampling sequence divided by t_{clk} must be one. Therefore, it is actually not required that the minimum distance between sampling instants is t_{clk}, which is also impossible because the sampling and conversion process takes longer than that. The GCD of one can also be achieved with other multiples k_i of the clock period, as long as some of them are coprime. An example for such a sampling scheme is shown in Fig. 7.3.

The delay can be produced by a counter which is seeded by a second PRNG, which is updated after every sampling instant. However, the pseudo-random channel selection PRNG in section 7.2 can only handle a total number of channels which is a power of two. Therefore, the remaining channels up to the next power of two can serve as dummy channels. These do not perform sampling and conversion, but only cause a delay of one clock cycle. Therefore, the increase in time granularity can be achieved without extra cost, but by an inherent property of the used channel selection mechanism. Furthermore, in the voltage monitoring system the combined sampling- and conversion time is not constant for all channels. This is due to different channel types with varying sampling times and the varying number of output signal attached to the channels. Therefore, for a specific channel and output configuration of the voltage monitoring system a special analysis must be carried out to verify that the static configuration and the measures to increase the effective sampling frequency do not interact with negative effects. The analysis process is described in section 7.4.

7.1.4. Introducing Sampling Clock Jitter

Figure 7.4 is similar to Fig. 7.3, but instead of definitive sampling points a constant probability density over one clock period is shown for each sampling instant. The displayed function $g(t)$

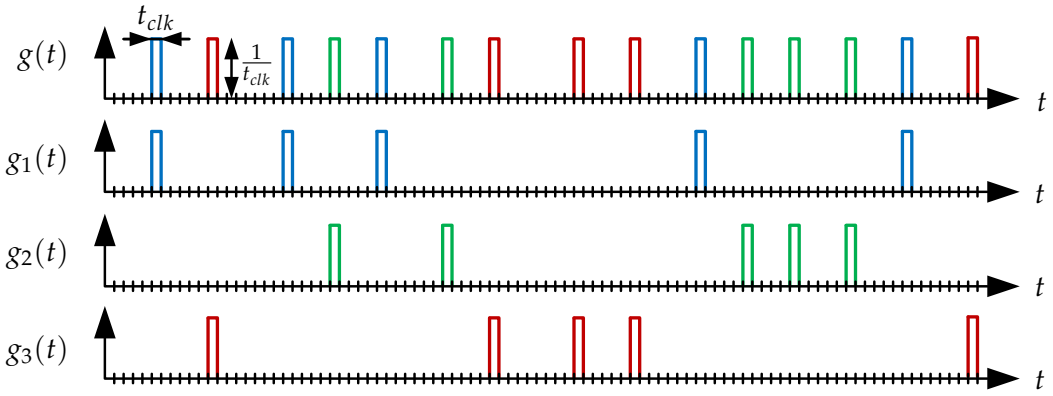

Figure 7.4.: Randomly permuted sampling scheme with clock period time granularity and clock jitter for $M = 3$ channels.

is the sum of all probability densities:

$$g(t) = \sum_{j=0}^{\infty} \mathcal{U}(t_j, t_j + t_{clk}) \,, \tag{7.5}$$

where the times t_i are the sampling time instants without jitter and the uniform distribution is the jitter.

The random channel permutation together with random delays at clock granularity ideally lead to a uniform distribution of the sampling point probability on the discrete grid with spacing t_{clk}. Adding the jitter ideally results in a uniform sampling probability in the continuous time domain. This effectively eliminates classic aliasing if the number of samples is large enough. This is verified in section 7.4.

7.2. Pseudo-Random Number Generator

The multi-channel random sampling scheme introduced earlier requires a deterministic method to perform a pseudo-random selection of channels. The periodicity of the pseudo-random sequence must be sufficiently long and the sequence must possess adequate statistical properties. Such a pseudo-random sequence could be generated by a state-machine with a lookup table (LUT), but the required silicon area makes it infeasible. Hence, a more efficient method to generate a pseudo-random sequence is needed.

7.2.1. Linear Feedback Shift Registers

This section is largely based on [13]. A linear feedback shift register (LFSR) can generate arbitrarily long pseudo-random sequences with little hardware effort. Figure 7.5 shows the basic structure of an LFSR with length L. It contains L 1-bit storage elements which constitute a shift register. At a state update the content of each register is shifted to the right. The new input s_j of the leftmost register is created by an exclusive-or combination of selected register

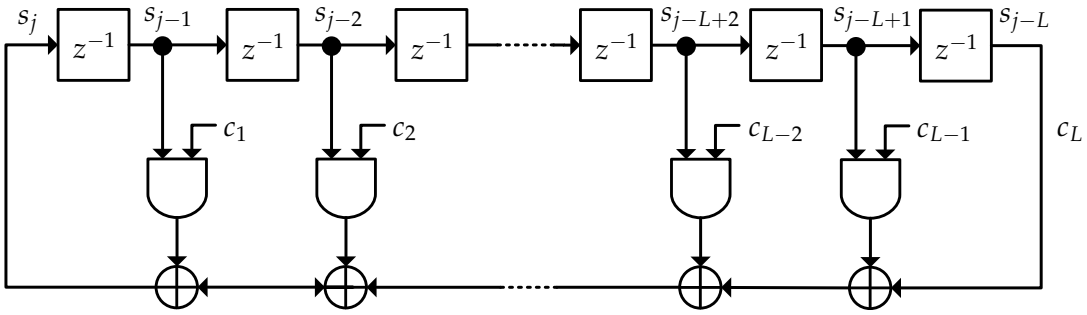

Figure 7.5.: Linear feedback shift register structure [13].

values. The output sequence is composed of the rightmost register's value s_{j-L}. The feedback path is configured by a certain assignment of $[c_1, c_2, \ldots, c_L] \in \{0,1\}^L$, where $c_i = 1$ means an active tap and $c_i = 0$ a deactivated tap. Together with the initial state $[s_{L-1}, s_{L-2}, \ldots, s_0]$ the output sequence is uniquely defined by the following recursion:

$$s_j = (c_1 s_{j-1} + c_2 s_{j-2} + \ldots + c_L s_{j-L}) \mod 2 \tag{7.6}$$

$$= \sum_{i=1}^{L} c_i s_{j-i} \mod 2\,. \tag{7.7}$$

7.2.2. Polynomial Representation

For the theoretical analysis it is practical to use a different mathematical representation by means of polynomials. Let the coefficients of a formal power series be elements of the finite field $\mathbb{F}_2$. This simplifies the equations, because in $\mathbb{F}_2$ the exclusive-or operation is the same as an addition. The connection polynomial is an equivalent representation of the feedback path:

$$C(D) = 1 + c_1 D + \ldots + c_L D^L\,. \tag{7.8}$$

Similarly, the output sequence is defined as

$$S(D) = s_0 + s_1 D + s_2 D^2 + \ldots\,, \tag{7.9}$$

and the initial state as

$$P(D) = p_0 + p_1 D + \ldots + p_{L-1} D^{L-1}\,. \tag{7.10}$$

Then the LFSR recursion can be described by the polynomial multiplication

$$C(D) \cdot S(D) = P(D)\,. \tag{7.11}$$

Due to the finite number of registers, the output of an LFSR is always periodic. An internal state of all zeros must be avoided because it will produce a zero sequence forever. For the LFSR to be nonsingular the degree of the connection polynomial must be $\deg(C(D)) = L$. If the LFSR is singular and the degree is $\deg(C(D)) = N < L$, the LFSR is actually a nonsingular LFSR of length N with an additional delay of $L - N$ at the output. Then the initial values $s_0, s_1, \ldots s_{L-N-1}$ are independent and precede the periodic output sequence.

7.2.3. Maximum Length Sequences

Depending on the choice of a feedback polynomial, the period and properties of the output sequence differ. For the generation of pseudo-random numbers with maximum hardware efficiency, the maximum period length is needed. A primitive polynomial $A(D)$ with $\deg(A(D)) \geq 1$ is irreducible if it cannot be split into a product of two polynomials. If the polynomial $A(D)$ with degree L is furthermore a generator of the group $\mathbb{F}_{2^L}$ it is called primitive. For a non-zero initial state, the output generated by a non-singular LFSR of length L with a primitive connection polynomial is called a maximum length sequence with a period of: [13]

$$m = 2^L - 1\,. \tag{7.12}$$

For a given degree there can be multiple primitive polynomials. Table 7.1 lists an incomplete selection of primitive polynomials up to degree 10.

degree	primitive polynomial	period
2	$1 + x + x^2$	3
3	$1 + x + x^3$	7
4	$1 + x + x^4$	15
5	$1 + x^2 + x^5$	31
6	$1 + x + x^6$	63
7	$1 + x + x^7$	127
8	$1 + x + x^5 + x^6 + x^8$	255
9	$1 + x^4 + x^9$	511
10	$1 + x^3 + x^{10}$	1023

Table 7.1.: List of primitive polynomials over $\mathbb{F}_2$ up to degree 10. [13]

7.2.3.1. Statistical Properties

Let s be a sequence, which was produced by a maximum length LFSR. The period of the sequence is then $m = 2^L - 1$ and it has the following statistical properties. One period of s contains exactly 2^{L-1} ones and $2^{L-1} - 1$ zeros, which gives a sequence period of $2^{L-1} + 2^{L-1} - 1 = 2^L - 1$. There is one less zero because the register state with all zeros must not occur due to its non-recoverable property.

For any subsequence $\bar{s}_k$ of length $2^L + k - 2$ the following holds. Every possible non-zero sequence of length k will show up 2^{L-k} times as a subsequence of $\bar{s}$, but the all-zero sequence of length k appears $2^{L-k} - 1$ times, which is one time less [13]. In summary this means that the pattern distribution in a maximum length sequence is almost uniform. Due to the linearity of the system, the output sequences are correlated, but for the voltage monitoring system this is not problematic.

$i = j-4$	s_{j-1}	s_{j-2}	s_{j-3}	s_{j-4}	s_{j-5}
1	0	1	1	1	1
2	0	0	1	1	1
3	1	0	0	1	1
4	1	1	0	0	1
5	0	1	1	0	0
6	1	0	1	1	0
7	0	1	0	1	1
8	0	0	1	0	1
9	1	0	0	1	0
10	0	1	0	0	1
11	0	0	1	0	0
12	0	0	0	1	0
13	0	0	0	0	1
14	1	0	0	0	0
15	0	1	0	0	0
16	1	0	1	0	0

$i = j-4$	s_{j-1}	s_{j-2}	s_{j-3}	s_{j-4}	s_{j-5}
17	0	1	0	1	0
18	1	0	1	0	1
19	1	1	0	1	0
20	1	1	1	0	1
21	0	1	1	1	0
22	1	0	1	1	1
23	1	1	0	1	1
24	0	1	1	0	1
25	0	0	1	1	0
26	0	0	0	1	1
27	1	0	0	0	1
28	1	1	0	0	0
29	1	1	1	0	0
30	1	1	1	1	0
31	1	1	1	1	1

Table 7.2.: Internal states over one period of a maximum length LFSR with 5 registers and primitive connection polynomial $C(D) = 1 + x^2 + x^5$.

7.2.3.2. Example Sequence

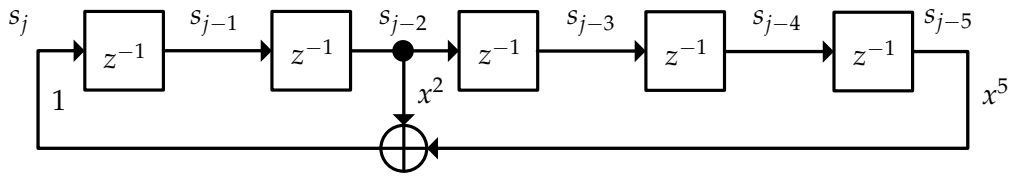

Figure 7.6.: Maximum length linear feedback shift register of length $L = 5$.

As an illustration a maximum length LFSR with the primitive connection polynomial $1 + x^2 + x^5$ is shown in Fig. 7.6. Table 7.2 shows the internal state of the registers over one full period of length $m = 2^5 - 1 = 31$ starting from an arbitrary non-zero initial state. The shift register behavior can be observed in the diagonal pattern of the states.

7.2.4. State Output Filter

The LFSR has the task of deciding which channel should be sampled. The output stream consisting of the shift register's least significant bit at each state update can be used to derive

the channel selection. If the system has more than two channels, more than one random bit is required. A multi-bit random number can be derived by updating the LFSR multiple times and storing the output bits. Unfortunately, when the GCD of the number of derived bits and the LFSR period is not one, the period of the multi-bit random number sequence is shorter than the LFSR period.

By tapping more than one of the L LFSR's registers, a new M-bit number can be obtained directly by only one state update operation. Hence, a new random number with any number of bits $M \leq L$ can be derived without reducing the effective period of the sequence. Due to the nature of the shift register, in each new random number only one bit value is actually new, while the others are only shifted. As it will be shown later, the resulting autocorrelation of the channel selection sequence does not have a negative influence on the anti-aliasing properties.

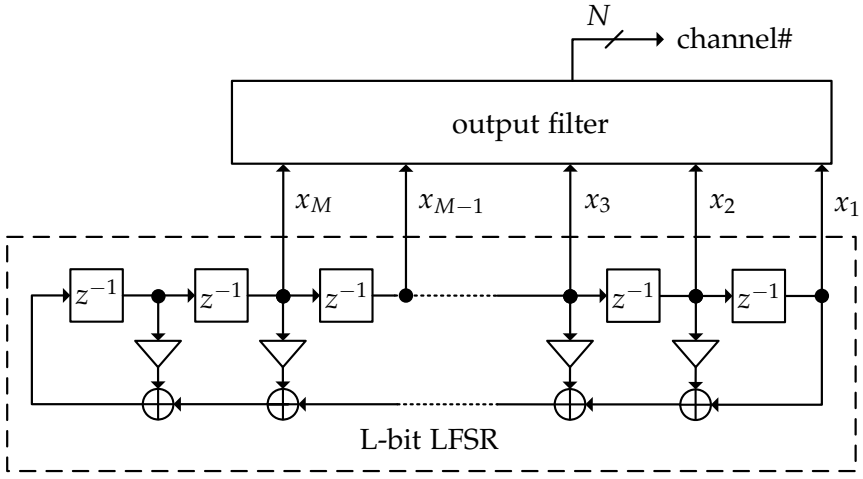

Figure 7.7.: Architecture of the random scheduler based on a Fibonacci linear feedback shift register with output filter.

The principle of tapping multiple LFSR registers is highlighted in Fig. 7.7. The extracted M-bit random number can be used directly or fed into an optional filter to derive an N-bit channel selection signal. The filter is an M-to-N-bit lookup table (LUT) which serves multiple purposes.

Firstly, a total number of channels Γ must be represented as a binary number with $N = \lceil \log_2 \Gamma \rceil$ bits. Hence, if Γ is not exactly a power of two, there are states which do not correspond to a channel. They can be mapped to a dummy channel which serves as an additional random delay of t_{clk} to assure clock period granularity (see section 7.1.3). Secondly, the channel numbers are defined by the ADC multiplexer. Hence, if the channel numbers are not assigned continuously, a mapping from random number to channel number is inevitable. Thirdly, the mapping allows adjusting the relative frequency at which channels are selected. The average sampling frequency of channels can be increased by having more random numbers point to the same channel number. An example for this is shown in Fig. 7.8. Finally, as will be elaborated in section 7.3.5, the permutation of channel mappings changes the properties of the pseudo-random sampling sequence. Such adjustments might be necessary to fulfill some timing requirements.

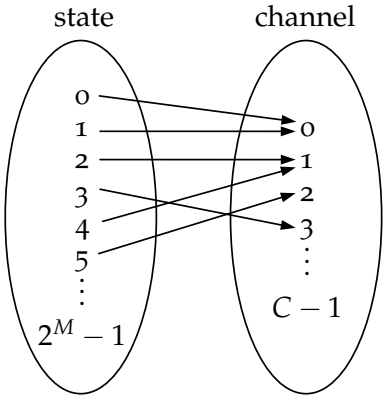

Figure 7.8.: Mapping from LFSR state to channel number.

7.3. Random Scheduler Synthesis

The structure of a random scheduler has been established. The configuration of all components depends on the concrete specifications for the channels and outputs of the voltage monitoring system. This section explains the synthesis based on generic specifications. There are a number of Γ channels $\boldsymbol{c} = [c_1, c_2, \ldots, c_\Gamma]$ and one additional dummy channel for the clock period time granularity. Attached to channel c_i are Ω_i output signals $\boldsymbol{o}_i = [o_{i,1}, o_{i,2}, \ldots, o_{i,\Omega_i}]$. Every output $o_{i,j}$ has rising and falling deglitch times $t_{r,i,j}$ and $t_{f,i,j}$ respectively.

7.3.1. Channel Priorities

If all channels were sampled with the same average sampling frequency, then outputs with shorter deglitch times would have to base decisions on less signal samples than outputs with longer deglitch times. Especially outputs with very short deglitch times would therefore suffer from very low disturbance immunity. Hence, the average sampling frequency of every channel must be adjusted to guarantee a minimum immunity.

For that purpose, each channel is assigned a priority $p_i \in \mathbb{N}$. It defines how many random number output states of the LFSR are mapped to the channel. For instance, channel 0 in Fig. 7.8 has a priority of $p_0 = 2$ because two states map to the channel. The ratio of the priorities signifies the relative sampling frequency of channels. The corresponding absolute sampling frequencies depend on the total number of channels and their priorities as well as on the speed of the ADC.

To avoid unnecessary complication of the scheduler, all outputs attached to a channel are updated each time the channel is sampled. Therefore, for a channel with multiple outputs and different deglitch times, the priority represents a compromise. To guarantee a minimum disturbance immunity, the lowest deglitch time of all outputs on a channel determines the priority. This means that outputs on the same channel with long deglitch times will have much more samples available for decisions, but still the minimum number of samples is guaranteed for the output with the lowest deglitch time.

Let the minimum deglitch time of an output $o_{i,j}$ on a channel c_i be defined as

$$t_{min,i,j} = \min(t_{r,i,j}, t_{f,i,j}) \,, \tag{7.13}$$

and the minimum deglitch time of the channel c_i as

$$t_{min,i} = \min(t_{min,i,1}, t_{min,i,2}, \ldots, t_{min,i,\Omega_i}) \,. \tag{7.14}$$

For the next calculations a resolution parameter r is required. A small r results in the smallest lookup table size, but the adjustment of relative sampling frequencies is coarse. A larger r means that the random number extracted from the LFSR has more bits, and that therefore more states are mapped to channel numbers. This allows finer trimming of timing parameters in the next sections. To save silicon area for the LUT, the parameter r should be as small as possible to still fulfill timing requirements. The inverse of a deglitch time is a frequency, and hence the priority of a channel is defined as the frequency normalized with the minimum frequency of all channels:

$$p_i = \left\lceil r \cdot \frac{t_{min,i}^{-1}}{\min_j \left(t_{min,j}^{-1} \right)} \right\rceil \,. \tag{7.15}$$

If one output has a much lower deglitch time than any other output, the resulting high priority of the channel means that the ADC will spend most of its time on sampling that channel. This would result in a very poor compromise on the number of samples and hence it is not recommended to use too short deglitch times.

The sum of all priorities $\sum_{i=1}^{\Gamma} p_i$ is equivalent to the required number of random numbers which are later mapped to the Γ channels. The total of different random numbers in binary representation is always a power of two. Therefore, a dummy channel with channel number $\Gamma + 1$ is introduced and its priority is the remainder to the next power of two. An additional purpose of the dummy channel is to assure a clock period time granularity of the random sampling process. It must be assured that at least one random state maps to the dummy channel, hence one is added in the following calculation:

$$p_{\Gamma+1} = 2^{\left\lceil \log_2\left(1+\sum_{i=1}^{\Gamma} p_i\right)\right\rceil} - \sum_{i=1}^{\Gamma} p_i \,. \tag{7.16}$$

7.3.2. Cycle Time and Average Sampling Rate

The average cycle time defines how long it takes to sample each channel i and convert all attached outputs p_i times. The time it takes to sample a channel i and convert all of its outputs is

$$t_i = t_{sample,i} + t_{conv} \cdot \Omega_i \,, \tag{7.17}$$

where $t_{sample,i}$ and $t_{conv,i}$ are already multiples of the clock period as defined in section 5.2.6. The sampling time depends on the specific channel type.

The average cycle time is then the sum of all channel times weighted with their priorities. The dummy channel is additionally accounted for.

$$t_{cycle} = \sum_{i=1}^{\Gamma} t_i \cdot p_i + t_{clk} \cdot p_{\Gamma+1} \tag{7.18}$$

Based on the priorities and the average cycle time, the average sampling rate f_i of a channel i can be calculated:

$$f_i = \frac{p_i}{t_{cycle}} . \tag{7.19}$$

7.3.3. Deglitch Filter Samples

Based on the knowledge of the average sampling frequencies of each channel, the deglitch filter lengths can be calculated. The length of the rising deglitch filter for output j on channel i is given as

$$n_{r,i,j} = \text{round}\left(t_{r,i,j} \cdot f_i\right) . \tag{7.20}$$

The falling deglitch time is obtained similarly.

7.3.4. Minimum Period Length of LFSR

A too short periodicity of the random number sequence decreases the anti-aliasing quality. The minimum periodicity must ensure that for the shortest deglitch times there is no repetition in the filter length. If better anti-aliasing quality is aspired for outputs with longer deglitch filters, then their sequences should also not contain repetitions. This is the way the periodicity of the LFSR is determined in the voltage monitoring system, because increasing the periodicity is cheap – adding one register to the LFSR doubles its periodicity (assuming a primitive connection polynomial).

The sum of all priorities including the dummy channel determines the bit width $M = \log_2\left(\sum_{i=1}^{\Gamma+1} p_i\right)$ of the random numbers. Hence, this is the minimum length L of an LFSR, which then produces a maximum length sequence with period $m = 2^M - 1$. Depending on the deglitch filter lengths, L must be longer to guarantee no repetitions also for a long deglitch filter.

$$\#\text{states} = \sum_{i=1}^{\Gamma+1} p_i \tag{7.21}$$

$$\tag{7.22}$$

Let the maximum deglitch filter length of an output $o_{i,j}$ on a channel c_i be defined as

$$n_{max,i,j} = \max(n_{r,i,j}, n_{f,i,j}) \,, \tag{7.23}$$

and subsequently the maximum deglitch filter length of the channel c_i as

$$n_{max,i} = \max(t_{max,i,1}, t_{max,i,2}, \ldots, t_{max,i,\Omega_i}) \,. \tag{7.24}$$

A channel i is associated with p_i states. Therefore, a channel's sampling sequence with the length of the deglitch filter could be made up of values stemming from different random number generator output states. The normalized value $\tilde{n}_{max,i}$ signifies how often each state occurs within one filter length on average:

$$\tilde{n}_{max,i} = \left\lceil \frac{n_{max,i}}{p_i} \right\rceil \,. \tag{7.25}$$

The maximum of these normalized filter lengths is the minimum period length for a single random number. The minimum sequence length of all random numbers is then the number of random output states multiplied with the maximum normalized filter length:

$$\text{sequence length} = \#\text{states} \cdot \max_i \left(\tilde{n}_{max,i}\right) \,. \tag{7.26}$$

Given the required sequence length, the bit width of the corresponding LFSR is

$$L = \lceil \text{sequence length} + 1 \rceil \,, \tag{7.27}$$

where the offset of one takes into account that the sequence length of a L-bit LFSR is only $m = 2^L - 1$.

7.3.5. Deglitch Time Variation

This section deals with the variation of deglitch times and measures to influence the variation.

7.3.5.1. Analysis of Deglitch Time Variation

The constant deglitch filter lengths are calculated based on the deglitch times and the average sampling frequencies. Due to the pseudo-random sampling sequence, there is an inherent periodic variation of the deglitch time over the LFSR period. The reason is sketched in Fig. 7.9 for an example sampling sequence where the deglitch filter length is $N = 3$. The varying pseudo-random time differences between the sampling points of a channel lead to varying deglitch times. Additionally, there is the clock jitter and the uncertainty of when an actual voltage threshold crossing happens.

Referring to Fig. 7.9, the minimum deglitch time $t_{d_{avg},i}$ can be calculated as follows. The first sample above the threshold voltage is taken at the latest jitter time $t_{i+1} + t_{clk}$ and this is

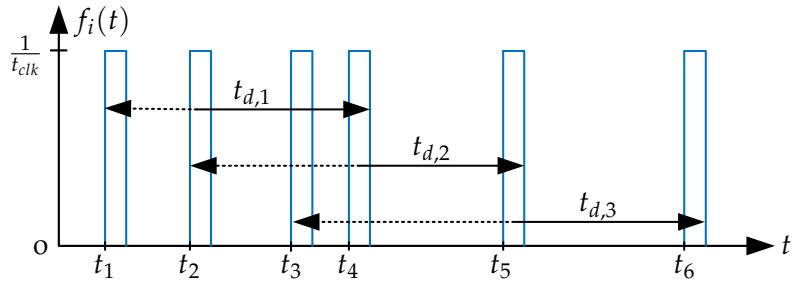

Figure 7.9.: Example for the inherent deglitch time variations with a deglitch filter length of $N = 3$.

also the first time when the signal was above the threshold in the continuous time domain. Provided that all subsequent samples are also above the voltage threshold and that the last sample is taken at the earliest jitter time t_{i+n}, the minimum deglitch time is obtained. Similar considerations can be made for the maximum possible deglitch time $t_{d_{max},i}$. The mean between those times is the average deglitch time $t_{d_{min},i}$.

$$t_{d_{min},i} = t_{i+n} - t_{i+1} - t_{clk} \tag{7.28}$$

$$t_{d_{max},i} = t_{i+n} - t_i + t_{clk} \tag{7.29}$$

$$\Rightarrow t_{d_{avg},i} = \frac{t_{d_{min},i} + t_{d_{max},i}}{2} = t_{i+n} - \frac{t_i + t_{i+1}}{2} \tag{7.30}$$

All deglitch times $t_{d_{min},i}$ of every channel must be evaluated for all i over a whole period of the LFSR sequence, as only then they start repeating themselves.

As the deglitch times are based on the clock period, a deviation of the clock frequency has a direct influence on the deglitch times. Therefore, the clock variation must be included in the analysis of worst-case minimum and maximum deglitch times.

7.3.5.2. Influence and Optimization of LFSR Output Mapping

A constant deglitch time can only be achieved by means of uniform sampling. Using random sampling, there is always an inherent variation if the deglitch time is based on a constant number of samples. If every channel has a priority of one, then the permutation of the assignment of LFSR states to channels already makes a difference for the timings, because the sampling and conversion times for the channels are different.

In a general voltage monitoring system with different priorities, there are many different ways to map LFSR states to channels. When due to a higher priority multiple states map to one channel, the resulting sampling sequence is the superposition of the sampling sequences associated with the states. By looking at the example for a 3-bit output of a 6-bit LFSR in Fig. 7.10, it can be seen that the combination of two arbitrary states can produce very different channel sampling sequences.

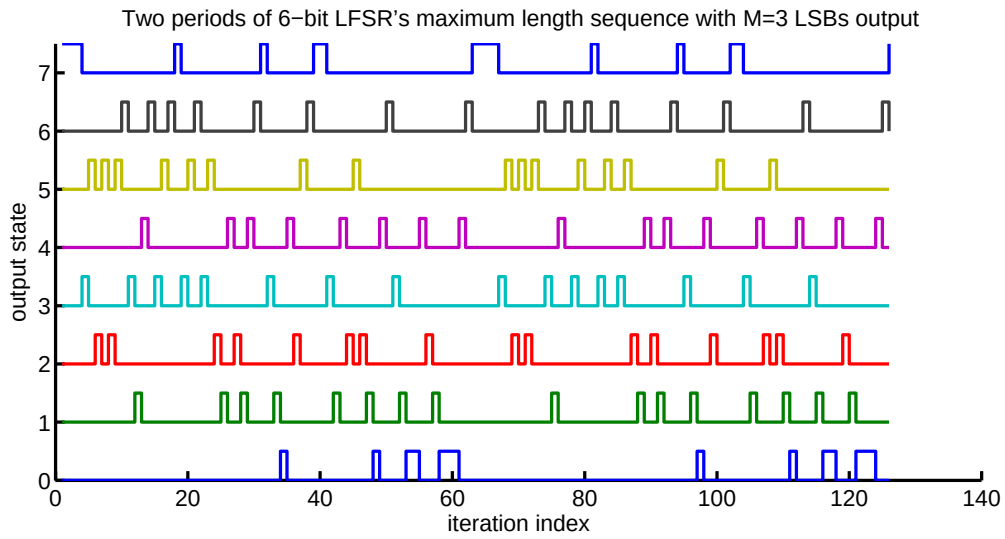

Figure 7.10.: Example for the 3-bit output of a 6-bit LFSR.

Figure 7.11 shows the deglitch times corresponding to the sampling sequences in Fig. 7.10 for a deglitch filter length of $n = 5$ samples. The deglitch time variations vary strongly between channels. This fact can be exploited to find an assignment of states to channels which fulfills the deglitch specification requirements (minimum and maximum deglitch times) – if such an assignment exists at all.

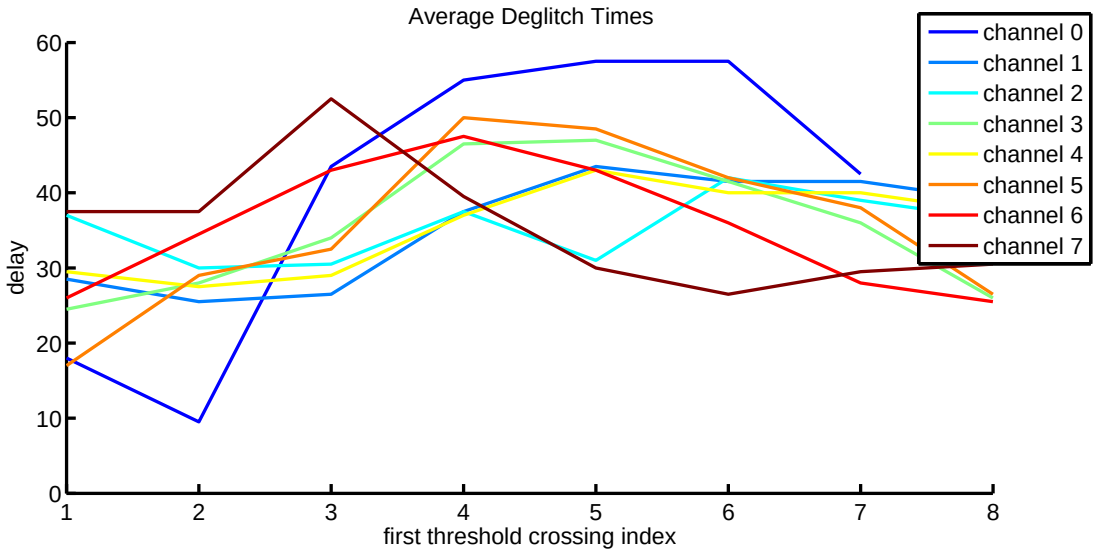

Figure 7.11.: Average deglitch times $t_{d_{avg},i}$ with deglitch filter length $n = 5$ for the 3-bit output of a 6-bit LFSR.

The number of distinguishable assignments of states to channels can be expressed by the

multinomial coefficient

$$\binom{\sum_{i=1}^{\Gamma+1} p_i}{p_1, p_2, \ldots, p_{\Gamma+1}} = \frac{\left(\sum_{i=1}^{\Gamma+1} p_i\right)!}{p_1! \cdot p_2! \cdot \ldots \cdot p_{\Gamma+1}!}, \tag{7.31}$$

which results in very high values for a realistic channel setup.

It is infeasible or even practically impossible to try all possible assignments until the deglitch times on all outputs meet their timing requirements. Hence a heuristic is proposed where only those states are randomly permuted where the assigned channels have outputs which violate their deglitch time specifications. This process ideally converges such that all deglitch time variations are within the limits.

Initially the algorithm starts with a random permutation of all state to channel assignments. Then the deglitch times are checked against the specifications for all outputs of all channels. Those channels which have no violations keep their mappings, but all other states are randomly permuted again. It is possible that due to permutations of other channel mappings an output which previously fulfilled its specification violates it again. This verification and permutation process continues until all specifications are met. If the algorithm gets stuck it must be restarted by permuting all states randomly.

7.4. Anti-Aliasing Analysis

The voltage monitoring system performs measurements on signals which are not bandlimited. The random sampling scheme guarantees the immunity against aliasing effects only for very high numbers of samples, but the real-time requirements of the system limits the number of samples. Therefore, a concrete configuration of the voltage monitoring system must be analyzed for its anti-aliasing properties.

7.4.1. Deglitch Filter

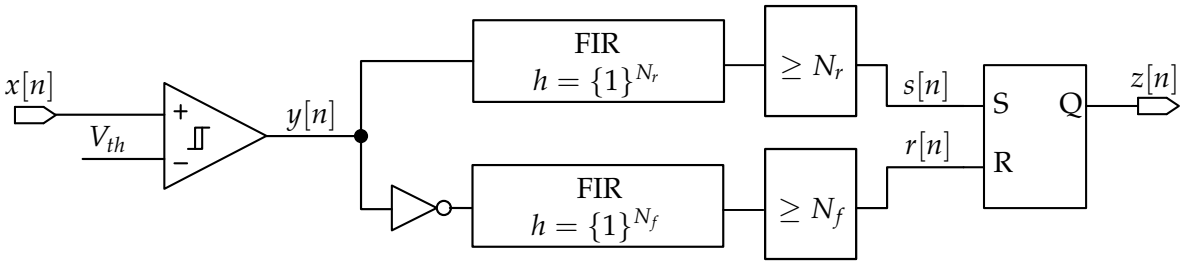

Figure 7.12.: Abstract signal processing chain of one output signal with different rising and falling delays and threshold voltage V_{th} without hysteresis.

The digital deglitch filter is completely specified by a length for the rising and falling filters. Figure 7.12 shows the abstract signal processing chain for one output signal of a channel with a specific threshold voltage V_{th}. This is not the way the filter is implemented, but it is useful for the behavior analysis. A new non-uniform sample $x[n]$ is processed whenever the

error case	ideal output	actual output	safe state 0	safe state 1
inadvertent rising	0	$0 \rightarrow 1$	↯	✓
omitted rising	$0 \rightarrow 1$	0	✓	↯
inadvertent falling	1	$1 \rightarrow 0$	✓	↯
omitted falling	$1 \rightarrow 0$	1	↯	✓

Table 7.3.: Summary of inadvertent or omitted output transitions.

corresponding channel is sampled at the time t_n. The fast compare conversion creates the 1-bit signal $y[n]$ by comparing the sampled voltage with the threshold V_{th} of the output:

$$y[n] = \begin{cases} 0 & \text{if } x[n] < V_{th} \\ 1 & \text{if } x[n] \geq V_{th} \end{cases} \tag{7.32}$$

Each of the FIR filters is chained with a digital comparator, which means that $s[n]$ and $r[n]$ are only high if all samples in the filter memory of N_r or respectively N_f samples are one. The filter for the falling edge is fed with the inverted signal $\overline{y}[n]$, which means that $s[n]$ and $r[n]$ can never be high at the same time. The output signal $z[n]$ is bistable, which is illustrated by a SR flip-flop. Hence, if neither a set $s[n] = 1$ or a reset $r[n] = 1$ is encountered, the output keeps its previous value.

Therefore, in case of doubt, that is if the signal $y[n]$ toggles its value and consequently $s[n]$ and $r[n]$ are zero, no decision is taken and the output remains as it is. This behavior is demanded for the voltage monitoring system for high immunity against disturbances. Consequently, if the DC voltage goes above or below the threshold, the presence of disturbances could prevent a change of the output $z[n]$ to the correct value. An important observation is that for the analysis of the switching behavior only the 1-bit threshold crossing signal $y[n]$ needs to be considered – all other information about the input signal is lost.

7.4.2. Voltage Level Monitoring as a Decision Problem

For each output signal the voltage monitoring system has to decide whether the voltage rail has been above or below the relevant threshold for the whole deglitch time. As this decision is only based on a limited number of samples, it naturally suffers from inaccuracies. For instance, pulses of short duration will most likely not be detected. This is not problematic, as due to the deglitch filter time no switching action is expected. However, while the analog system would reset the filter time due to the pulse, the digital voltage monitoring system is not aware of this at all.

Table 7.3 summarizes the possible differences between expected output states (as dictated by the analog system) and the actual output states determined by the digital system. In an inadvertent decision the digital system changes the output value when it should not have changed. On the other hand, an omitted decision means that the digital system does not change the output when it should.

Which of these errors is critical in a system depends on the existence of safe states. For instance, an output representing an overvoltage comparator has a safe state of one, that is the state where an overvoltage is assumed. Therefore, if an omitted falling error or an inadvertent rising error occurs, they influence the reliability of the circuit, but not its safety.

In the next sections it is analyzed which signals can cause false decisions due to aliasing. The omitted decisions are not treated explicitly, because in the presence of disturbances the analog system's deglitch filter always resets and hence prevents any change of the output. Therefore, there is no golden reference against which the digital voltage monitoring system could be compared. Either the digital voltage monitoring system also keeps the output value untouched, which is equivalent to the analog system, or it changes the output and hence makes the correct decision.

7.4.3. Additive White Gaussian Noise

Additive white Gaussian noise (AWGN) is uncorrelated and hence allows a simple calculation of the decision error probability. Therefore, the probability that a decision is made due to noise is independent of the random sampling scheme. Let the variable $x[n]$ represent the n-th sample of the input signal, which consists of a DC voltage A and AWGN ν with variance σ. The comparator output $y[n]$ is the result of comparing $x[n]$ to the threshold voltage V_{th}.

$$x[n] \sim \mathcal{N}(A, \sigma^2) \tag{7.33}$$

$$\tag{7.34}$$

The probability to get a low or high output of the comparator for one sample is then

$$P(y[n] = 0) = P(x[n] < V_{th}) = \Phi\left(\frac{V_{th} - A}{\sigma}\right) \tag{7.35}$$

$$P(y[n] = 1) = P(x[n] \geq V_{th}) = 1 - \Phi\left(\frac{V_{th} - A}{\sigma}\right), \tag{7.36}$$

where Φ is the CDF of the standard normal distribution. A deglitch filter with length N only makes a decision if either all N samples are zero or if all samples are one. Due to the independence of the noise samples, the probability that the output is switched to high by the set signal $s[n] = 1$ or switched to low by the reset signal $r[n] = 1$ is then:

$$P(r[n] = 1) = \prod_{i=n-N+1}^{n} P(y[i] = 0) = P(y[n] = 0)^N = \Phi\left(\frac{V_{th} - A}{\sigma}\right)^N \tag{7.37}$$

$$P(s[n] = 1) = \prod_{i=n-N+1}^{n} P(y[i] = 1) = P(y[n] = 1)^N = \left(1 - \Phi\left(\frac{V_{th} - A}{\sigma}\right)\right)^N. \tag{7.38}$$

If the DC voltage A is the same as the threshold voltage, then the probability $P(y[n] = 0) = P(y[n] = 1) = 0.5$. Consequently, the probability that the noise causes $P(r[n] = 1) = P(s[n] = 1) = 2^{-N}$. Therefore, the probability for such a wrong decision is significantly decreased with an increased deglitch filter length N.

7.4.4. Sinusodial Signals

A common type of disturbance signals are of sinusoidal nature, which makes them interesting for the anti-aliasing analysis. Additionally, the direct power injection immunity test method for integrated circuits described in the next subsection makes use of sinusoidal test signals.

7.4.4.1. Electromagnetic Immunity – Direct Power Injection

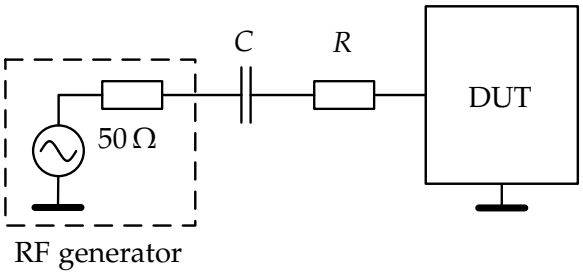

Figure 7.13.: The principle of the direct power injection (DPI) method applied on a single pin of a device under test (DUT) according to IEC 62132-4 [8].

The voltage monitoring system is a part of an automotive SoC. Therefore, a high immunity against EMI is important to prevent wrong voltage measurements due to aliasing. Hence, the concept should be proven to withstand the susceptibility tests of the applicable EMC standard for automotive electronic systems. In the susceptibility tests the test signals can be injected via galvanic coupling or via electromagnetic waves [17]. The galvanic coupling yields higher disturbance amplitudes and hence is exclusively dealt with in this section as a representative test method for immunity.

The direct power injection (DPI) method is described in part 4 of the IEC 62132-4 standard [8]. It features a simple test-setup and has a high measurement repeatability. Figure 7.13 shows the basic principle for the single pin DPI. The capacitance C is used to block DC and its value is typically 6.8 nF, while the resistor R is by default 0 Ω unless more attenuation is wanted. The connections are established by means of coaxial cables with a wave impedance of 50 Ω, which results in impedance matching at the radio frequency (RF) generator. The impedance mismatch at the integrated circuit (IC) pin usually leads to a big portion of the injected power to be reflected.

The test signal is usually a continuous wave (CW), which can optionally be amplitude modulated with 80 %, but for the IC level test the peak power is the same as for the unmodulated signal. The test is performed for signal frequencies from 150 kHz up to 1 GHz. At every inspected frequency the signal power is successively increased until either the maximum power level has been reached or the DUT fails. Table 7.4 shows the immunity level ranges, of which zone 1 is applicable to all externally available voltage supply rails of the voltage monitoring system.

The actual voltage level at the internal nets which might be connected to the tested pin cannot be stated in general, because it depends on all connected circuitry. Sinusoidal disturbance signals in the range from 150 kHz up to 1 GHz are used for the anti-aliasing analysis in the

Zone	Power (dBm)	External device protection	Device examples
1	30 ... 37	None (or small capacitor)	Power supply circuits, bus transceiver
2	20 ... 27	Passive low-pass filter	Communication line driver
3	10 ... 17	No direct connection	Microcontrollers

Table 7.4.: Example of immunity level ranges defined by IEC 62132-4 [8].

next sections. Only relative amplitudes are considered, which makes the analysis applicable to any circuit net.

7.4.4.2. Disturbance Signal Model

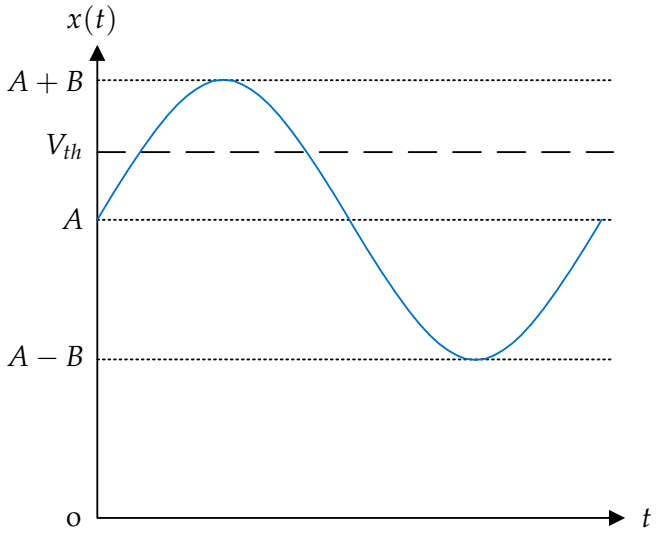

Figure 7.14.: Signal model for sinusoidal disturbances.

The continuous-time sinusoidal signal model shown in Fig. 7.14 is later on used for the aliasing analysis. It has five independent variables:

$$x(t) = A + B \cdot \sin(\omega t + \varphi) \ . \tag{7.39}$$

The cumulative density function (CDF) of the sinusoidal signal (7.39) and the PDF, which is the derivative, is given below. Both are depicted in Fig. 7.15.

$$F_X(x) = P(X \leq x) = \begin{cases} 0 & \text{if } x \leq A - B \\ 1 - \frac{\arccos\left(\frac{x-A}{B}\right)}{\pi} & \text{if } A - B < x < A + B \\ 1 & \text{if } A - B \leq x \end{cases} \tag{7.40}$$

$$f_X(x) = \frac{\mathrm{d}F_X(x)}{\mathrm{d}x} = \begin{cases} \frac{1}{B \cdot \pi} \cdot \frac{1}{\sqrt{1 - \left(\frac{x-A}{B}\right)^2}} & \text{if } A - B < x < A + B \\ 0 & \text{otherwise} \end{cases} \ . \tag{7.41}$$

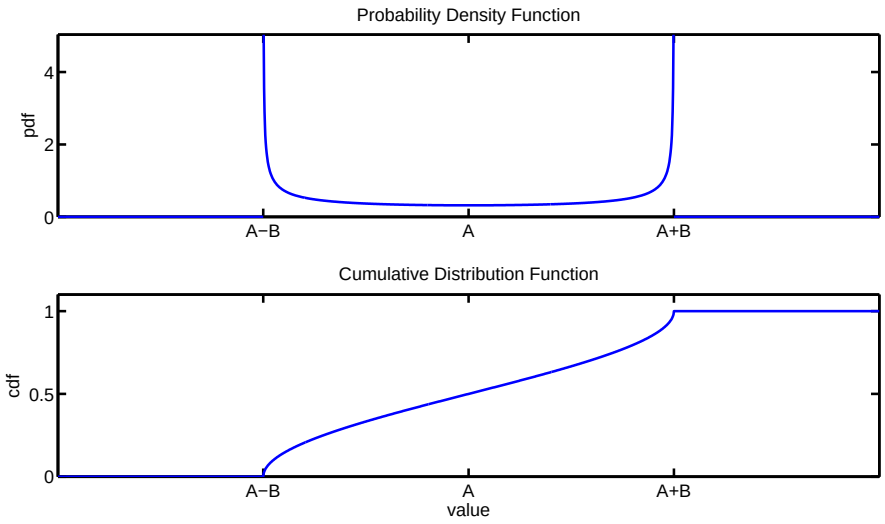

Figure 7.15.: PDF and CDF for sinusoidal with amplitude B and offset A.

If a random sampling scheme results in a constant sampling probability over time as for instance with ARS, then the sampling instants are uncorrelated with the signal. This independence permits to calculate the probability for making a decision in the same way as it was done in section 7.4.3 for noise. The probability to get a low or high output of the comparator for one sample is

$$P(y[n] = 0) = P(x[n] < V_{th}) = F_X(V_{th}) \tag{7.42}$$
$$P(y[n] = 1) = P(x[n] \geq V_{th}) = 1 - F_X(V_{th})\,, \tag{7.43}$$

and the switching probabilities for deglitch filters with length N are subsequently

$$P(r[n] = 1) = \prod_{i=n-N+1}^{n} P(y[i] = 0) = P(y[n] = 0)^N = F_X(V_{th})^N \tag{7.44}$$
$$P(s[n] = 1) = \prod_{i=n-N+1}^{n} P(y[i] = 1) = P(y[n] = 1)^N = (1 - F_X(V_{th}))^N\,. \tag{7.45}$$

7.4.4.3. Monte Carlo Analysis

The calculation of the switching probabilities due to aliasing in the previous section is only valid for an ideal constant sampling probability. However, the random sampling method used in the voltage monitoring system is not ideal and therefore a concrete configuration of the system must be verified regarding its anti-aliasing properties. This section explains an approach for such an analysis using a Monte Carlo method.

The following inequality determines if a signal sample according to (7.39) is above the threshold. The consolidation of the threshold voltage V_{th}, the DC voltage A and the sine amplitude B into the relative voltage A^* simplifies the analysis by reducing the number of variables. The relative voltage A^* is the distance of the DC voltage from the threshold voltage normalized by the sine amplitude. Hence, it makes sense to limit the range to $-1 \leq A^* \leq 1$.

$$x[n] \geq V_{th} \tag{7.46}$$

$$A + B \cdot \sin(\omega \cdot t_n + \varphi) \geq V_{th} \tag{7.47}$$

$$\sin(\omega t + \varphi) \geq \underbrace{\frac{V_{th} - A}{B}}_{A^*} \tag{7.48}$$

A closed expression is obtained when a function $\psi_b(a)$ is defined which serves as a nonlinear threshold. Then the comparator output signal $y[n]$ can be directly stated as:

$$\psi_b(a) = \begin{cases} 1 & \text{if } a \geq b \\ 0 & \text{otherwise} \end{cases} \tag{7.49}$$

$$\Rightarrow y[n] = \psi_{A^*}(\sin(\omega \cdot (t_n + \tau) + \varphi)) \ . \tag{7.50}$$

The number of signal samples in the deglitch filter of length N, which are above the threshold, is defined as $V[n]$:

$$\tau \sim \mathcal{U}(0, t_{clk}) \tag{7.51}$$

$$V[n] = \sum_{i=n-N+1}^{n} \psi_{A^*}(\sin(\omega \cdot (t_i + \tau) + \varphi)) \ . \tag{7.52}$$

The sampling time instants t_i are on the clock period time grid, which is why the random variable τ is explicitly added. The result is a mathematical model for the sampling scheme from section 7.1.4.

Relative Threshold Choice The question is which relative thresholds are relevant for the Monte Carlo simulation. If only a DC voltage A without disturbances is applied to a channel, then after the deglitch time the output will go to $z[n] = 0$ when $A < V_{th}$. When subsequently a sinusoidal disturbance is added, then the output should ideally remain at $z[n] = 0$. In case it rises to $z[n] = 1$ by $s[n] = 1$, this is an inadvertent decision caused by aliasing.

Specifically, the random samples must all happen to fall in the time instants when the sinusoidal disturbance makes the signal value greater than the threshold. Such a case is depicted in Fig. 7.16, where for a filter length of $N = 5$ the shown sine wave causes inadvertent switching of the output to $z[n] = 1$, even though the DC level is below the threshold V_{th}. Obviously, a longer deglitch filter time would have saved the day in this example. If the random sampling scheme is of sufficient quality, then increasing the number of samples in the deglitch filter also decreases the probability for wrong decisions due to aliasing.

Ideally, in the presence of a disturbance the output remains at its previous value. However, if the output is low and $r[n] = 1$, then this results in no change of the status quo as the output was already zero. Only aliasing which leads to $s[n] = 1$ is relevant then.

If A is below the threshold and $z[n] = 0$, then the probability for switching the output high by $s[n] = 1$ because of aliasing is maximal at $A = V_{th}$, which is equivalent to the relative voltage

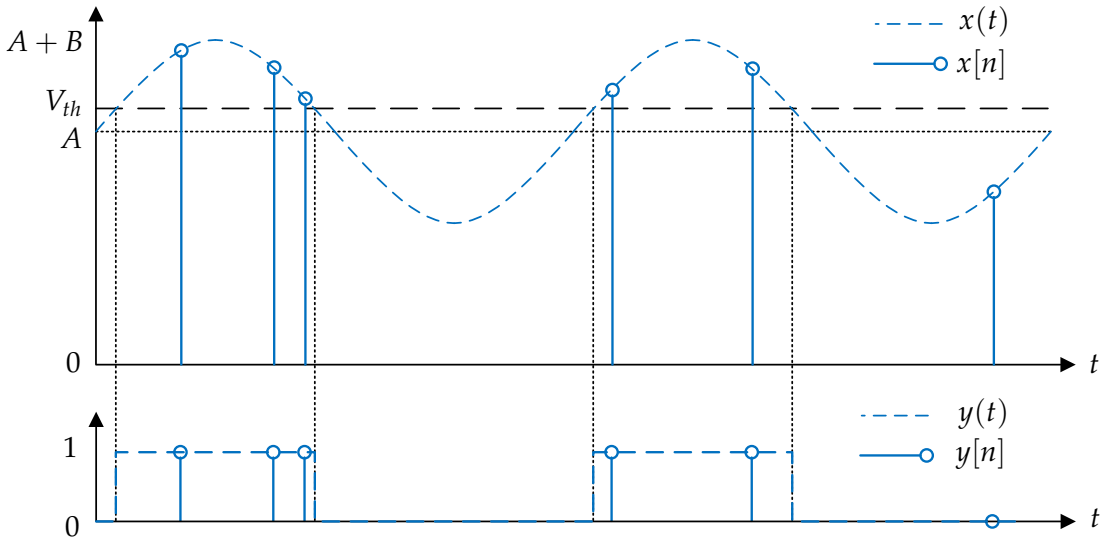

Figure 7.16.: Example for wrong decision due to sine aliasing with $N = 5$ deglitch samples.

$A^* = 0$. If the voltage was any higher, then the probability for $s[n] = 1$ is higher, but it is not aliasing anymore because then the DC voltage $A > V_{th}$ and hence switching to high is expected. Similar reasoning leads to the conclusion with an initial output value of $z[n] = 1$ and switching to $z[n] = 0$ due to aliasing. In summary, performing the aliasing analysis for $A^* = 0$ is representative for the worst-case aliasing switching probability for the rising and for the falling case. The analysis at the relative threshold voltage $A^* = 0$ gives equal probabilities for the signal value being above or below the threshold. Furthermore, it does not matter if the signal is amplitude modulated or not, because at $A^* = 0$ the amplitude envelope is irrelevant.

Monte Carlo Runs In each Monte Carlo run a realization of every random variable is drawn from its respective distribution. The minimum reasonable sinusoidal frequency is limited by the length of the deglitch filter, as signals with a longer period will likely cause a change of the output. For every run, (7.52) must be evaluated. The pseudo-random sampling time instants t_i can be calculated deterministically. If the LFSR has a period length of m, then for one run the finite impulse response (FIR) is calculated for m time shifts. This guarantees that the transition region between two LFSR periods is also taken care of. If the filter length is N, then the convolution of the FIR filter requires $m + N - 1$ samples. This requires drawing as many jitter values $\tau \sim \mathcal{U}(0, t_{clk})$, which then remain constant for calculating $V[n]$ for the different n. The phase and frequency of the sine are drawn from a uniform distribution:

$$\omega \sim \mathcal{U}(2\pi f_{min}, 2\pi f_{max}) \tag{7.53}$$

$$\varphi \sim \mathcal{U}(0, 2\pi)\,. \tag{7.54}$$

In each run the histogram of $V[n]$ is calculated by evaluating (7.52). An example for 1000 runs of the Monte Carlo simulation for an arbitrary channel in a voltage monitoring system configuration is shown in Fig. 7.16. The frequency range is restricted to $\omega \sim 2\pi \cdot \mathcal{U}(150\,\text{kHz}, 1\,\text{GHz})$. The filter length has a realistic value of $N = 24$ and the LFSR length is $m = 2^{11} - 1$. The

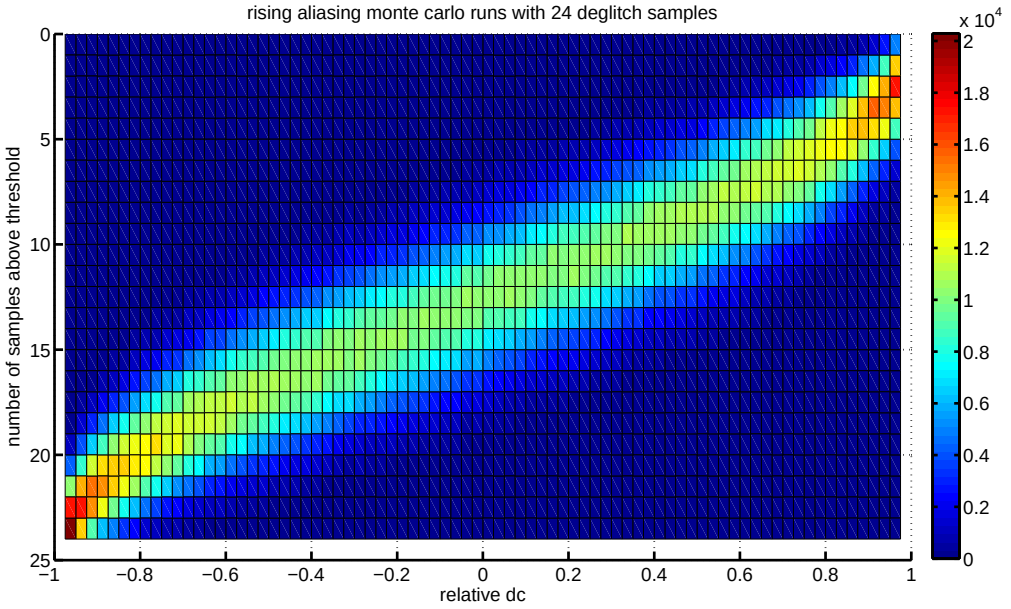

Figure 7.17.: Example of 1000 Monte Carlo runs for every relative voltage.

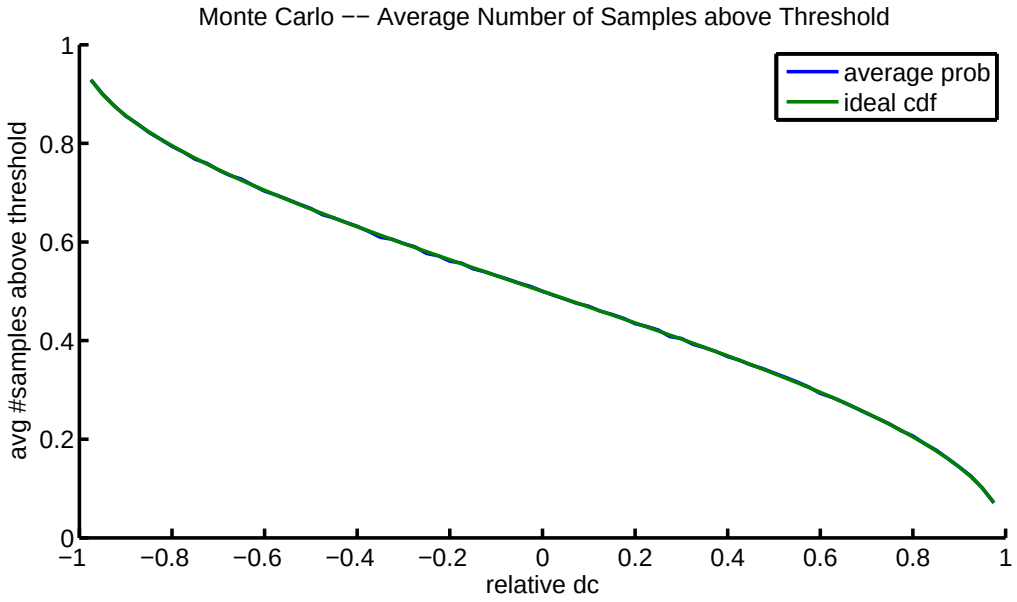

Figure 7.18.: Average number of samples above threshold voltage for 1000 Monte Carlo runs for multiple relative voltages.

results of all Monte Carlo runs are shown in a three-dimensional histogram plot for multiple relative threshold voltages. As explained, the analysis for $A^* = 0$ is sufficient, but in this example some relationships become more apparent. Figure 7.18 show the normalized average of $V[n]$ for the different threshold voltages. Apparently, it fits very well with the theoretical values as obtained from the ideal CDF of the sinusoidal signal.

The number of Monte Carlo runs must be sufficiently high in order to get reliable results. If a situation is encountered where $V[n] = N$ (for the rising deglitch filter analysis) or $V[n] = 0$ (for the falling deglitch filter analysis), then aliasing has caused an inadvertent output change. As can be seen in Fig. 7.16, this does not happen often. According to (7.45), the aliasing switching probability at $A^* = 0$ is ideally 2^{-N}. Hence, increasing the number of samples increases the aliasing immunity. If in a Monte Carlo run a case of aliasing is detected, it is verified if this is a systematic problem of the random scheduler or if it is a coincidence. For that purpose, the frequency ω is kept constant, while only the other variables are drawn freshly from their distributions. If no more aliasing cases occur, then the initially encountered aliasing case was only a coincidence. In chapter 10 the analysis and the results are shown for a real voltage monitoring system.

Unfortunately, the nonlinearity of the system does not allow generalizing from the analysis of one sinusoidal signal, because the superposition principle is not valid. Nevertheless, the results of the sinusoidal signal are valid for all signals which lead to the same threshold crossing signal. If a specific disturbance signal occurs in an application, then a Monte Carlo simulation can be performed for that signal type to check the immunity against inadvertent decisions. However, the simulation in section 10.3.4 exemplifies that a good resistance against sine disturbances can lead to good immunity against other disturbance signals, as long as they are not intentionally correlated with the sampling sequence.

Part III.

Proof of Concept

8. Digital Implementation

This part explains selected implementation aspects of the concept introduced in part II. The main purpose is a proof of concept, which is why not every feature is implemented. In this chapter the implementation structure of the main digital logic is shown. The actual implementation is realized by means of the Very High Speed Integrated Circuit Hardware Description Language (VHDL). In chapter 9 an automated workflow for creating the digital logic of a voltage monitoring system based on a specification is presented. Finally, in chapter 10 a basic voltage monitoring system without the extra safety features of section 4.4 is simulated. The results are used as a proof of concept for the new voltage monitoring system.

8.1. Digital Modules Overview

Except for the ADC, the whole voltage monitoring system operates purely in the digital domain. The implementation of the main functionality excluding the functional safety features (see section 4.4) is presented in this section. Figure 8.1 gives an overview of the proposed digital module hierarchy. Referring to the concept overview in Fig. 4.1, the digital part covers all modules but the ADC itself. Furthermore, the BIST control signals are missing because the BIST is not implemented in the digital model.

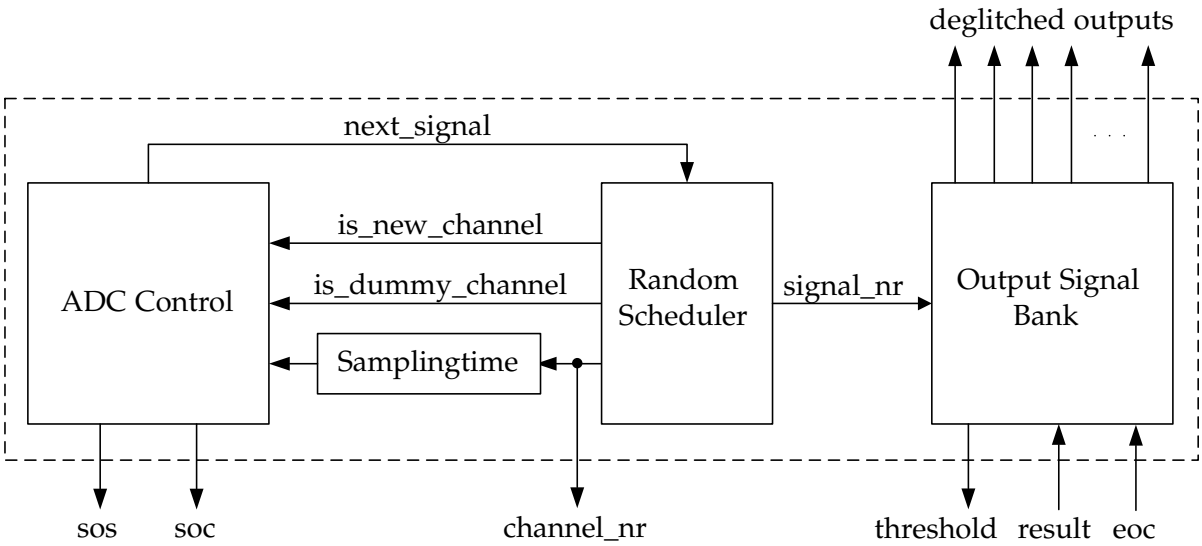

Figure 8.1.: Overview of the digital implementation.

The external connections of the digital part are the ADC interface signals sos (start of sampling), soc (start of conversion), eoc (end of conversion), the channel number, the threshold

code for the fast compare conversion and the conversion result. Furthermore, the digital part provides outputs which represent the deglitched comparators.

Hierarchically, the digital implementation is split into three main blocks and one auxiliary block. The ADC control block is a state machine which performs sampling and conversion in an infinite loop. Which channel is sampled is dictated by the random scheduler block, which randomly updates the channel number and subsequently addresses all output signals of a channel. The output signal bank contains all deglitch filters and also provides the threshold code depending on the selected signal and its current output state.

8.2. Random Scheduler

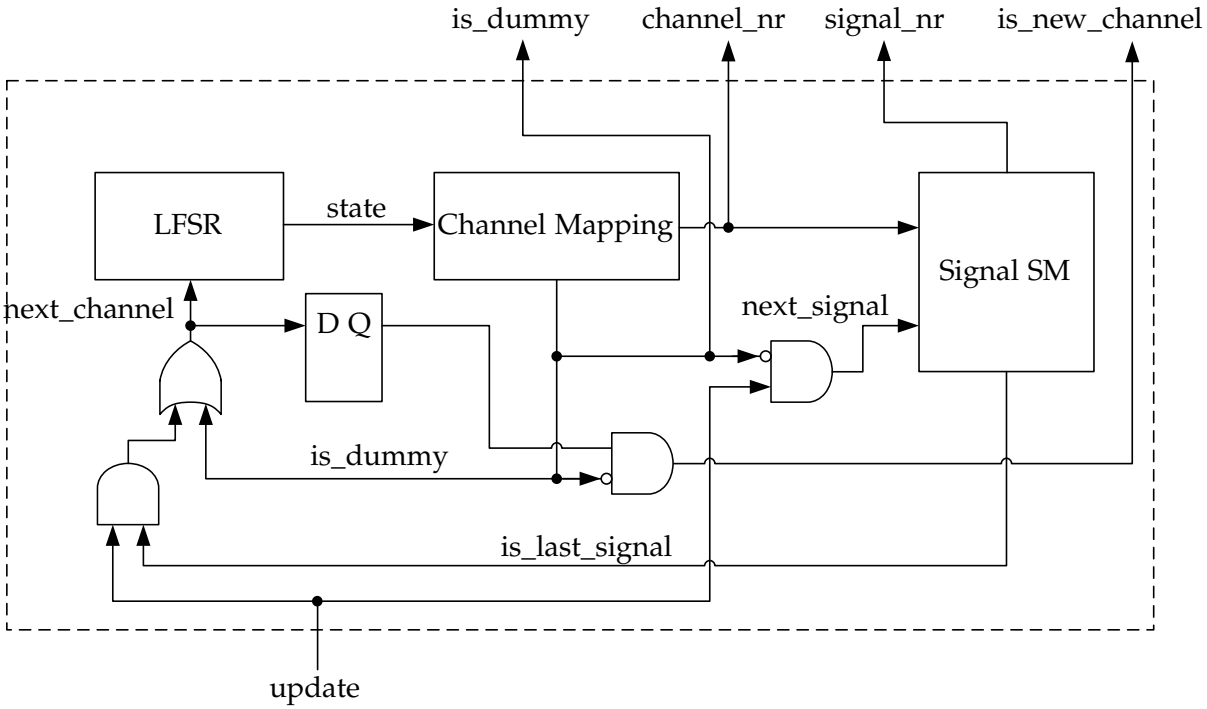

Figure 8.2.: Implementation of the random scheduler.

When requested via the input signal *update*, the random scheduler provides a new signal number or a new channel and a new signal number. Each channel is assigned a sequential list of signal numbers which are attached to it. At each update, the next signal is provided. When the last signal in the list is reached, a new pseudo-random channel number is provided along with the first signal number of the new channel.

The described functionality is split into the submodules visible in Fig. 8.2. The LFSR together with the channel mapping pseudo-randomly generates new channels according to the concept in section 7.2. The signal state machine contains the sequences of outputs for each channel and also indicates when the last signal of a channel is reached. The logic gates determine whether the LFSR is updated or the signal state machine is advanced. The channel mapping block maps the LFSR output state to a channel number. Surplus random states are mapped to a dummy

channel, which provides the time granularity of a clock period for the random sampling scheme. Therefore, if a dummy channel is encountered, an additional signal *is_dummy* is set. It causes an immediate update of the LFSR at the next clock edge and hence a delay of one clock cycle.

The output signal *is_new_channel* tells the ADC control block that a channel update has happened. As a consequence, the new channel (which could be the same channel number as before) must be sampled before the fast compare conversions can start.

8.2.1. Linear Feedback Shift Register

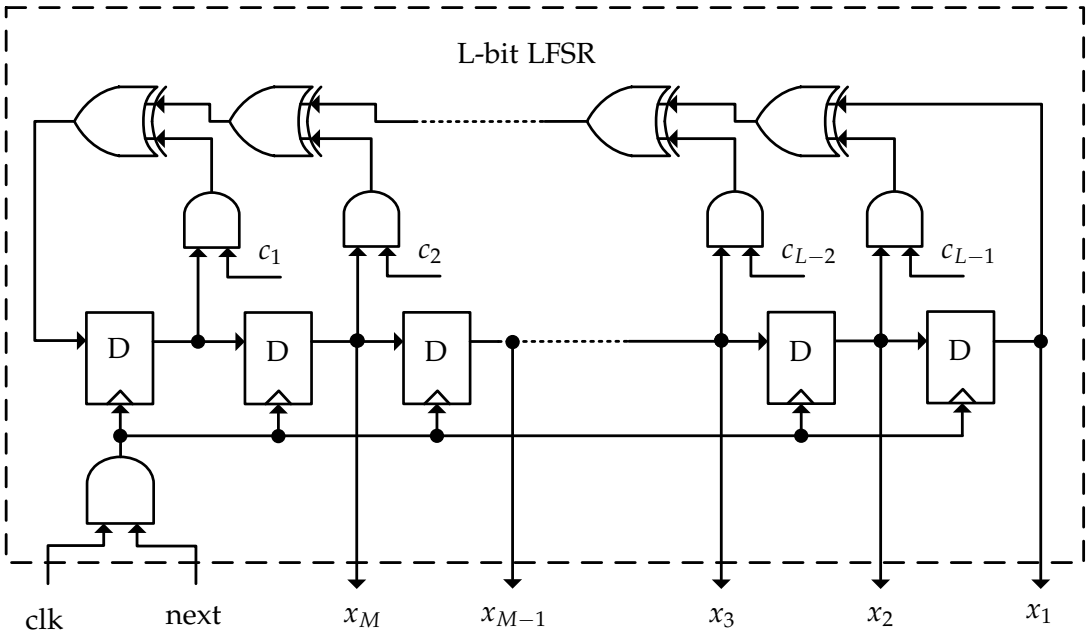

Figure 8.3.: Implementation of the linear feedback shift register.

The linear feedback shift register implementation is shown in Fig. 8.3, but D-flipflops are explicitly used as delay and storage elements. To obtain a synchronous circuit, the LFSR clock is gated with the update signal *next*. The feedback connection can either be dynamically set via the values $c_1, c_2, \ldots, c_{L-1}$, or it can be hardwired.

8.2.2. State to Channel Mapping

The state to channel mapping consists of a simple lookup table (LUT). It maps from an M-bit random state to an N-bit channel number. Depending on the properties, the logic synthesis can optimize the size of the lookup table significantly. In an application where all channels need the same sampling frequency and have a priority of one, the random state to channel number mapping is not required. However, the logic for detecting the dummy channel is still necessary.

8.2.3. Signal State Machine

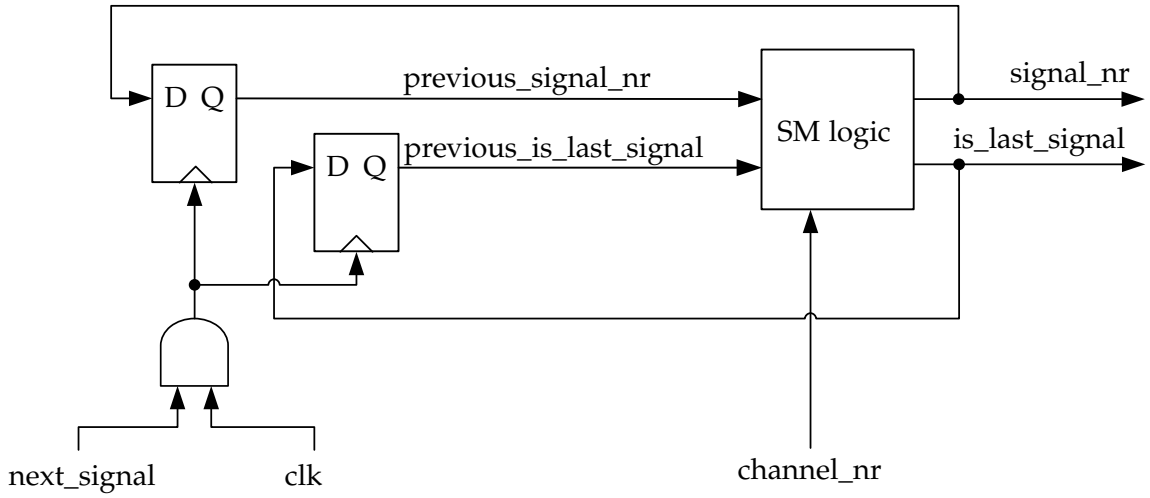

Figure 8.4.: Implementation of the signal state machine on register transfer level.

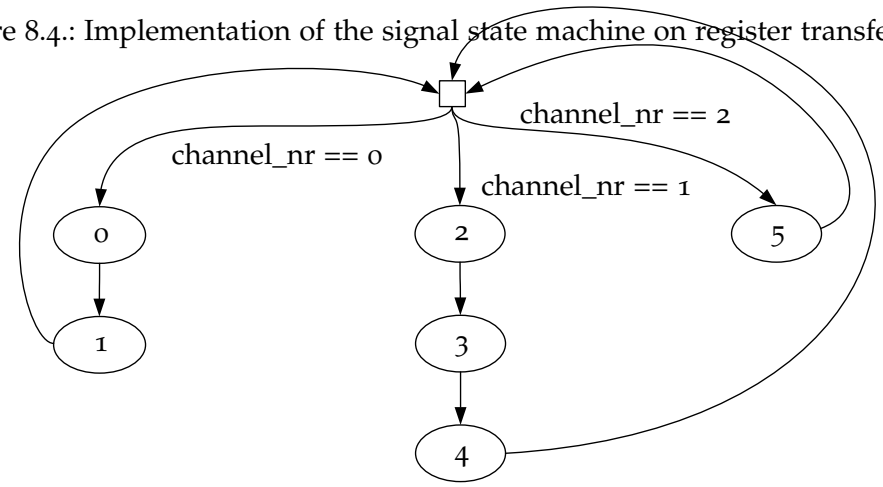

Figure 8.5.: Implementation of the signal state machine.

The signal state machine has the task of successively providing the signal numbers of all signals attached to a channel. When the last signal number is encountered, this is signified by an additional output *is_last_signal*. Figure 8.4 shows that the output is a function of the previous signal state and of whether the previous signal number was the last for the channel. In this case, the new signal number is determined by the channel number, as is visualized in Fig. 8.5. Otherwise, the state machine simply switches to the next signal of a channel. This implementation permits to determine the new signal number immediately when a new channel is selected.

8.3. ADC Control

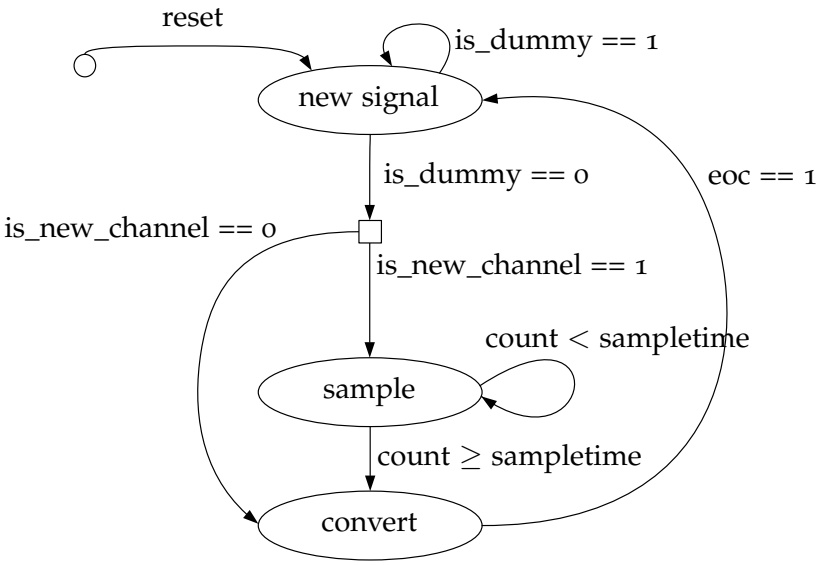

Figure 8.6.: ADC control state machine.

The ADC control block is a state machine as shown in Fig. 8.6. It initially starts in the *new signal* state, where it signals the random scheduler to provide the next signal number. If this request results in a channel update in which the dummy channel is selected, then the update signal is applied until a valid channel is encountered. This leads to a delay of one clock period per encountered dummy channel and hence guarantees a clock period time granularity.

When a valid channel number is obtained, the new channel is sampled. The sampling time is provided by a lookup-table on the basis of the channel number. This value is used as the limit for a counter. When the sampling is finished, the fast compare conversion is performed. Then the next signal is requested. If it is on the same channel, the conversion is performed directly. Otherwise, the new channel is sampled and the process starts anew.

8.4. Deglitch Filter Bank

The deglitch filter bank shown in Fig. 8.7 is a container for all individual deglitch filters. It connects the result output of the ADC to all filters, but depending on the selected signal number, the eoc (end of conversion) signal enables only the concerned filter via a one-hot encoding. All output signals of the filters are available outside, but additionally also the threshold code for the current signal number is provided. It is used in the ADC for the fast compare conversion. To provide hysteresis behavior, the threshold code depends on the output signal as indicated in Fig. 8.8.

The filter itself is depicted in Fig. 8.9. It consists of a D-flipflop to provide a bistable memory for the output state. At each update, the counter is increased whenever the input signal is equal to the output signal or reset otherwise. Depending on the output state, the counter value is compared to the rising or falling deglitch filter length value. When the respective

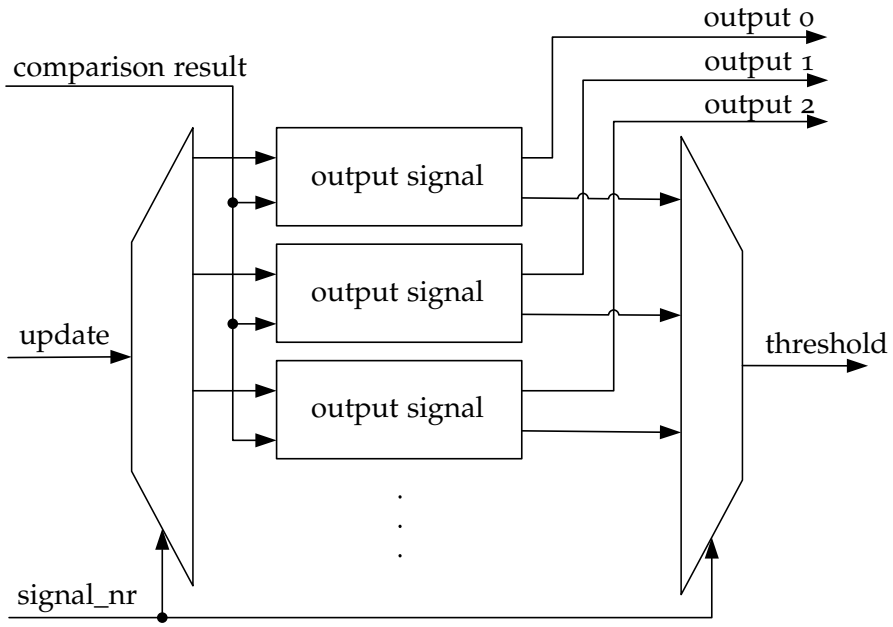

Figure 8.7.: Implementation of the deglitch filter bank with multiplexed output signals.

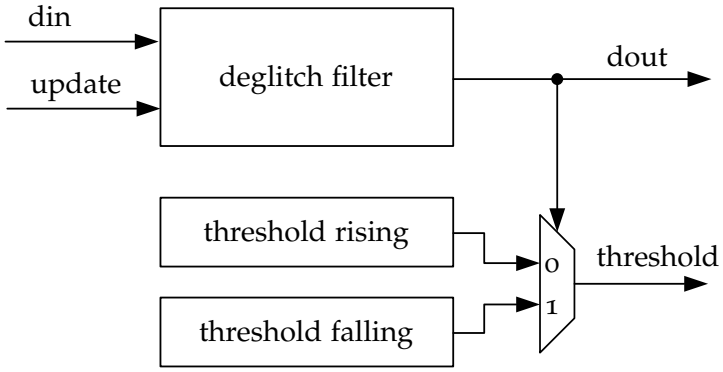

Figure 8.8.: Implementation of the output signal with state dependent threshold (hysteresis).

threshold is reached, the counter is also reset, but additionally the output signal is updated to the value of the input signal.

8.5. Testbench Setup

Figure 8.10 shows the setup for the test bench. A functional model of the ADC with customized input channels is connected to the voltage monitoring system. Each channel gets its input from a signal source with configurable offset voltage, sine frequency and amplitude as well as a superposed triangle wave. The analog sampling clock jitter is not implemented in the

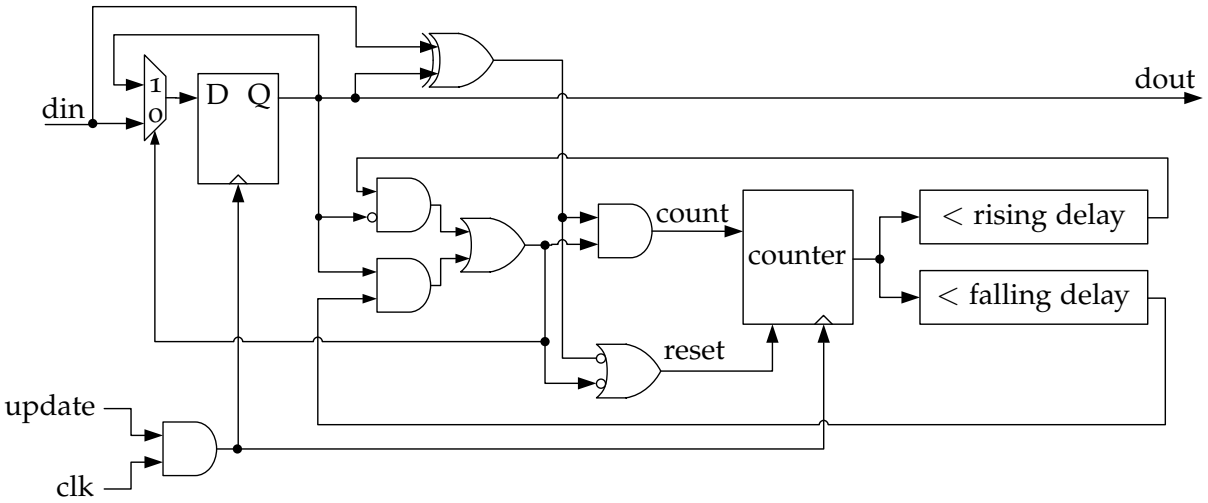

Figure 8.9.: Implementation of the deglitch filter with independent rising and falling edge delay.

functional model, which means that in the simulation the expected effective Nyquist frequency is half of the system clock frequency.

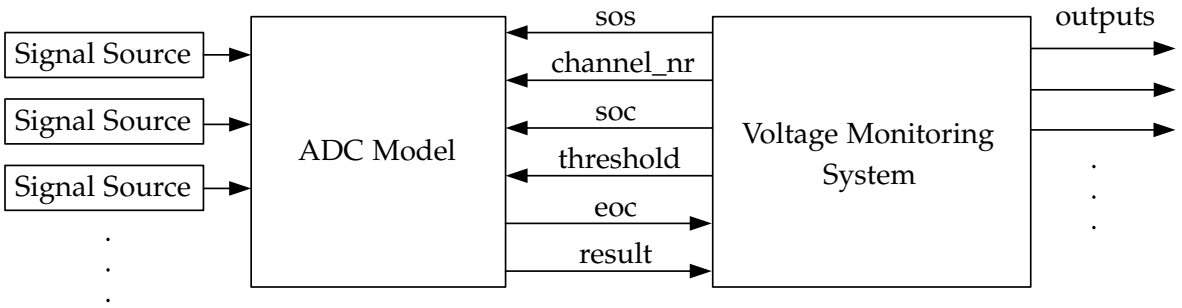

Figure 8.10.: Testbench for the voltage monitoring system.

9. Voltage Monitoring System Design and Analysis Tool

The implementation of all required digital blocks is described in chapter 8. To design a complete voltage monitoring system, parameters such as threshold codes and deglitch filter lengths must be calculated. Then it must be verified if all threshold voltages and deglitch times remain within the specification windows in all corner cases. Furthermore, the anti-aliasing immunity must be checked.

Performing all these steps manually is an strenuous task. If only one parameter is changed, all checks must be performed once again. Therefore, in this chapter an automated workflow is presented which empowers the automatic design of the digital part of a voltage monitoring system. Furthermore, it automatically performs all described analysis and verification steps. Finally, the system is automatically exported as a set of VHDL files together with a testbench, which also contains a generated functional model of the ADC with the required number of channels. This allows performing a digital simulation of the system without extra effort. The workflow tool is implemented using Matrix Laboratory (MATLAB). The specifications are provided as a spreadsheet file, which is organized as a relational database.

9.1. Specification

The spreadsheets containing the voltage monitoring specifications are organized like a relational database. Each of the following tables has a primary key, which uniquely identifies a row in the table. The tables refer to other tables by means of foreign keys, which must point to a valid row in another table. The next subsections describe the tables and their columns.

9.1.1. Operating Mode

An operating mode represents a configuration of the voltage monitoring system. It is associated with a clock source and a bandgap reference voltage. Furthermore, the domains of active outputs can be selected. Altogether, this permits to account for different operating situations of a chip such as a full operation mode and a sleep mode. They commonly differ in the accuracy of available reference voltages and clock sources, which motivates adapted widths of the specification windows.

mode_name	Unique name of the operating mode (primary key).
clock_name	Name of the used clock source (foreign key).
adc_name	Name of the used ADC (foreign key).
bandgap_name	Name of the used reference voltage source (foreign key).
active_outputs	Names of the active output domains (foreign keys separated by commas).

9.1.2. ADC

All relevant parameters can be specified in the ADC table. Theoretically, multiple ADCs can be defined, but typically only one ADC is used in a voltage monitoring system. If more outputs and channels are required than can be handled by one voltage monitoring system, it is recommended to split the system into smaller instances.

adc_name	Unique name of the ADC (primary key).
num_bits	Integer signifying the resolution in bits.
caparray	Value of the total capacitor array in multiples of the unit capacitance.
fast_compare_clocks	Number of clock cycles required for a fast compare conversion.
error_offset_lsb	Absolute value of worst case offset error.
error_gain_percent_fsr	Absolute value of worst case gain error in percent.
error_inl_lsb	Absolute value of worst-case INL error.

9.1.3. Bandgap

Multiple bandgap voltage references with different accuracies can be present in a SoC. An operating mode must select one of them to be used in the voltage monitoring system.

bandgap_name	Unique name of the bandgap reference source (primary key).
bandgap_voltage	Nominal voltage.
bandgap_error_percent	Absolute worst-case error in percent.

9.1.4. Clock

Similar to the bandgap voltage sources, also multiple clock sources can be available in a SoC.

clock_name	Unique name of the clock source (primary key).
clock_freq_logic_mhz	Logic clock frequency in MHz.
clock_freq_adc_mhz	ADC clock frequency in MHz. It can be the same as the logic clock frequency or lower due to a frequency divider.
clock_tolerance_percent	Absolute worst-case error of the clock frequency in percent.

9.1.5. Channel Type

A channel type of the ADC is associated with a sampling time as well as minimum and maximum operating voltages. The parameters depend on the used type of transistors and protective circuitry.

type_name	The channel type name serves as a primary key.
sample_time_ns	Minimum required sampling time.

9.1.6. Channel

Each input channel of an ADC is of a certain channel type and has its own gain capacitance which in combination with the total capacitor array size determines the channel's full scale range (FSR) voltage.

channel_nr	Unique integer to identify the ADC channel.
node	Unique name of the node in the schematic (primary key).
type	Type of the channel (foreign key).
gain_cap	Gain capacitance value in multiples of unit capacitance (LSB capacitance).

9.1.7. Output Signal

An output signal is a virtual comparator which is attached to an ADC input channel (and consequently to a node in a schematic). Specification windows are defined for the comparator and deglitch filter parameters. The association with an operating domain allows selectively enabling of a subset of output signals in a specific operating mode.

output_nr	Unique integer to identify the output signal.
domain_name	The output signal is part of this operating domain (foreign key).
signal_name	Unique name identifying the output signal (primary key).
node_name	Name of the channel (foreign key).
threshold_rising_min	Minimum rising threshold voltage.
threshold_rising_max	Maximum rising threshold voltage.
threshold_falling_min	Minimum falling threshold voltage.
threshold_falling_max	Maximum falling threshold voltage.
hysteresis_min	Minimum hysteresis voltage.
hysteresis_max	Maximum hysteresis voltage.
deglitch_rising_min_us	Minimum rising deglitch time in microseconds.
deglitch_rising_max_us	Maximum rising deglitch time in microseconds.
deglitch_falling_min_us	Minimum falling deglitch time in microseconds.
deglitch_falling_max_us	Maximum falling deglitch time in microseconds.

9.2. Automated Workflow

The specifications as defined in the previous section are at first parsed. This leads to an object-oriented representation of the specifications as indicated in the left box of the class diagram in Fig. 9.1. The subsequent automated digital design is performed for each specified operating mode separately. Therefore, for each operating mode a separate digital part is created. In a future version of the tool, a combined synthesis can prevent redundant structures such as deglitch filters.

In the right box of the class diagram in Fig. 9.1 the implementation classes are shown. They do not represent a functional implementation, but rather the actual parameters for an implementation. For instance the deglitch filter specification contains a minimum and maximum filter time, but the implementation has an actual filter length value. The operating mode object is at the top of the implementation hierarchy.

9.2.1. Design and Analysis

In the design step the implementation parameters are derived from the specifications. At first, depending on the active domains in the specific operating mode, the subset of activated outputs is determined. Channels without active outputs are ignored in the following steps.

For each output, the deglitch filter lengths and threshold codes must be calculated. The ideal rising and falling threshold voltage are each assumed to be in the middle of their specification window: $V_{th} = \frac{1}{2}(V_{th,min} + V_{th,max})$. If this assumption is not true for a specific voltage monitoring system, then an additional specification parameter for the typical threshold voltage must be added.

The threshold voltages are converted to the nominal digital codes using $\varepsilon = 0$ as explained in section 5.3. Assuming the worst-case maximum and minimum errors with $\varepsilon = \pm 1$, the codes are converted back to voltages. Those are compared with the specification windows to check if any violations could occur. The hysteresis voltage is checked in a similar way according to the equations in section 5.4. All calculations assume a worst-case correlation of $\rho = 1$ as explained in section 5.5.

The random scheduler is constructed as explained in section 7.3 by successively calculating channel priorities, the cycle time and the deglitch filter lengths. For the ideal deglitch filter time once again the average of the minimum and maximum time is assumed. After the LFSR length is calculated, a primitive connection polynomial of the required length is looked up in a table to make the LFSR produce a maximum length sequence. The LFSR random output to channel number mapping is performed as demonstrated in section 7.3.5.2. Hence, the deglitch time variation is analyzed and only states mapping to channels where output signals violate their specifications are permuted. However, the number of iterations is limited. If no assignment to fulfill the specifications is found, the resolution parameter r can be increased for more freedom in the state assignment. This comes at the cost of an increased lookup table size. Alternatively, the specification windows can be adapted to permit a higher deglitch time variation.

Finally, the anti-aliasing analysis is carried out for each output signal as presented in section 7.4.4.3. The Monte Carlo simulation is performed for rising and falling deglitch filters

separately, because they can have different lengths. Additionally, the definition of aliasing-related inadvertent output states differs for them. The number of runs is configurable and should be high enough to reach a high confidence level in the aliasing immunity by inspection of the distribution parameters.

9.2.2. VHDL Export

After the design and analysis the digital part of the voltage monitoring system can be automatically exported to Very High Speed Integrated Circuit Hardware Description Language (VHDL). The digital implementation using VHDL is shown in chapter 8. The automatic export uses templates to create the digital hierarchy. The implementation parameters are inserted into the components of the VHDL code. Multiple instances are created by means of components with generic ports to configure them. Additionally, the bit widths of all signals are calculated based on the highest number to be represented. For instance, the channel number bit width N is determined as

$$N = \log_2 \max(\text{channel number} + 1)\,, \tag{9.1}$$

assuming that the numbers start from zero. The numbers do not need to be consecutive, as they depend on the channel configuration of the ADC and on the subset of active channels in a certain operating mode.

9.3. Class Diagram

The class diagram of the MATLAB tool is shown in Fig. 9.1. Only the most important properties and methods are shown due to space restrictions.

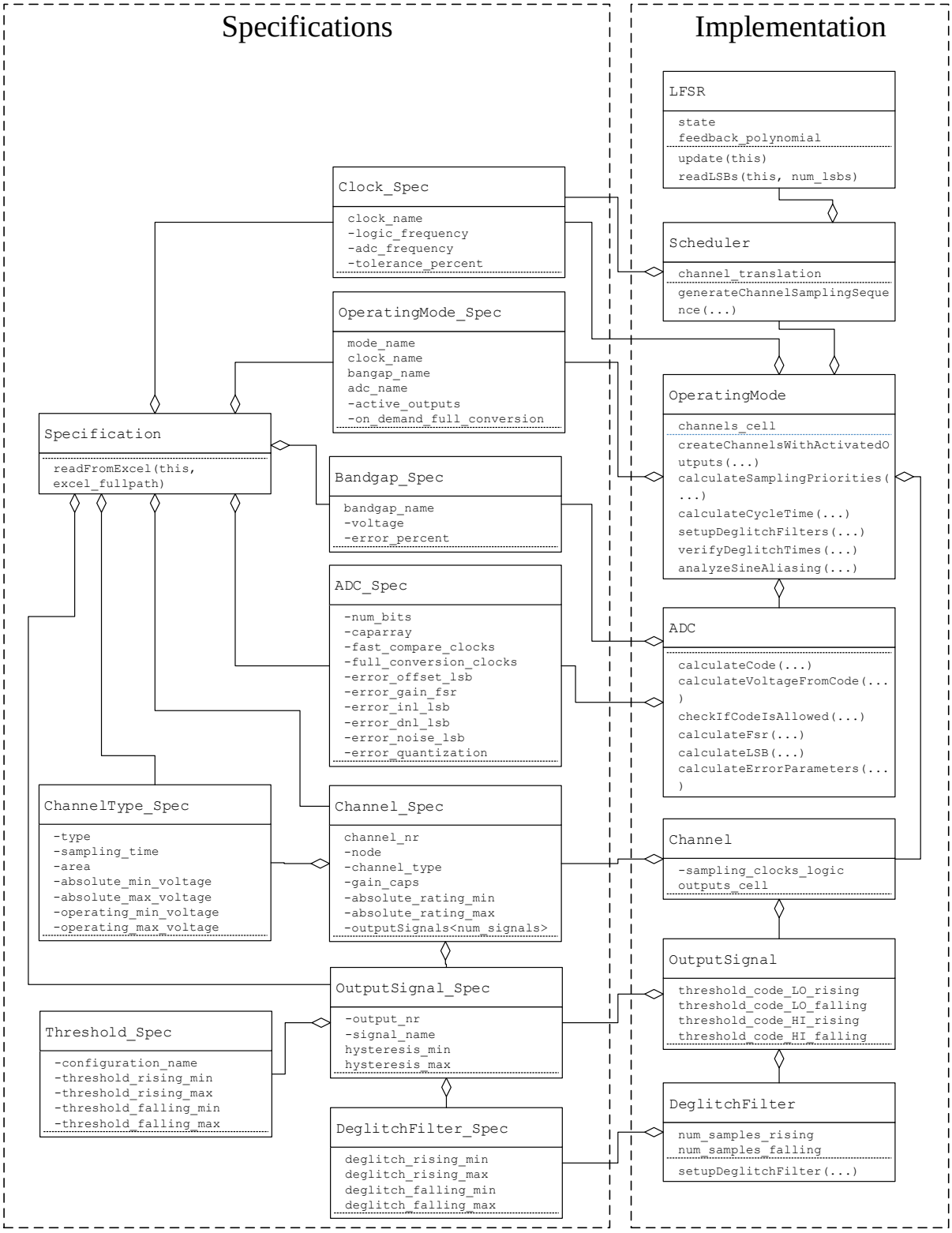

Figure 9.1.: Class diagram of the design tool for voltage monitoring systems.

10. Concrete Example

This chapter provides the proof of concept by means of a simulation. An exemplary voltage monitoring system is specified and processed by the automated workflow tool. In a simulation ramp signals are applied to the resulting VHDL testbench to verify the correct voltage thresholds and the correct deglitch times. Furthermore, sinusoidal disturbance signals are superposed to show that the system does not make wrong decisions in the presence of disturbances.

10.1. Specifications

Tables 10.1 to 10.5 show the exemplary specifications for a voltage monitoring system. A 10-bit ADC with 4 channels is used, whereof each has two attached output signals. They serve as an over- and undervoltage comparator respectively. Two channels are rated up to 40 V and require longer sampling times than the 10 V channels. The output signals are assigned to two different domains, but in the defined operating mode both domains and hence all outputs are active.

The output signals in the `startup` domain have no defined falling threshold voltages, which means that they do not have a hysteresis. Furthermore, they have symmetrical rising and falling deglitch times. On the other hand, the `boost_35` and `buck_25` channels have output signals with different deglitch times. For instance, the overvoltage signals have a lower rising deglitch time and a longer falling deglitch time. Therefore, once the overvoltage condition is indicated, it takes longer to go back to the normal state. This behavior is enhanced by the hysteresis voltage, which requires the voltage to go back to an even lower level to remove the overvoltage condition.

mode_name	clock_name	adc_name	bandgap_name	active_outputs
full_operation	CLK_normal	ADC10	BG_normal	startup, normal

Table 10.1.: Operating mode specification.

bandgap_name	bandgap_voltage	bandgap_error_percent
BG_normal	1.215	1

Table 10.2.: Bandgap reference voltage specification.

channel_nr	node	type	gain_cap
0	lvr_3	C10	256
1	lvr_5	C10	192
2	boost_35	C40	32
3	buck_25	C40	46

Table 10.3.: Channel specifications.

type_name	sample_time_ns
C10	200
C40	500

Table 10.4.: Channel type specifications.

10.2. Synthesis and Analysis

The specification spreadsheets given in the last section are used as an input for the automated synthesis and analysis workflow. The following figures represent screenshots of tables and diagrams which are automatically produced by the MATLAB tool of section 9.

output name	channel name	rising min	rising max	falling min	falling max	hysteresis min	hysteresis max
boost_35_ov	boost_35	36.5915 V	38.24 V	35.8097 V	37.4263 V	670.0368 mV	929.8469 mV
boost_35_uv_n	boost_35	32.6829 V	34.1719 V	31.9012 V	33.3583 V	670.0368 mV	929.8469 mV
buck_25_ov	buck_25	26.18 V	27.3564 V	25.688 V	26.8443 V	414.3223 mV	592.9459 mV
buck_25_uv_n	buck_25	23.2279 V	24.2838 V	22.7359 V	23.7717 V	414.3223 mV	592.9459 mV
lvr_3_ov	lvr_3	3.1268 V	3.2738 V	3.1268 V	3.2738 V	0 V	0 V
lvr_3_uv_n	lvr_3	2.736 V	2.867 V	2.736 V	2.867 V	0 V	0 V
lvr_5_ov	lvr_5	5.1369 V	5.3724 V	5.1369 V	5.3724 V	0 V	0 V
lvr_5_uv_n	lvr_5	4.6468 V	4.8623 V	4.6468 V	4.8623 V	0 V	0 V

Figure 10.1.: Results of the threshold voltage analysis.

Figure 10.1 shows the results of the threshold and hysteresis voltage analysis. If any of the calculated minimum and maximum voltages were outside of the specification window, the respective cell would have a red background. Apparently, all voltages are within the specification windows, which can be manually verified by comparing with the specification tables. In case of specification violations there are multiple points of action: adjusting the channel input gain, improving of ADC or bandgap reference parameters, or expanding of specification windows.

A summary of the channel properties is given in Fig. 10.2. The dummy channel has a priority of two, which means that on average two clock period delays are inserted per cycle where each channel occurs as often as its priority dictates. The two channels for high voltages also have a priority of two, which makes them dominant for the total cycle time due to the slower channel type. The channels for the LVR voltages have a lower priority as their deglitch times are comparatively longer than the lowest times of the other channels. The total of priorities is 8, which means that $\log_2(8) = 3$ LSBs of the LFSR are used and mapped to one of the four

output_nr	domain	signal_name	node_name	threshold (V)						deglitch (µs)			
				rising		falling		hysteresis		rising		falling	
				min	max	min	max	min	max	min	max	min	max
0	startup	lvr_3_uv_n	lvr_3	2.7	2.9			0	0	60	140		
1	startup	lvr_3_ov	lvr_3	3.1	3.3			0	0	60	140		
2	startup	lvr_5_uv_n	lvr_5	4.6	4.9			0	0	60	140		
3	startup	lvr_5_ov	lvr_5	5.1	5.4			0	0	60	140		
4	normal	boost_35_uv_n	boost_35	32.5	34.3	31.7	33.5	0.3	1	60	140	40	80
5	normal	boost_35_ov	boost_35	36.5	38.3	35.7	37.5	0.3	1	40	80	60	140
6	normal	buck_25_uv_n	buck_25	23	24.5	22.5	24	0.3	1	60	140	40	80
7	normal	buck_25_ov	buck_25	26	27.5	25.5	27	0.3	1	40	80	60	140

clock_name	clock_freq_logic_mhz	clock_freq_adc_mhz	clock_tolerance_percent
CLK_normal	40	40	2

adc_name	num_bits	caparray	fast_compare_clocks	error_offset_lsb	error_gain	error_inl
ADC10	10	1024	3	0.5	1	1

Table 10.5.: Output signals, clock and ADC specifications.

channel name	#outputs	priority	sampling+conversions time	FSR
dummy	0	2/8	50 ns	
boost_35	2	2/8	1.35 us	38.88 V
buck_25	2	2/8	1.35 us	27.047 V
lvr_3	2	1/8	375 ns	4.86 V
lvr_5	2	1/8	375 ns	6.48 V
ALL CHANNELS	8	8/8	3.5 us	

Figure 10.2.: Channel properties summary.

output name	channel	priority	avg. freq.	#rising	#falling	rising min	rising avg	rising max	falling min	falling avg	falling max
boost_35_ov	boost_35	2	571.4286 kHz	34	57	49.0196 us	58.6119 us	69.3367 us	90 us	98.8529 us	107.2704 us
boost_35_uv_n	boost_35	2	571.4286 kHz	57	34	90 us	98.8529 us	107.2704 us	49.0196 us	58.6119 us	69.3367 us
buck_25_ov	buck_25	2	571.4286 kHz	34	57	52.1078 us	58.6119 us	65.6633 us	92.1078 us	98.8529 us	105.7398 us
buck_25_uv_n	buck_25	2	571.4286 kHz	57	34	92.1078 us	98.8529 us	105.7398 us	52.1078 us	58.6119 us	65.6633 us
lvr_3_ov	lvr_3	1	285.7143 kHz	29	29	78.6275 us	99.7277 us	111.0459 us	78.6275 us	99.7277 us	111.0459 us
lvr_3_uv_n	lvr_3	1	285.7143 kHz	29	29	78.6275 us	99.7277 us	111.0459 us	78.6275 us	99.7277 us	111.0459 us
lvr_5_ov	lvr_5	1	285.7143 kHz	29	29	89.1667 us	99.7277 us	108.8265 us	89.1667 us	99.7277 us	108.8265 us
lvr_5_uv_n	lvr_5	1	285.7143 kHz	29	29	89.1667 us	99.7277 us	108.8265 us	89.1667 us	99.7277 us	108.8265 us

Figure 10.3.: Results of deglitch times analysis.

channels according to the priorities.

The deglitch time analysis summary is depicted in Fig. 10.3. The absence of red cells means that there are no violations of the specifications. For the outputs on the two LVR channels the number of rising and falling deglitch filter lengths is the same. The other outputs have different rising and falling deglitch filter lengths. It is important to note that the minimum deglitch filter length on all outputs is around 30. This compromise on a minimum deglitch filter length is important to achieve similar minimum aliasing immunities for all outputs.

Figure 10.4 visualizes the deglitch time variation of the rising and falling deglitch time for one output signal. According to Fig. 10.3 the rising deglitch filter length for the output signal `boost_35_ov` is 34 and the falling deglitch filter length is 57. The longer deglitch filter length leads to a smoother deglitch curve with less variation, as the sampling function variations are smaller in relation to the filter length. The minimum curve shows the case with the maximum clock frequency and the latest possible occurence of the overvoltage in the interval between sampling time instants. In contrast, the maximum deglitch time is due to the lowest clock frequency and the assumption that the overvoltage occurs at the earliest possible time in between the sampling time instants. In this example the LFSR has 8 bit and hence the maximum length sequence is $2^8 - 1 = 255$. Each of the 3-bit output states occurs $256/8 = 32$ times, only the state `000` occurs one time less. The channel `boost_35` has a priority of 2 and hence in one period of the LFSR it is sampled 64 times (or 63 times if the state `000` maps to it, but here this is not the case). Therefore, in the deglitch times plot, 64 distinct rising and falling deglitch times are shown with their minimum and maximum values. The average of the sequences are shown in Fig. 10.3 – they match the targeted times almost exactly.

Finally, the tool performs the aliasing analysis for all output signals according to section 7.4.4.3 with $\omega \sim 2\pi \cdot \mathcal{U}(150\,\text{kHz}, 1\,\text{GHz})$. The used number of Monte Carlo runs per channel is 10^6. In every run the deglitch filter time is calculated over a whole LFSR period. Hence, for channel `boost_35` there are 64 new values for the histogram, where each represents the

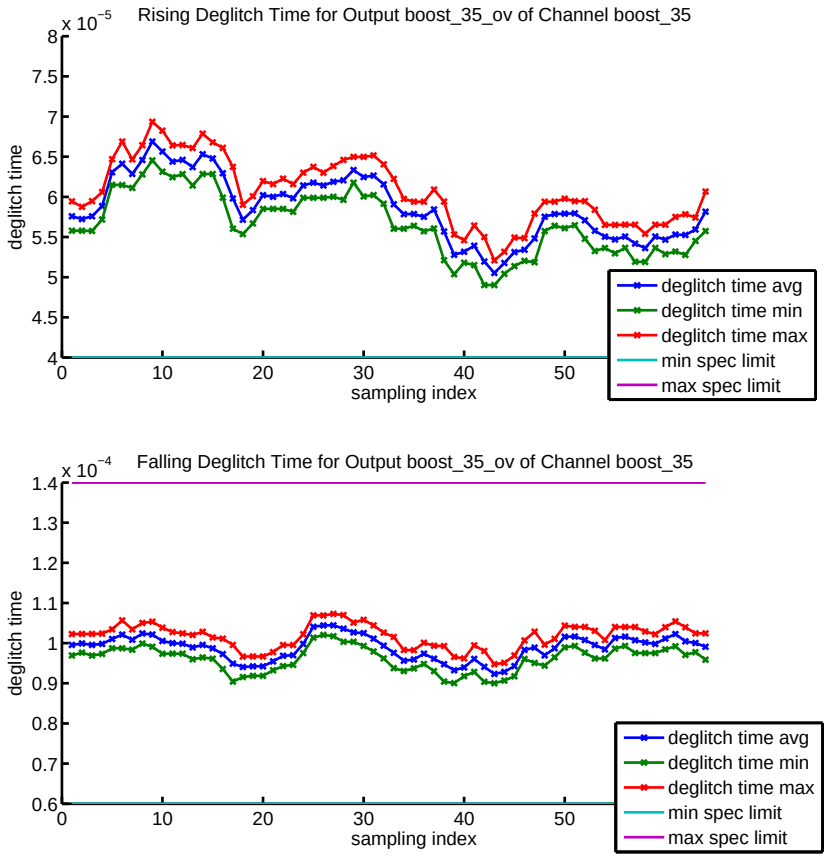

Figure 10.4.: Deglitch time variation and specification windows for the output signal `boost_35_ov`.

number of samples which are above the relative threshold voltage of zero. Therefore, the total number of occurrences in the histogram is $64 \cdot 10^6$ for the rising and separately for the falling deglitch time analysis. Figure 10.5 shows the histogram plot for the Monte Carlo analysis. The rising deglitch filter length is 34 and the mean of the number of high samples is half of this. Aliasing would happen if the analysis of the rising decision would result in the maximum of 34 samples above the threshold. Obviously, this does not happen in the 10^6 runs. For the falling decision the filter length is even longer, which makes it more improbable that aliasing occurs due to all samples being below the threshold.

10.3. Simulation

The previous section shows the synthesis and theoretical analysis of the exemplary voltage monitoring system. This section shows the digital simulation of the model.

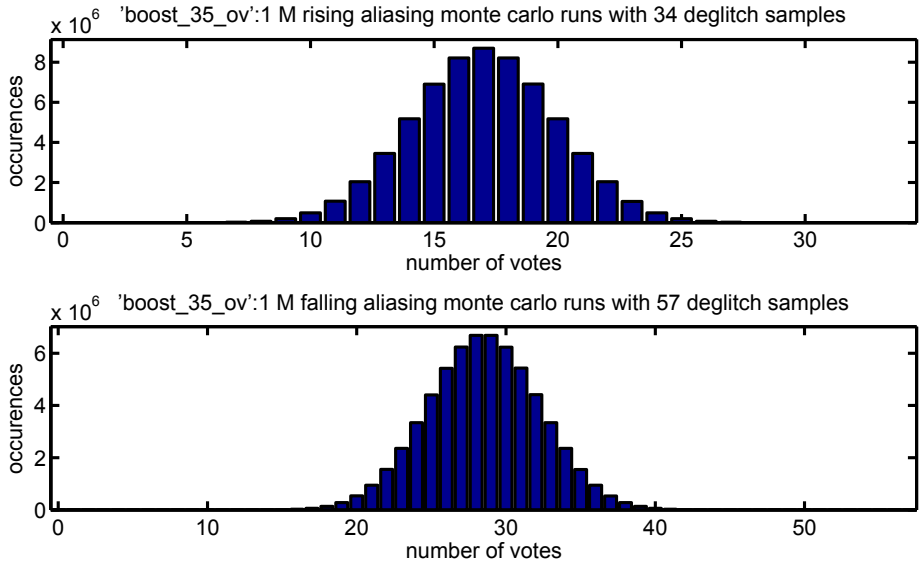

Figure 10.5.: Results of aliasing Monte Carlo simulation with 10^6 runs.

10.3.1. Random Scheduler

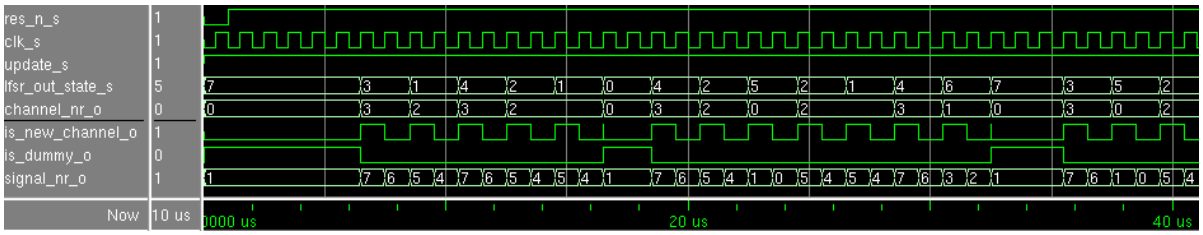

Figure 10.6.: Simulation of random scheduler in a separate test bench.

The random scheduler is simulated separately in a test bench to verify its functionality. Figure 6.34 shows the result when the update signal is constantly applied. The LFSR is initialized with all ones, which is why for the first few updates the 3-bit output state is always 7. This state is mapped to the dummy channel. For each channel both associated output signal numbers occur in sequence. It can also be seen that channel numbers 2 and 3 have a priority of two, because two different LFSR output states map to them.

10.3.2. ADC Control

Figure 10.7 displays the inputs and outputs of the ADC control block at an arbitrary point in time. The random scheduler is updated with each end of conversion represented by `eoc_i`. Then either a dummy channel occurs and the next update follows immediately, or a valid channel and signal number combination is obtained. If the obtained signal is on the same channel, the conversion is performed directly. If a new channel was obtained, the sampling is started with a pulse on `sos_o`. This also happens when the new channel is the same as before.

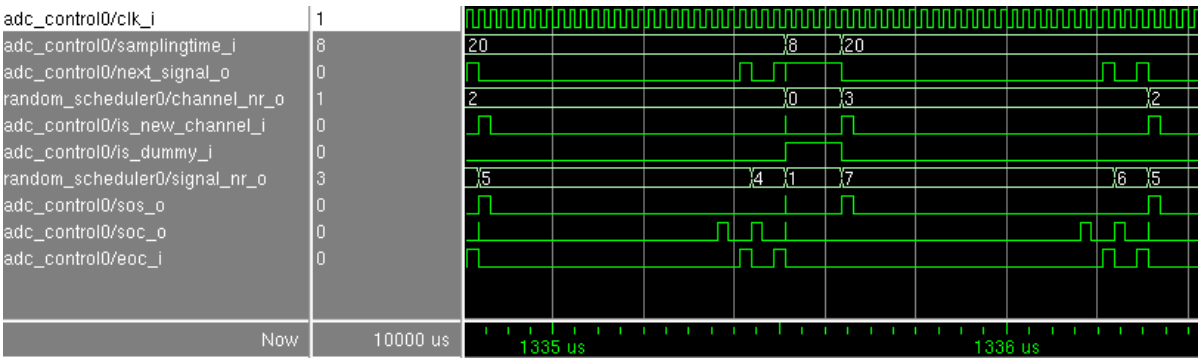

Figure 10.7.: Behavior of the ADC control block.

After the correct sampling time, the `soc_o` signal starts a fast compare conversion with the provided threshold code.

10.3.3. Ideal Environment

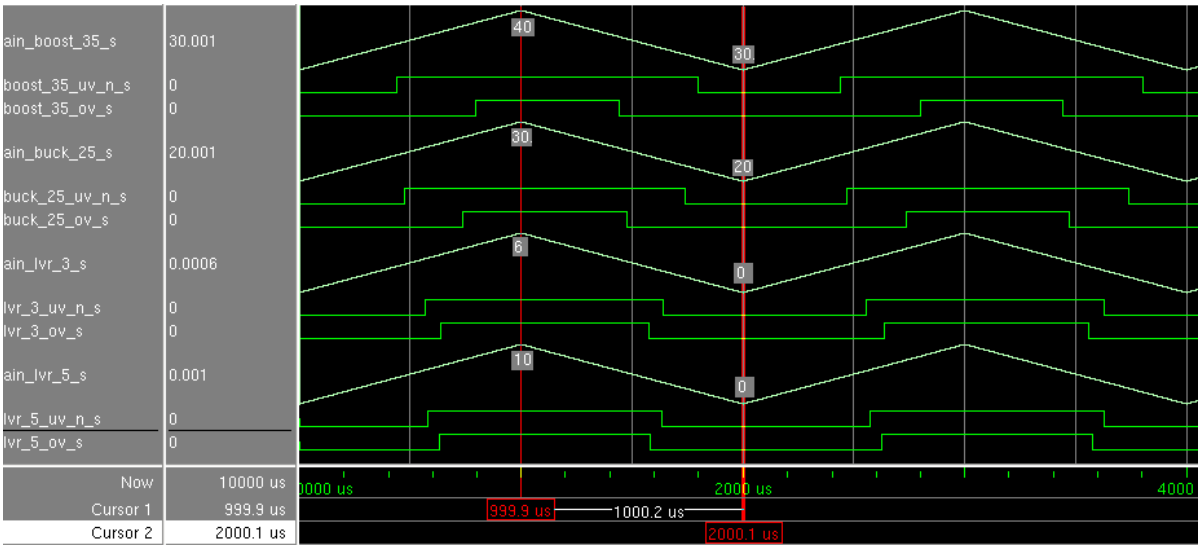

Figure 10.8.: Simulation of the complete voltage monitoring system with ramp signals applied to all channels.

In the ideal environment there are no disturbance signals. Triangle wave signals are applied to the channels to allow measuring of the threshold voltages, hysteresis voltages and the deglitch times. Figure 10.8 depicts the top-level simulation with marked minimum and maximum voltages of the applied triangle waves.

A detailed measurement of the deglitch times is shown in Fig. 10.9 for the output `boost_35_ov`. The signal `count_s` is the value of the counter inside the deglitch filter. The counter starts to increase when the input signal has the voltage 37.411 V, which is the first sample above the threshold voltage. When the 34th sample in a row is above the threshold, the output `boost_35_ov_o` is set high and the counter is reset. The time between the first sample above

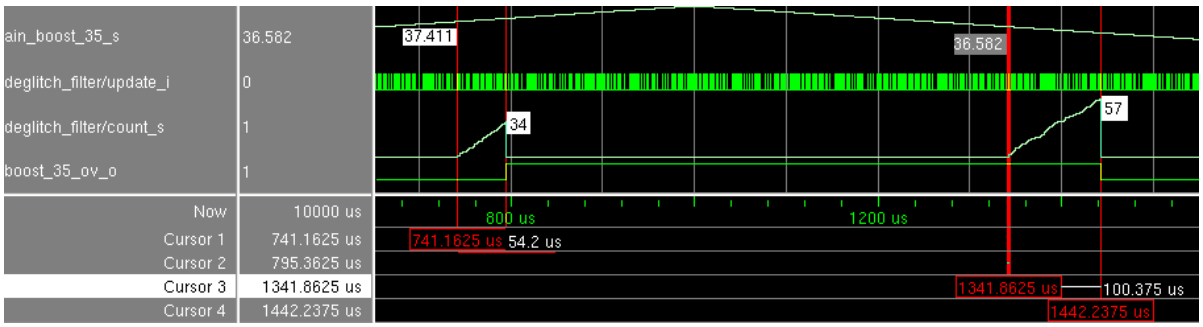

Figure 10.9.: Deglitch times for output `boost_35_ov`.

the threshold and the switching of the output signal is the deglitch time. Here it is 54.2 µs, which is inside the specification window. Depending on the LFSR state when the voltage is above the threshold for the first time, the deglitch time varies as explained in section 7.3.5. The falling deglitch time of 100.375 µs is also within the specification. The falling threshold voltage of 36.582 V results in a calculated hysteresis of 892 mV.

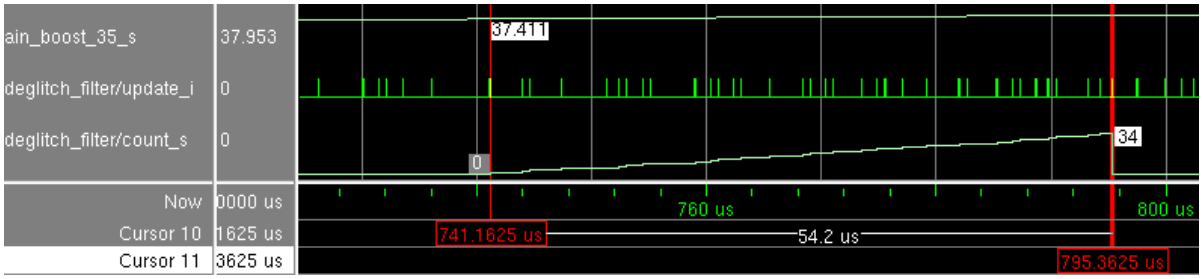

Figure 10.10.: Rising deglitch time for output `boost_35_ov`.

An enlarged view of the same rising deglitch time is provided in Fig. 10.10. The update signal of the deglitch filter shows when a new fast compare conversion results is added to the filter. However, this should not be interpreted as the sampling time instant, which occurs a little bit earlier when the sampling switch is closed before the conversion. The input voltage crosses the ideal threshold voltage somewhere before the sampling instant corresponding with the update impulse at 741.1635 µs and the previous one. This ambiguity cannot be resolved due to the sampled nature of the system, but the introduced additional deglitch time error is small compared to the whole deglitch time. In section 7.3.5.1 the calculations of the minimum and maximum worst-case deglitch time take this already into account.

10.3.4. Disturbance Test Signals

The previous section shows the simulation results for ideal environments without disturbances. This section shows the behavior of the voltage monitoring system in the presence of sinusoidal disturbances at the example of the output signal `boost_35_ov`.

Figure 10.11 shows the output signal for a DC input voltage of 37.4 V. This voltage is the exact rising threshold voltage of the output. The disturbance signal is a superposed sinusoidal with 2 V amplitude and a frequency of 10 MHz. The expected behavior is that no change of

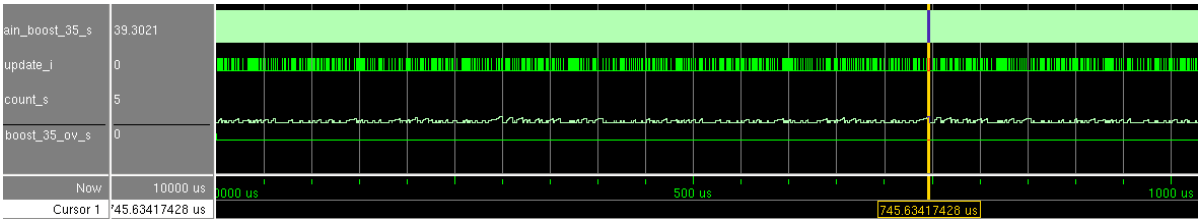

Figure 10.11.: Simulation with sinusoidal disturbance signal with 10 MHz.

the output signal should happen, as with an ideal random sampling system the disturbance prevents a unanimous decision.

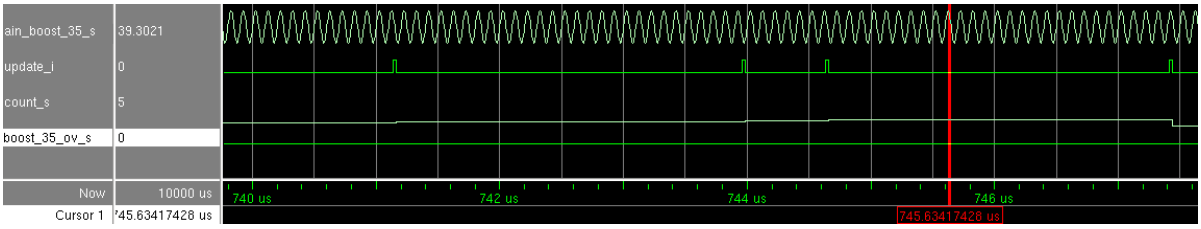

Figure 10.12.: Enlarged view of the simulation with a sinusoidal disturbance with 10 MHz.

In the simulation it is visible that the output signal remains at zero. The inspection of the counter value inside the deglitch filter reveals that the counter is reset very early and it never gets close to its maximum value of 34, which would set the output high. An enlarged view where the input voltage is clearly identifiable is given in Fig. 10.12. As mentioned in the previous section, the filter update pulses do not correspond to the sampling time instants.

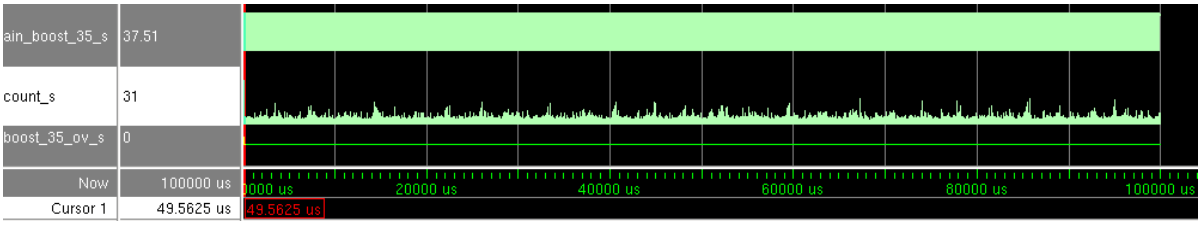

Figure 10.13.: Simulation with a linear chirp signal from 0 Hz to 20 MHz as disturbance.

The functional ADC model used in all shown simulations does not implement the analog jitter function. Therefore, the resistance against aliasing is only given up to half of the system clock of 40 MHz. Based on that, the aliasing immunity of the system is analyzed using the same setup as before, but now a linear frequency sweep of the disturbance signal from 0 Hz to 20 MHz within 100 ms is used. Figure 10.13 points out that the output signal shows no inadvertent aliasing-related switching. The value of the counter inside the deglitch filter shows behavior similar to noise. Except for an initial pulse, it never gets close to the filter length value of 34, which would cause a change of the output.

An enlarged view of the initial time period is provided in Fig. 10.14. The start of the chirp signal is slow and hence the voltage is initially above the threshold voltage for only a little less than the rising deglitch time. When the frequency increases, the maximum counter values

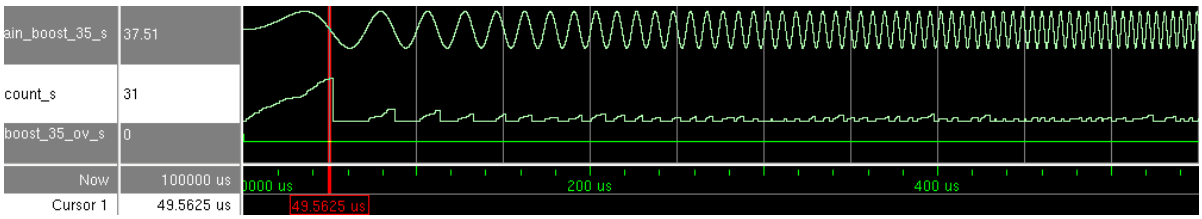

Figure 10.14.: Beginning of the chirp signal.

decrease more and more. In summary, it can be stated that for the analyzed disturbances no aliasing occurs on the output under test. However, for a specific configuration of a concrete voltage monitoring system all outputs must be examined. This is done by means of a Monte Carlo simulation in the automated workflow for constant disturbance frequencies. The simulation of the chirp signal in this section exemplifies that the results of the anti-alasing analysis for constant frequency signals permits to assume also useful anti-aliasing properties for arbitrary signals as long as they are not correlated with the actual sampling sequences of the channels.

11. Conclusion and Outlook

11.1. Conclusion

The new concept for a voltage monitoring system is based on an analog-to-digital converter with an input multiplexer. The modularity enables an automated design-flow for the digital part on the basis of a specification. The multiplexed random sampling method permits to process non-bandlimited signals, which saves silicon area for the otherwise required anti-aliasing filters. In comparison with the old voltage monitoring system based on distributed comparators, the new concept provides better testability due to the developed ADC BIST method. Not only does it allow faster production tests, but it also enables the verification of the ADC functionality during power-up and even during operation. Hence, a high diagnostic coverage of latent faults in the monitoring system can be achieved, which is important for applications in automotive environments.

However, the application of the ADC BIST method is not restricted to the voltage monitoring system. It can be used in any capacitive charge redistribution SAR ADC to reduce test costs or even enable new applications where a high diagnostic coverage is needed. One of the most outstanding characteristics of the presented BIST method is the little area overhead, as only an additional discharging capacitor is required in the analog circuit. The accuracy and measurement time trade-off can be configured by choosing different discharging capacitors. Furthermore, it is assessed whether the bottom-plate switches and the internal comparator are functional. For an on-chip test, a simple method empowers to check if the mismatch is within specified bounds. A more sophisticated method allows estimating the weights of the capacitor array accurately. This requires computational power that can be provided by an on-chip processing unit or by external equipment such as an ATE.

A multiplexed random sampling method permits to operate on input signals without anti-aliasing filters. A linear feedback shift register operates as a pseudo-random number generator to select channels, which makes it possible to guarantee certain deglitch filter times deterministically. By means of a Monte Carlo analysis it can be determined if a given voltage monitoring system provides sufficient immunity against inadvertent output states due to aliasing.

The test costs of integrated circuits represent a significant share of their total costs. Therefore, the improved testability of the new voltage monitoring system alone can justify its use. If the new voltage monitoring system requires less silicon area than a comparator-based system cannot be answered in general. However, the qualitative relationship is depicted in Fig. 11.1. In the comparator based system, there is no fixed area cost. In the ADC based voltage monitoring system, the ADC core and the digital ADC control block account for fixed area costs. Additional channels only require additional sampling switches. Multiple outputs on a channel are almost for free, as they only require more digital logic. For given component

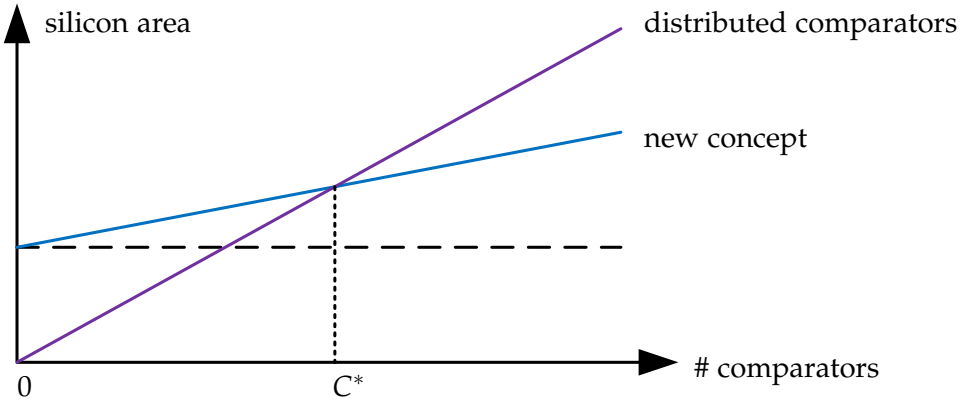

Figure 11.1.: Silicon area comparison of old and new voltage monitoring concept.

sizes there is a number of comparators where both systems require the same area (break-even point). For more comparators, the new voltage monitoring system requires comparatively less area. However, the new voltage monitoring system cannot emulate an arbitrary number of comparators. Depending on the concrete sampling and conversion speeds of the ADC, the maximum possible number of channels and outputs is limited. The main reason is the aliasing immunity, which decreases with decreasing deglitch filter lengths. Therefore, a voltage monitoring system with a large number of output signals on different channels can be feasible if all deglitch times are long enough such that the deglitch filter lengths provide sufficient aliasing immunity. Another option is to split a system into two independent ADC-based voltage monitoring systems.

All these considerations show that for each application the feasibility of the new voltage monitoring system must be analyzed individually. The key factors are the achievable performance in the amplitude and time domain, the testability, and the required silicon area.

11.2. Outlook and Future Work

The voltage monitoring concept can substitute comparators not only in the power supply system of an SoC, but also in other applications. However, if no high-voltage signals have to be processed, then the size of anti-aliasing filters might be tolerable. In such a case the random-sampling approach is not required and regular uniform sampling can be used, which saves the random scheduler and eliminates deglitch time variations.

The used pseudo-random number generation method is based on a linear feedback shift register with an output filter. Even though the achievable aliasing immunity seems sufficient, it could be investigated if other methods provide even better results. The aliasing immunity analysis for sinusoidal disturbance signal is based on a Monte-Carlo simulation. The analysis could be extended to other classes of signals. Furthermore, it could be examined if a closed expression for calculating the anti-aliasing properties of a sampling sequence can be found. This would save the effort to perform Monte Carlo simulations for analyzing the aliasing immunity.

At the end of this thesis a digital simulation of a voltage monitoring system is presented. However, the used ADC model is only a functional model without the analog clock jitter. Therefore, measurements of a real implementation of the whole system must be used to verify the theoretical results.

The ADC BIST method where the exact weights are estimated is based on a nonlinear equation. So far, this equation is solved by the iterative Newton's method. It can be examined if better estimators exist, especially when the time measurement uncertainty for each bit weight is known. The assumption for the BIST simulation was that the noise is white with a Gaussian distribution. Therefore, other noise models such as flicker noise should be analyzed, as it is dominant in semiconductors. The considered filtering approaches all use the comparator output signal while the discharging process goes on. Therefore, the filtering introduces delays which distort the threshold crossing times. An alternative would be to repeat a comparator measurement without discharging in between. Hence, if the comparator detects a threshold crossing, the discharging is stopped and a short median filter could be used to determine if the threshold has indeed been reached. This method would increase the measurement time, which could be critical regarding leakage currents that influence the charge. Consequently, it is only feasible for very low numbers of hits per weight.

For the ADC BIST only concept simulations have been performed. There is a large gap between the assumptions in the simulation and the parasitic effects in a real ADC. For instance, leakage currents could distort the charge on the capacitor array and influence the threshold crossing time measurement results. Therefore, the feasibility of the BIST method must still be verified in a real implementation.

Part IV.

Appendix

Glossary

2oo3 two out of three.

ADC analog-to-digital converter.

ARS additive random sampling.

ATE automated test equipment.

AWGN additive white Gaussian noise.

BIST built-in self-test.

CDF cumulative density function.

CW continuous wave.

DAC digital-to-analog converter.

DC direct current.

DFT discrete Fourier transform.

DNL differential nonlinearity.

DPI direct power injection.

DUT device under test.

EMC electromagnetic compatibility.

EMI electromagnetic interference.

FIR finite impulse response.

FSR full scale range.

GCD greatest common divisor.

i.i.d. independent and identically distributed.

IC integrated circuit.

INL integral nonlinearity.

JRS jittered random sampling.

LBIST logic built-in self-test.

LFSR linear feedback shift register.

LSB least significant bit.

LUT lookup table.

LVR linear voltage regulator.

MATLAB Matrix Laboratory.

MSB most significant bit.

PDF probability density function.

PMF probability mass function.

PRNG pseudo-random noise generator.

RF radio frequency.

RSS random skip sampling.

SAR successive approximation register.

SNR signal-to-noise ratio.

SoC system on chip.

VHDL Very High Speed Integrated Circuit Hardware Description Language.

Bibliography

[1] F. Azais et al. "A low-cost adaptive ramp generator for analog BIST applications". In: *VLSI Test Symposium, 19th IEEE Proceedings on. VTS 2001*. 2001, pp. 266–271. DOI: 10.1109/VTS.2001.923449.

[2] Hans-Jochen Bartsch. *Kleine Formelsammlung Mathematik*. Carl Hanser Verlag, 2007. ISBN: 9783446411135.

[3] I. Bilinskis. *Digital Alias-free Signal Processing*. Wiley, 2007. ISBN: 9780470511626.

[4] P. Bogner and F. Kuttner. "Measurement Apparatus". Pat. US 2011/0254569 A1. Oct. 20, 2011.

[5] Daniel Carrica, Mario Benedetti, and Roberto Petrocelli. "Random Sampling Applied to the Measurement of a DC Signal Immersed in Noise". In: *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT* 50.5 (Oct. 2001), pp. 1319–1323.

[6] An-Sheng Chao et al. "A Capacitance-Ratio Quantification Design for Linearity Test in Differential Top-Plate Sampling SAR ADCs". In: *International Journal of Circuit Theory and Applications* (2014).

[7] Hung kai Chen, Chih-Hu Wang, and Chau-Chin Su. "A self calibrated ADC BIST methodology". In: *VLSI Test Symposium, 2002. (VTS 2002). Proceedings 20th IEEE*. 2002, pp. 117–122. DOI: 10.1109/VTS.2002.1011121.

[8] International Electrotechnical Commission. *International Standard IEC 62132-4, Integrated circuits – Measurement of electromagnetic immunity 150 kHz to 1 GHz – Part 4: Direct RF power injection method*. Feb. 2006.

[9] Xuan-Lun Huang et al. "A self-testing and calibration method for embedded successive approximation register ADC". In: *Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific*. Jan. 2011, pp. 713–718. DOI: 10.1109/ASPDAC.2011.5722279.

[10] Xuan-Lun Huang et al. "Testing and Calibration of SAR ADCs by MCT-Based Bit Weight Extraction". In: *Mixed-Signals, Sensors and Systems Test Workshop (IMS3TW), 2012 18th International*. May 2012, pp. 1–4. DOI: 10.1109/IMS3TW.2012.11.

[11] IEEE Instrumentation and Measurement Society. "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters". In: *IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000)* (Jan. 2011), pp. 1–139. DOI: 10.1109/IEEESTD.2011.5692956.

[12] C. Luo. *Non-Uniform Sampling: Algorithms and Architectures*. PhD Thesis. Georgia Institute of Technology, Dec. 2012.

[13] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Oct. 1996. ISBN: 0849385237.

[14] A.V. Oppenheim and R.W. Schafer. *Discrete-time Signal Processing*. 3rd international. Pearson Prentice Hall, 2010. ISBN: 9780132067096.

[15] A. Papoulis and S.U. Pillai. *Probability, Random Variables, and Stochastic Processes*. 4th ed. McGraw-Hill series in Electrical and Computer Engineering. McGraw-Hill, 2002. ISBN: 978007122615.

[16] R. Sarpeshkar, T. Delbruck, and C.A. Mead. "White noise in MOS transistors and resistors". In: *Circuits and Devices Magazine, IEEE* 9.6 (Nov. 1993), pp. 23–29. ISSN: 8755-3996. DOI: 10.1109/101.261888.

[17] A. Schwab and W. Kürner. *Elektromagnetische Verträglichkeit*. 6th. Springer-Verlag, 2011.

[18] International Organization for Standardization. *ISO 26262, Road vehicles – Functional safety – Part 1: Vocabulary*. 2011.

[19] U. Tietze, C. Schenk, and E. Gamm. *Halbleiter-Schaltungstechnik*. 13th ed. Springer-Verlag, 2010. ISBN: 9783642016219.

[20] Widrow, B. and Kollar, I. and Liu, M.-C. "Statistical Theory of Quantization". In: *IEEE Transactions on instrumentation and measurement* (Apr. 1996), pp. 353–361.