Martin Ebetshuber, BSc

Programmable capacitor for pressure MEMS emulation

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Electrical Engineering

submitted to

Graz University of Technology

Supervisor

Mario Auer, Assoc.Prof. Dipl.-Ing. Dr.techn.
Institute of Electrical Measurement and Sensor Systems

Stefan Gansinger, Dipl.-Ing. Dr.techn.
Infineon Austria

Graz, February 2024

Abstract

The verification of an ASIC (application-specific integrated circuit), which measures a capacitive MEMS pressure sensor, has stringent requirements regarding the accurate setting as well as the linearity of the pressure to be measured. Since the laboratory environment shows limitations with respect to the accuracy of the pressure as well as noise, the capacitive pressure sensor should be emulated by an integrated programmable capacitor.

Due to the high requirement on linearity, different integrated capacitors are investigated with respect to their matching behaviour. Also different models on how to weigh those capacitors in capacitor-arrays are created and simulated. After that, several circuits are derived that interconnect capacitor-arrays while staying close to the configuration of the MEMS pressure sensor. Simulation results indicate that an attenuated full-bridge with thermometer-coded capacitor-arrays shows the best behaviour regarding linearity. The programmable capacitor can be controlled by a microcontroller that shifts bits into chip, depending on the programmed value. Subsequently, this work shows the implementation of the attenuated full-bridge along with the required logic blocks and the creation of a layout of the programmable capacitor.

Final top-level simulations of the extracted layout verify that the specifications of the programmable capacitor are met.

Kurzfassung

Die Verifizierung eines ASIC (Application-Specific-Integrated-Circuit), welcher einen kapazitiven MEMS Drucksensor misst, erfordert eine äußerst genaue Einstellung und Linearität des zu messenden Druckes. Da die Laborumgebung, durch den begrenzt genau einstellbaren Druck und des Rauschens, eine Limitierung aufweist, soll der kapazitive Durcksensor durch einen integrierten programmierbaren Kondensator emuliert werden. Bedingt durch die hohe Anforderung an die Linearität, werden verschiedene integrierte Kondensatoren hinsichtlich ihres matching-Verhaltens untersucht. Zusätzlich werden Modelle für die Gewichtung von Kondensatoren in Kondensator-Arrays erstellt und simuliert. Anschließend werden verschiedene Schaltungen untersucht, welche die Kondensator-Arrays verschalten und ähnlich wie der MEMS Drucksensor aufgebaut sind. Simulationsergebnisse zeigen dass eine gedämpfte Vollbrücke mit thermometer-kodierten Kondensatoren die besten Eigenschaften hinsichtlich Linearität aufweist. Der programmierbare Kondensator kann mit einem Microcontroller angesteuert werden, welcher die Bits, abhängig vom programmierten Wert, in den programmierbaren Kondensator schiebt.

Die Arbeit zeigt anschließend die Implementierung der gedämpften Vollbrücke und die Logik des Chips sowie die Erstellung eines Layouts des programmierbaren Kondensators. Abschließend verifizieren Simulationen mit dem extrahierten Layout die korrekte Funktionsweise und die erreichte Spezifikation des programmierbaren Kondensators.

Acknowledgements

First of all, I want to express my deepest gratitude to my supervisors, Mario Auer from Graz University of Technology and Stefan Gansinger from Infineon Austria, for their unlimited support in all technical and academical matters throughout the whole Master Thesis.

I further want to thank Stefan Mendel from Infineon Austria for giving me the opportunity to do my Master Thesis in cooperation with Infineon, which turned out as an excellent decision.

Additionally, a big thank you to my colleagues, who not only supported me with the top-level layout, but also gave me ideas and hints on design and simulations.

Last but not least, I want to express my gratitude to my family and friends who supported me apart from the Master Thesis and always had my back. A special thank you to my parents, who gave me the opportunity and resources to study in this interesting field of science.

Contents

1. Introduction **12**
1.1. Motivation 12
1.2. Pressure Sensing Circuit to be Emulated 13
1.2.1. Current Setup 13
1.2.2. Sensing Circuit Read-Out Principle 15
1.3. Specification for Programmable Capacitor 16

2. Structure and Modelling of Programmable Capacitors **17**
2.1. Integrated Capacitors 17
2.1.1. Types of Integrated Capacitors 17
2.1.2. Mismatch of Integrated Capacitors 20
2.2. Capacitor Weighting 24
2.2.1. Binary-Weighted Capacitor-Array 25
2.2.2. Thermometer-Coded Capacitor-Array 27
2.2.3. Hybrid-Coded Capacitor-Array 29
2.3. Applications of State-of-the-Art Capacitor-Arrays 31
2.3.1. Capacitor-Arrays in DACs 32
2.3.2. Other Applications 33

3. Concept for Sensor Emulation **35**
3.1. Bridge Configuration 35
3.1.1. Half-Bridge Configuration 35
3.1.2. Attenuated Half-Bridge Configuration 36
3.1.3. Full-Bridge Configuration 37
3.1.4. Attenuated Full-Bridge Configuration 39
3.2. Final Bridge Concept Decision 42
3.2.1. Chosen Capacitor Weighting 42
3.2.2. Chosen Bridge Configuration 42
3.2.3. Final Decision on Unit-Capacitor Type and Unit-Capacitance 47
3.2.4. Design Trade-Off 50
3.3. Programming Concept 50
3.3.1. Addressing of Capacitors 51
3.3.2. Binary-to-Thermometer Encoder 51

4. Implementation of the Programmable Capacitor **53**
4.1. Types of Implemented Programmable Capacitors 53

4.2. Top-Level Configuration 53
4.2.1. Block Diagram 53
4.3. Capacitor-Array Configuration 54
4.3.1. Block Diagram 54
4.3.2. Clamping Diodes at Floating Nodes 56
4.3.3. Control Signals and Addressing 57
4.4. Capacitor-Switch Cell (CS-Cell) 59
4.4.1. Schematic of the Ideal CS-Cell 59
4.4.2. Schematic of the Implemented CS-Cell 60
4.4.3. Capacitor and Switches 60
4.4.4. Control Logic 62
4.5. Dummy-Cell 62
4.5.1. Description 62
4.5.2. Consequences for Attenuation Capacitors 63
4.6. Bridge Logic 64
4.6.1. Column- and Row-Bridge-Logic 65
4.6.2. Row_set-Logic 67
4.7. Layout of Programmable Capacitor 67
4.7.1. Design Rule Issue and Adjustment of C_u 68
4.7.2. Parasitic Extraction and Redesign of C_s 68

5. Final Simulation and Verification 70
5.1. Final Simulation Setup 70
5.1.1. Block: Reference Modulator 70
5.1.2. Block: Programmable Capacitor 70
5.1.3. Block: PGA 71
5.1.4. Block: ADC 71
5.1.5. Calculation of $\mathrm{C_{eq}}$ from the Simulation Results 71
5.2. Programming Signals 72
5.3. Simulation Results 73

6. Conclusion and Outlook 76

A. Appendix 79
A.1. AC Signal Derivations of Bridge-Configurations 79
A.1.1. Attenuated Half-Bridge 79
A.1.2. Full-Bridge Configuration 81
A.1.3. Ideal Attenuated Full-Bridge Configuration 81
A.1.4. Real Attenuated Full-Bridge Configuration 83
A.1.5. Parasitic Capacitances in Attenuated Full-Bridge 84
A.2. Logic Functions of Binary-to-Thermometer Encoder 86
A.3. Schematic of Binary-to-Thermometer Encoder 86
A.4. MATLAB Model 88

A.5. Layout 90
A.5.1. Layout of CS-Cell 90
A.5.2. Layout of Smaller Programmable Capacitor (625 CS-cells per Branch) 91
A.5.3. Layout of Larger Programmable Capacitor (1600 CS-cells per Branch) 92
A.5.4. Layout of Top-Level Programmable Capacitor 93

Bibliography **94**

List of Figures

1.1. MEMS Pressure Sensor Principle 12
1.2. Capacitive Bridge Readout 14
1.3. AC signal derivation for both branches of the MEMS pressure sensor. Each branch is investigated separately by splitting up both the branches and the PGA. 15

2.1. Example of a VPP capacitor 18
2.2. Example of a Poly-Poly capacitor 18
2.3. Example of a MOM capacitor 19
2.4. Example of a MIM capacitor 20
2.5. Relative standard deviation of vertical MOM capacitors 23
2.6. Binary-Weighted Capacitor-Array 25
2.7. Linearity of Binary-Weighted Capacitor-Array 26
2.8. Thermometer-coded Capacitor-Array 27
2.9. Linearity of Thermometer-Coded Capacitor-Array 28
2.10. Hybrid-Coded Capacitor-Array 30
2.11. Linearity of a Hybrid-Coded Capacitor-Array 31
2.12. Principle of a SAR-ADC 32

3.1. Capacitive Half-Bridge 36
3.2. Attenuated capacitive Half-Bridge 37
3.3. Capacitive Full-Bridge 38
3.4. Attenuated capacitive Full-Bridge 39
3.5. Attenuated capacitive Full-Bridge (realistic) 41
3.6. Entangled switching scheme including both branches 43
3.7. Dominant Parasitics in Capacitor-Array 45
3.8. Graphical Concept Result Representation 49
3.9. Programming Logic Concept 52

4.1. Bridge Configuration 54
4.2. Block diagram of implemented matrix configuration of single cells 55
4.3. Clamping Diodes at Floating Nodes 56
4.4. Schematic of ideal CS-cell 59
4.5. Schematic of implemented cap-switch-cell 60
4.6. Schematic of Implemented Dummy-Cell 63
4.7. Schematic of the implemented 6 bit shift register 65
4.8. Schematic of Row_set Logic 67

5.1. Final simulation setup 70
5.2. Simulation Input Example 72
5.3. Simulation Result of C_{eq} 74
5.4. Simulation Result for DNL(C_{eq}) 74

A.1. AC signal derivation for both branches of the attenuated half-bridge. . . . 79
A.2. AC signal derivation for both branches of the full-bridge. 81
A.3. AC signal derivation for both branches of the ideal attenuated full-bridge. 82
A.4. AC signal derivation for both branches of the real attenuated full-bridge. . 83
A.5. AC signal derivation for both branches of the attenuated full-bridge including parasitic capacitances 84
A.6. Schematic of Binary-to-Thermometer Encoder 87
A.7. Layout of CS-Cell 90
A.8. Layout of Smaller Programmable Capacitor (625 CS-cells) 91
A.9. Layout of Larger Programmable Capacitor (1600 CS-cells) 92
A.10.Layout of Top-Level Programmable Capacitor 93

List of Tables

1.1. Specifications for Programmable Capacitor 16

2.1. Capacitor Mismatch Data from Literature 24

3.1. Conceptional Results of Bridge Configurations 42
3.2. Concept Simulation Results 48

4.1. Control Logic of Single Cell 57
4.2. Control Logic of CS-Cell 62
4.3. Binary-to-Thermometer Encoder 66

List of Acronyms and Symbols

Abbreviation	**Explanation**
ADC	Analog-to-Digital Converter
AC	Alternating Current
ASIC	Application-Specific-Integrated-Circuit
CDC	Capacitance-to-Digital Converter
DAC	Digital-to-Analog Converter
DNL	Differential Nonlinearity
DRC	Design Rule Check
FSR	Full Scale Range
LSB	Least Significant Bit
LVS	Layout versus Schematic
MEMS	Micro-Electro-Mechanical System
MIM	Metal-Insulator-Metal
MOM	Metal-Oxide-Metal
PGA	Programmable-Gain Amplifier
RF	Radio Frequency
SAR	Successive-Approximation-Register
SPDT	Single-Pole-Double-Throw
VPP	Vertical Parallel Plate

1. Introduction

1.1. Motivation

Capacitive MEMS pressure sensors are used in a wide field of applications, such as blood pressure monitoring in medical applications, tracking the pressure of gas or liquid in engines, observing the pressure in car tires, and many more. Compared to their piezoresistive counterpart, capacitive MEMS pressure sensors are superior in terms of stability, power consumption and packaging. Additionally, they show a lower sensitivity with respect to temperature [1].

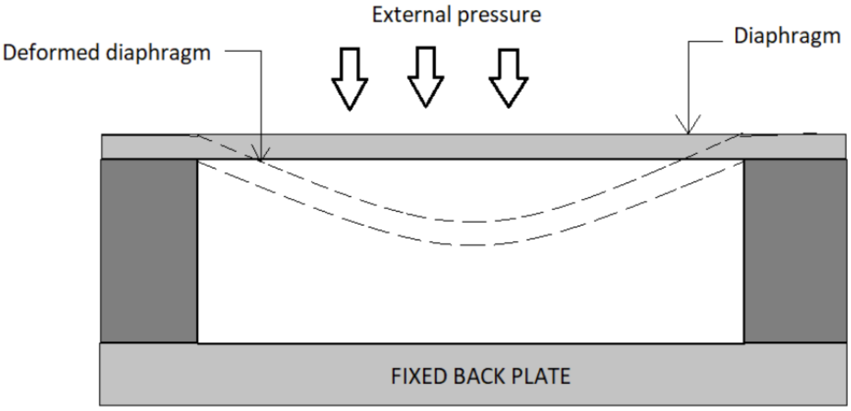


Figure 1.1.: Principle of a capacitive MEMS pressure sensor. Application of pressure leads to the deformation of the diaphragm which further leads to a change in capacitance between two electrodes [1]. ©2021 IEEE

The basic working principle behind a simple capacitive MEMS pressure sensor is shown in Figure 1.1. Basically, the sensor consists of two electrodes, denoted as diaphragm and fixed back plate. The electrodes are separated by an insulating material. If an external pressure is applied to the sensor, the diaphragm is deformed which leads to a change in distance between the electrodes and this further changes the capacitance between them. Therefore, by measuring the change of capacitance, it is possible to calculate the external pressure [1]. Typically, a capacitive MEMS pressure sensor is part of a measurement circuit and acts as the sensing element, for example in a capacitive bridge setup containing also reference capacitors. The output of the measurement circuit can be determined by a so called capacitance-to-digital converter (CDC) read-out circuit. This circuit converts the measured capacitance into a voltage by using a capacitive interface

circuit, and the voltage is then converted into a digital word by an ADC. Examples of capacitance-to-digital converters are provided in [2] and [3].
However, in order to verify the correct measurement of an ASIC that implements a capacitance-to-digital converter for the read-out of a pressure sensing circuit, the applied pressure to the sensing circuit has to be very accurate. For high accuracy pressure sensors, the limitations in verifying the accuracy and linearity can be the accuracy of the pressure chamber in the laboratory. This is due to the complexity of the setup containing multiple regulation loops for air pressure and temperature. Also the reference sensor might limit the accuracy. Due to these limitations, the aim of this work is to create an integrated circuit, denoted as programmable capacitor, that is connected to the ASIC instead of the pressure sensing circuit and therefore emulates the latter. This emulation would assist the verification of the mentioned ASIC.
In the currently used pressure sensing circuit, the capacitive MEMS pressure sensors are arranged in a half-bridge configuration and the bridge is read-out by the ASIC, which uses a fully-differential programmable-gain amplifier as an interface. The next section describes the current setup in more detail and defines what needs to be emulated in this work.

1.2. Pressure Sensing Circuit to be Emulated

1.2.1. Current Setup

Currently, the pressure measurement solution consists of two parts, an ASIC that implements a capacitance-to-digital converter and a pressure sensing circuit that is bonded to the ASIC. During operation, the ASIC converts the measured change in capacitance of the sensing circuit into a voltage, using a programmable-gain amplifier as a capacitive interface. This voltage is then converted into a digital word, which is proportional to the pressure applied to the sensing circuit. Figure 1.2 shows the current setup, including the pressure sensing circuit and the ASIC.

Pressure Sensing Circuit

In the pressure sensing circuit, two capacitive MEMS pressure sensors C_s are configured in a half-bridge configuration along with two reference capacitors C_r, shown by the left box in Figure 1.2. If a pressure is applied to the pressure sensing circuit, the capacitance of the MEMS pressure sensors will change by ΔC, which further leads to a change of charge at the output of the pressure sensing circuit. The bridge configuration is excited by an alternating voltage between the nodes $v_{ref,p}$ and $v_{ref,n}$. The excitation voltage is provided by the ASIC.

ASIC

As depicted in the right box in Figure 1.2, the first essential part in the ASIC is a fully-differential programmable-gain amplifier (PGA). Due to its capacitive feedback, the PGA integrates the charge generated by the pressure sensing circuit. The integration of the charge leads to a differential voltage at the output of the PGA which is proportional to the differential capacitance between the branches of the pressure sensing circuit. Following the integration stage, a fully-differential analog-to-digital converter translates the differential output voltage of the PGA into a digital word. Since the differential output of the PGA is proportional to the change of capacitance in the pressure sensing circuit, a specific attention is paid to the programmable-gain amplifier, especially in defining the relation between its output and the capacitance change in the pressure sensing circuit.

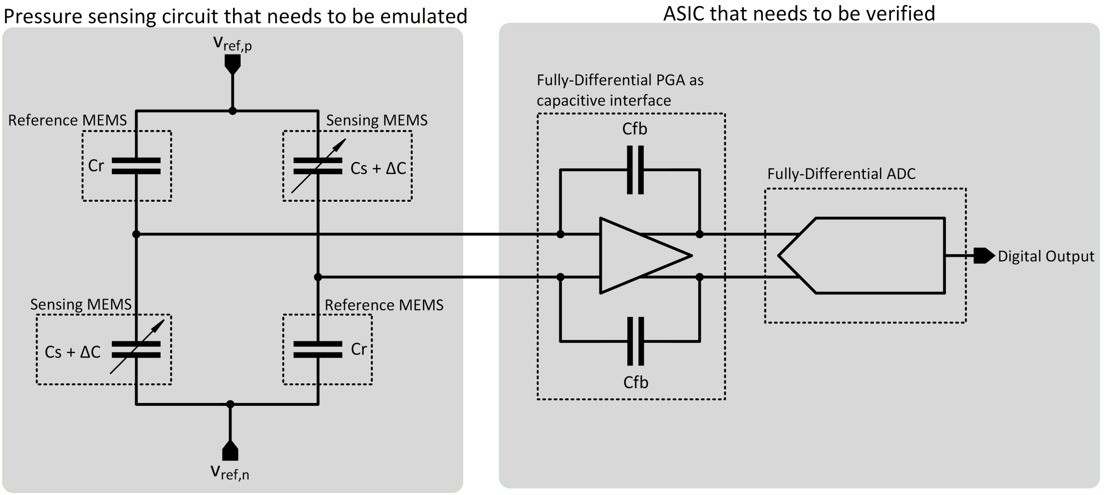


Figure 1.2.: Capacitive pressure measurement solution. The left box shows the schematic of the sensing circuit and thus the block that needs to be emulated, while the right box shows the essential stages of the ASIC that needs to be verified.

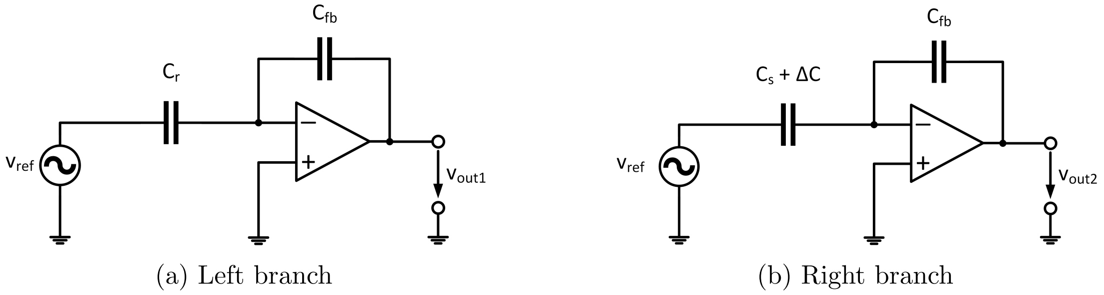
Aim of this Work

The aim of this work is to create a ”programmable capacitor” as an integrated circuit, that emulates the pressure sensing circuit. Therefore, the integrated circuit should behave in the same way as the pressure sensing circuit, with the difference that the differential capacitance at the output should depend on a programmed value instead of pressure. Additionally, the programmable capacitor should be excited by the same alternating reference voltage as the pressure sensing circuit, in order to maintain the correctness of the phase between the programmable capacitor and the ASIC. Inherently, the programmable capacitor has to have a fully-differential behaviour at its output so that the fully-differential behaviour of the ASIC is met. Different concepts on how to design such a programmable capacitor have to be investigated, where the strongest requirement lays on the linearity of the capacitance change along with the resolution of ΔC. The specifications are given in more detail in Table 1.1. Once a suitable concept is derived, an implementation regarding schematic and layout should be carried out.

Finally, the implemented integrated circuit should be verified by means of simulations which include the relevant blocks of the ASIC. If the simulation results meet the requirements, it is favourable to prepare the physical design of the integrated circuit for production. In summary, the declared aim of this work is to create an integrated circuit from the concept to the physical design, so that the programmable capacitor can be used in the laboratory at a later point in time.

1.2.2. Sensing Circuit Read-Out Principle

Before deriving a suitable concept for the programmable capacitor, it is essential to understand the relation between the differential output voltage of the programmable-gain amplifier and the change of capacitance in the pressure sensing circuit of the currently used setup. In order to derive the relation, both branches of the bridge setup are investigated separately by means of an AC signal analysis. Therefore, also the PGA is split up so that it can be seen as a inverting-amplifier with capacitive feedback. The circuits used for the derivation are shown in Figure 1.3. The inputs of the PGA are at a common DC voltage, which means that from an AC signal point of view, each branch can be reduced to one capacitor, which further simplifies the calculation. Since the capacitors are considered to be ideal, sine- and cosine-terms would cancel when calculating the equivalent capacitance at the output, and therefore the derivation is shown only for the amplitudes. All voltages in lower-case are AC voltages.



(a) Left branch

(b) Right branch

Figure 1.3.: AC signal derivation for both branches of the MEMS pressure sensor. Each branch is investigated separately by splitting up both the branches and the PGA.

Each output voltage can be calculated by applying the equation of an inverting amplifier:

$$v_{out1} = -\frac{(j\omega C_{fb})^{-1}}{(j\omega C_r)^{-1}} v_{ref} = -\frac{C_r}{C_{fb}} v_{ref} \tag{1.1}$$

$$v_{out2} = -\frac{(j\omega C_{fb})^{-1}}{[j\omega(C_s + \Delta C)]^{-1}} v_{ref} = -\frac{C_s + \Delta C}{C_{fb}} v_{ref} \tag{1.2}$$

With the difference of the voltages that are applied to the bridge as it is shown in Figure 1.2:

$$v_{ref} = v_{ref,p} - v_{ref,n} \tag{1.3}$$

The differential output voltage of the PGA is the difference between both outputs:

$$v_{out,diff} = v_{out1} - v_{out2} = \frac{C_s - C_r + \Delta C}{C_{fb}} v_{ref} = \frac{\Delta C}{C_{fb}} v_{ref} + \frac{C_s - C_r}{C_{fb}} v_{ref} \tag{1.4}$$

This result allows the calculation of the equivalent differential capacitance between both branches, which is proportional to the differential output voltage of the PGA:

$$v_{out,diff} = -C_{eq} \frac{v_{ref}}{C_{fb}} \longrightarrow \boxed{C_{eq} = -C_{fb} \frac{v_{out,diff}}{v_{ref}} = C_r - C_s - \Delta C} \tag{1.5}$$

One can immediately see, that the output of the PGA is directly proportional to the change of capacitance ΔC and shows a linear behaviour. Additionally, the result shows that the equivalent capacitance seen from the PGA reduces to -ΔC when the nominal values of C_r and C_s are equal. The next task is to find a concept for the programmable capacitor that also provides a linear behaviour between the output of the PGA and a change in capacitance ΔC.

1.3. Specification for Programmable Capacitor

In Table 1.1, the specification for the programmable capacitor is provided. As specified, the programmable capacitor should provide a full-scale range (FSR) of 160fF for C_{eq} with a resolution of LSB(C_{eq}) = 100aF. However, the specification for the linearity is more stringent than that for the FSR and the resolution. Therefore, as long as the specification for the linearity is met, the specified FSR and resolution can be seen as reference values, which are allowed to deviate to a certain extent in the low percentage region (e.g. the resolution is $\sim$ 110aF instead of the specified value).

Specification	**Value**	**Explanation**
$\mathrm{C_{eq,min}}$	-80fF	minimum programmable value of equivalent capacitance
$\mathrm{C_{eq,max}}$	80fF	maximum programmable value of equivalent capacitance
$\mathrm{C_{eq,FSR}}$	160fF	programmable full-scale-range of equivalent capacitance
$\mathrm{LSB(C_{eq})}$	100aF	programmable LSB value of equivalent capacitance
$\sigma\left(\mathrm{DNL_{C_{eq}}}\right)$	0.15%	Standard deviation of differential nonlinearity of $\mathrm{C_{eq}}$
VDD	1.8V	single supply voltage (VSS = 0V)
Metal Layers	4	Number of available metal layers for physical design

Table 1.1.: Specifications for the programmable capacitor. The linearity represents the most stringent specification and the specification for FSR and resolution can be relaxed to a certain extent, in order to achieve the high linearity.

2. Structure and Modelling of Programmable Capacitors

Before a suitable circuit is selected, the literature is investigated in order to select applicable integrated capacitors beforehand and study them with respect to their performance and limitations. Also ways to weight and switch integrated capacitors are investigated.

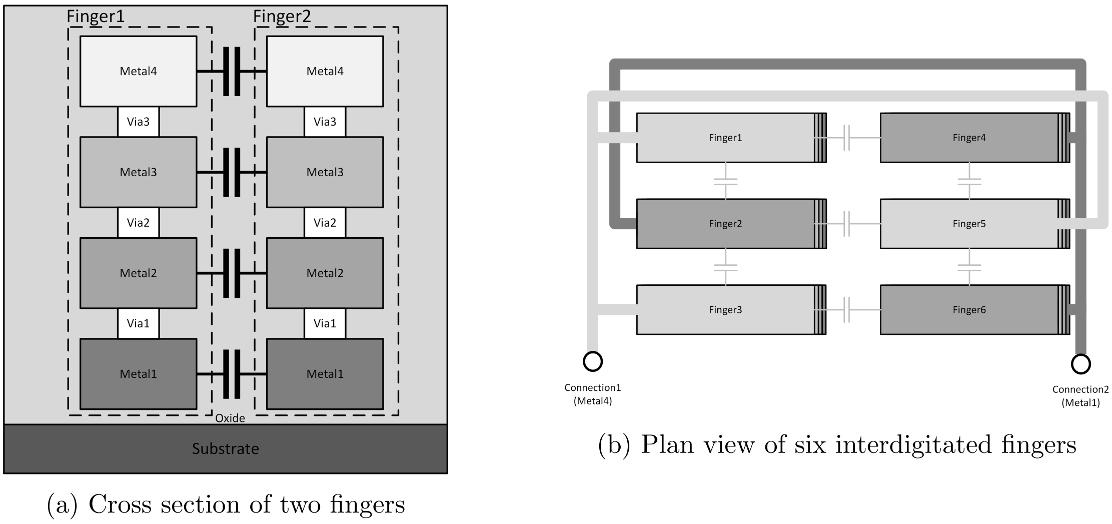
2.1. Integrated Capacitors

Depending on the used technology package, different types of integrated capacitors are available. However, each type has its advantages and disadvantages and one has to select a type that fits the application the best. Additionally, a trade-off between capacitance, area and matching is unavoidable as this section will point out. The following sections describe different types of integrated capacitors, showing their benefits and limitations. Furthermore, matching between integrated capacitors is discussed in detail.

2.1.1. Types of Integrated Capacitors

Vertical Parallel Plate Capacitor

A VPP (vertical parallel plate) capacitor is a commonly used integrated passive device that offers a high capacitance density as well as a high quality factor. Basically, the VPP capacitor consists of multiple interconnected metal layers, where each layer consists of interdigitated fingers that represent the "top-" and "bottom-" plate of a conventional capacitor. By using many metal layers and a high number of interdigitization, such a structure can achieve a high capacitance density [4][5]. For example, the VPP capacitor reported in [4] achieves a capacitance density of $1.76\mathrm{fF}/\mu\mathrm{m}^2$.
Figure 2.1 shows an example of a VPP capacitor that is constructed by using four metal layers.
The technology package that is used to design the programmable capacitor offers different types of VPP capacitors. Unfortunately, all available types use the lowest available metal layer (Metal1), which is a big drawback, because switches, logic and interconnections, that are used to address and drive the capacitor, cannot be placed below the capacitor. This does not only mean that the area would increase due to the placing of these parts elsewhere on the chip, but also the interconnection of the capacitors to the other parts of the circuit is challenging for a large number of them. Even though simulations show a very good matching behaviour of the VPP capacitors (subsection 2.1.2), the named drawbacks might outweigh the good matching for some applications.



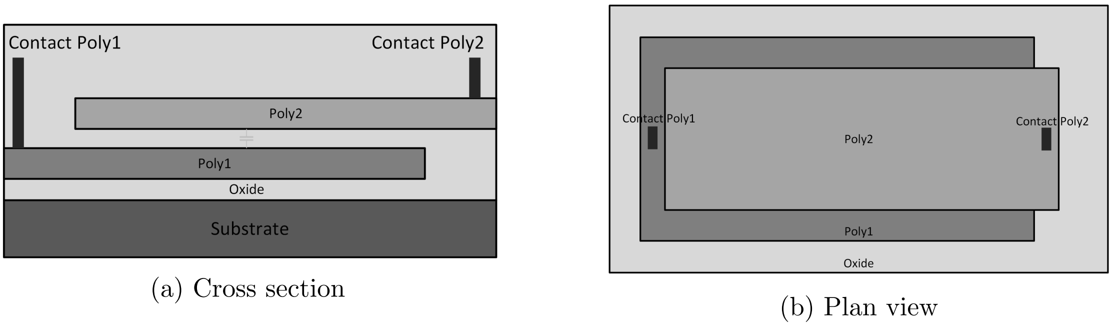
(a) Cross section of two fingers

(b) Plan view of six interdigitated fingers

Figure 2.1.: Example of a VPP capacitor made out of four metal layers. The contributing capacitances are shown between the fingers.

Poly-Poly Capacitor

Another type of integrated capacitors is the poly-poly capacitor, which is available in technologies with two poly layers. The capacitor is created by a vertical parallel connection between both poly layers and the layers are separated by silicon-dioxide (SiO_2). Because of the proximity of the capacitor's bottom-plate to the substrate, the bottom-plate can have a parasitic capacitance in the range of 10% to 30% of the capacitor's nominal value [6]. Additionally, the tolerance on the capacitor's absolute value is in the same range. Nevertheless, the matching (subsection 2.1.2) between equally sized poly-poly capacitors on the same die is more accurate by achieving values between 0.05% to 1% [6]. Regarding the capacitance density, only a limited number of values can be found in literature. In [7], a capacitance density of 0.85fF/µm^2 is reported for a poly-poly capacitor. Figure 2.2 shows an example of a poly-poly capacitor including contacts to higher metal layers.



(a) Cross section

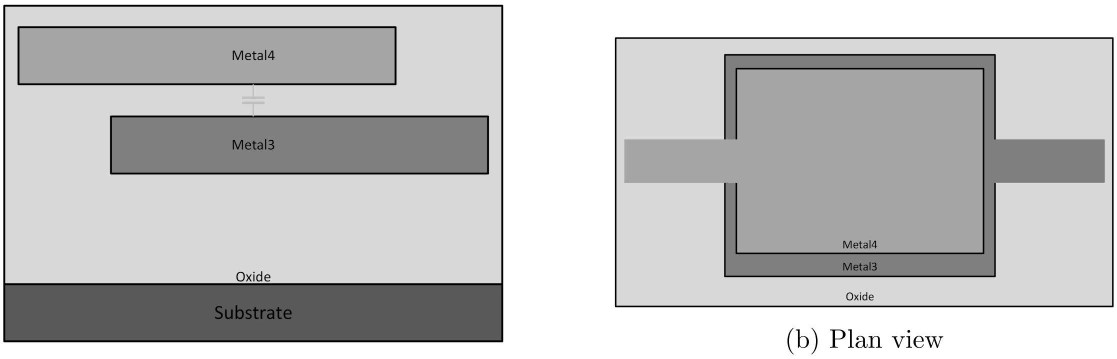
(b) Plan view

Figure 2.2.: Example of a poly-poly capacitor with contacts to higher metal layers.

Unfortunately, the used technology package for the programmable capacitor does not provide a second poly layer and therefore this type of capacitor cannot be used.

Metal-Oxide-Metal Capacitor

MOM (metal-oxide-metal) capacitors are implemented by using the standard metal connection layers available in every CMOS process. Additionally, there are two types of MOM capacitors, depending on the direction of the electric field between the capacitor's electrodes: lateral-field fringing capacitors and vertical-field parallel-plate capacitors [8]. On the one hand, the parallel-plate MOM capacitor is quite similar to the poly-poly capacitor, with the poly layers exchanged by the metal layers. On the other hand, a single layer of a VPP capacitor is similar to a lateral-field fringing MOM capacitor. In [9], a novel MOM capacitor structure is reported which achieves a capacitance density of $0.283\text{fF}/\mu\text{m}^2$.



(a) Cross section

(b) Plan view

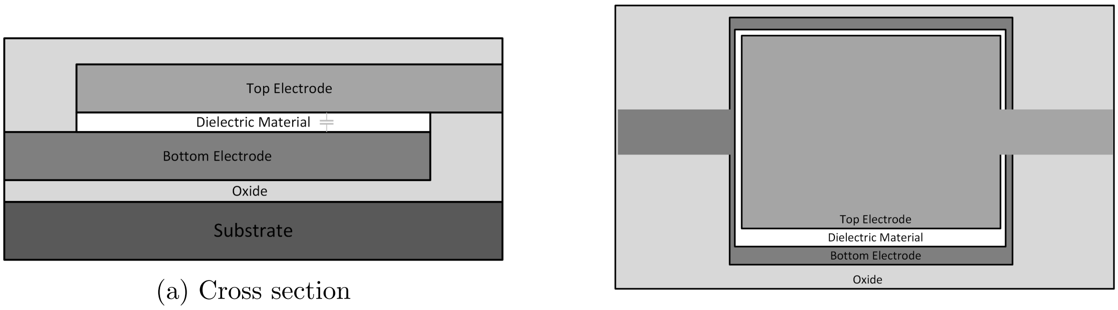
Figure 2.3.: Example of a parallel-plate MOM capacitor constructed with Metal3 and Metal4 layers. For MOM capacitors, there are no special connections, as a wire of the regarding metal-layer itself acts as a connection to other parts of the circuit.

In the used technology package, only parallel-plate capacitors are available and therefore only this type of capacitor is shown in Figure 2.3. However, for the available MOM capacitor, the metal layers of the plates can be chosen (e.g. Metal2 and Metal1 or Metal4 and Metal3). Considering the fact that each capacitor needs switches and logic devices, the Metal4 - Metal3 capacitor is the only reasonable configuration to choose as a MOM capacitor for further investigation, as it allows the placement of devices and wiring beneath it. Additionally, the larger distance of the bottom-plate (Metal3) to the substrate decreases the bottom-plate's parasitic capacitance which is also an advantage.

Metal-Insulator-Metal Capacitor

MIM (metal-insulator-metal) capacitors are similar to MOM capacitors, with the difference that there is a special dielectric material, such as Tantalum-oxide (Ta_2O_5) or

Alumina (Al_2O_3) between the plates of the capacitor instead of the standard oxide (typically SiO_2) in MOM capacitors. These special oxides have a higher dielectric constant κ than the standard oxide layer which results in a higher unit-capacitance [10].
The higher unit-capacitance and therefore the higher capacitance density is of large interest for many integrated applications, since integrated capacitors usually consume large chip area. Therefore, MIM capacitors, and especially their high κ dielectric materials, are subject to research in order to achieve high capacitance while consuming a small area [11]. For example, in [11] a capacitance density of $10\text{fF}/\mu\text{m}^2$ is reported for a MIM capacitor. In [12], the development of four layer MIM capacitors with an even higher achieved capacitance density of about $70\text{fF}/\mu\text{m}^2$ is reported.
Figure 2.4 shows an example of a MIM capacitor. In literature, different metal materials and dielectric layers are used for the MIM capacitor and therefore the figure is kept general and shows only the basic principle behind this type of capacitors.
Unfortunately, the used design package does not provide MIM capacitors. Nevertheless, due to their high potential for future applications and the high level of research interest, they are presented in this section.



(a) Cross section

(b) Plan view

Figure 2.4.: Example of a MIM capacitor. The electrodes are connected to metal layers or provide a connection to other parts of the circuit themselves.

2.1.2. Mismatch of Integrated Capacitors

General

In general, different formulations which describe the mismatch of integrated devices can be found in literature. Pelgrom et. al [13] describe the term mismatch as time-independent random variations in physical quantities of identical designed devices caused by fabrication of those devices, excluding variations of absolute values from wafer-to-wafer or batch-to-batch [13]. Applying this definition to integrated capacitors, only the random difference between the capacitance of equal capacitors on the same wafer is described as mismatch. However, Shyu et al. [14] differentiate between two types of errors of the values of integrated capacitors which can be observed. On the one hand there are systematic errors, which affect each equal capacitor on a chip in the same way and these correlated errors can be reduced for example by layout techniques. On the other hand, there are uncorrelated random errors, which affect each capacitor on the same chip in a different way and therefore represent an absolute limit on the achievable accuracy of circuits that rely on the ratio of those capacitors. Comparing both definitions, it can be seen that the definition of the uncorrelated random errors in [14] matches with the previously described definition in [13]. The systematic errors only change the absolute value of capacitors, but they do not change the ratio between the capacitors. So, the systematic error of the capacitance is of no interest, as it would introduce only gain and offset errors which could be calibrated by means of post-processing of the measurement data. However, due to the absolute limitation on the achievable linearity, the random errors of the capacitance are of highest interest and will be denoted as ”random mismatch” in this thesis.

Reasons for Random Mismatch between Capacitors

Regarding parallel-plate capacitors, there are three main reasons for the random mismatch between equally designed capacitors, namely the random variation of the plate's edges, the random variation of the thickness, and variation in the permittivity of the oxide between the plates of the capacitor. A variation in the ideally straight edge of the plates results in a random variation of the plate's area and therefore in the capacitance of the capacitor. Additionally, a variation in the thickness or permittivity of the oxide between the plates also causes a deviation of the capacitors ideal capacitance [14] [10]. Shyu et. al [14] provide a deep analytical investigation of the effects of edge- and oxide-variation on the value of MOS capacitors. Additionally, Marin et. al [10] provide an analytical investigation of edge-variations for MIM capacitors.

Characterization of Mismatch

In order to describe the mismatch between capacitors in a comprehensive way, a model for the normally distributed capacitor's value has to be introduced [15]:

$$C = C_\mu + \Delta C \tag{2.1}$$

The capacitance is defined as the sum of the capacitor's mean value C_μ and the deviation of this value expressed by ΔC. The latter term stems from the previously described variations of the capacitor's physical parameters. The standard deviation of the capacitance is equal to the standard deviation of ΔC [15]:

$$\sigma\left(C\right) = \sigma\left(\Delta C\right) \tag{2.2}$$

and the relative standard deviation is denoted as $\sigma\left(\frac{\Delta C}{C_\mu}\right)$ [15].
If a capacitor is constructed by connecting a number of N unit-capacitors in parallel, the absolute standard deviation of this equivalent capacitance is given by [15]:

$$\sigma\left(C_{eq}\right) = \sqrt{N} \cdot \sigma\left(\Delta C\right) \tag{2.3}$$

Equation 2.3 indicates that it is sufficient to measure the absolute standard deviation of a large capacitor made out of unit-capacitors in order to determine the absolute standard

deviation of those unit-capacitors. This approach is extensively shown in [8] and [15].
Pelgrom et al. showed in their famous and often cited paper [13], that the matching between transistor parameters (threshold-voltage V_t, current factor β and substrate factor K) is proportional to the square root of the transistor's inverse area. Hence, larger devices show a smaller standard deviation of the transistor parameters and therefore a better matching behaviour. Shyu et al. [14] indicate that the matching of integrated capacitors behaves similar to the model proposed by Pelgrom et al. In fact, Shyu et al. [14] show that the relative standard deviation of integrated MOS capacitors is proportional to $C^{-\frac{3}{4}}$ when edge-effects are considered and proportional to $C^{-\frac{1}{2}}$ when oxide variations are considered, which leads to the conclusion that the standard deviation of large capacitors is mainly dependent on random oxide variations while the standard deviation of small capacitors is mainly dependent on random edge variations. However, Omran et al. [15] show how matching between integrated poly-poly capacitors can be measured and conclude both [13] and [14] by giving a lean expression for a general model that allows the calculation of a capacitor's relative standard deviation [15]:

$$\sigma\left(\frac{\Delta C}{C_\mu}\right) = \sqrt{\frac{K_A^2}{A} + \frac{K_E^2}{A^{\frac{3}{2}}}} \tag{2.4}$$

where A denotes the area of the capacitor's top-view, K_A is the area-dependence Pelgrom's coefficient and K_E is the edge-dependence Pelgrom's coefficient. Equation 2.4 shows that for large areas, the area-effect caused by oxide variation, dominates the matching between capacitors, while for small areas the matching is dominated by the edge-effect which is caused by edge variations [8]. Interestingly, Omran et al. [8] do not see an increased mismatch due to edge effects in [8], when they investigate the matching properties of MOM capacitors. In their work, they provide an equation for the standard deviation of vertical parallel-plate capacitors [8]:

$$\sigma\left(\frac{\Delta C}{C}\right) = \frac{K_A}{\sqrt{A}} \tag{2.5}$$

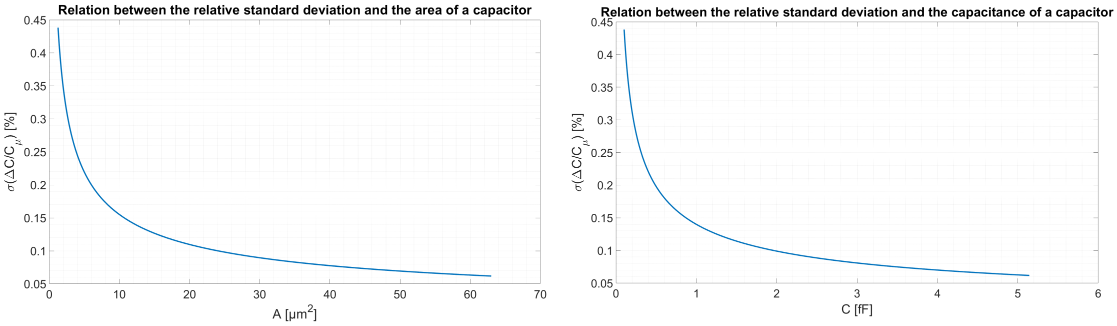
Similar to Equation 2.4, K_A is the area-dependence Pelgrom's coefficient of the capacitor's top-area A. Since the area is proportional to the capacitance, the authors provide a second equation for the capacitor's standard deviation [8]:

$$\sigma\left(\frac{\Delta C}{C}\right) = \frac{K_C}{\sqrt{C}} \tag{2.6}$$

with K_C the capacitance-dependence Pelgrom's coefficient for $\sigma\left(\frac{\Delta C}{C}\right)$ and C the capacitance of the capacitor.
Both Pelgrom's coefficients, K_A and K_C, are extracted by means of curve fitting of measured mismatch data [8]. If ΔC is not the variation of the unit-capacitor but the standard deviation of the difference between two measured capacitors with the same nominal value, then the standard deviation of the single capacitor is by a factor of $\sqrt{2}$ smaller when there is no correlation [16]. This has to be considered when the mismatch

between capacitors is determined by measuring their difference.

Figure 2.5 shows the dependency of the relative standard deviation of a vertical MOM capacitor on the corresponding area and capacitance, respectively. Basically, Figure 2.5a plots Equation 2.5 while Figure 2.5b plots Equation 2.6. The Pelgrom's coefficients $K_A = 0.49\% \cdot \mu\mathrm{m}$ and $K_C = 0.14\% \cdot \sqrt{\mathrm{fF}}$ where taken from [8], who extracted these values for vertical MOM capacitors. The plots show that the relative standard deviation strongly increases with decreasing capacitance. However, for larger capacitors, the relative standard deviation does not decrease that strong when the capacitance is increased. This is an important finding regarding the selection of the capacitor's value at a later point: From a certain value on, the increment of the capacitance improves the matching between capacitors only by a small value, but the size of the chip is drastically increased.



(a) Relative standard deviation depending on area

(b) Relative standard deviation depending on capacitance

Figure 2.5.: Relative standard deviation of vertical MOM capacitors. The Pelgrom coefficients K_A and K_C are taken from [8].

Availability of Capacitor Mismatch Data

As this section already indicated, reliable capacitor mismatch data is essential for the design of highly linear circuits that rely on accurate ratios between capacitors. However, the design kits usually provide extrapolated mismatch data or for measured capacitor only. If extrapolated mismatch data is available, the extrapolation often only contains the area-dependency in the Pelgrom's model, but not the edge-effect. Therefore, the mismatch of very small capacitors can be larger than expected [15][17]. For example, Omran et al. [15] show that their measured mismatch for very small poly-poly capacitors ($C_u = 2\mathrm{fF}$) was much higher than predicted by the extrapolated models.

Quantitative Mismatch Data Published in Literature

Table 2.1 shows published mismatch data for different capacitor types, values and processes. For the second and third column it can be clearly seen that the relative standard deviation of the capacitor's value decreases with increasing capacitance, similar to the described Pelgrom's model. Interestingly, the poly-poly capacitor with a unit-capacitance of 2fF shows a much higher relative standard deviation than the MOM capacitor with a unit-capacitance of 1.95fF. However, the comparison of relative mismatch data is only fair for capacitors with equal capacitance. In general, the Pelgrom's coefficient K_A allows a fairer comparison of relative capacitor mismatch over different unit-capacitor sizes, processes and capacitor types. One can see that the published Pelgrom's coefficient of $K_A = 0.49\% \cdot \mu\text{m}$ in [8] indicates that the capacitors in that work match much better than they do in the work of [18] with a published Pelgrom's coefficient of $K_A = 1.9\% \cdot \mu\text{m}$. Unfortunately, no Pelgrom's coefficient was published in [15].

	TCASS-II '16 [15]	**TCASS-I '16 [8]**			**TCASS-I '14 [18]**	
Process	350nm	180nm			32nm	
Type	Poly-Poly	MOM (vertical)			MOM (lateral)	
$\mathbf{K_A}$	-	0.49%·µm			1.9%·µm	
$\mathbf{C_\mu}$	2fF	0.48fF	0.94fF	1.95fF	1.2fF	0.45fF
$\boldsymbol{\sigma\left(\frac{\Delta C}{C_\mu}\right)}$	0.43%	0.21%	0.15%	0.10%	0.8%	1.2%

Table 2.1.: Capacitor mismatch data published in literature

Summary of Literature Study on Capacitor Mismatch

All investigated papers verify by measurements of real capacitors, that the relative standard deviation of capacitors is proportional to the square root of the inverse capacitor area, and therefore verifying the often stated "Pelgrom's model". For this work, this means that the area of the used capacitors provide an absolute limitation for the maximum achievable linearity. However, in some publications, additional mismatch can be seen due to the domination of the edge variations for very small capacitors, which also has to be kept in mind. Also the limited availability of mismatch data in a design-package represents a risk when a circuit is designed. Even if mismatch data is available, it has to be considered that the data is usually extrapolated from results of mismatch measurement of larger structures and therefore might not be applicable for very small capacitors.

Since mismatch of capacitors gives a limitation for the achievable linearity of the programmable capacitor, special attention is paid to this topic in this work.

2.2. Capacitor Weighting

In the pressure sensing circuit to be emulated, the capacitance change ΔC is realized by means of changing the distance between the electrodes of a capacitive MEMS pressure sensor by applying a certain pressure. When the applied pressure changes continuously, also the capacitance changes continuously. However, in case of the programmable capacitor, a digitally programmed value has to be converted into a certain change of capacitance. One possible way to do this is to place capacitors in parallel and switch them on or off in order to achieve a specific change of capacitance ΔC. This section introduces different methods to distribute the overall capacitance over switchable capacitors, based on findings in literature. Each method has its advantages and disadvantages and requires a careful selection which strongly depends on the application of these capacitor-arrays.

2.2.1. Binary-Weighted Capacitor-Array

The first weighting concept to be investigated is the so called binary-weighted capacitor-array. It consists of binary-weighted capacitors, which can be switched in a binary pattern, as shown in Figure 2.6. At a first glance, the big advantage of this concept is the low number of needed switches to cover a large range of switchable capacitance. This section deals with the further investigation of this concept.

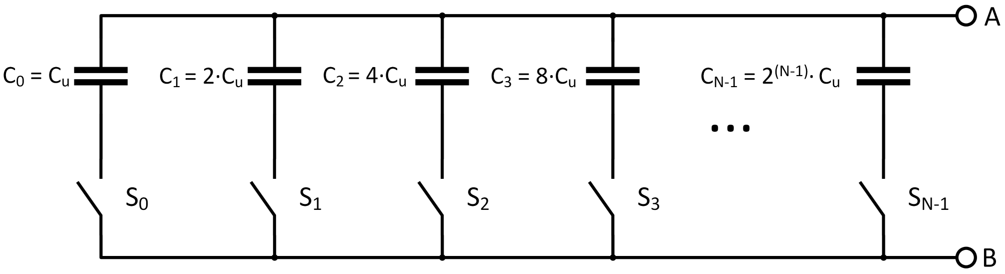


Figure 2.6.: Example of a binary-weighted capacitor-array. The total capacitance of the array is present between the terminals A and B. The switches are controlled in a binary pattern.

Scaling of Capacitors

For a binary-weighted capacitor-array, a number of N capacitors is needed, where N is the number of bits. The LSB capacitor has a unit-capacitance C_u and the capacitance of all other capacitors is a binary-scaled value of the unit-capacitance [19]. In order to calculate the needed capacitance for each capacitor in the array, the following equation is used:

$$C_i = 2^i \cdot C_u \tag{2.7}$$

The index i represents an integer which runs from 0 to N-1. In case of a 10 bit binary-weighted capacitor-array, the LSB capacitor has a value of C_u and the MSB capacitor has a value of $C_9 = 2^9 \cdot C_u = 512 \cdot C_u$. One can already see that adding one bit to the array doubles its size. In other words, each subsequent capacitor has to have the same capacitance as the sum of all previous capacitors. In order to reduce the influence of systematic errors in the ratios between the capacitors, usually all capacitors in the array are constructed as a parallel connection of unit-capacitors [14].

Impact of Random Mismatch on Linearity

In order to check the impact of random mismatch on the linearity of the binary-weighted capacitor-array, a MATLAB model is created. This model generates 500 arrays where each array represents a 10 bit binary-weighted capacitor-array. In these arrays, each capacitor consists of multiples of the unit-capacitor, similar as it is done in real applications. The unit-capacitor is considered to be normally distributed with a unit-capacitance of $C_u = 2\text{fF}$ and the associated relative standard deviation of $\sigma\left(\frac{\Delta C}{C_u}\right) = 0.1\%$, where both values are taken from [8]. The simulation consecutively switches on the capacitors in the array, thus increasing the capacitance at the output. For every change of capacitance, the simulation determines the differential nonlinearity (DNL) in each array. Additionally, the simulation determines the standard deviation of the differential nonlinearity over all 500 arrays, which is plotted in Figure 2.7. The differential nonlinearity, especially its standard deviation, allows a statement on the linearity of the capacitor-array and is useful to compare the binary-weighting scheme with the other investigated weighting-schemes.

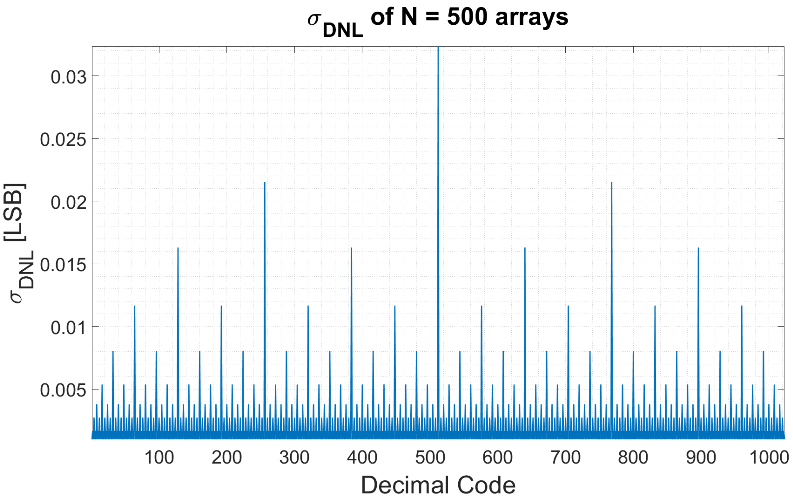


Figure 2.7.: Standard deviation of the DNL of a 10 bit binary-weighted capacitor-array. The unit-capacitance $C_u = 2\text{fF}$ and its standard deviation of $\sigma\left(\frac{\Delta C}{C_u}\right) = 0.1\%$ are taken from [8].

The plot of the simulation result shows, that the largest nonlinearities occur at transitions where large capacitors are switched, in case of the 10 bit capacitor-array for example at the transition of "0b0111111111" to "0b1000000000", which leads to the large spike in the middle of the plot. Additionally, it can be seen that the largest capacitor is responsible for the largest nonlinearity, the second largest capacitor is responsible for the second largest nonlinearity and so on. This behaviour was expectable, as Equation 2.3 indicates that the absolute random mismatch of a capacitor scales with the square-root of the number of interconnected unit-capacitors the capacitor is made of. Ideally, the capacitance of a binary-weighted capacitor is equal to the sum of all smaller capacitors. However, due to random mismatch, this is not achievable and thus leads to the nonlinearities visible in the plot.

Discussion on Binary-Weighted Capacitor-Array

A major advantage of a binary-weighted capacitor-array is the low number of needed switches, which is equal to the number of implemented bits. This results in a low complexity of the logic that controls these switches which further can reduce the overall space and the routing expenses of the array. Additionally, there are sophisticated algorithms reported in literature, that improve the placement of the unit-capacitors in the physical design in a way so that the impact of unmatched routing parasitics is minimized [20][21]. However, the binary-weighted capacitor-array also has a big drawback, namely the discussed influence of capacitor mismatch on linearity. As the section on matching pointed out, the impact of mismatch can be reduced by increasing the size of the unit-capacitors. However, this increases the smallest switchable capacitance and the capacitance of the overall array for a fixed number of bits, resulting in a trade-off between the LSB capacitance and matching.

2.2.2. Thermometer-Coded Capacitor-Array

Another concept to weigh capacitors is used in the thermometer-coded capacitor-array, also denoted as "unit capacitor-array" [22], where each switched capacitor has the same capacitance as the LSB capacitor. This results in a high number of switches, because the capacitors are not switched on in a binary pattern anymore, but in a thermometer pattern, meaning that each capacitor is consecutively switched on and stays in that state. Pun et al. [22] show that the thermometer-coded capacitor-array is superior to the binary-weighted capacitor-array regarding linearity, because as opposed to the latter, only one capacitor is switched at a time and no large bit transitions occur anymore. Figure 2.8 shows the basic principle behind a thermometer-coded capacitor-array.

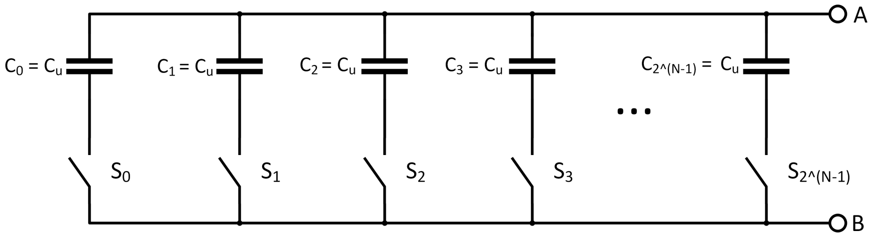


Figure 2.8.: Example of a thermometer-coded capacitor-array. The total capacitance of the array is present between the terminals A and B. The switches are controlled in a thermometer pattern.

Scaling of Capacitors

All capacitors in a thermometer-coded capacitor-array are equal to the unit-capacitor, hence they have the same capacitance as the LSB capacitor:

$$C_i = C_u \tag{2.8}$$

where the variable i is an integer number between 0 and 2^{N-1} and N is the number of bits to be implemented. Due to the high number of switchable capacitors, it is evident that the concept of thermometer-coded capacitor-arrays requires a large number of switches to cover a large range of switchable capacitance. In fact, a 10 bit thermometer-coded capacitor-array requires $2^{10} = 1024$ unit-capacitors. This is the same number of unit-capacitors as it is needed in binary-weighted capacitor-arrays for the same number of bits. However, the thermometer-coded capacitor-array requires 1024 switches instead of 10 as in the binary-weighted capacitor-array. Additionally, the thermometer-coded capacitor-array requires a binary-to-thermometer encoder, in order to control the thermometer-coded switches with a binary value [22].

Impact of Random Mismatch on Linearity

Similar as for the binary-weighted capacitor-array, a MATLAB model is created that implements 500 thermometer-coded capacitor-arrays, where each capacitor in a single array is consecutively switched on. In order to enable a fair comparison, the same unit-capacitance $C_u = 2\text{fF}$ and its standard deviation $\sigma\left(\frac{\Delta C}{C_u}\right) = 0.1\%$ of [8] are used, similar to the values of the binary-weighted capacitor-array. Figure 2.9 shows the standard deviation of the differential nonlinearity of the thermometer-coded capacitor-array.

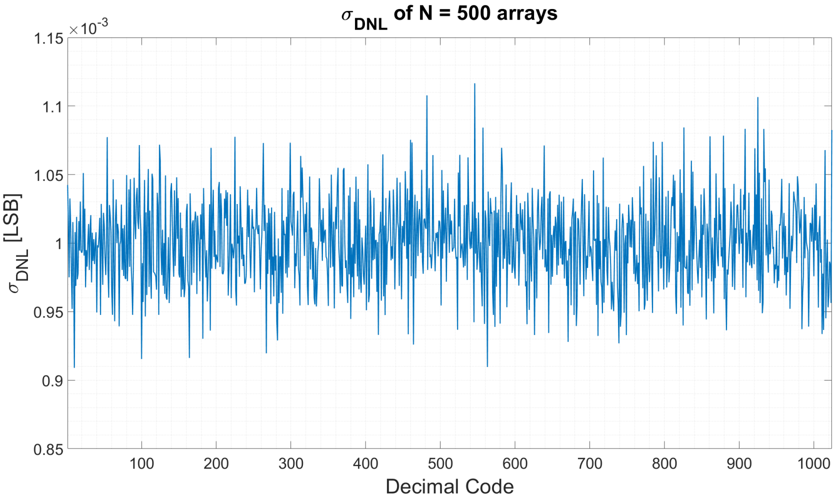


Figure 2.9.: Standard deviation of the DNL of a 10 bit thermometer-coded capacitor-array. The unit-capacitance $C_u = 2\text{fF}$ and its standard deviation of $\sigma\left(\frac{\Delta C}{C_u}\right) = 0.1\%$ are taken from [8].

By comparing the standard deviation of the differential nonlinearity of the thermometer-coded capacitor-array with the one of the binary-weighted capacitor-array, it is visible that the linearity of the former is much better. Actually, the standard deviation of the differential nonlinearity of the thermometer-coded capacitor-array is equal to the standard deviation of the used unit-capacitor, in this case 0.1% of the unit-capacitance. This was expectable, because the differential nonlinearity determines the difference between consecutive steps related to the average step-size, and for the case of the thermometer-coded capacitor-array, the deviation of the step-size is limited by the standard deviation of the unit-capacitor only. However, this is not true for the binary-weighted capacitor-array, because for each step, usually more than one capacitor is switched.

Discussion on Thermometer-Coded Capacitor-Array

Regarding linearity, which is a tight specification in this work, the thermometer-coded switching of capacitors is superior to the binary-weighted switching of capacitors, as not only the literature but also the conducted simulations proofed. Another advantage is the robustness against defects in the capacitor-array. If there is a defect in the capacitor-array which would lead to the failure of one switch, the impact on the linearity is by far not as large as for binary-weighted capacitor-arrays. In fact, a defect would cause a linearity in the transfer function due to a missing code. However, the defect would not influence the linearity of the previous and following codes and it would just result in a lower overall switchable capacitance. Nevertheless, one has to keep in mind that thermometer-coded capacitor-arrays require a large number of switches, which further requires a large logic overhead in order to control them.

2.2.3. Hybrid-Coded Capacitor-Array

The third possibility to realize the capacitor-array is to combine the functionality of the binary-weighted and the thermometer-coded capacitor-arrays. In [23], this concept is called "partially thermometer-coded" capacitor-array. However, in order to avoid confusion with the thermometer-coded capacitor-array, this concept is called "hybrid-coded" capacitor-array in this work. As the simulation result of the binary-weighted capacitor-array indicates, the largest nonlinearities occur when the largest capacitors of the array are switched. In order to overcome this issue, those large capacitors are split up into equally sized smaller capacitors in the hybrid-coded capacitor-array and they are switched in a thermometer-coded sense, while the smaller capacitors are realized in a binary-weighted sense [23]. Therefore, the hybrid-coded capacitor-array combines the advantage of the low number of switches of the binary-weighted capacitor-array with the high linearity of the thermometer-coded capacitor-array. Figure 2.10 shows an example of such a configuration.

Due to the fact that a smaller number of thermometer-coded capacitors is used compared to the thermometer-coded capacitor-array, the required binary-to-thermometer decoder results in a smaller logic overhead.

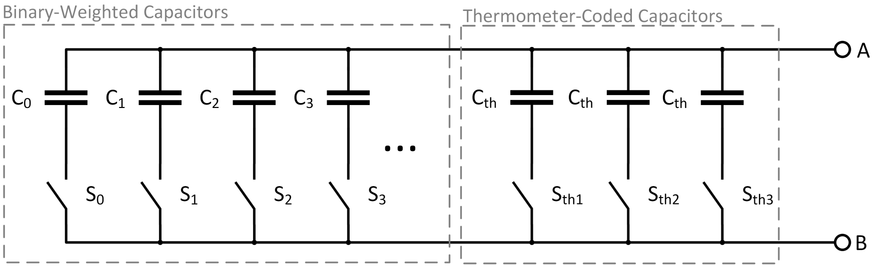


Figure 2.10.: Example of a hybrid-coded capacitor-array. The capacitors in the right rectangle are of equal size and are switched in a thermometer-coded sense, while the capacitors in the left rectangle are binary-weighted and switched in a binary-weighted sense. The total capacitance of the array is present between the terminals A and B.

Scaling of Capacitors

In hybrid-coded capacitor-arrays, the number of split-up thermometer-coded capacitors N_{th} depends on the number M, which is the number of the most significant capacitors of the binary-weighted capacitor-array that should be replaced by the thermometer-coded capacitors [23]:

$$N_{th} = 2^M - 1 \tag{2.9}$$

Further, the equivalent capacitance C_{th} for a single thermometer-coded capacitor is determined as [23]:

$$C_{th} = 2^{N-M} C_u \tag{2.10}$$

For example, a hybrid-coded capacitor-array of $N = 10$ bit and $M = 2$ bit would need $2^2 - 1 = 3$ segments to substitute the two largest capacitors by thermometer-coding and the equivalent capacitance of each thermometer-coded capacitor would be $2^{10-2} C_u = 256 \cdot C_u$.

In this example, only one additional switch is needed compared to the binary-weighted case. However, the 3 thermometer-coded capacitors have to be controlled by a binary-to-thermometer encoder that translates a 2 bit binary value into a 3 digit thermometer-code.

Impact of Random Mismatch on Linearity

Also for this concept, a MATLAB model that implements five hundred 10 bit hybrid-coded capacitor-arrays with a unit-capacitance of $C_u = 2\text{fF}$ and its standard deviation of $\left(\frac{\Delta C}{C_u}\right) = 0.1\%$ is created. Again, both values are taken from [8]. Similar to the example given above, the two largest capacitors are substituted by three equivalent sized thermometer-coded capacitors and all other capacitors are binary-weighted. Figure 2.11 shows the simulated standard deviation of the differential nonlinearity of the hybrid-coded capacitor-array, and indicates that the nonlinearities caused by the most significant capacitors are reduced. Additionally, for the given parameters the simulation result shows that the linearity of the hybrid-coded capacitor-array is about two times better than the linearity of the binary-weighting capacitor-array and about 16 times worse than the linearity of the thermometer-coded capacitor-array. Therefore, it provides a compromise between a moderate number of switches and achievable linearity.

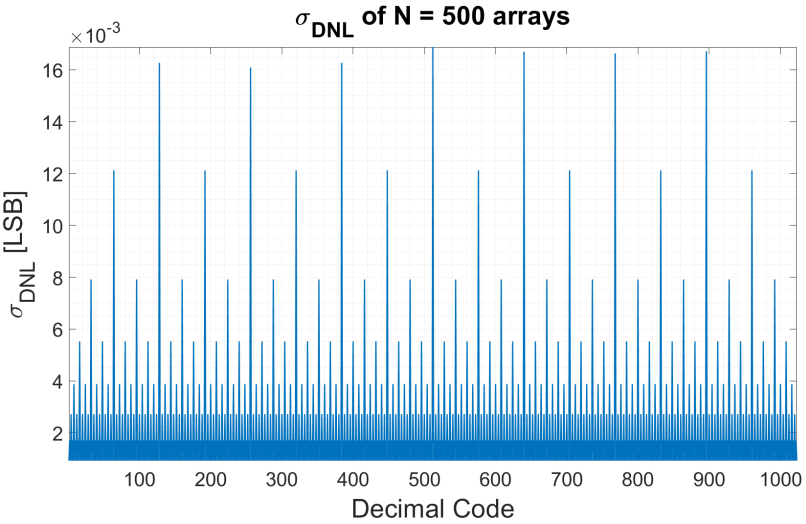


Figure 2.11.: Standard deviation of the DNL of a 10 bit hybrid-coded capacitor-array, where the two largest capacitors are substituted by three thermometer-coded capacitors and the remaining capacitors are binary-weighted. The unit-capacitance $C_u = 2\text{fF}$ and its standard deviation of $\sigma\left(\frac{\Delta C}{C_u}\right) = 0.1\%$ are taken from [8].

Discussion on Hybrid-Coded Capacitor-Array

As expected prior to the simulations, the hybrid-coded capacitor switching scheme is a trade-off between the switching of binary-weighted and thermometer-coded capacitors. It provides a good linearity while limiting the number of needed switches. When designing a hybrid-coded capacitor-array, the number of bits that should be thermometer-coded can be chosen depending on the specification. By choosing a larger number of thermometer-coded capacitors, the hybrid-coded capacitor-array approaches the thermometer-coded capacitor-array and therefore looses the advantage of a medium number of needed switches. However, by choosing a very small number of thermometer-coded capacitors, the hybrid-coded capacitor-array gets similar to the binary-weighted capacitor-array, which results in a lower number of switches but degradation of the achievable linearity for a given unit-capacitor.

2.3. Applications of State-of-the-Art Capacitor-Arrays

An examination of the literature reveals that capacitor-arrays are nowadays mainly used in analog-to-digital converters that use the successive-approximation-register principle.

However, there are also other applications that require a programmable capacitance, for example to provide a wide tuning range of resonant circuits. The following sections present some of the found applications and evaluate their usability for this work.

2.3.1. Capacitor-Arrays in DACs

Basic Functionality

Capacitor-arrays are extensively used as a DAC (digital-to-analog converter) in SAR-ADCs (successive-approximation-register analog-to-digital converters), as shown in Figure 2.12. Basically, an SAR-ADC consists of a sample-and-hold circuit, a comparator, a successive-approximation-register and a DAC. After the input voltage was sampled, it is compared to a fraction of a reference voltage. This fraction is controlled by the SAR and created by the DAC and it follows a binary-search algorithm until the difference between the input voltage and the output voltage of the DAC is smaller than 1 LSB [24]. Hence, the latest created digital word to control the DAC is simultaneously the converted digital value of the ADC. However, the linearity of an SAR-ADC suffers from the mismatch of the capacitors in the DAC [25]. Therefore, the literature on SAR-ADCs gives a valuable insight not only on the application of capacitor-arrays but also on the effect of their nonlinearity on applications.

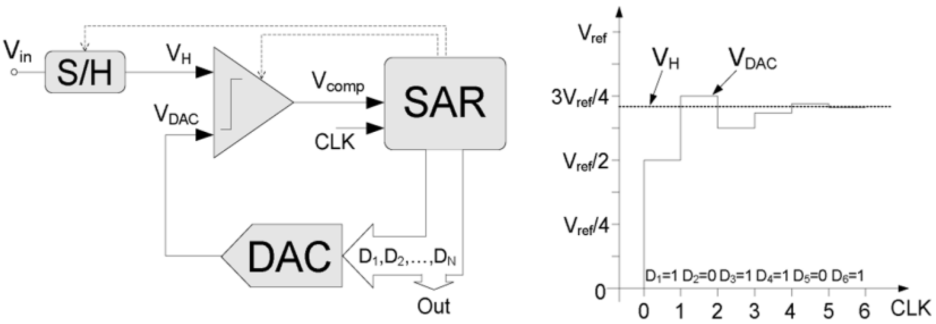


Figure 2.12.: Basic principle of a successive-approximation-register analog-to-digital converter including one conversion cycle [24]. ©2011 IEEE

Binary-Weighted Capacitor-Array in SAR-ADCs

In literature, many SAR-ADCs that utilize binary-weighted capacitor-arrays can be found and usually two different concepts are described. On the one hand there are conventional binary-weighted capacitor-arrays [25][19], similar to the one shown in subsection 2.2.1. On the other hand, there is a special configuration described, where the capacitor-array is split into two arrays and separated by a capacitor. There are different names for this concept, such as "split-capacitor" DAC [19] or "segmented-capacitor" array [26]. The splitting of the capacitor-array with a capacitor decreases the size of the MSB capacitors which certainly decreases the number of needed unit-capacitors and therefore reduces the space occupied by the capacitor-array [19]. However, in [19] it is shown that the linearity of the split-capacitor DACs heavily depends on the matching of the unit-capacitors, due to the presence of the splitting capacitor. In fact, the authors show that for a given linearity, the conventional binary-weighted DAC consumes less space than the split-capacitor DAC, because in order to achieve the given linearity, the unit-capacitors would have to be made larger for the latter. For this project, these are valuable findings and will be considered at a later decision on how to weigh the capacitors in the array.

Thermometer-Coded Capacitor-Array in SAR-ADCs

Only a limited number of publications can be found that present thermometer-coded capacitor-arrays in SAR-ADCs, most likely because the linearity of binary-weighted DACs is "good enough" and the large logic overhead of thermometer-coded capacitor-arrays can therefore be avoided. However, Pun et. al [22] show that the linearity of a thermometer-coded capacitive DAC is superior to both a conventional and a split-capacitor binary-weighted DAC. Additionally, the authors outline that for a given DNL, the thermometer-coded DAC shows the highest area efficiency and consumes the least switching energy under the investigated concepts. These findings are valuable, as the high linearity is a stringent requirement in this work.

Hybrid-Coded Capacitor-Array in SAR-ADCs

Similar to the thermometer-coded capacitive DACs, only limited publications on hybrid-coded capacitive DACs are available. Nevertheless, Hänzsche et al. [23] propose a hybrid-coded DAC implemented in an SAR-ADC. The authors state that it would not be feasible to implement thermometer-coded DAC, due to the large logic complexity which grows exponentially with number of bits and thus would require small technologies. Therefore, they present a 12 bit binary-weighted capacitive DAC where the largest 3 bit are substituted by 7 thermometer-coded capacitors. Their simulation results show that the linearity could be improved compared to binary-weighted DAC. Additionally, they state that it is sufficient if only a few of the MSB bits are thermometer-coded, in order to reduce the nonlinearity at large bit transitions. In summary, the authors underpin the same trade-off between complexity and linearity as it is described in subsection 2.2.3.

2.3.2. Other Applications

Apart from the usage in SAR-ADCs, capacitor-arrays are also used in other applications, for example in circuits that address RF technology. In [27], the authors propose a binary-weighted capacitor-array that is used in wide tuning range resonant circuits, where the main focus lays on the quality factor of the implemented capacitor-array. In [28] the authors present a capacitor-array for high-frequency antenna tuning, including a comprehensive investigation of RF switches. However, both publications show a strong focus on RF, which is not of interest in this work, and little or no focus is placed on matching and linearity of those arrays. Nevertheless, it is noteworthy that capacitor-arrays are also implemented in other fields of microelectronics.

3. Concept for Sensor Emulation

This chapter contains the description of the different investigated bridge configurations including their relation between output and switched capacitance. Based on these findings, a suitable bridge configuration including a capacitor-weighting scheme is chosen and a corresponding model is created and simulated. This model is further used not only to underpin the decision on the bridge configuration and the weighting-scheme, but it is also used to determine the required unit-capacitance.

3.1. Bridge Configuration

Since the aim of this work is to create a programmable capacitor that emulates the capacitive MEMS pressure sensor in the closest way possible, a bridge configuration similar to the pressure sensing circuit is preferable. This section investigates different bridge configuration concepts, where the main focus is to stay concept-wise as close to the pressure sensing circuit as possible.

3.1.1. Half-Bridge Configuration

Schematic

The half-bridge configuration depicted in Figure 3.1 is the first concept to be investigated, as it emulates exactly the pressure sensing circuit that contains the capacitive MEMS pressure sensor. In each branch of the bridge, one capacitor-array is connected in series to a fixed capacitor. Additionally, the capacitor-arrays are arranged in a differential manner, as their positions are interchanged between both branches.

Relation of △C to Output of PGA

The derivation is similar to subsection 1.2.2. Due to the fact that all caps have the same nominal value ($C = C_r = C_s$), the equation of the differential output voltage of the PGA is reduced to:

$$v_{out,diff} = \frac{\Delta C}{C_{fb}} v_{ref} \tag{3.1}$$

where v_{ref} is:

$$v_{ref} = v_{ref,p} - v_{ref,n} \tag{3.2}$$

The equivalent capacitance seen from the input of the PGA is calculated by:

$$C_{eq} = -C_{fb} \frac{v_{out,diff}}{v_{ref}} \longrightarrow \boxed{C_{eq} = -\Delta C} \tag{3.3}$$

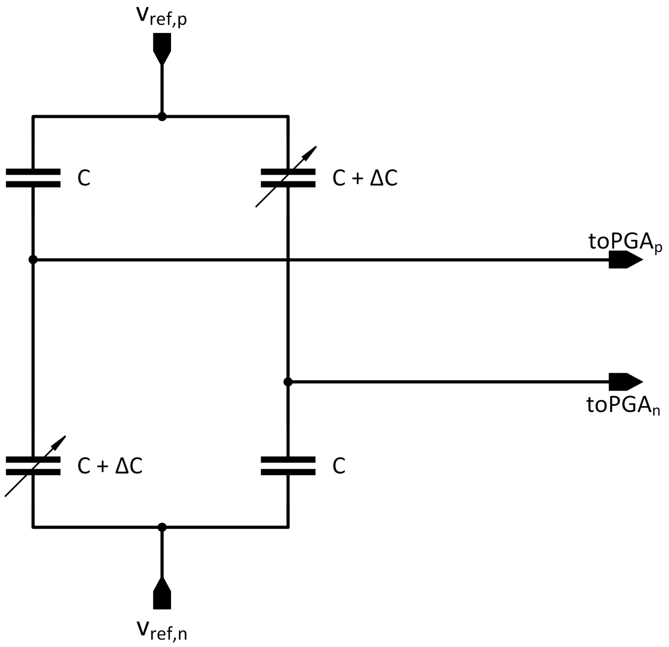


Figure 3.1.: Capacitive Half-Bridge. This setup is closest to the setup of the MEMS device the programmable capacitor should emulate.

Discussion

Equation 3.3 indicates that the equivalent capacitance C_{eq} equals the change of the sensing capacitors ΔC in the bridge. This implies that the LSB capacitance of ΔC would have to be in the sub-fF range in order to achieve the specified resolution. However, such small capacitors are very challenging to handle with regards to matching and therefore achieving an acceptable linearity. Additionally, the used design package allows the implementation of such small capacitors only by custom created MOM capacitors, which involves the risk of inaccurate mismatch data for simulations of circuits that implement such small capacitors (subsection 2.1.2).

3.1.2. Attenuated Half-Bridge Configuration

Schematic

Since a good matching between the capacitors is essential to achieve a high linearity, it is favourable to implement large capacitors. However, the previous investigation of the half-bridge showed that this requires very small changes of ΔC to meet the specification. In order to implement large capacitors for good matching, while keeping the LSB value of the equivalent capacitance low, the bridge is attenuated by two series capacitor C_s at its output. This section examines the impact of this attenuation on linearity.

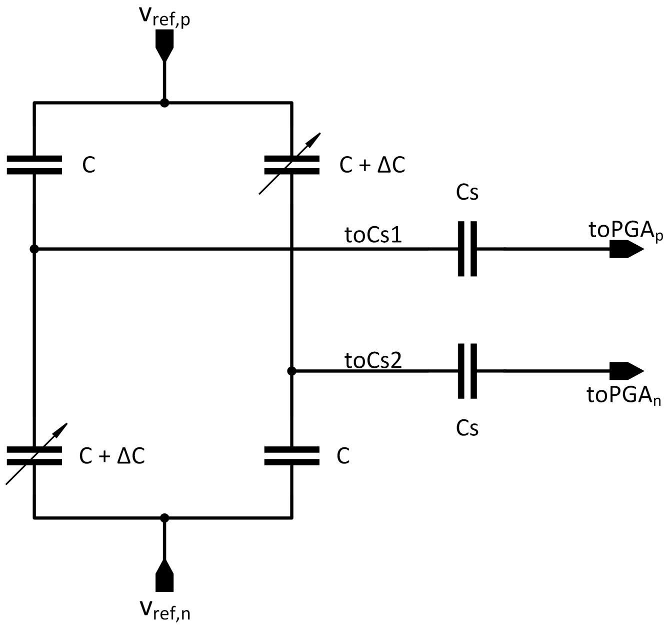


Figure 3.2.: Attenuated capacitive Half-Bridge. The half-bridge configuration is attenuated by the attenuation capacitors C_s.

Relation of ΔC to Output of PGA

The differential equivalent capacitance between both branches can be calculated by means of an AC signal derivation, shown in subsection A.1.1, and the following result can be obtained:

$$C_{eq} = C_{eq1} - C_{eq2} \longrightarrow \boxed{C_{eq} = C_s \frac{\Delta C}{2C + \Delta C + C_s}} \tag{3.4}$$

Discussion

By looking at Equation 3.4, one can immediately see that the differential capacitance between both branches C_{eq} is in a non-linear dependency on the change of capacitance ΔC in the branches, because ΔC occurs in a sum in the denominator. Even though the capacitance change ΔC is effectively attenuated, which would allow the switching of larger capacitors in the branches, the inherent nonlinearity makes this bridge configuration unusable for this work.

3.1.3. Full-Bridge Configuration

Schematic

If the value of the capacitors in each branch is changed in the opposite direction but by the same value, and if this is done inversely in the second branch, the bridge-configuration is described as a "Full-Bridge", depicted in Figure 3.3.

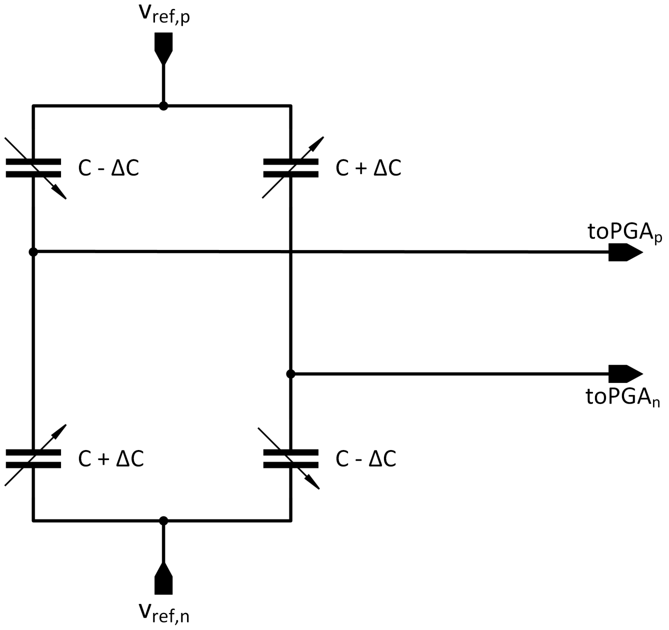


Figure 3.3.: Capacitive Full-Bridge with four changeable capacitors. The complementary switching maintains a differential behaviour between the left and the right branch.

Relation of ΔC to Output of PGA

The derivation of the differential capacitance seen by the PGA is quite similar to the derivation of the half-bridge and is shown in subsection A.1.2. For simplicity, only the final equation is shown here:

$$\boxed{C_{eq} = -2\Delta C} \tag{3.5}$$

Discussion

Equation 3.5 shows that the full-bridge is linear. However, the capacitance change ΔC occurs twice as large in the equivalent capacitance C_{eq} as for the case of the half-bridge. This further means, that this full-bridge configuration requires an even smaller LSB capacitance than the half-bridge, because in order to create a given change of C_{eq} for a specified resolution, the minimum switchable capacitor would need to be half as large as the minimum switchable capacitor of the half-bridge. The limitations of very small capacitors have been discussed in subsection 2.1.2.

3.1.4. Attenuated Full-Bridge Configuration

Schematic

This section shows the concept of an attenuated full-bridge configuration which shows, on the contrary to the attenuated half-bridge configuration, a linear behaviour between the change of capacitance in the branches and the output of the PGA, while giving the opportunity to attenuate the differential capacitance between the branches . Figure 3.4 shows the schematic of the attenuated full-bridge configuration. It is assumed that all capacitors have the same nominal value and the absolute change of capacitance ΔC is the same for all capacitors. Additionally, both attenuation capacitors C_s are considered to be equal.

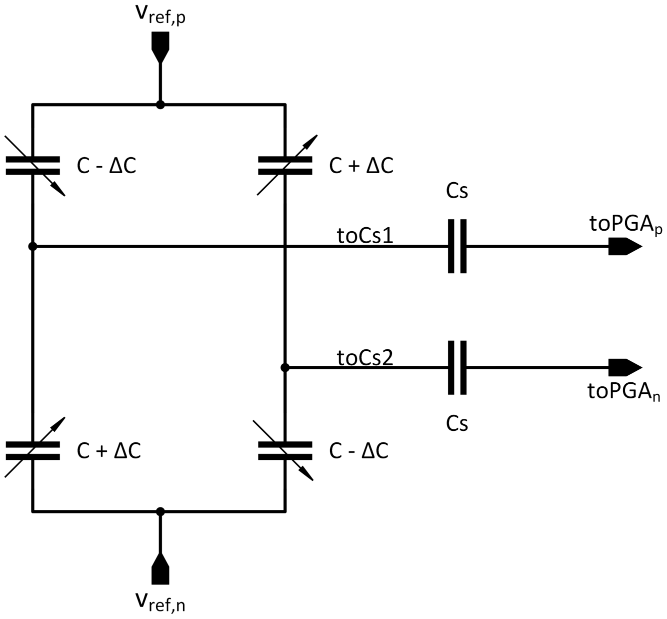


Figure 3.4.: Attenuated capacitive Full-Bridge. The capacitors switch complementary between both branches.

Relation of △C to Output of PGA

The derivation shown in subsection A.1.3 gives the differential equivalent capacitance between both branches:

$$C_{eq} = C_{eq1} - C_{eq2} \longrightarrow \boxed{C_{eq} = \frac{2C_s}{2C + C_s}\Delta C} \tag{3.6}$$

Discussion on Ideal Configuration

By looking at Equation 3.6, one can see that the relation between the differential equivalent capacitance and the capacitance change ΔC is not only linear, but is also attenuated by a constant attenuation factor A:

$$A = \frac{2C_s}{2C + C_s} \tag{3.7}$$

The attenuation factor relates a switched capacitance ΔC to the equivalent capacitance C_{eq} read by the PGA. It depends on the total capacitance of one branch $2C$, as well as the attenuation capacitor C_s. For a given capacitance per branch, the desired attenuation factor can be set by the attenuation capacitors:

$$C_s = 2C\frac{A}{2 - A} \tag{3.8}$$

Here is an arbitrary example: The total capacitance of a branch is $2C = 1\text{pF}$ and it can be switched by a unit-capacitor of $C_u = 10\text{fF}$. In order to achieve a 100aF change of the equivalent capacitance C_{eq} for a single step, the attenuation has to be $A = \frac{100\text{aF}}{10\text{fF}} = \frac{1}{100}$. According to Equation 3.12, this would require attenuation capacitors with a capacitance of $C_s = 100\text{fF} \cdot \frac{\frac{1}{100}}{2 - \frac{1}{100}} = 5.025\text{fF}$. Due to the effective attenuation, this concept would allow to switch larger capacitors having smaller mismatch, while keeping the setup linear. Thus, the concept of the attenuated full-bridge has only shown advantages so far and is therefore investigated deeper in the next sections.

Schematic of Realistic Configuration

Since the ideal configuration of this concept is very promising, the concept is investigated in a deeper way. In reality, not all capacitors would have the same nominal value and absolute value of capacitance change. Therefore, the real configuration of the attenuated full-bridge will deviate from the ideal configuration. In order to investigate the effect of these deviations, a different nomenclature of the capacitors is used, shown in Figure 3.5. This allows the derivation of a more realistic relation between a programmed capacitor change and the resulting capacitor change represented at the output of the PGA.

Relation of ∆C to Output of PGA

The derivation shown in subsection A.1.4 gives the following relation between capacitance change and the equivalent capacitance determined by the PGA:

$$C_{eq} = C_{eq1} - C_{eq2} \longrightarrow \boxed{\begin{aligned} C_{eq} = -\frac{1}{2}C_{s1}&\frac{C_1 - C_2 - \Delta C_1 - \Delta C_2}{C_1 + C_2 - \Delta C_1 + \Delta C_2 + C_{s1}} + \\ &+\frac{1}{2}C_{s2}\frac{C_3 - C_4 + \Delta C_3 + \Delta C_4}{C_3 + C_4 + \Delta C_3 - \Delta C_4 + C_{s2}} \end{aligned}} \tag{3.9}$$

C_i represents the nominal value of the capacitor, while ΔC_i denotes the absolute change of capacitance of the capacitor, where $i \in 1, 2, 3, 4$.

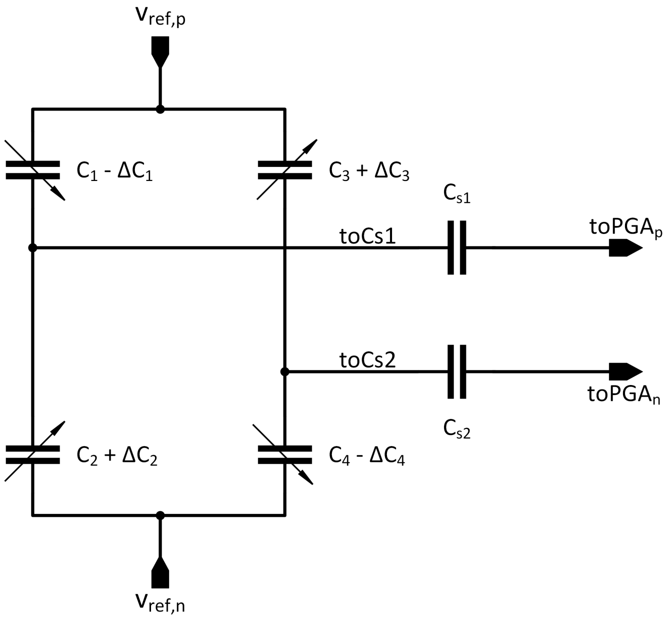


Figure 3.5.: Attenuated capacitive Full-Bridge with different nomenclature of the capacitors, as their nominal values and the change of capacitance is not considered to be equal anymore.

Discussion on Real Configuration

In this section, Equation 3.9 is investigated thoroughly. One can see that for both terms, the capacitance changes ΔC_i have the same sign per numerator. However, they have a differential sign in the denominator in each term. This further means, that if $\Delta C_1 = \Delta C_2 = \Delta C_{12}$ and $\Delta C3 = \Delta C_4 = \Delta C_{34}$, the capacitance changes vanish in the denominator while staying in the numerator, and the equation can be rewritten to:

$$\boxed{C_{eq} = -\frac{1}{2}C_{cs1}\frac{C_1 - C_2 - 2\Delta C_{12}}{C_1 + C_2 + C_{s1}} + \frac{1}{2}C_{s2}\frac{C_3 - C_4 + 2\Delta C_{34}}{C_3 + C_4 + C_{s2}}} \tag{3.10}$$

By looking at Equation 3.10, one can obtain that the given constrain of equal absolute capacitance change is mandatory for the bridge to be linear, because a change of capacitance only affects the denominator. Further notice, that if $C_1 = C_2 = C_3 = C_4$ and $C_{s1} = C_{s2}$, the equation reduces to the ideal equation (Equation 3.6). Equation 3.10 further shows, that a mismatch between the attenuation capacitors C_{s1} and C_{s2} results in an offset and gain error, but does not affect the linearity.

Conceptional Results of Bridge Configurations

Table 3.1 summarizes the determined conceptional results of the different bridge configurations. In addition to linearity, it shows the proportionality between the equivalent capacitance C_{eq} determined by the PGA and the capacitance change ΔC in the branches. This table is intended as a decision-making aid to choose one of the bridge configurations for the final concept.

Type	**Section**	**Linearity**	$\mathbf{C_{eq} \propto \Delta C}$
Half-Bridge	subsection 3.1.1	linear	ΔC
Attenuated Half-Bridge	subsection 3.1.2	non-linear	$\frac{C_s}{2C+\Delta C+C_s}\Delta C$
Full-Bridge	subsection 3.1.3	linear	$2\Delta C$
Attenuated Full-Bridge	subsection 3.1.4	linear	$\frac{2C_s}{2C+C_s}\Delta C$

Table 3.1.: Conceptional results of different bridge configurations

3.2. Final Bridge Concept Decision

Decisions regarding the capacitor-weighting, the bridge configuration and the unit-capacitance are presented in this section. These decisions are based on the investigated theory and simulation results. Additionally, the final concept will be checked by creating a model and conducting concept verification simulations with this model.

3.2.1. Chosen Capacitor Weighting

In section 2.2, different capacitor weighting methods were investigated. Within this work, the most stringent requirement on the chip is linearity and both logic complexity and occupied space are considered to be subordinated. Therefore, the way that fits the requirements the best is to weigh the capacitors by the thermometer-coded weighting scheme, as it gives the highest achievable linearity under all three investigated weighting schemes. Even though a thermometer-coded capacitor-array requires many switches and thus a large overhead for addressing each capacitor, the linearity is superior to the binary-weighted and hybrid-coded weighting schemes. This was not only confirmed by conducted simulations, but also stated by literature. Additionally, for the desired number of switchable cells, the logic- and routing-complexity is considered to be acceptable within this work.

3.2.2. Chosen Bridge Configuration

As Table 3.1 in the previous section indicates, only the attenuated full-bridge configuration allows the switching of capacitors that are larger than the required capacitive step-size of C_{eq}, while providing a linear behaviour between the change of capacitance ΔC in the branches and the differential capacitance C_{eq} determined by the PGA. Due to its attenuation and linear behaviour, only this bridge configuration is applicable for this work and therefore it is chosen for the final concept. Gain- and offset-errors, which might arise due to a deviation of the attenuation capacitors from their intended value, are compensated by means of post-processing of the measurement data. Therefore, these errors are taken into account when implementing this configuration.

Ensuring Equal Change of Capacitance in Each Branch

Equation 3.9 assumes that $\Delta C_1 = \Delta C_2$ and $\Delta C_3 = \Delta C_4$ so that those capacitances vanish in the denominator and the attenuated full-bridge becomes linear. Taking into account that the capacitors C_1 to C_4 are realized as switchable capacitor-arrays, this would mean that exactly the same capacitance has to be switched in both arrays of one branch. For example in the left branch, if one capacitor is switched off in the array that forms C_1, exactly the same value has to be switched on in the array that forms the capacitor C_2. In the right branch, it is the opposite, so that the differential behaviour between both branches is maintained. However, to switch exactly the same capacitance in two different arrays is not possible in reality, because due to mismatch, the capacitors that are switched will not have exactly the same capacitance. This leads to a non-linear behaviour, as the ΔC_i will not vanish in the denominator of Equation 3.9. To overcome this issue, the capacitors in each branch are not switched on or off, but they are switched between $v_{ref,p}$ and $v_{ref,n}$. In this thesis, this switching scheme is called "entangled switching scheme" because exactly the same capacitor that is switched from $v_{ref,p}$ is switched to $v_{ref,n}$ and vice versa. This switching scheme inherently forces $\Delta C_1 = \Delta C_2$ and $\Delta C_3 = \Delta C_4$. By looking at Figure 3.5, one can see that for the left branch C_1, C_2 and C_{s1} are connected to a common node. Therefore, two switch a capacitor e.g. from C_1 to C_2, only the bottom-plate of the capacitor has to be switched from $v_{ref,p}$ to $v_{ref,n}$, which allows a very effective and convenient setup, as it is shown in Figure 3.6.

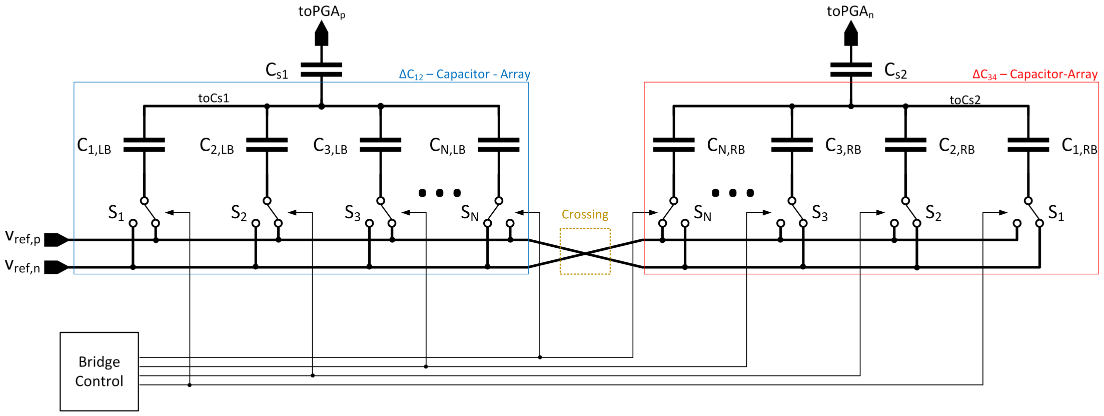


Figure 3.6.: Entangled switching scheme including both branches. Each branch consists of thermometer-coded capacitors whose bottom-plates are either switched to $v_{ref,p}$ or $v_{ref,n}$. The capacitors are switched inversely between both branches. For clarity, only a few capacitors are shown.

This setup effectively enables the usage of only one array per branch, which is another advantage. The ΔC_{12} array represents the switchable capacitors C_1 and C_2 and therefore the left branch, while the ΔC_{34} array represents the switchable capacitors C_3 and C_4 and thus the right branch. It is important to mention that each switch is either switched to $v_{ref,p}$ or $v_{ref,n}$ and does not have a floating state where the bottom-plate of the capacitor would be neither switched to $v_{ref,p}$ nor to $v_{ref,n}$. Therefore, the overall capacitance of each branch stays constant.

Crossing of $\mathbf{v_{ref,p}}$ and $\mathbf{v_{ref,n}}$ between Capacitor-Arrays

Due to the required fully-differential behaviour of the programmable capacitor, the capacitors have to be switched in an opposite way between both branches, e.g. when C_1 increases then C_3 must decrease. In context of the arrays ΔC_{12} and ΔC_{34}, this would mean that if for example a capacitor in ΔC_{12} is switched from $v_{ref,n}$ to $v_{ref,p}$, a capacitor in ΔC_{34} has to be switched from $v_{ref,p}$ to $v_{ref,n}$. In order to control the switches of both arrays with the same signal for each switch, $v_{ref,p}$ and $v_{ref,n}$ are crossed between both arrays. Therefore, each equivalent switch, for example the switch of the first capacitor, can be controlled by the same signal in both arrays. This drastically reduces the logic complexity and wiring expenses when it comes to the physical design of the circuit later on.

Qualitative Analysis of Parasitic Capacitances

In this analysis, the qualitative impact of parasitic capacitances of the capacitors onto the linearity of the capacitor-array is investigated. During the analysis, it is considered that all capacitors are Metal4-Metal3 MOM capacitors, however, the discussion why these capacitors are used follows at a later point in subsection 3.2.3. To simplify the analysis, it is considered that each unit-capacitor has the same parasitic capacitances, hence only a part of the capacitor-array is shown in Figure 3.7.

Both the unit-capacitors C_u and the attenuation capacitor C_s have a parasitic capacitance from metal layer Metal4 to ground, denoted as $C_{par,M4}$ and $C_{s,par,M4}$, and from metal layer Metal3 to ground, denoted as $C_{par,M3}$ and $C_{s,par,M3}$. Additionally, the unit-capacitors have parasitic capacitances between their top- and bottom-plates, referred to as $C_{par,M4M4}$ and $C_{par,M3M3}$, respectively. During operation, the voltages $v_{ref,p}$ and $v_{ref,n}$ alternate between 0V and 1.2V with a phase-shift of 180°. Consequently, the parasitic capacitance $C_{par,M3}$ is shorted for half of a cycle and is in parallel to 1.2V for the other half of the cycle. This further means that $C_{par,M3}$ is always driven during operation and therefore does not affect the linearity of the capacitor-array. This is the reason why the unit-capacitors are placed as in Figure 3.7, as this placement allows the elimination of the impact of these parasitic capacitances.

For each branch of the capacitor-array, the parasitic capacitances $C_{par,M4}$ are in parallel and can be added together with the parasitics of the wiring to ground $C_{par,wire}$ and the parasitic capacitance $C_{s,par,M4}$ of the attenuation capacitor to represent the parasitic

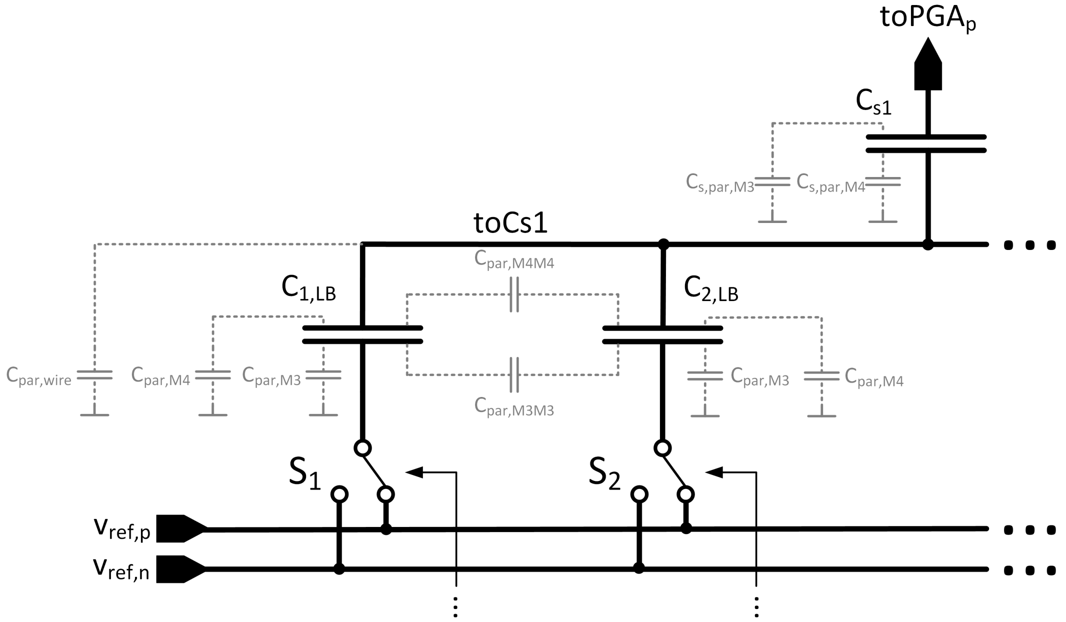


Figure 3.7.: Dominant parasitic capacitances in the capacitor-array are shown in gray. The unit-capacitors C_u and the attenuation capacitor C_s are considered to be Metal4-Metal3 MOM capacitors.

capacitance C_{par} of the node $toCs1$ to ground:

$$C_{par} = C_{s,par,M4} + C_{par,wire} + \sum_{n=1}^{N} C_{par,M4} \tag{3.11}$$

where N is the total number of unit-capacitors in each branch. Considering that this capacitance is equal for both branches, the impact on linearity of the ideal attenuated full-bridge is derived in subsection A.1.5. The result of this derivation shows that these parasitics only cause a gain error, but do not influence the linearity:

$$C_{eq} = \frac{2C_s}{2C + C_{par} + C_s} \Delta C \tag{3.12}$$

Hence, the only consequence is that the attenuation capacitors C_s would have to be recalculated, so that the gain fits the requirements again. This can be done by modifying Equation 3.12 as follows:

$$C_s = (2C + C_{par}) \frac{A}{2 - A} \tag{3.13}$$

In general, the parasitic capacitances are unknown before the chip is physically designed, thus C_s has do be adapted at a later point once C_{par} has been determined. Compared to the unit-capacitors, the attenuation capacitor is placed "upside-down", so that its larger

$C_{s,par,M3}$ is driven by the common-mode input of the PGA and does not additionally load the node *toCs*. Regarding the parasitic capacitances between unit-capacitors, it is evident that $C_{par,M4M4}$ is shorted, because all top-plates are interconnected. Similar to $C_{par,M3}$, the parasitic capacitance $C_{par,M3M3}$ either loads the input voltages or is shorted, depending on the position of the switches of neighbouring unit-capacitors. Therefore, $C_{par,M3M3}$ does not affect the linearity of the capacitor-array.

Due to the large distance between the Metal4-layer, the overall parasitic capacitance C_{par} between the node *toCs* and ground is considered to be small, however, as long as there is no physical design implemented, its value is unknown. Finally, it is noteworthy that there are also rather small parasitic capacitances between the top-plate of a unit-capacitors and the bottom-plate of its neighbouring capacitor. However, these capacitances can be modelled to be parallel to the unit-capacitor. If the spacing in the physical design is constant between each capacitor, these parasitic capacitances are equal for each unit-capacitor, hence they increase the capacitance of each unit-capacitor by the same value. Thus, only a benign gain error is observed and the linearity is not affected. For clearness and due to their small value, these parasitic capacitances are not shown in Figure 3.7.

Final Equation for Equivalent Capacitance

Equation 3.10 was derived in a general way for the real attenuated full-bridge configuration, where the values of C_1 to C_4 could be arbitrarily chosen. In order to be implemented by a model later, this equation is modified so that it fits the setup depicted in Figure 3.6, where C_1 to C_4 are directly coupled to the switched capacitors in the array, as the following derivation shows.

As an initial condition, it is considered that all capacitors are switched to $v_{ref,p}$ in the left branch, and to $v_{ref,n}$ in the right branch. Thus, C_1 and C_4 are equivalent to the total capacitance of the left and right branch, respectively:

$$C_1 = C_{tot,LB} = \sum_{i=1}^{N} C_{i,LB} \tag{3.14}$$

$$C_4 = C_{tot,RB} = \sum_{i=1}^{N} C_{i,RB} \tag{3.15}$$

where N is the total number of capacitors in each branch. Both capacitors C_2 and C_3 are equal to zero, as all capacitors are switched to C_1 and C_4. With these assumptions, Equation 3.10 reduces to:

$$C_{eq} = -\frac{1}{2}\frac{C_{tot,LB} - 2\Delta C_{12}}{C_{tot,LB} + C_{s1}} + \frac{1}{2}\frac{-C_{tot,RB} + 2\Delta C_{34}}{C_{tot,RB} + C_{s2}} \tag{3.16}$$

Since ΔC_{12} and ΔC_{34} are realized as capacitor-arrays, these capacitances depend on the number of switched capacitors N_{sw}:

$$\Delta C_{12} = \sum_{i=1}^{N_{sw}} C_{i,LB} \tag{3.17}$$

$$\Delta C_{34} = \sum_{i=1}^{N_{sw}} C_{i,RB} \tag{3.18}$$

By increasing the number N_{sw}, C_1 and C_4 decrease while C_2 and C_3 increase, however, the sum of C_1 and C_2 , as well as of C_3 and C_4 always stays constant.
Finally, the equation for the equivalent capacitance determined by the PGA can be written as:

$$\boxed{C_{eq} = -\frac{1}{2}\frac{C_{tot,LB} - 2\sum_{i=1}^{N_{sw}} C_{i,LB}}{C_{tot,LB} + C_{s1}} - \frac{1}{2}\frac{C_{tot,RB} - 2\sum_{i=1}^{N_{sw}} C_{i,RB}}{C_{tot,RB} + C_{s2}}} \tag{3.19}$$

It is noteworthy that Equation 3.19 is just a different notation of Equation 3.10 that takes the entangled switching scheme, shown in Figure 3.6, as well as the number of switched capacitors in both arrays into account. Additionally, it is worth to mention that the equation for the attenuation capacitors is not affected, as it only depends on the attenuation and the total capacitance in each branch.

3.2.3. Final Decision on Unit-Capacitor Type and Unit-Capacitance

The selection of a suitable capacitor type along with its unit-capacitance is a crucial step prior to the implementation. Not only the matching between the unit-capacitors and therefore the unit-capacitance have to be decided, but also a focus must be placed on physical design considerations, as some capacitor types do not support an efficient and sufficiently small layout later. The following sections show on which results the final decisions on the capacitor type and the unit-capacitance are based on.

Model

In order to make a decision on the capacitor type and the unit-capacitance that is supported by reliable simulation data, the mismatch of different capacitor types and values is determined by means of simulations in Cadence Design Environment. After that, a MATLAB model is created, that basically implements Equation 3.19 and this model is fed by the unit-capacitance C_u and the mismatch $\sigma\left(\frac{\Delta C}{C_u}\right)$ determined by the previous simulations in Cadence. More precisely, based on the given unit-capacitance and its standard deviation, the model implements the arrays ΔC_{12} and ΔC_{34} and calculates C_{eq} for every switched capacitor in the arrays. Both capacitor-arrays consist of 1024 unit-capacitors, which consecutively leads to 1024 values for C_{eq}. Quite similar as it was done in the section on capacitor-weighting (section 2.2), the MATLAB model determines the standard deviation of the DNL of C_{eq} over 500 runs. It is important to mention that the attenuation capacitors C_s are considered to be ideal, because their deviation would only cause a benign gain error which would further lead to an offset in the standard deviation of the DNL. The code of the implemented model is provided in section A.4.

Simulation Results

Multiple simulations have been conducted with two different types of capacitors, namely the vertical-parallel-plate capacitor (VPP) and the Metal4-Metal3 MOM capacitor (M4M3 MOM). For each capacitor type, several unit-capacitances have been investigated. Depending on the unit-capacitance, the attenuation capacitors have to be adapted according to Equation 3.12 so that the LSB of C_{eq} stays constant. Table 3.2 shows the simulation results depending on the input values of the model. As the capacitor-arrays are thermometer-coded, the standard deviation of the DNL is randomly distributed around a certain value, and therefore the average value of the standard deviation is provided, which allows a fairer comparison between the simulation results. In the used technology package, the MOM capacitor can have smaller values than the VPP capacitor.

	Model Input			Model Output
Capacitor Type	$\mathbf{C_u}$ **[fF]**	$\sigma\left(\frac{\mathbf{\Delta C}}{\mathbf{C_u}}\right)$**[%]**	$\mathbf{C_s}$ **[fF]**	$\overline{\sigma\left(\mathbf{DNL_{C_{eq}}}\right)}$**[%]**
VPP	25	0.0840	51.30	0.0594
	35	0.0757	51.27	0.0534
	50	0.0660	51.25	0.0466
	100	0.0532	51.23	0.0376
M4M3 MOM	1	0.763	53.89	0.5397
	25	0.138	51.30	0.0972
	35	0.123	51.27	0.0869
	50	0.111	51.25	0.0784
	100	0.0926	51.23	0.0655

Table 3.2.: Simulation results for different capacitor types and unit-capacitances

Figure 3.8 gives a graphical interpretation of the simulation results provided in Table 3.2. The simulation results show that the VPP capacitor would fulfil the specification for every investigated unit-capacitance C_u. The same is true for the M4M3 MOM capacitor except for the unit-capacitance of $C_u = 1\text{fF}$, which shows a large impact on linearity due to its high mismatch. Such a small value was only simulated to show that small capacitors drastically increase the nonlinearity of the circuit. However, the simulation results also show that increasing the capacitance from a certain capacitance on does not have a large effect on the linearity of C_{eq} anymore. This further means that if the unit-capacitance is chosen too large, a large area is needed while improving the linearity only by a small factor. For example, increasing a M4M3 MOM capacitor from a value of 25fF to a value of 100fF, the standard deviation of the differential nonlinearity only decreases by approximately 0.03%.

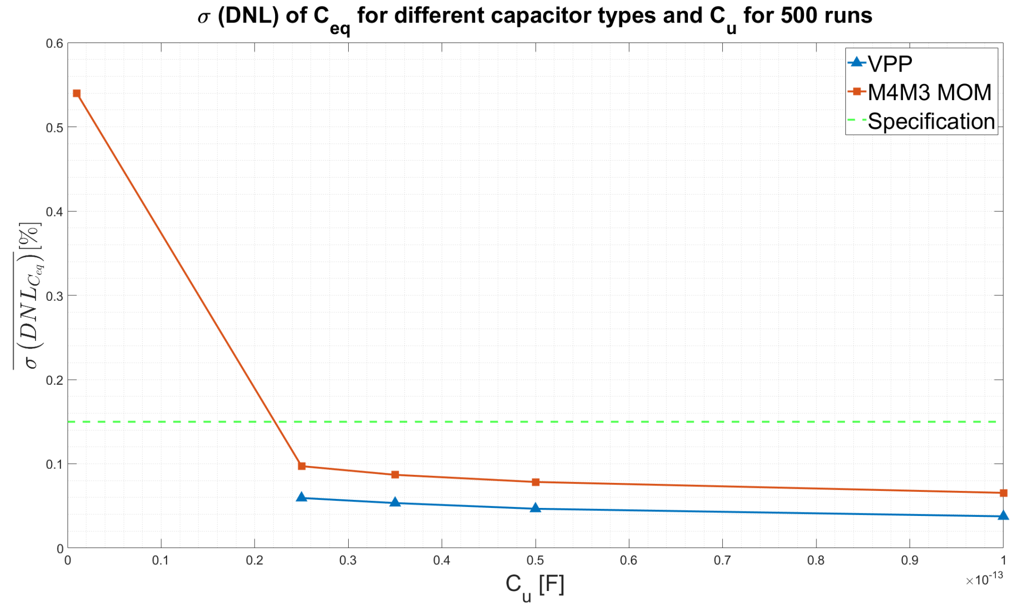


Figure 3.8.: Simulation results of the MATLAB model for different capacitor types and unit-capacitance C_u for 500 runs. Graphical representation of Table 3.2.

Decision on Capacitor-Type and Value

As Figure 3.8 indicates, the VPP capacitor shows a better linearity than the M4M3 MOM capacitor for the same unit-capacitance. However, the VPP capacitor has a major drawback because it is constructed out of multiple metal layers, starting from the lowest metal layer Metal1. Since these metal layers are also used to interconnect devices on the chip and only 4 metal layers are available, the implementation of VPP capacitors would have the consequence that a lot of chip area would be "wasted", because the control circuit (logic, switches, etc.) of each capacitor would have to be placed elsewhere on the chip. However, the M4M3 MOM capacitor shows a worse linearity for a given unit-capacitance, but it does not occupy the two lowest metal layers, Metal2 and Metal1. This further means that these layers can be used to connect the capacitors to their underlying circuitry. Regarding the unit-capacitance, the simulation results show that a capacitance of C_u = 25fF would be sufficient for both types of capacitors, while maintaining a margin to the specification limit. A larger capacitance would consume more area while improving the linearity of the circuit only by a small value. Considering all addressed circumstances, the M4M3 MOM capacitor with a unit-capacitance of C_u = 25fF is chosen as a unit-capacitor.

Number of Unit-Capacitors

The number of unit-capacitors per branch, from here on denoted as N, can be determined by calculating the ratio between the full-scale-range (FSR) of the equivalent capacitance determined by the PGA and its LSB capacitance:

$$N = \frac{C_{eq,FSR}}{C_{eq,LSB}} = \frac{160\text{fF}}{100\text{aF}} \longrightarrow \boxed{\text{N} = 1600} \tag{3.20}$$

Attenuation Capacitors $\mathbf{C_s}$

Since the value C_u of the unit-capacitors has been defined, the value of the required attenuation A can be calculated:

$$A = \frac{C_{eq,LSB}}{C_u} = \frac{100\text{aF}}{25\text{fF}} = \frac{1}{250} \tag{3.21}$$

With the determined values for the number of capacitors N and the unit-capacitance C_u, the value of the attenuation capacitors C_{s1} and C_{s2} can be determined by using Equation 4.10. However, C_{par} is considered to be zero as no information on parasitics is available at this point. Now, the value of the attenuation capacitors can be determined by solving Equation 3.12 and keeping in mind that $2C$ is equal to the overall capacitance of one branch:

$$C_s = (2C + C_{par})\frac{A}{2-A} = N \cdot C_u \frac{A}{2-A} = 1600 \cdot 25\text{fF}\frac{\frac{1}{250}}{2-\frac{1}{250}} = 80.16\text{fF} \tag{3.22}$$

3.2.4. Design Trade-Off

During the derivation of the concept for the bridge configuration, it became clear that there is a trade-off between switchable capacitance range, LSB capacitance and matching. On the one hand, large capacitors load the ASIC while consuming a large area. On the other hand, they provide a good matching and therefore contribute to a high linearity. However, from a specific capacitance on, the increment of capacitance does not show a strong contribution to linearity anymore, following the described "Pelgrom's model". On the contrary, the unit-capacitors cannot be made too small, because of matching problems and therefore a higher nonlinearity. However, smaller capacitors consume less area and hence a higher number of them could be implemented for a given chip size. Additionally, this would result in a higher switchable range of capacitance. Another advantage would be the lower capacitive loading of the ASIC. It becomes clear that the trade-off strongly depends on the prioritization of the parameters. In context of this work, the linearity is considered to be the most stringent parameter of the programmable capacitor, followed by the switchable range of capacitance. Therefore, the consumed area is considered to be less stringent to a certain extent, and the unit-capacitor and its value are chosen the way it is described in the previous chapter.

3.3. Programming Concept

The last part of the concept process is the derivation of a suitable concept that provides an efficient way of programming the chip. Once the chip is bonded to the ASIC, the programmable capacitor receives binary values from an external microcontroller and according to these values, the programmable capacitor switches the unit-capacitors. Thus, a main part of the logic complexity can be outsourced to the microcontroller. The following sections describe the derivation of a concept that provides a simple way to program the programmable capacitor. However, a more detailed perspective on the logic part is given in the chapter of implementation.

3.3.1. Addressing of Capacitors

Each branch of the programmable capacitor consists of 1600 switchable capacitors, which means that 1600 control signals would be needed if every capacitor is addressed at its own. Due to the crossing of $v_{ref,p}$ and $v_{ref,n}$ between the left and the right branch, these control signals could be used for both branches. Nevertheless, 1600 control signals would not only require a large logic overhead, but also a large wiring expense would have to be considered. This indicates the drawback of the thermometer-coded switching-scheme. However, this issue can be tackled by addressing each capacitor not by its own control signal, but by row and column signals. Considering that the capacitors are arranged in a square configuration in each branch, denoted as "matrix", only 40 row and 40 column signals are required to address each capacitor. Since the control signals are used for both branches, a number of 80 control signals are in total sufficient to address the capacitors in both matrices of the programmable capacitor.

3.3.2. Binary-to-Thermometer Encoder

As 80 control signals would represent a high number of needed pins for the programmable capacitor, it is favourable to translate a programmed binary value into the row and column signals to address the capacitors. In order to achieve this, the binary values for the rows and the columns are programmed into the programmable capacitor by means of shifting the bits into two shift registers, one for the rows and one for the columns. Basically, these shift register convert a serial bit stream into a parallel bit stream and the latter is fed into a binary-to-thermometer encoder. In general, a binary-to-thermometer encoder is a logic circuit that converts a binary value into a thermometer-coded value [29]. For example, if a binary value of the decimal value 30 is applied to the input of a binary-to-thermometer encoder, 30 consecutive outputs of the encoder are high. Hence, the maximum number of outputs of the encoder is limited by the number of bits at its input. For the case of the programmable capacitor, each binary-to-thermometer encoder must have 40 outputs, in order to address the 40 row and column signals respectively. Therefore, the following number of bits are needed, where N_R and N_C are the number of row and column signals:

$$N_{bin} = \log_2(N_R) = \log_2(N_C) = \log_2(40) = 5.322 \text{ bit} \longrightarrow N_{bin} = 6 \text{ bit} \tag{3.23}$$

Therefore, each binary-to-thermometer encoder requires 6 bit at its inputs, which leads to a maximum number of outputs of:

$$N_{th} = 2^{N_{bin}} - 1 = 2^6 - 1 = 63 \tag{3.24}$$

The "-1" in the equation stems from the fact that if the binary value represents the decimal value "0", no thermometer-coded outputs are active and therefore no output is needed for that code.

For the given number of rows and columns, only 40 out of the maximum 63 thermometer-coded outputs are needed, which simplifies the logic of the binary-to-thermometer encoder. However, deriving the logic equations for a binary-to-thermometer encoder is still not straight forward, because there is no specific "mapping" between the binary and the thermometer-coded value. A way to derive those equations is shown in subsubsection 4.6.1 at a later point.

Figure 3.9 shows the basic programming concept. A microcontroller serially sends binary values for the number of rows and columns to switch on. These bits are converted from serial to parallel and are then applied to the input of the respective binary-to-thermometer encoder. According to the programmed value, the encoders set the addressed rows and columns that further address the capacitors in the matrices. However, the description of the implemented logic for each capacitor along with a more detailed explanation of the control signals and the corresponding logic can be found in the chapter of implementation.

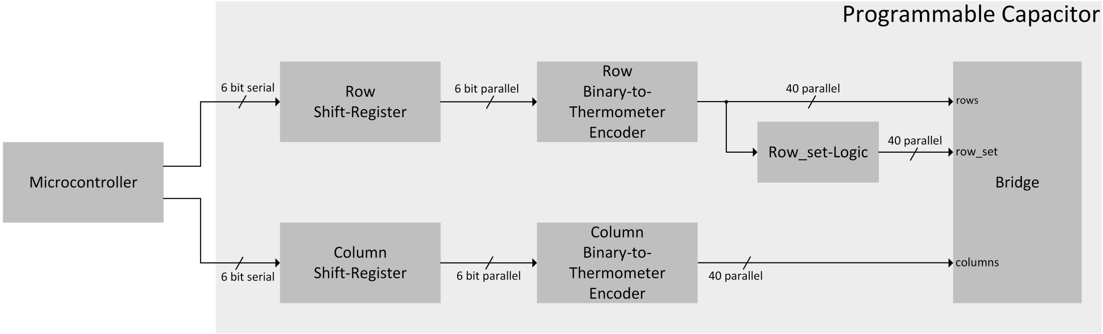


Figure 3.9.: Concept to program the programmable capacitor. The microcontroller serially sends binary values for rows and columns. These signals are encoded to a thermometer-code that sets the rows and columns in the bridge. For simplicity, the blocks along with their relevant connections are shown on an abstract level.

4. Implementation of the Programmable Capacitor

This chapter describes the implementation of the previously derived concept for the programmable capacitor. Since the implemented circuit consists of several sub-blocks, it is considered to be clearer if a "top-to-bottom" approach is used to describe the chip. This means that first the top-level circuit and its functionality is described in a more abstract way, followed by the description of each sub-block in more detail.

4.1. Types of Implemented Programmable Capacitors

It is of interest how the chip area influences the matching between the capacitors, as the chip might show some process gradients when a larger area is considered. However, such process effects are not taken into account by means of simulation, which leads to the fact that the chip might not fulfil the specifications later. Therefore, two versions of the programmable capacitor are implemented. One version consists of 625 switchable capacitors per branch while the other version consists of 1600 switchable capacitors per branch. Consequently, this results in different areas per type of programmable capacitor. Even though the smaller programmable capacitor does not fulfil the specification of the switchable capacitance range, it would still be helpful to verify the ASIC when the process gradients would make the larger bridge unusable. However, the programmable capacitor with 1600 capacitor per array is considered as the main device and the description of the implementation amounts to this version. The smaller device basically consists of the same parts as the large one, with the difference that only 25 outputs of the binary-to-thermometer encoder logic is used instead of the 40 used in the larger device.

4.2. Top-Level Configuration

As a "top-to-bottom" approach is used to describe the implemented programmable capacitor, its top-level configuration is shown in this section first. In the following sections, the block "Left Matrix" represents the ΔC_{12} capacitor-array, while the block "Right Matrix" represents the ΔC_{34} capacitor-array.

4.2.1. Block Diagram

From a top-level point of view, the programmable capacitor consists of a logic block, denoted as "Bridge Logic", as well two capacitor-arrays that represent the branches of the attenuated full-bridge. The firmer block is used to address the capacitors according to the programmed value and consists of several sub-blocks that are described in more detail in section 4.6. Each capacitor-array consists of multiple capacitor-switch cells which are described in section 4.3. Figure 4.1 shows the top-level block diagram of the programmable capacitor including the mentioned blocks. On the left hand side of the figure, all inputs of the programmable capacitor are shown, while at the right side, both outputs are depicted. For clarity, the connections for the supply voltages VDD and VSS are not shown. The attenuation capacitors C_{s1} and C_{s2} are connected to the capacitor-arrays, respectively. Another important thing to note is the crossing of $v_{ref,p}$ and $v_{ref,n}$ between both capacitor-arrays, whose purpose has already been described in subsection 3.2.2.

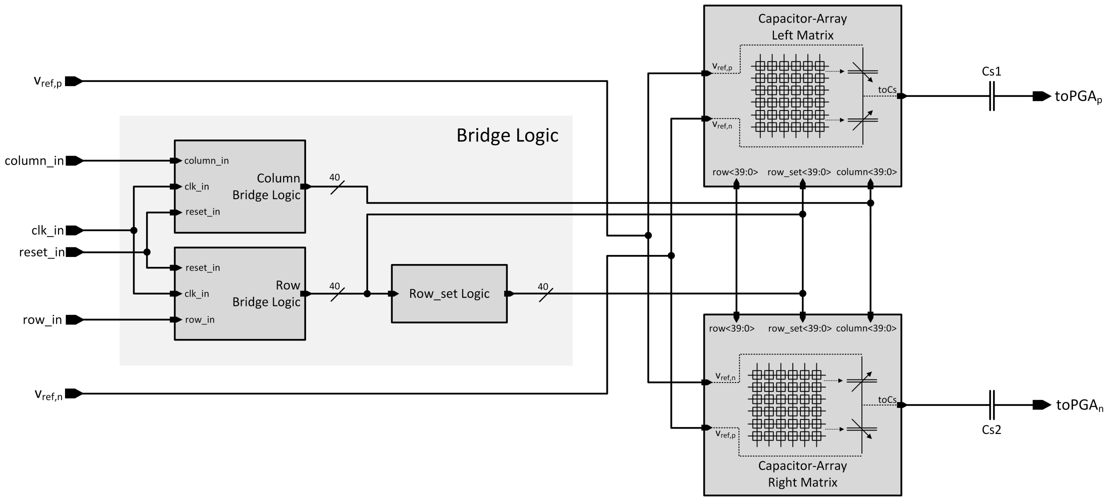


Figure 4.1.: Top-Level view of the implemented programmable capacitor. The bridge logic addresses the capacitors according to the programmed value. Each capacitor-array consists of multiple capacitor-switch cells.

4.3. Capacitor-Array Configuration

Both capacitor-arrays of the programmable capacitor are equal and represent the main part of the chip. In each array, the capacitors are differentially either switched to $v_{ref,p}$ or $v_{ref,n}$, depending on the programmed value. Due to the crossing of $v_{ref,p}$ and $v_{ref,n}$ between the arrays, they can be addressed by the same logic while still providing a differential behaviour between them.

4.3.1. Block Diagram

Each capacitor-array consists of 1600 capacitor-switch cells (CS-cell) which are described in section 4.4 as well as dummy-cells, described in section 4.5. The CS-cells are arranged in a quadratic matrix configuration, which allows to use a logic block later that is identical for both rows and columns. The output *toCs* of each cell (dummy- and CS-cell) is interconnected throughout the whole matrix. For a better visibility, this node is not interconnected in the block diagram, but the interconnection is indicated by giving each output of the cells the same name. As Figure 4.2 shows, each CS-cell interconnects its regarding input signals to its row-and column-neighbours, respectively. The figure shows one matrix only, however, the second matrix is identical to the matrix shown below. In fact, the second matrix uses the same row- and column-connections, only the nodes *toCs* are different between the matrices, as they connect to the attenuation capacitors C_{s1} and C_{s2}, respectively.

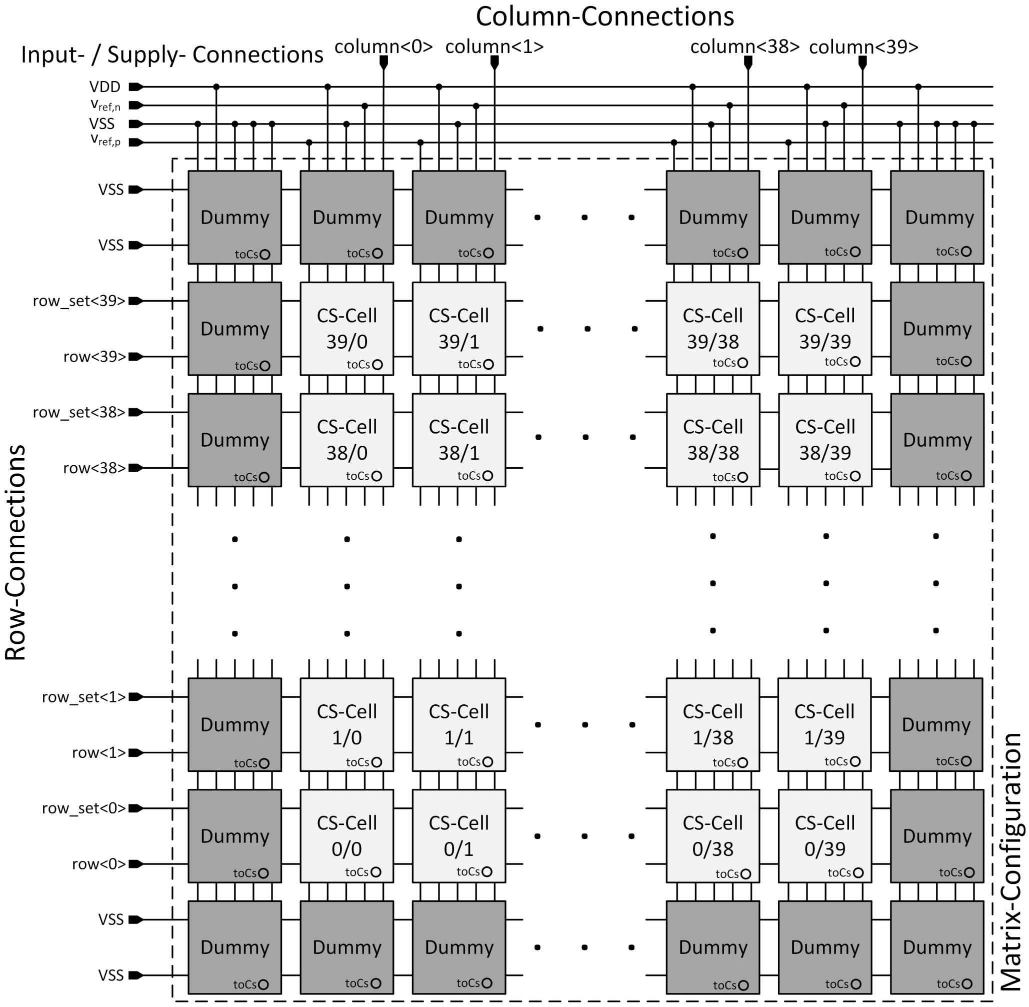


Figure 4.2.: Block diagram of implemented matrix configuration of single cells. Both matrices are identical and use the same row- and column signals. Each matrix consists of 1600 CS-cells, and for clarity this figure only shows the cells at the corners.

Each CS-cell is denoted with two numbers that represent the connected row- and column signals. For example, the CS-cell 39/1 is connected to the row control signal *row*<39> and the column control signal *column*<1>. The purpose of the dummy cells is described at a later point in section 4.5.

4.3.2. Clamping Diodes at Floating Nodes

When looking at the attenuated full-bridge configuration, one can see that the nodes $toCs1$ and $toCs2$ are basically floating, because these nodes are placed between two capacitors, namely the capacitor-array and the regarding attenuation capacitor of the matrix. Consequently, charge cannot enter or exit these nodes and is therefore conserved between the capacitors. This leads to the issue that charge, which might accumulate on the floating nodes during production of the programmable capacitor, can cause an electrostatic potential that threatens the maximum voltage rating of the MOM capacitors. Therefore, the potential of the floating nodes is limited by placing them between clamping diodes, as it is depicted in Figure 4.3. Hence, the potential of the floating nodes is limited to a value between VSS and VDD.
Since the diodes are reverse biased, the nodes $toCs1$ and $toCs2$ are still floating, but their potential is limited and no charge should enter or exit the nodes. Simulations conducted on the extracted layout of the programmable capacitor show no noticeable influence of these diodes onto the linearity.

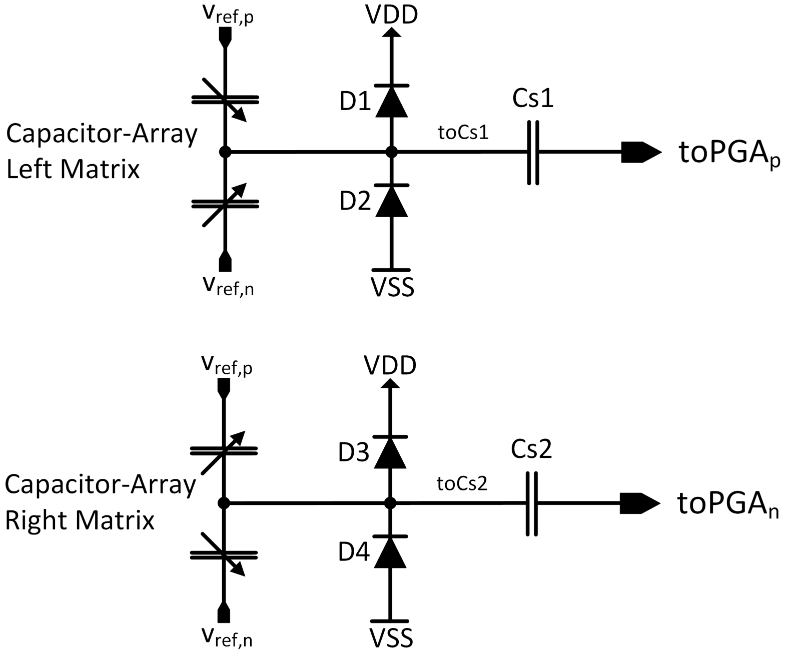


Figure 4.3.: The clamping diodes limit the electrostatic potential of the floating nodes between the capacitor-arrays and their attenuation capacitors to a value between VSS and VDD.

4.3.3. Control Signals and Addressing

General Addressing Description

The purpose of the row- and column-control signals is to address the cells according to the programmed value of the chip. Per default, meaning no value is programmed, all CS-cells in the left matrix are switched to $v_{ref,p}$ while all CS-cells in the right matrix are switched to $v_{ref,n}$. This results in the minimum absolute capacitance of the switchable range. Once values are programmed for the number of rows and columns to switch, these values are converted from binary to thermometer-code (see section 4.6) and are applied to the bridge by means of three different control signals: *row*, *row_set* and *column*. Each CS-cell decodes the control signals applied to its inputs and determines whether it is addressed or not. If the CS-cell is addressed, the capacitor inside the CS-cell switches from $v_{ref,p}$ to $v_{ref,n}$ in the left matrix and inversely in the right matrix. In general, a cell is addressed if both the row and the column signal of the regarding cell is logic high. However, a third signal is needed, named *row_set*, that addresses all CS-cells in a row as soon as the next row is set to logic high. This approach is necessary, because if only row and column signals are used to address the cells in a thermometer-coded way, cells that have already been addressed would also switch and so the differential capacitance would be corrupted. However, this approach ensures that the signals *row* and *column* are only used for the switching of CS-cells in the latest row while the *row_set* signal keeps all previous CS-cells switched. An example provided later in this section makes this addressing concept more understandable and points out its importance.
Table 4.2 shows the truth table of the recently explained addressing concept. The bottom-plate of a capacitor inside a cell is only switched if both *row* and *column* signals are high, or the *row_set* signal is high. The differential behaviour between the left and the right matrix is also noticeable in the table.

Logic Input			**Cap in CS-cell switched to**	
row	**column**	**row_set**	**Left Matrix**	**Right Matrix**
0	0	0	$\mathrm{v_{ref,p}}$	$\mathrm{v_{ref,n}}$
0	0	1	$\mathrm{v_{ref,n}}$	$\mathrm{v_{ref,p}}$
0	1	0	$\mathrm{v_{ref,p}}$	$\mathrm{v_{ref,n}}$
0	1	1	$\mathrm{v_{ref,n}}$	$\mathrm{v_{ref,p}}$
1	0	0	$\mathrm{v_{ref,p}}$	$\mathrm{v_{ref,n}}$
1	0	1	$\mathrm{v_{ref,n}}$	$\mathrm{v_{ref,p}}$
1	1	0	$\mathrm{v_{ref,n}}$	$\mathrm{v_{ref,p}}$
1	1	1	$\mathrm{v_{ref,n}}$	$\mathrm{v_{ref,p}}$

Table 4.1.: Truth table for control logic of a single cell. Even though the same control signals are used for both matrices, the differential behaviour is achieved due to the crossing of $v_{ref,p}$ and $v_{ref,n}$.

Microcontroller Considerations

As described in the introduction of this work, the programmable capacitor is later programmed by a microcontroller. In more detail, the microcontroller writes the number of rows and columns to the programmable capacitor, depending on the number N_{sw} of CS-cells to switch. However, the translation of the number of CS-cells to switch into the regarding number of rows and columns is done in the program of the microcontroller. Basically, the program of the microcontroller stores the absolute number of rows N_{rows} as well as the absolute number of columns $N_{columns}$ with both having a value of 40. During operation, the program translates a number of CS-cells to switch N_{sw} into the number of row and column signals that needs to be set to logic high on the programmable capacitor. In order to determine the number of rows that needs to be set to logic high row_{switch}, the program uses the following equation:

$$row_{switch} = \lceil \frac{N_{sw}}{N_{columns}} \rceil \tag{4.1}$$

The ceiling is required so that the next row is set to logic high too. The number of columns that need to be switched to logic high col_{switch} are calculated the following way:

$$col_{switch} = (\frac{N_{sw}}{N_{columns}} - \lfloor \frac{N_{sw}}{N_{columns}} \rfloor) \cdot N_{columns} \tag{4.2}$$

It is noteworthy that Equation 4.2 gives a result of 0 if N_{sw} is a multiple of $N_{columns}$, which is undesired because all columns would need to be set to high. Therefore, the program running on the microcontroller has to query such cases and when they occur, all columns have to be set.

On the microcontroller, both row_{switch} and col_{switch} are converted into binary values and are then given to the programmable capacitor, which is described in more detail at a later point in this work.

Addressing Example

Consider the following example: The chip is programmed so that a number of $N_{sw} =$ 73 CS-cells are switched. By using Equation 4.1 and Equation 4.2, the software on the microcontroller determines that $row_{switch} = 2$ and $col_{switch} = 23$. These values are transferred to the programmable capacitor in a binary way, where they are converted into a thermometer code. Consequently, the signals *row*<0> and *row*<1> are set to high. Since those consecutive signals are set to high, the signal *row_set*<0> is also set to high, causing the switching of the whole first row. For this row, the signal *row_set* overrules the signals *row* and *column*. For the second row, where *row_set*<1> is low, only those CS-cells are switched where both *row* ad *column* signals are set to high. Since *row*<1> and *column*<0> - *column*<22> are set to high, 23 CS-cells are switched in the second row, leading to a total number of 73 CS-cells switched, as intended. This example should point out the necessity of three signals: The signals *row* and *column* are responsible to address the last CS-cells to switch, while *row_set* ensures to switch all previous CS-cells.

4.4. Capacitor-Switch Cell (CS-Cell)

4.4.1. Schematic of the Ideal CS-Cell

An ideal single CS-cell consists of following elements:

- Unit-Capacitor $\mathrm{C_u}$ (Metal4-Metal3 capacitor) whose top-plate is connected to the node *toCs*
- Switch that either connects the capacitor's bottom-plate to $v_{ref,p}$ or $v_{ref,n}$
- Logic that controls the switch depending on logical input signals *row*, *column* and *row_set*

An abstract schematic of an ideal single CS-cell is shown in Figure 4.4, which should help to point out the basic functionality. The switch is implemented as a spdt (single-pole-double-throw) switch, meaning that the capacitor is either switched to $v_{ref,p}$ or to $v_{ref,n}$ with no other state possible.

Basically, the control logic decodes the logic input signals, and depending on their values switches the bottom plate of the capacitor to either $v_{ref,p}$ or $v_{ref,n}$, resulting in a very simple configuration. In case of the abstracted schematic, the capacitor's bottom-plate is arbitrarily switched towards $v_{ref,p}$. The description of the control logic follows in subsection 4.4.4.

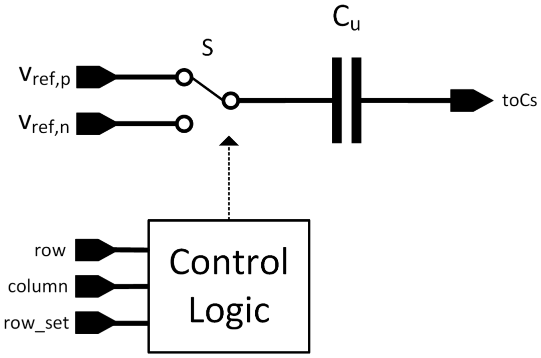


Figure 4.4.: Schematic of an ideal CS-cell. Depending on the logic input signals, the bottom-plate of the capacitor is either switched to $v_{ref,p}$ or $v_{ref,n}$.

Based on this ideal configuration, the CS-cell is implemented as it is described in the next section.

4.4.2. Schematic of the Implemented CS-Cell

Figure 4.5 shows the schematic of the implemented CS-cell and a description is provided in the following sections. For clarity, the supply voltages VDD and VSS of the control logic, as well as the bulk-connections of the transistors are not shown in the schematic.

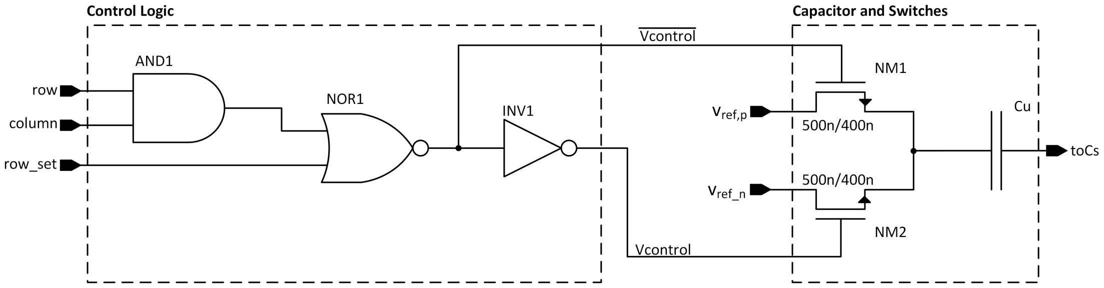


Figure 4.5.: Schematic of implemented CS-Cell including the unit-capacitor, the switches as well as the control logic.

4.4.3. Capacitor and Switches

In an ideal CS-cell, the capacitor is switched by a spdt-switch, which basically connects one terminal to one of two other terminals. In case of the capacitor-switch cell, the spdt-switch connects the bottom-plate of the capacitor to either $v_{ref,p}$ or $v_{ref,n}$. However, in a real implementation, the spdt-switch has to be constructed using two single switches, requiring that these switches switch in an opposite way and therefore maintaining that only one of the two voltages $v_{ref,p}$ and $v_{ref,n}$ is connected to the capacitor's bottom plate at any time. This can be achieved by two different approaches:

- Approach 1: Use the same logic control signal for both switches, and the switches are inherently opposite, for example a NMOS transistor for the one switch and a PMOS transistor for the other switch.
- Approach 2: Use two control signals which are inverted to each other. This allows the use of switches of the same type, for example PMOS or NMOS for both of the switches.

When a switch is turned on, it has to switch the whole voltage range of $v_{ref,p}$ and $v_{ref,n}$, therefore voltages between 0V and 1.2V. Considering the first approach, this would require a negative voltage at the gate of the PMOS transistor in order to switch a voltage of 0V. However, the programmable capacitor is supplied with a unipolar supply voltage of 1.8V only, which means that this approach is inconvenient to realize in this work. Moreover, the second approach realized with two NMOS transistors seems to be the most feasible way to create the spdt-switch, because inverted logic signals are easy to create and NMOS transistors provide a low ohmic path for a positive gate voltage. Therefore, this approach is used inside the implemented CS-cell. Both transistors NM1 and NM2, which represent the spdt-switch, as well as their inverted control signals $Vcontrol$ and $\overline{Vcontrol}$ are depicted in Figure 4.5 . The creation of these control signals is described later in subsection 4.4.4.

Requirements on Transistors

The task of the transistors NM1 and NM2 is to switch the signal $v_{ref,p}$ or $v_{ref,n}$ to the bottom-plate of the unit-capacitor, while avoiding the corruption of the signal to switch. Since the gates of the transistors are connected to the logic signals $Vcontrol$ and $\overline{Vcontrol}$, the voltages at the gates are either 0V or 1.8V. Considering that the analog signals $v_{ref,p}$ and $v_{ref,n}$ are alternating between 0V and 1.2V and the control signal for the switch is logic high (1.8V), the minimum gate-source voltage $V_{GS,min}$ for the closed state of the switch can be calculated as:

$$V_{GS,min} = V_{G,on} - V_{S,max} = 1.8\text{V} - 1.2\text{V} = 0.6\text{V} \tag{4.3}$$

where $V_{G,on}$ is the voltage at the gate for the on state of the transistor and $V_{S,max}$ is the maximum voltage at the source, both with respect to ground. Simulations verify that this gate-source voltage is sufficient to provide a low ohmic path between the nodes $v_{ref,p}$ and $v_{ref,n}$ to the bottom-plate of the capacitor. The maximum gate-source voltage $V_{GS,max}$ for the conducting case of the switch is the difference between the gate voltage $V_{G,on}$ and the minimum source voltage $V_{S,min}$:

$$V_{GS,max} = V_{G,on} - V_{S,min} = 1.8\text{V} - 0\text{V} = 1.8\text{V} \tag{4.4}$$

This voltage is well below the limit of the maximum allowed gate-source-voltage of the used NMOS device. When the control signal of a transistor is logic low (0V), the switch has to operate in the open state and shall not provide any low ohmic path from its drain to its source. In this case, the minimum and maximum gate-source voltages of the switching transistor can also be calculated. Starting with the maximum gate-source voltage for the open state:

$$V_{GS,max} = V_{G,off} - V_{S,min} = 0\text{V} \tag{4.5}$$

where $V_{G,off}$ represents the gate voltage in the open state with respect to ground. Additionally, the minimum gate-source-voltage for the open state is calculated by:

$$V_{GS,min} = V_{G,off} - V_{S,max} = 0\text{V} - 1.2\text{V} = -1.2\text{V} \tag{4.6}$$

Simulations show that both gate-source voltages are sufficiently low to keep the transistor turned off within the whole voltage range of $v_{ref,p}$ and $v_{ref,n}$, hence providing a very high-ohmic path between the nodes $v_{ref,p}$ and $v_{ref,n}$ to the bottom-plate of the capacitor. The dimensions of the transistors are chosen so that they have a width $W = 500$nm and a length of $L = 400$nm, as simulations show a good behaviour for transistors with these dimensions for the voltages determined above.

4.4.4. Control Logic

Each CS-cell contains a control logic block which decodes 3 logic input signals (*row*, *column* and *row_set*) into 2 control signals (*Vcontrol* and $\overline{Vcontrol}$) that are used to control the switches in the CS-cell. Basically, the logic input signals are used to address a single CS-cell in a matrix configuration as it was previously described in section 4.3. For the left matrix, a CS-cell has to connect the bottom-plate of the unit-capacitor to the $v_{ref,n}$ terminal when it is addressed, and to the $v_{ref,p}$ terminal for the default state (non-addressed). Since $v_{ref,p}$ and $v_{ref,n}$ are crossed between the left and the right matrix, exactly the same is true for the right matrix. Again, a CS-cell is considered to be addressed, if its *row* and *column* signals are logic high, or its *row_set* signal is logic high. This leads to the following logic equations for the control voltages *Vcontrol* and $\overline{Vcontrol}$ of the transistors:

$$\begin{aligned} Vcontrol &= row \cdot column + row\_set \\ \overline{Vcontrol} &= \overline{row \cdot column + row\_set} \end{aligned} \tag{4.7}$$

These expressions can be realized by interconnecting an AND, NOR and INVERTER logic block, as it is shown in Figure 4.5. Additionally, Table 4.2 shows the truth table for these logic equations along with the information of the transistor states.

Logic Input			**Logic Output**		**Transistor States**	
row	**column**	**row_ set**	**Vcontrol**	$\overline{\textbf{Vcontrol}}$	**NM1**	**NM2**
0	0	0	0	1	on	off
0	0	1	1	0	off	on
0	1	0	0	1	on	off
0	1	1	1	0	off	on
1	0	0	0	1	on	off
1	0	1	1	0	off	on
1	1	0	1	0	off	on
1	1	1	1	0	off	on

Table 4.2.: Truth table for control logic of a CS-cell, including the states of the transistors NM1 and NM2.

4.5. Dummy-Cell

4.5.1. Description

To achieve a high linearity when consecutively switching through the CS-cells, it is mandatory that the values of the cell's unit-capacitors are as similar as possible. This was already introduced by the term "matching" in subsection 2.1.2.

However, not only the value of the capacitor, but also the values of the capacitor's parasitic capacitances should be as similar as possible for all cells in the matrix. For example, the used Metal4-Metal3 capacitor has a parasitic capacitance from Metal3 to ground as well as from Metal4 to ground among many other parasitic capacitances. These parasitics should be the same for every cell in order not to cause additional nonlinearities. If multiple cells are connected in a matrix configuration, as it is done in this work, then there are additional inter-cell parasitic capacitances introduced. This further means that the value of each unit-capacitor is corrupted to a certain extent by the capacitors of its neighbouring cells. If the corruption is equal in every CS-cell in the matrix, the linearity is not affected. However, a CS-cell placed at the edge of the matrix is corrupted differently than a CS-cell placed in the center of the matrix, because it lacks of neighbouring CS-cells and thus does not exhibit the same parasitics as a CS-cell in the center. At first, the physical design (layout) was done with no countermeasures, and the simulations on the extracted layout showed that the linearity severely aggravated when CS-cells at the edges and corners of the matrices where switched. These issues lead to the design of a so called "dummy-cell", whose purpose is to provide the same parasitic capacitances for the CS-cells placed at the edges and corners of the matrices, as their neighbours placed further inside the matrices. However, these dummy-cells are not switchable between $v_{ref,p}$ and $v_{ref,n}$ and pass all matrix signals to the CS-cells. In order to create a large similarity, the schematic of the dummy-cell, depicted in Figure 4.6, is almost the same as the schematic of the CS-cell, with the only difference that the bottom-plate of the dummy-cell's unit-capacitor is constantly grounded. The top-plate of the dummy-cell's unit-capacitor is connected to the top-plates of all other cells within a matrix.

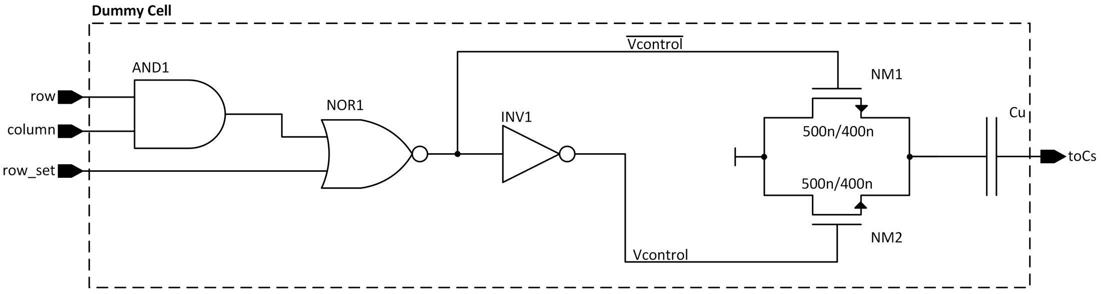


Figure 4.6.: Schematic of the implemented dummy-cell. For clarity, the voltage supply of the logic gates and the bulk connections of the transistors are not shown.

4.5.2. Consequences for Attenuation Capacitors

Since the top-plate of each dummy-cell's unit-capacitor is connected to the top-plates of all other unit-capacitors within a matrix, and the bottom-plates are always grounded, a dummy-cell can be seen as a parasitic capacitance from the node *toCs* to ground, with the same value as a unit-capacitor C_u. Thus, the total capacitance of the dummy-cells to ground in each matrix can be determined as:

$$C_{dummy} = \sum_{n=1}^{N_{dummy}} C_u = N_{dummy} C_u \tag{4.8}$$

where N_{dummy} is the number of dummy-cells in each matrix. In case of 1600 CS-cells in a matrix, this leads to 40 dummy-cells for each edge plus 1 dummy-cell for each corner of the matrix, hence resulting in a total of 164 dummy-cells for each matrix.

In subsection 3.2.2, it was shown that parasitic capacitances from the node *toCs* to ground only result in a gain error. Because the dummy-cells are now considered as parasitic capacitances too, the relation between the change of capacitance ΔC and the equivalent capacitance determined by the PGA changes to:

$$C_{eq} = \frac{2C_s}{2C + C_{par} + C_{dummy} + C_s} \Delta C \tag{4.9}$$

In order to achieve the same attenuation as before and thus eliminating the gain error, the attenuation capacitors have to be redesigned according to:

$$C_s = (2C + C_{par} + C_{dummy}) \frac{A}{2 - A} \tag{4.10}$$

It is worth to mention that the overall capacitance of the dummy-cells C_{dummy} is much larger than the parasitic capacitance C_{par}, which can only be determined after the physical design anyway and therefore leading to an unknown small value as for now.

In Equation 3.22, the value of the attenuation capacitors was determined to $C_{s1} = C_{s2} =$ 80.16fF, however, this value is not valid anymore, as the attenuation would be different. Therefore the attenuation capacitors are redesigned by considering the dummy-capacitors C_{dummy} in the overall capacitance of each matrix and neglecting the parasitic capacitance C_{par}:

$$C_s = (N + N_{dummy}) \cdot C_u \cdot \frac{A}{2 - A} = (1600 + 164) \cdot 25\text{fF} \cdot \frac{\frac{1}{250}}{2 - \frac{1}{250}} = 88.83\text{fF} \tag{4.11}$$

where $N \cdot C_u = 2C$ is the total capacitance in each matrix without the dummy-cells.

4.6. Bridge Logic

As stated earlier, the programmable capacitor receives binary-coded values for rows and columns from the microcontroller. These values represent the number of rows and columns to address. However, in order to address the regarding CS-cells, the values for rows and columns have to be converted into a thermometer-code first, which is the main purpose of the bridge logic. As it is depicted in the top-level-configuration in Figure 4.1, the bridge logic block consists of three sub-blocks: column-bridge-logic, row-bridge-logic and row_set-logic. The first two sub-blocks are identical and convert the programmed binary signals *row_in* and *column_in* into the thermometer-coded signals *row* and *column* which are directly applied to the matrices. The row_set-bridge logic sets the signal *row_set*, depending on the *row* signal.

4.6.1. Column- and Row-Bridge-Logic

Since the column-bridge-logic and the row-bridge-logic are identical, the following section shows a general description of this sub-block. The first step in the logic is to convert the serial data from the microcontroller into parallel signals, which can be realized with a shift register. After that, the parallel output signals of the shift register are applied to a binary-to-thermometer encoder, whose outputs are further applied to the rows or columns of the matrices.

Shift Register

As derived in Equation 3.23 earlier, 6 bit are needed for the addressing of rows and columns, respectively. In order to convert the serial bit stream from the microcontroller into a parallel output, a simple 6 bit shift register represents an efficient solution, not only for the conversion from a serial to a parallel bit-stream, but also for buffering the binary data. Basically, a 6 bit shift register consists of 6 D-FF (data flip-flop) connected in series. For each rising clock-edge, each D-FF switches its input through to its output. Consequently, it requires 6 rising clock-edges to transfer a 6 bit serial stream into its parallel equivalent [30, p. 698]. Figure 4.7 shows the implemented shift register with its three inputs *data_in*, *clk_in* and *reset_in* as well as its 6 outputs *bin_out*<6 : 1>. For the column-bridge-logic, the signal *data_in* represents the top-level signal *column_in*, and in the row-bridge-logic it represents the top-level signal *row_in*. All input signals are connected to the microcontroller, which generates the according signals. In order to be able to reset the whole shift register, each D-FF is connected to the reset line *reset_in*.

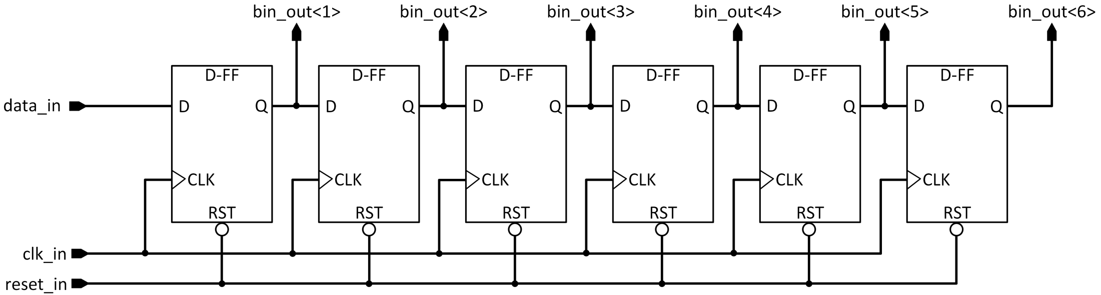


Figure 4.7.: Schematic of the implemented 6 bit shift register, which converts a binary bit-stream into a parallel bit-stream and additionally offers a buffer for the data.

During operation, the microcontroller programs the values for the rows and columns by applying the binary values to *data_in* while also applying a positive clock-edge for each bit, hence, after 6 clock cycles, the data is available at the output of the shift register and can be further processed by the binary-to-thermometer encoder. It is worth to mention that the clock line *clk_in* is used for both the row- and column-bridge-logic, which allows the microcontroller to program the programmable capacitor with one clock only. Also the reset line *reset_in* is used for both logics. Since it is a negative logic, *reset_in* has to be set to logic low in order to reset the flip-flops in the shift register and to logic high during the programming phase.

Binary-to-Thermometer Encoder

As explained in subsection 3.3.2, the purpose of a binary-to-thermometer encoder is to take a binary code and convert it into a thermometer-code. This is basically the reverse function of a thermometer-to-binary encoder, which is heavily used in Flash - ADCs. In this work, the binary-to-thermometer encoder allows the convenient programming of the chip with a 6 bit binary code for rows and columns, respectively. This binary code is translated into 40 thermometer coded outputs by the encoder, which further allows the addressing of the rows and columns in a thermometer coded sense. However, the derivation of the logic functions as well as the implementation of the decoder are not that straight forward, as there exists no logical mapping between a binary- and thermometer-code, which would allow the convenient scaling of the encoder. In fact, each added bit leads to a very large additional complexity in the binary-to-thermometer encoder. Nevertheless, the complexity for the implemented binary-to-thermometer encoder is considered to be acceptable in this work, because instead of 64 possible thermometer-coded outputs, only 40 are needed, which drastically reduces the complexity. Therefore, setting up a truth table for the encoder and deriving the logic equations is manageable by using additional software tools that determine the minimum logic expression for each thermometer-coded output of the encoder. Unfortunately, such a truth table is rather large, and hence a smaller truth table for only 3 bit is shown in Table 4.3.

Decimal	**Binary**			**Thermometer**						
d	**b3**	**b2**	**b1**	**t7**	**t6**	**t5**	**t4**	**t3**	**t2**	**t1**
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	1	1
3	0	1	1	0	0	0	0	1	1	1
4	1	0	0	0	0	0	1	1	1	1
5	1	0	1	0	0	1	1	1	1	1
6	1	1	0	0	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1

Table 4.3.: Truth table for a 3 bit binary-to-thermometer encoder.

For a more convenient placement in the schematics later, the logic equations are brought into a form so that they can be realized with NAND gates only. All derived logic equations can be found in section A.2. Additionally, the implemented schematic for the row- and column-bridge-logic is depicted in section A.3. The schematic is a high resolution pdf, which allows to scroll in for a more detailed view.

4.6.2. Row_set-Logic

In subsection 4.3.3 the necessity of the *row_set* signals has been pointed out. Basically, the generation of these signals is very simple, as a *row_set* signal is only logic high, when the current and the next *row_signal* are logic high. This relation can be defined by the following logic expression:

$$row\_set{<}i{>} = row{<}i{>} \cdot row{<}i+1{>} \tag{4.12}$$

where i is an integer between 0 and 39. Figure 4.8 shows a part of the implemented row_set-logic, which is repeatedly the same for all other signals.

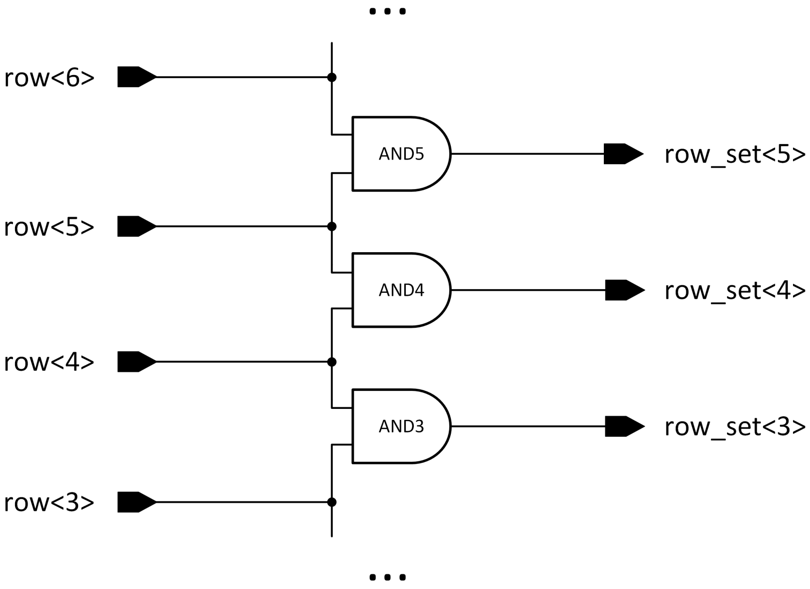


Figure 4.8.: Schematic of logic that creates the *row_set* signals depending on the *row* signals. Only a part of the logic is shown, as the logic is repeatedly the same for all other signals.

It is noteworthy that the signal *row_set*<39> would not be needed, because the last row is addressed by the *row* and *column* signals anyway. However, since unit CS-cells are later used in the layout, this signal is set to logic low in order to provide a defined potential for the wires in the regarding CS-cells.

4.7. Layout of Programmable Capacitor

Each of the described sub-blocks of the programmable capacitor has been physically designed and merged together to a top-level layout. The layout of the CS-cell, the two versions of the programmable capacitors as well as the top-level layout of the whole chip can be found in section A.5. The most effort was spent on designing the CS-cells so that they can be placed easily in a matrix configuration. Additionally, the top-level layout has been verified by means of DRC (design rule check) and LVS (layout versus schematic).

4.7.1. Design Rule Issue and Adjustment of C_u

In production, the metal density of metal layers is usually controlled by two process steps. If the metal density is too high in a specific region, it is reduced by cutting holes into the respective layers. Additionally, in regions where the metal density is too low, extra metal is filled into those regions in order to increase the metal density. However, this is unwanted for the programmable capacitor, because due to cutting, the values of the unit-capacitors would not be equal anymore. Furthermore, filling metal between capacitors might cause parasitic capacitances which could have a random value with unknown impact on the linearity.

Therefore, these production processes are turned off for the programmable capacitor by means of indication layers in the layout. Unfortunately, this results in the necessity that the metal layers meet the design rules regarding the metal density, which consequently leads to the fact that the unit-capacitance of $C_u = 25\text{fF}$ is too large to meet the metal density rules. The largest unit-capacitance which does not cause violations of the design rules is $C_u = 14.4\text{fF}$. By using the mismatch data from previous investigations, it was determined that this unit-capacitance is still sufficiently large enough to meet the requirements on linearity. Therefore, the unit-capacitors are changed to that value in order for all design rules to be met.

4.7.2. Parasitic Extraction and Redesign of C_s

The change of the unit-capacitance from $C_u = 25\text{fF}$ to $C_u = 14.4\text{fF}$ requires the change of the attenuation from $A = \frac{1}{250}$ to $A = \frac{1}{144}$ in order to keep the LSB of C_{eq} constant at 100aF. Thus, the attenuation capacitors C_s have to be adapted accordingly. Since the parasitics in the layout can be extracted at this point, their impact on the value of C_s can also be considered now. Over the Metal4-layer of the programmable capacitor, an aluminium layer is placed and connected to ground, which allows a proper shielding of the chip. However, this layer causes a large parasitic capacitance from the node *toCs* to ground of $C_{par,alu} = 8.393\text{pF}$. Since this parasitic capacitance has a constant value to ground, it can be treated like the dummy-cells. Additionally, this parasitic capacitance is much larger than than the parasitics discussed in subsection 3.2.2, hence the latter are neglected in the redesign of the attenuation capacitors C_s:

$$\begin{aligned} C_s &= (2C + C_{par,alu} + C_{dummy})\frac{A}{2-A} = ((N + N_{dummy}) \cdot C_u + C_{par,alu}) \cdot \frac{A}{2-A} \\ &= ((1600 + 164) \cdot 14.4\text{fF} + 8.393\text{pF}) \cdot \frac{\frac{1}{144}}{2 - \frac{1}{144}} = 117.750\text{fF} \end{aligned} \tag{4.13}$$

For the sake of completeness, the attenuation capacitors for the smaller programmable capacitor with 625 CS-cells per matrix was adapted to a value of $C_s = 48.741\text{fF}$ by considering a parasitic capacitance of $C_{par,alu} = 3.491\text{pF}$ as well as 104 dummy-cells.

5. Final Simulation and Verification

5.1. Final Simulation Setup

Figure 5.1 shows the final simulation setup to verify the functionality of the programmable capacitor and check if there are any systematic nonlinearities. The shown additional blocks are the same as they are used in the ASIC. In the simulations, the programmable capacitor is placed at the position of the MEMS it needs to emulate. This gives a simulation capability that is the closest to reality. In the following sections, the used additional blocks are briefly described.

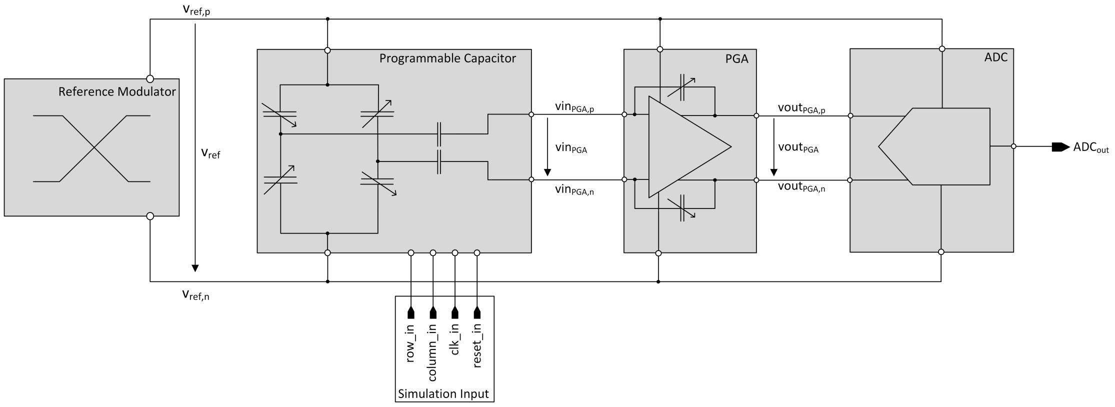


Figure 5.1.: Final simulation setup including the refmod, programmable capacitor, PGA and ADC.

5.1.1. Block: Reference Modulator

The reference modulator generates the excitation signals $v_{ref,p}$ and $v_{ref,n}$. These signals are inverted to each other and alternate between 0V and 1.2V. Since these voltages are applied to all main blocks, the equivalent capacitance C_{eq} is independent of the amplitude of those signals, as it will be shown in the equations later.

5.1.2. Block: Programmable Capacitor

This block represents the implemented programmable capacitor, thus the result of this work. Simulations can either be conducted with the schematic, which is basically the ideal case, or with the extracted layout, which gives a more realistic simulation. During simulations, the programmable capacitor can be programmed the same way as the chip later, by applying the following signals at the input: *row_in*, *column_in*, *clk_in* and *reset_in*. In the programmable capacitor, these signals are applied to the shift registers for rows and columns, respectively. The speed of the programming can be chosen arbitrarily. Usually, a full sweep through the switchable capacitance is done during a simulation, which consequently leads to a ramp of C_{eq}.

5.1.3. Block: PGA

In reality, the PGA represents the interface between the programmable capacitor and the ASIC. The main purpose of the PGA is to integrate the charge at the output of the programmable capacitor which leads to a voltage at the output of the PGA. The programmable gain of the PGA can be set by its feedback capacitors C_{fb}. However, the programmable capacitor is designed in a way so that the LSB of C_{eq} is 100aF for a default feedback capacitor value of $C_{fb} = 1.04$pF. Therefore, the feedback capacitors are set to this value in the simulations too. Nevertheless, the possibility to change the value of the feedback capacitors gives some sort of reliability in the real application, as the impact of a deviation of the attenuation capacitors C_s might be diminished by setting the gain accordingly. From the output voltage of the PGA $vout_{PGA}$, a direct calculation of C_{eq} is possible, as it has been shown extensively in chapter 3.

5.1.4. Block: ADC

In the ASIC, the output voltage of the PGA $vout_{PGA}$ is converted into the digital domain by an ADC. This ADC has a gain of $G_{ADC} = 0.775$, which has to be considered when calculating C_{eq} from the output ADC_{out} of the ADC later. Basically, ADC_{out} represents the output $vout_{PGA}$ of the PGA referred to the reference voltage v_{ref} and scaled with the gain of the ADC.

5.1.5. Calculation of $\mathbf{C_{eq}}$ from the Simulation Results

It is noteworthy that the PGA and the ADC will introduce nonlinearities by themselves, thus it is of interest to check if the overall nonlinearity contains systematic impacts. Therefore, the output ADC_{out} of the ADC is the most valuable simulation result, as it represents the last value of the simulated chain of blocks. The input of the ADC is basically the output of the PGA and the following relation can be obtained:

$$ADC_{out} = G_{ADC} \cdot \frac{vout_{PGA}}{v_{ref}} \longrightarrow vout_{PGA} = \frac{ADC_{out} \cdot v_{ref}}{G_{ADC}} \tag{5.1}$$

From previous derivations, the following relation between the output $vout_{PGA}$ of the PGA and the equivalent capacitance C_{eq} between the branches is known:

$$vout_{PGA} = v_{ref} \cdot \frac{C_{eq}}{C_{fb}} \longrightarrow C_{eq} = C_{fb} \cdot \frac{vout_{PGA}}{v_{ref}} \tag{5.2}$$

By inserting Equation 5.1 into Equation 5.2, a direct relation between the output ADC_{out} of the ADC and the equivalent capacitance C_{eq} can be established:

$$C_{eq} = C_{fb}\frac{vout_{PGA}}{v_{ref}} = C_{fb}\frac{ADC_{out} \cdot \cancel{v_{ref}}}{G_{ADC} \cdot \cancel{v_{ref}}} = C_{fb}\frac{ADC_{out}}{G_{ADC}} = 1.04\text{pF} \cdot \frac{ADC_{out}}{0.775} \tag{5.3}$$

Equation 5.3 shows that the result of C_{eq} is independent of v_{ref}. This is achieved by relating the ADC to the same voltages $v_{ref,p}$ and $v_{ref,n}$ the bridge is excited with.

5.2. Programming Signals

In simulations, the programmable capacitor is programmed the same way as it will later be programmed in the laboratory, namely by applying time dependent signals at its inputs. As it was derived in subsection 4.3.3, the microcontroller calculates the values col_{switch} and row_{switch} for a given number N_{sw} of CS-cells to switch. These values are converted into binary values by the microcontroller and are then shifted into the programmable capacitor by means of simultaneously clocking them into the *row_in* and *col_in* input. In order to ensure stable signals for the rows and columns at every rising clock edge, the clock rises in the middle of every applied bit. Since the shift registers in the bridge-logics are built in a way that they expect the MSB of *row_in* and *col_in* at their last output, the MSB values have to be shifted into the programmable capacitor first, so that after 6 clock cycles, the outputs of the shift registers represent *row_in* and *col_in*, respectively. Figure 5.2 shows a programming example for a number of $N_{sw} = 1386$ CS-cells to switch. The arbitrarily chosen number of N_{sw} leads, according to Equation 4.1 and Equation 4.2, to $row_{switch} = 35$ ($\widehat{=}$0b100011) and $col_{switch} = 26$ ($\widehat{=}$0b011010). For each value of N_{sw}, the signals are applied as shown in Figure 5.2. The time T_{6bit} to program the 6 bit can be chosen arbitrarily.

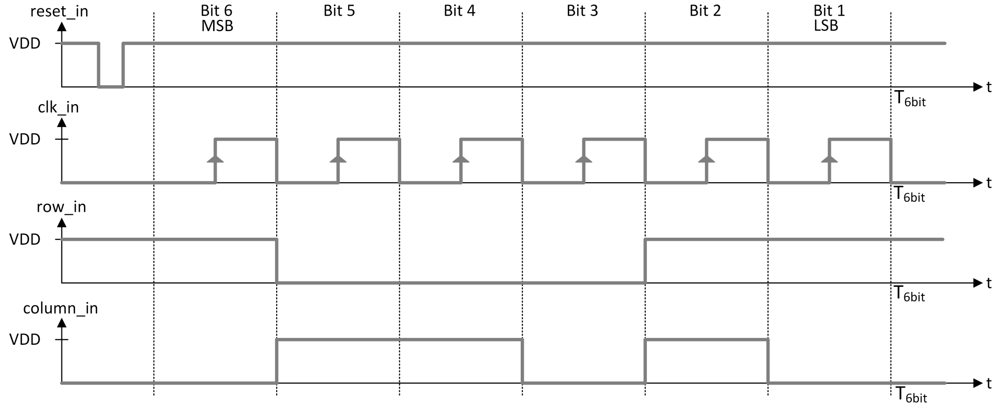


Figure 5.2.: Examples of programming signals as a simulation input. The signal *row_in* represents a value of 35 and the signal *column_in* represents a value of 26. Thus, a number of $N_{sw} = 1386$ CS-cells are switched to $v_{ref,n}$ in the left branch and to $v_{ref,p}$ in the right branch.

5.3. Simulation Results

During a simulation run, the output ADC_{out} of the ADC is determined for every number N_{sw} of switched CS-cells and is stored in a table. By implementing Equation 5.3 in a MATLAB model, C_{eq} can be determined for every value of N_{sw} which further allows to calculate the DNL(C_{eq}) and therefore the possibility to check the linearity for any systematic nonlinearities. It is important to mention, that such simulations have been extensively carried out before the layout of the programmable capacitor was done. However, it is considered that the extracted version, including all parasitics, gives the behaviour that is closest to reality. Therefore, only simulation results for this case are shown in this section. Another important notice is, that the mismatch of the capacitors cannot be simulated for the extracted view. It would be possible for the schematic, however, it would take an unacceptable amount of time to check a large number of implemented programmable capacitors on linearity issues caused by mismatch of the unit-capacitors. Therefore, the unit-capacitors were chosen in a way that there is margin for nonlinearities caused by the PGA and ADC.

$\mathrm{C_{eq}}$ for Full Sweep Through CS-Cells

In order to verify the functionality of the programmable capacitor, a sweep between $N_{sw} = 0$ and $N_{sw} = 1600$ CS-cells for the extracted post-layout schematic of the programmable capacitor is performed and C_{eq} is calculated. From Figure 5.3 it can be seen that C_{eq} slightly exceeds the specified full-scale range of 160fF, because the average value of its LSB is $\overline{\mathrm{LSB}\,(\mathrm{C_{eq}})} = 102.53\mathrm{aF}$ instead of 100aF. However, the slight deviation of the resolution is considered to be neglectable.

Differential Nonlinearity of $\mathrm{C_{eq}}$

The plot for C_{eq} shows that the programmable capacitor achieves the specification on the full-scale range of C_{eq}. Additionally, no significant nonlinearities can be obtained in the plot. However, in order to check if the linearity is not impacted by systematic nonlinearities, the differential nonlinearity of C_{eq} is calculated with the same simulation data and the result is plotted in Figure 5.4. The plot shows that the DNL(C_{eq}) of the programmable capacitor does not contain any systematic nonlinearities, and only random nonlinearities are introduced by the PGA and ADC.

Discussion of Simulation Results

Due to long simulation times (approximately one week for a whole sweep through all 1600 CS-cells), it is not possible to also consider the mismatch of each capacitor in the simulations. Hence, the plots for the simulation results are simulated with capacitors with no mismatch model but with extracted parasitics. This means that the plotted differential nonlinearity stems from introduced nonlinearities of the PGA and ADC, for example the quantization noise of the latter. Also the limited simulation accuracy is considered to cause a random nonlinearity. However, the model simulations in MATLAB

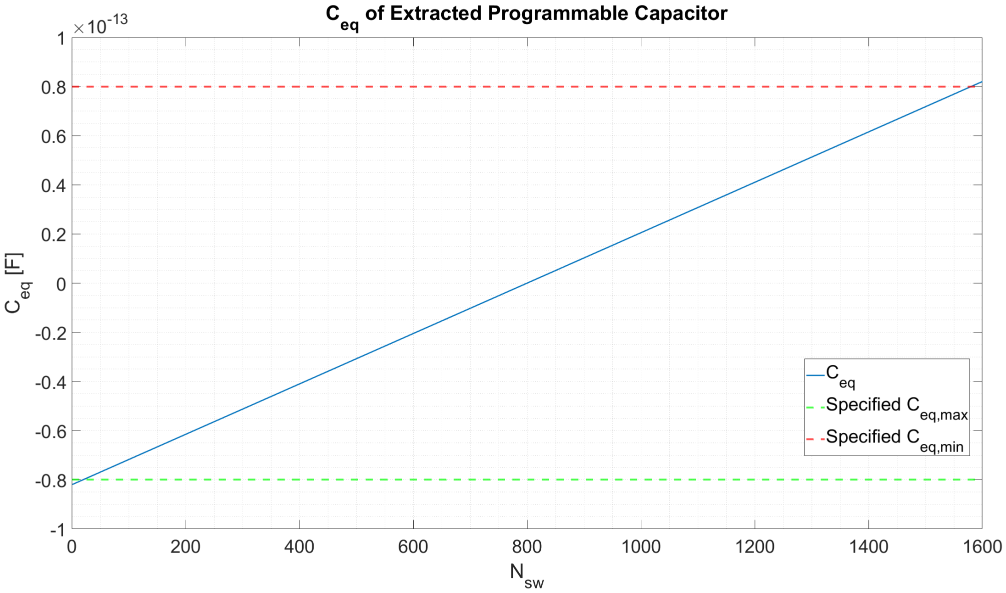


Figure 5.3.: Simulation result for C_{eq} for a whole sweep through all 1600 CS-cell per branch of the extracted programmable capacitor. The result shows that the programmable capacitor meets the required full-scale range for C_{eq} and the average resolution is determined to be $\overline{\mathrm{LSB}\,(\mathrm{C_{eq}})} = 102.53\mathrm{aF}$.

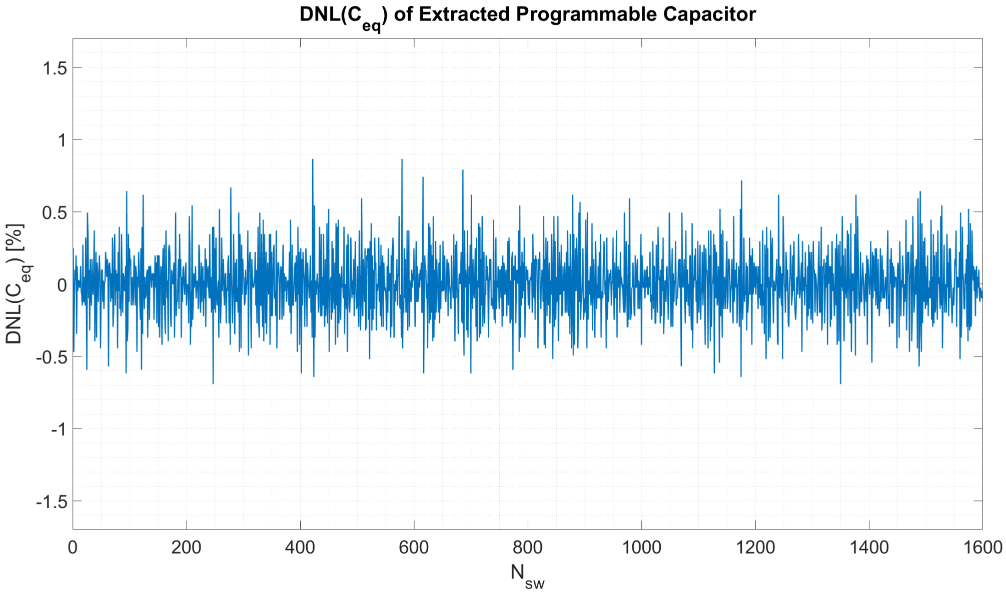


Figure 5.4.: Simulation result for DNL(C_{eq}) for a whole sweep through all 1600 CS-cell per branch of the extracted programmable capacitor. The result shows that the programmable capacitor does not cause systematic nonlinearities and only random nonlinearities by the PGA and ADC are observed.

showed that the used capacitors show a very good matching and therefore a low impact on the linearity. Thus, the linearity of the programmable capacitor was confirmed by MATLAB simulations, while these final simulations confirms that there are no systematic nonlinearities observable. Further, it is considered that averaging measurement results in the laboratory will reduce the quantization noise of the ADC.

6. Conclusion and Outlook

In this work, the design and implementation of a programmable capacitor, that emulates a MEMS pressure sensor, has been shown. First, different types of integrated capacitors were investigated and their advantages and disadvantages have been discussed. Considering all investigated types, the MOM capacitor was chosen to be the capacitor type that fits the requirements, as it shows acceptable matching properties and allows the placement of devices beneath it when higher metal layers are used for the MOM capacitor.

After the investigation of the types of capacitors, different ways on how to switch those capacitors by means of capacitor-arrays are discussed. The binary-weighted capacitor-arrays show a low number of switches, but the DNL is large. On the contrary, thermometer-coded capacitor-arrays show excellent linearity but each unit-capacitor requires a switch and therefore the wiring overhead is drastically increased. Hybrid-coded capacitor-arrays represent the trade-off between binary-weighted and thermometer-coded capacitor-arrays, but their linearity is still too large. Since the specification on the linearity is the most stringent in this work, it was decided that the thermometer-coded capacitor-array is the capacitor-weighting scheme that is able to meet the specifications.

Due to the fact that the programmable capacitor is excited by the same voltages as the MEMS pressure sensor and should provide the same output behaviour, it is favourable that the configuration of the programmable capacitor does not deviate too much from the half-bridge configuration of the MEMS pressure sensor. Therefore, different bridge configurations have been investigated. Starting with the half-bridge configuration, which exactly matches the MEMS pressure sensor, it was derived that the equivalent capacitance is directly proportional to the change of capacitance in the branches of the bridge. This would require to switch very small capacitors of only 100aF in order to meet the specifications. Since such small capacitors show a very bad matching behaviour, this configuration was discarded. The same is true for the full-bridge configuration which would require even smaller capacitors. However, also the attenuation of both the half- and full-bridge have been investigated, with the conclusion that the attenuated half-bridge is nonlinear and the attenuated full-bridge is linear. Therefore, the attenuated full-bridge allows the switching of larger capacitors, which show a better matching, while offering a low change in equivalent capacitance at its outputs. Hence, the attenuated full-bridge was determined to be the most suitable configuration to be implemented.

In order to determine the value of the unit-capacitor, a MATLAB model was created. This model implements the derived equation for the attenuated full-bridge and realizes the change in capacitance by a thermometer-coded capacitor-array. In this array, each capacitor is normally distributed with different unit-capacitances and their according standard deviations, where the values for the standard deviations were taken from pre-

vious mismatch simulations. By calculating the standard deviation of the DNL over 500 simulated runs, a decision on the required unit-capacitor value was made. However, due to design rule issues in the layout at a later point, the value of the unit-capacitor was decreased, while still providing sufficient linearity. As a last step in the concept phase, a suitable concept for programming the programmable capacitor was derived. It was determined that a matrix configuration of the capacitor-arrays would make the addressing of each capacitor much easier, as less control signals are required. This further allows the programming of the programmable capacitor by binary signals that represents the row and columns of a matrix configuration. These binary signals are converted into thermometer-coded signals and are then applied to the row and columns.
The implementation of the programmable capacitor was described in a top-to-bottom approach, meaning that the top-level has been discussed first, followed by the matrix configuration and its logic blocks and finally concluded by the descriptions of a single capacitor-switch-cell and a dummy-cell. The top-level of the programmable capacitor consists of two matrices, which represent the capacitor-arrays of the bridge's left and right branch. Both matrices are connected in a way so that they can be addressed by the same logic signals while maintaining their differential behaviour. However, each matrix consists of 1600 CS-cells and each cell can be addressed by logic signals. These logic signals are generated in the "Bridge-Logic" block, where binary values from the microcontroller are translated into thermometer-coded signals, that allows to address each cell in the matrices. In order to be able to switch each CS-cell on its own, a third logic signal is generated beside the signals for the row and columns. This signal switches a whole row, once two neighboured rows are switched. In other words, if a whole row should be switched on, all CS-cells in this row are switched and the signals for the row and columns are used for the next row. This provides an efficient thermometer-coded switching of the CS-cells. The lowest implemented layer is represented by the CS-cell, which basically consists of the unit-capacitor, the switches as well as a logic circuit. The CS-cell uses the logic circuit to determine whether it is addressed or not depending on the three logic signals at its input. Depending on the outcome of the logic, the capacitor is switched two one of the two reference voltages. After the first matrix was created by means of physical design, it was found out that the CS-cells at the edges and corners of the matrix introduce nonlinearities. Due to missing neighbouring CS-cells, their capacitance is different from the CS-cells located in the center of the matrix. In order to overcome this issue, a dummy-cell was created. Basically, each dummy-cell is similar to a CS-cell, with the difference that its capacitor's bottom-plate is not switchable but is constantly switched to ground. Those dummy-cells are placed around the matrix and simulations of extracted matrix layouts showed that linearity issues have vanished.
After all schematics have been verified, the physical design of the programmable capacitor was done. Here, a strong focus was laid on the layout of the single CS-cell, as a lean layout allows the convenient placement of 1600 CS-cells in the layout of the matrix configuration. Once the layouts of the sub-blocks was created, they were placed in the top-level configuration that also consists of pad structures and an aluminium layer. The extraction of this top-level layout considers the most important parasitics and therefore it can be considered to be closest to the produced programmable capacitor later.

Finally, simulations have been conducted on the extracted layout of the programmable capacitor. The simulations have been carried out with sub-blocks that are implemented in the ASIC, to keep the simulations closest to reality as possible. From the simulation results it can be seen that neither the PGA nor the ADC introduces large systematic nonlinearities. However, it is important to mention that the capacitors and their parasitics have the value that was extracted from the layout and therefore no mismatch is modelled during these simulations. If the mismatch would be modelled for each number of switched CS-cell, the simulation would take an unacceptable amount of time. Since the impact of mismatch of the capacitors on the linearity has been investigated by the MATLAB model, it can be estimated that the overall linearity, including programmable capacitor, PGA and ADC is well within the specifications also including a margin.

By the time this paragraph is written, the programmable capacitor has been taped-out and masks are currently created. Once the masks are available, the programmable capacitor will be produced and shipped to the laboratory. There, its functionality will be verified. If the specifications are met, the programmable capacitor will then be used to verify the ASIC.

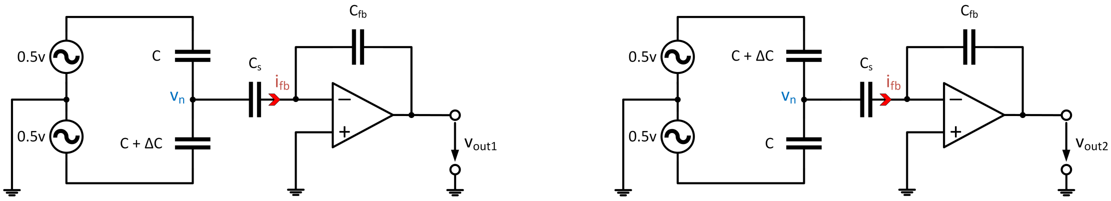
A. Appendix

A.1. AC Signal Derivations of Bridge-Configurations

In the following subsections, the relation between the output of the PGA and the differential change of capacitance in the investigated bridge configurations is derived. As a convention, all lower-case letters are AC signals. Since the frequency is the same for all parameters, it is omitted in the most derivations, which should increase the clearance. However, in derivations where the frequency is considered (e.g. impedance of attenuation capacitors), it can be seen that the frequency cancels in the subsequent calculations.

A.1.1. Attenuated Half-Bridge

The analysis of the relation between the capacitance change ΔC and the output of the PGA is a bit more sophisticated because of the series connection of the attenuation capacitors C_s. Therefore, the AC signal equivalent circuit, as it is shown in Figure A.1, is used.



(a) Left branch

(b) Right branch

Figure A.1.: AC signal derivation for both branches of the attenuated half-bridge.

Derivation of the Left Branch

Since the non-inverting input of the PGA is fixed to a common-mode voltage, it is considered as GND from an AC point of view. The voltage at node v_n can be determined by super-positioning both voltage sources and determining the capacitive voltage divider:

$$v_n = v_n' + v_n'' = \frac{1}{2}v\frac{C}{2C + \Delta C + C_s} - \frac{1}{2}v\frac{C + \Delta C}{2C + \Delta C + C_s} = -\frac{1}{2}v\frac{\Delta C}{2C + \Delta C + C_s} \quad \text{(A.1)}$$

where v_n' and v_n'' represent the superpositioned values of the node v_n. Since the PGA tries to minimize the voltage between non-inverting and inverting input, the inverting input is considered to be quasi-zero. This allows the calculation of the current through the attenuation capacitor C_s:

$$i_{fb} = \frac{v_n}{\frac{1}{j\omega C_s}} = -\frac{1}{2} v j\omega C_s \frac{\Delta C}{2C + \Delta C + C_s} \tag{A.2}$$

The input of the PGA is considered to have infinite impedance, which means that the recently calculated current also flows through the feedback capacitor of the PGA. The output voltage is equal to the voltage over the feedback capacitor, but with opposite sign:

$$v_{out1} = -\frac{1}{j\omega C_{fb}} i_{fb} = \frac{1}{2} v \frac{C_s}{C_{fb}} \frac{\Delta C}{2C + \Delta C + C_s} \tag{A.3}$$

Again, this result can be used to calculate the equivalence capacitance the PGA "sees" at the left branch:

$$v_{out1} = v \frac{C_{eq1}}{C_{fb}} \longrightarrow C_{eq1} = \frac{1}{2} \frac{C_s \Delta C}{2C + \Delta C + C_s} \tag{A.4}$$

Equation A.4 shows that the result only depends on the values of the capacitors and is independent on frequency and amplitude of the input source.

Derivation of the Right Branch

The derivation of the right branch is done similar to the left branch, hence only the resulting equations are shown below:

$$v_n = v_n' + v_n'' = \frac{1}{2} v \frac{C + \Delta C}{2C + \Delta C + C_s} - \frac{1}{2} v \frac{C}{2C + \Delta C + C_s} = \frac{1}{2} v \frac{\Delta C}{2C + \Delta C + C_s} \tag{A.5}$$

$$i_{fb} = \frac{v_n}{\frac{1}{j\omega C_s}} = \frac{1}{2} v j\omega C_s \frac{\Delta C}{2C + \Delta C + C_s} \tag{A.6}$$

$$v_{out2} = -\frac{1}{j\omega C_{fb}} i_{fb} = -\frac{1}{2} v \frac{C_s}{C_{fb}} \frac{\Delta C}{2C + \Delta C + C_s} \tag{A.7}$$

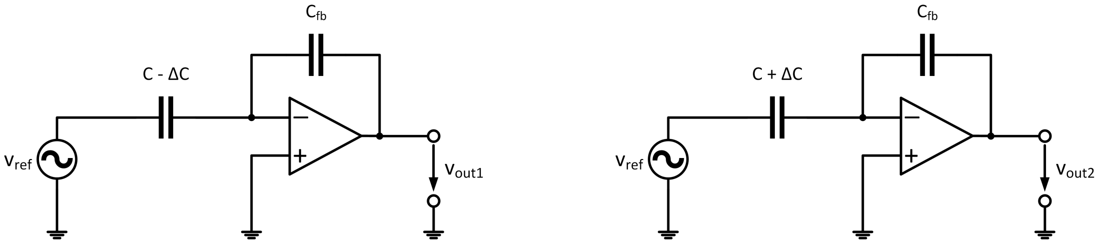
$$v_{out2} = v \frac{C_{eq2}}{C_{fb}} \longrightarrow C_{eq2} = -\frac{1}{2} \frac{C_s \Delta C}{2C + \Delta C + C_s} \tag{A.8}$$

The differential equivalent capacitance between both branches can now be calculated by determining the difference between Equation A.4 and Equation A.8:

$$C_{eq} = C_{eq1} - C_{eq2} \longrightarrow \boxed{C_{eq} = C_s \frac{\Delta C}{2C + \Delta C + C_s}} \tag{A.9}$$

A.1.2. Full-Bridge Configuration

Since the nodes between the capacitors in each branch are connected to the inputs of the PGA, which have a fixed common-mode voltage, a similar equivalent circuit as for the half-bridge (MEMS setup) can be used to derive the relationship between the capacitance change and the output of the PGA. The equivalent circuit is shown in Figure A.2.



(a) Left branch

(b) Right branch

Figure A.2.: AC signal derivation for both branches of the full-bridge.

Derivation:

Same as for the half-bridge, the output voltages of the PGA can be calculated by applying the equation for a capacitive inverting amplifier:

$$v_{out1} = -\frac{C - \Delta C}{C_{fb}} v_{ref} \tag{A.10}$$

$$v_{out2} = -\frac{C + \Delta C}{C_{fb}} v_{ref} \tag{A.11}$$

With:

$$v_{ref} = v_{ref,p} - v_{ref,n} \tag{A.12}$$

The differential output voltage of the PGA is the difference between both outputs:

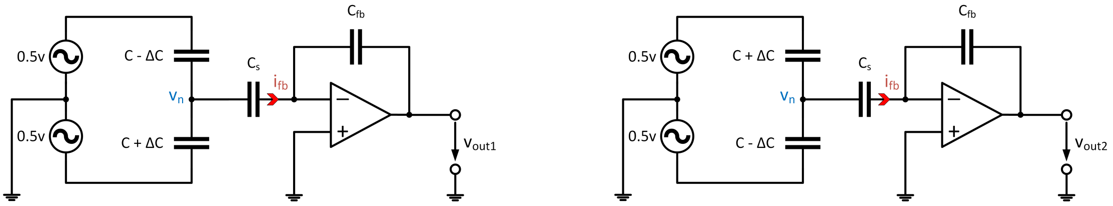
$$v_{out,diff} = v_{out1} - v_{out2} = \frac{-C + \Delta C + C + \Delta C}{C_{fb}} v_{ref} = \frac{2\Delta C}{C_{fb}} v_{ref} \tag{A.13}$$

Now, the differential capacitance determined by the PGA can be calculated as:

$$v_{out,diff} = -\frac{C_{eq}}{C_{fb}} v_{ref} \longrightarrow \boxed{C_{eq} = -C_{fb} \frac{v_{out,diff}}{v_{ref}} = -2\Delta C} \tag{A.14}$$

A.1.3. Ideal Attenuated Full-Bridge Configuration

The derivation of the relation between the capacitance change in the branches and the output of the PGA for the attenuated full-bridge is done similar to the derivation of the attenuated half-bridge with the exception that now both capacitors in each branch change inversely by the same value.



(a) Left branch

(b) Right branch

Figure A.3.: AC signal derivation for both branches of the ideal attenuated full-bridge.

Derivation of the Left Branch:

The derivation starts with the calculation of the node-voltage v_n by super-positioning both voltage sources:

$$v_n = v_n' + v_n'' = \frac{1}{2}v\frac{C - \Delta C}{2C + C_s} - \frac{1}{2}v\frac{C + \Delta C}{2C + C_s} = -v\frac{\Delta C}{2C + C_s} \tag{A.15}$$

By knowing the voltage of the node, one can calculate the current through the feedback path of the PGA:

$$i_{fb} = \frac{v_n}{\frac{1}{j\omega C_s}} = -j\omega C_s v\frac{\Delta C}{2C + C_s} \tag{A.16}$$

The first output voltage of the PGA is the feedback current times the feedback-impedance, but with opposite sign:

$$v_{out1} = -\frac{1}{j\omega C_{fb}}i_{fb} = v\frac{C_s}{C_{fb}}\frac{\Delta C}{2C + C_s} \tag{A.17}$$

Again, this single-ended setup can be seen as an inverting amplifier, which allows to calculate the equivalent capacitance the input of the amplifier "sees" (negative sign is already implied in v_{out1}):

$$v_{out1} = \frac{C_{eq1}}{C_{fb}}v \longrightarrow C_{eq1} = v_{out1}\frac{C_{fb}}{v} = C_s\frac{\Delta C}{2C + C_s} \tag{A.18}$$

Derivation of the Right Branch

The equations for the right branch are derived equivalently to the left branch and lead to the following equations:

$$v_n = v_n' + v_n'' = \frac{1}{2}v\frac{C + \Delta C}{2C + C_s} - \frac{1}{2}v\frac{C - \Delta C}{2C + C_s} = v\frac{\Delta C}{2C + C_s} \tag{A.19}$$

$$i_{fb} = \frac{v_n}{\frac{1}{j\omega C_s}} = j\omega C_s v\frac{\Delta C}{2C + C_s} \tag{A.20}$$

$$v_{out2} = -\frac{1}{j\omega C_{fb}}i_{fb} = -v\frac{C_s}{C_{fb}}\frac{\Delta C}{2C + C_s} \tag{A.21}$$

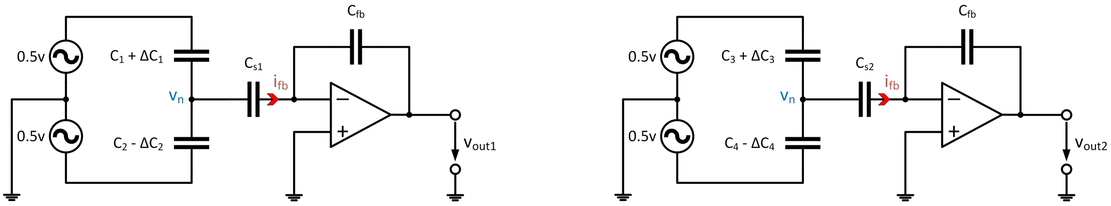
$$v_{out2} = \frac{C_{eq2}}{C_{fb}}v \longrightarrow C_{eq2} = v_{out2}\frac{C_{fb}}{v} = -C_s\frac{\Delta C}{2C + C_s} \tag{A.22}$$

Calculating the difference between Equation A.18 and Equation A.22 gives the differential equivalent capacitance between both branches:

$$C_{eq} = C_{eq1} - C_{eq2} \longrightarrow \boxed{C_{eq} = \frac{2C_s}{2C + C_s}\Delta C} \tag{A.23}$$

A.1.4. Real Attenuated Full-Bridge Configuration

Despite the different notation of the capacitors, Figure A.4 basically shows the same setup as Figure A.3.



(a) Left branch

(b) Right branch

Figure A.4.: AC signal derivation for both branches of the real attenuated full-bridge.

Derivation of the Left Branch

Compared to the ideal configuration, the equations change to:

$$\begin{aligned} v_n = v_n' + v_n'' = \frac{1}{2}v\frac{C_1 - \Delta C_1}{C_1 + C_2 - \Delta C_1 + \Delta C_2 + C_{s1}} - \\ -\frac{1}{2}v\frac{C_2 + \Delta C_2}{C_1 + C_2 - \Delta C_1 + \Delta C_2 + C_{s1}} = \frac{1}{2}v\frac{C_1 - C_2 - \Delta C_1 - \Delta C_2}{C_1 + C_2 - \Delta C_1 + \Delta C_2 + C_{s1}} \end{aligned} \tag{A.24}$$

$$i_{fb} = j\omega C_{s1}v_n = j\omega C_{s1}\frac{1}{2}v\frac{C_1 - C_2 - \Delta C_1 - \Delta C_2}{C_1 + C_2 - \Delta C_1 + \Delta C_2 + C_{s1}} \tag{A.25}$$

$$v_{out1} = -\frac{1}{j\omega C_{fb}} = -\frac{1}{2}\frac{C_{s1}}{C_{fb}}\frac{C_1 - C_2 - \Delta C_1 - \Delta C_2}{C_1 + C_2 - \Delta C_1 + \Delta C_2 + C_{s1}} \tag{A.26}$$

$$v_{out1} = \frac{C_{eq1}}{C_{fb}}v \longrightarrow C_{eq1} = v_{out1}\frac{C_{fb}}{v} = -\frac{1}{2}C_{s1}\frac{C_1 - C_2 - \Delta C_1 - \Delta C_2}{C_1 + C_2 - \Delta C_1 + \Delta C_2 + C_{s1}} \tag{A.27}$$

Derivation of the Right Branch:

The derivation of the equation of the right branch is done similar:

$$\begin{aligned} v_n = v_n' + v_n'' = \frac{1}{2}v\frac{C_3 + \Delta C_3}{C_3 + C_4 + \Delta C_3 - \Delta C_4 + C_{s2}} - \\ -\frac{1}{2}v\frac{C_4 - \Delta C_4}{C_3 + C_4 + \Delta C_3 - \Delta C_4 + C_{s2}} = \frac{1}{2}v\frac{C_3 - C_4 + \Delta C_3 + \Delta C_4}{C_3 + C_4 + \Delta C_3 - \Delta C_4 + C_{s2}} \end{aligned} \tag{A.28}$$

$$i_{fb} = j\omega C_{s2} v_n = j\omega C_{s2}\frac{1}{2}v\frac{C_3 - C_4 + \Delta C_3 + \Delta C_4}{C_3 + C_4 + \Delta C_3 - \Delta C_4 + C_{s2}} \tag{A.29}$$

$$v_{out2} = -\frac{1}{j\omega C_{fb}} i_{fb} = -\frac{1}{2}v\frac{C_{s2}}{C_{fb}}\frac{C_3 - C_4 + \Delta C_3 + \Delta C_4}{C_3 + C_4 + \Delta C_3 - \Delta C_4 + C_{s2}} \tag{A.30}$$

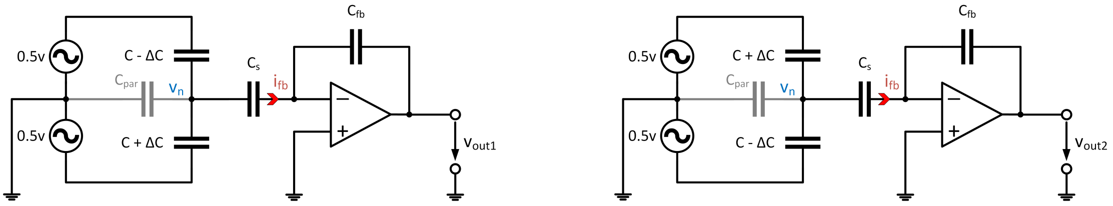
$$v_{out2} = \frac{C_{eq2}}{C_{fb}}v \longrightarrow C_{eq2} = v_{out2}\frac{C_{fb}}{v} = -\frac{1}{2}C_{s2}\frac{C_3 - C_4 + \Delta C_3 + \Delta C_4}{C_3 + C_4 + \Delta C_3 - \Delta C_4 + C_{s2}} \tag{A.31}$$

The difference between Equation A.27 and Equation A.31 gives the differential equivalent capacitance between the left and the right branch of the bridge:

$$C_{eq} = C_{eq1} - C_{eq2} \longrightarrow \boxed{\begin{aligned} C_{eq} = -\frac{1}{2}C_{s1}\frac{C_1 - C_2 - \Delta C_1 - \Delta C_2}{C_1 + C_2 - \Delta C_1 + \Delta C_2 + C_{s1}} + \\ +\frac{1}{2}C_{s2}\frac{C_3 - C_4 + \Delta C_3 + \Delta C_4}{C_3 + C_4 + \Delta C_3 - \Delta C_4 + C_{s2}} \end{aligned}} \tag{A.32}$$

A.1.5. Parasitic Capacitances in Attenuated Full-Bridge

For simplicity, the impact of the parasitic capacitances C_{par} in each branch onto the relation between C_{eq} and ΔC is only shown for the ideal attenuated full-bridge configuration. However, this derivation is also valid for the real attenuated full-bridge. Figure A.5 shows the setup including the parasitic capacitances C_{par} in each branch. The node v_n is the same node as $toCs$ in the body text.



(a) Left branch

(b) Right branch

Figure A.5.: AC signal derivation for both branches of the attenuated full-bridge including parasitic capacitances

Since the parasitic capacitances are always connected to ground, C_{par} only appears in the denominator of the equations of the attenuated full-bridge. The already derived equations of the ideal attenuated full-bridge just have to be modified, as it is shown below.

Derivation of the Left Branch:

$$v_n = v_n' + v_n'' = \frac{1}{2}v\frac{C-\Delta C}{2C+C_{par}+C_s} - \frac{1}{2}v\frac{C+\Delta C}{2C+C_{par}+C_s} = -v\frac{\Delta C}{2C+C_{par}+C_s} \tag{A.33}$$

$$i_{fb} = \frac{v_n}{\frac{1}{j\omega C_s}} = -j\omega C_s v\frac{\Delta C}{2C+C_{par}+C_s} \tag{A.34}$$

$$v_{out1} = -\frac{1}{j\omega C_{fb}}i_{fb} = v\frac{C_s}{C_{fb}}\frac{\Delta C}{2C+C_{par}+C_s} \tag{A.35}$$

$$v_{out1} = \frac{C_{eq1}}{C_{fb}}v \longrightarrow C_{eq1} = v_{out1}\frac{C_{fb}}{v} = C_s\frac{\Delta C}{2C+C_{par}+C_s} \tag{A.36}$$

Derivation of the Right Branch

$$v_n = v_n' + v_n'' = \frac{1}{2}v\frac{C+\Delta C}{2C+C_{par}+C_s} - \frac{1}{2}v\frac{C-\Delta C}{2C+C_{par}+C_s} = v\frac{\Delta C}{2C+C_{par}+C_s} \tag{A.37}$$

$$i_{fb} = \frac{v_n}{\frac{1}{j\omega C_s}} = j\omega C_s v\frac{\Delta C}{2C+C_{par}+C_s} \tag{A.38}$$

$$v_{out2} = -\frac{1}{j\omega C_{fb}}i_{fb} = -v\frac{C_s}{C_{fb}}\frac{\Delta C}{2C+C_{par}+C_s} \tag{A.39}$$

$$v_{out2} = \frac{C_{eq2}}{C_{fb}}v \longrightarrow C_{eq2} = v_{out2}\frac{C_{fb}}{v} = -C_s\frac{\Delta C}{2C+C_{par}+C_s} \tag{A.40}$$

Calculating the difference between Equation A.36 and Equation A.40 gives the differential equivalent capacitance between both branches:

$$C_{eq} = C_{eq1} - C_{eq2} \longrightarrow \boxed{C_{eq} = \frac{2C_s}{2C+C_{par}+C_s}\Delta C} \tag{A.41}$$

A.2. Logic Functions of Binary-to-Thermometer Encoder

This section gives the relationships between the inputs and the outputs of the implemented binary-to-thermometer encoder. The variables $b1$ to $b6$ represent the binary inputs, while the variables $t1$ to $t40$ represent the thermometer-coded outputs. The equations were derived by feeding the truth-table of the binary-to-thermometer encoder into a tool.

$$
\begin{aligned}
t1 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4}\cdot\overline{b3}\cdot\overline{b2}\cdot\overline{b1}} & t21 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4}\cdot\overline{b5\cdot b3\cdot b2}\cdot\overline{b5\cdot b3\cdot b1}}\\
t2 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4}\cdot\overline{b3}\cdot\overline{b2}} & t22 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4}\cdot\overline{b5\cdot b3\cdot b2}}\\
t3 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4}\cdot\overline{b3}\cdot\overline{b2\cdot b1}} & t23 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4}\cdot\overline{b5\cdot b3\cdot b2\cdot b1}}\\
t4 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4}\cdot\overline{b3}} & t24 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4}}\\
t5 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4}\cdot\overline{b3\cdot b2}\cdot\overline{b3\cdot b1}} & t25 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4\cdot b3}\cdot\overline{b5\cdot b4\cdot b2}\cdot\overline{b5\cdot b4\cdot b1}}\\
t6 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4}\cdot\overline{b3\cdot b2}} & t26 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4\cdot b3}\cdot\overline{b5\cdot b4\cdot b2}}\\
t7 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4}\cdot\overline{b3\cdot b2\cdot b1}} & t27 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4\cdot b3}\cdot\overline{b5\cdot b4\cdot b2\cdot b1}}\\
t8 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4}} & t28 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4\cdot b3}}\\
t9 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4\cdot b3}\cdot\overline{b4\cdot b2}\cdot\overline{b4\cdot b1}} & t29 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4\cdot b3\cdot b2}\cdot\overline{b5\cdot b4\cdot b3\cdot b1}}\\
t10 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4\cdot b3}\cdot\overline{b4\cdot b2}} & t30 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4\cdot b3\cdot b2}}\\
t11 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4\cdot b3}\cdot\overline{b4\cdot b2\cdot b1}} & t31 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4\cdot b3\cdot b2\cdot b1}}\\
t12 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4\cdot b3}} & t32 &= b6\\
t13 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4\cdot b3\cdot b2}\cdot\overline{b4\cdot b3\cdot b1}} & t33 &= \overline{\overline{b6\cdot b4}\cdot\overline{b6\cdot b3}\cdot\overline{b6\cdot b2}\cdot\overline{b6\cdot b1}}\\
t14 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4\cdot b3\cdot b2}} & t34 &= \overline{\overline{b6\cdot b4}\cdot\overline{b6\cdot b3}\cdot\overline{b6\cdot b2}}\\
t15 &= \overline{\overline{b6}\cdot\overline{b5}\cdot\overline{b4\cdot b3\cdot b2\cdot b1}} & t35 &= \overline{\overline{b6\cdot b4}\cdot\overline{b6\cdot b3}\cdot\overline{b6\cdot b2\cdot b1}}\\
t16 &= \overline{\overline{b6}\cdot\overline{b5}} & t36 &= \overline{\overline{b6\cdot b4}\cdot\overline{b6\cdot b3}}\\
t17 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4}\cdot\overline{b5\cdot b3}\cdot\overline{b5\cdot b2}\cdot\overline{b5\cdot b1}} & t37 &= \overline{\overline{b6\cdot b4}\cdot\overline{b6\cdot b3\cdot b2}\cdot\overline{b6\cdot b3\cdot b1}}\\
t18 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4}\cdot\overline{b5\cdot b3}\cdot\overline{b5\cdot b2}} & t38 &= \overline{\overline{b6\cdot b4}\cdot\overline{b6\cdot b3\cdot b2}}\\
t19 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4}\cdot\overline{b5\cdot b3}\cdot\overline{b5\cdot b2\cdot b1}} & t39 &= \overline{\overline{b6\cdot b4}\cdot\overline{b6\cdot b3\cdot b2\cdot b1}}\\
t20 &= \overline{\overline{b6}\cdot\overline{b5\cdot b4}\cdot\overline{b5\cdot b3}} & t40 &= \overline{\overline{b6\cdot b4}}
\end{aligned}
\tag{A.42}
$$

A.3. Schematic of Binary-to-Thermometer Encoder

The schematic of the encoders is a high resolution pdf-file, which allows to scroll into the schematic for a more detailed view.

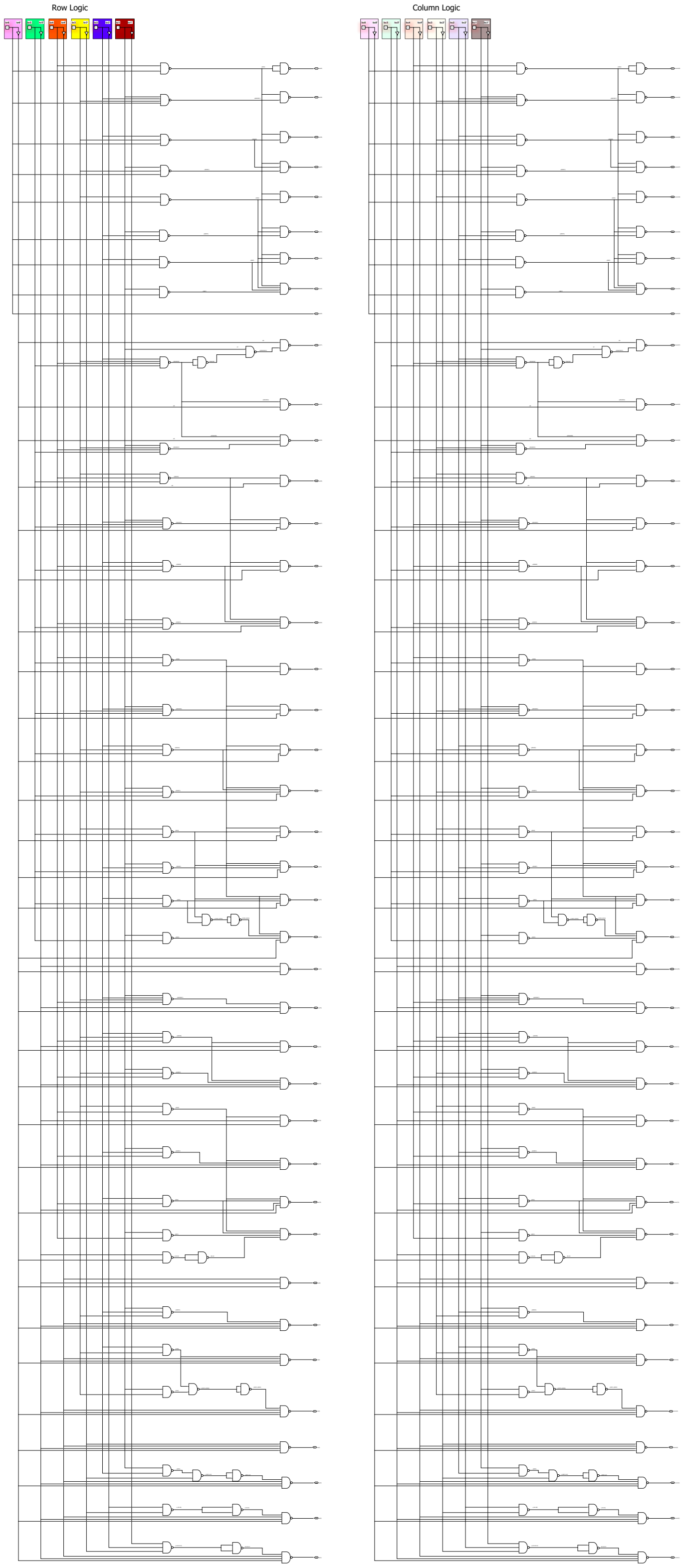


Figure A.6.: This schematic shows the binary-to-thermometer encoders for the rows (left) and columns (right). Both encoders are identical.

A.4. MATLAB Model

The following listing shows the code of the implemented MATLAB model that is used to calculate the standard deviation of the differential nonlinearity of C_{eq} of the attenuated full-bridge. The model is used to decide which unit-capacitance is needed in order that the circuit fulfils the specification.

```matlab
%define number of bits
sN = 10;

%define attenuation capacitors
sCs = 51.23e-15;
sCs1 = sCs;
sCs2 = sCs;

%define number of runs
sRuns = 500;

vDNL = zeros(sRuns,1024);

%create 500 independent bridges
for i = 1:sRuns

    %define unit-capacitor with corresponding standard deviation
    sCu = 100e-15;
    sCsigma = sCu*0.000926;

    %define thermometer-coded capacitor-arrays
    vC2_therm = normrnd(sCu,sCsigma,[2^sN,1]);
    vC3_therm = normrnd(sCu,sCsigma,[2^sN,1]);

    %determine switched on capacitors
    vC2 = zeros([2^sN+1, 1]);
    vC3 = zeros([2^sN+1, 1]);

    %determine all possible switched positions
    for j=2:numel(vC2)
        vC2(j) = sum(vC2_therm(1:j-1));
        vC3(j) = sum(vC3_therm(1:j-1));
    end

    %determine total capacitance of each branch
    sC12 = sum(vC2_therm);
    sC34 = sum(vC3_therm);

    %determine remaining capacitors
    vC1 = sC12 - vC2;
    vC4 = sC34 - vC3;

    %calculate the equivalent capacitance for every switched capacitor
    vC_eq = -0.5*sCs1*(vC1-vC2)/(sC12 + sCs1) + 0.5*sCs2*(vC3-vC4)/(sC34 + sCs2);

    %determine the differential nonlinearity (DNL)
    vDNL(i,:) = diff(vC_eq)/(mean(diff(vC_eq))) - 1;

end

%calculate the standard deviation of the differential nonlinearity
vDNL_std = std(vDNL);

%calculate the average standard deviation of the DNL in %
sDNL_std_mean = mean(vDNL_std)*100;
```

A.5. Layout

A.5.1. Layout of CS-Cell

The layout of the dummy-cell is similar to the layout of the CS-Cell.

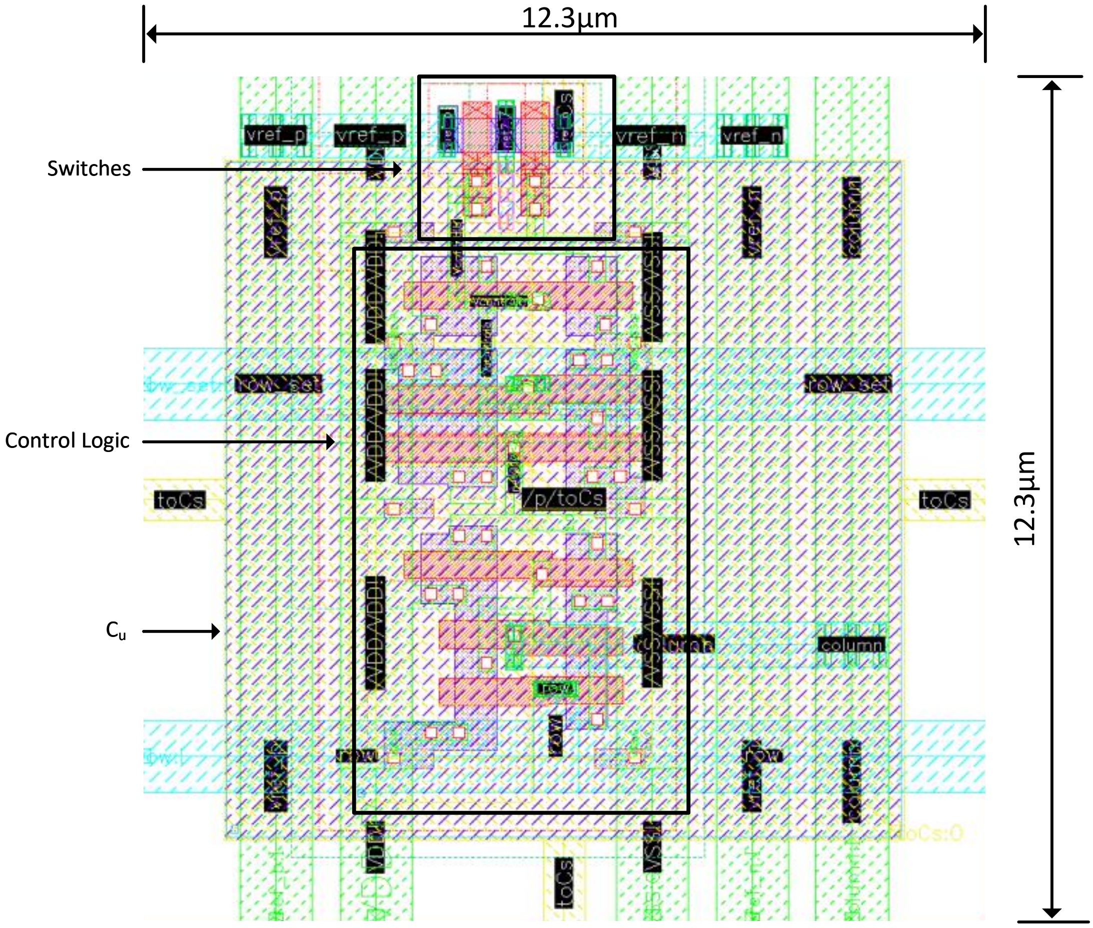


Figure A.7.: Layout of the implemented CS-cell. The layout is done in a way, so that the cells are easily interconnected when placed. The interconnections to the other cells can be seen at the edges of the CS-cell.

A.5.2. Layout of Smaller Programmable Capacitor (625 CS-cells per Branch)

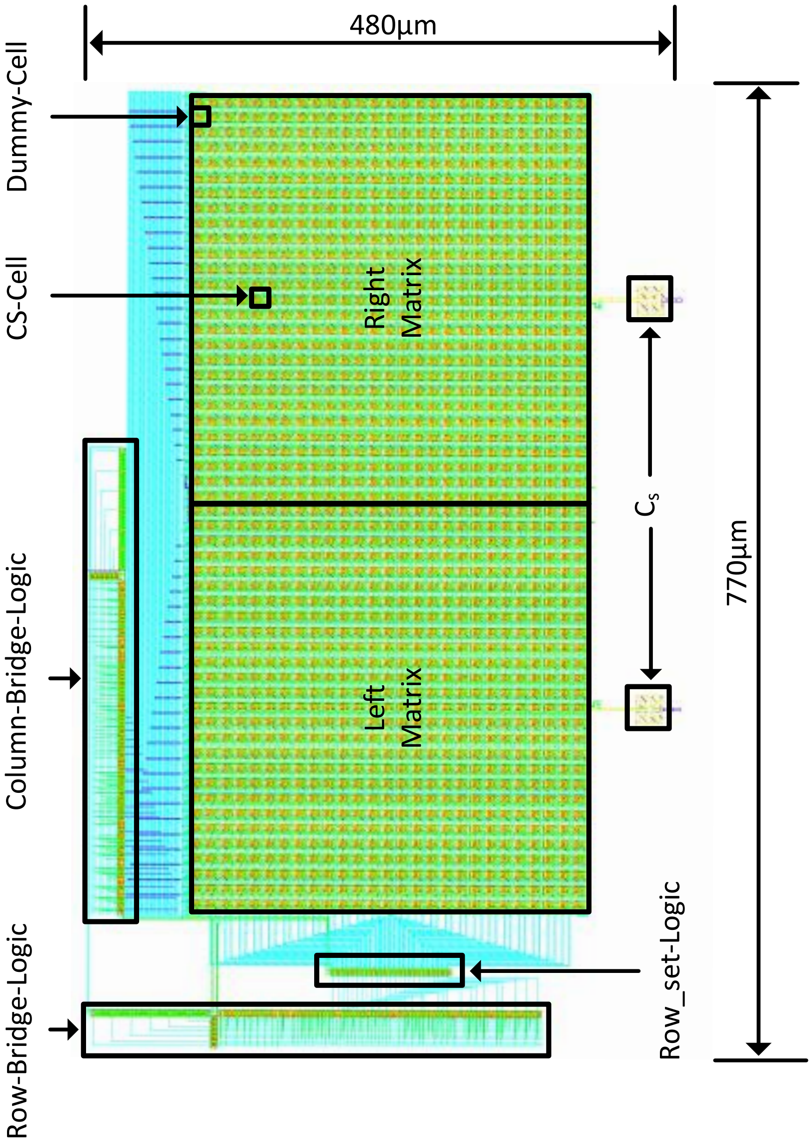


Figure A.8.: Layout of the implemented smaller programmable capacitor with 625 CS-cells per branch.

A.5.3. Layout of Larger Programmable Capacitor (1600 CS-cells per Branch)

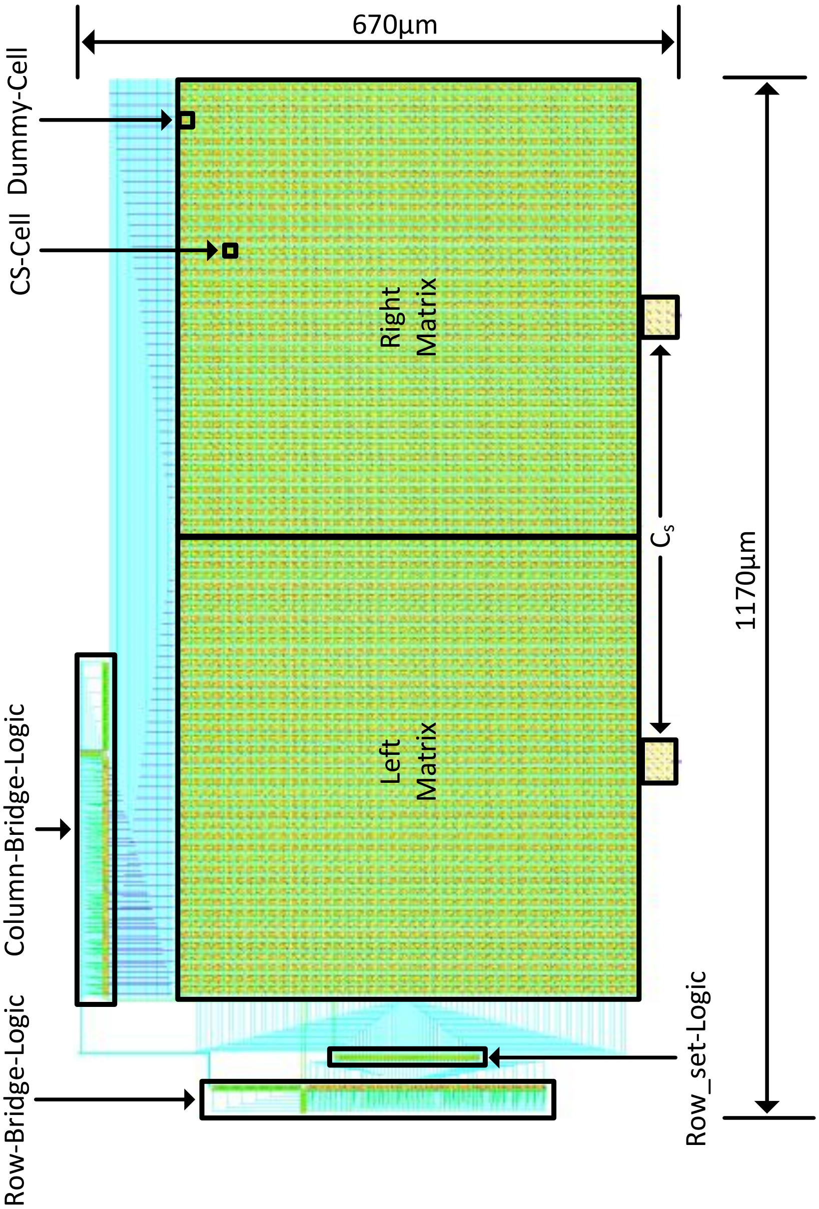


Figure A.9.: Layout of the larger implemented programmable capacitor with 1600 CS-cells per branch.

A.5.4. Layout of Top-Level Programmable Capacitor

The top-level layout of the programmable capacitor consists of the layouts of the smaller and larger programmable capacitors, including pads and a aluminium layer at the top.

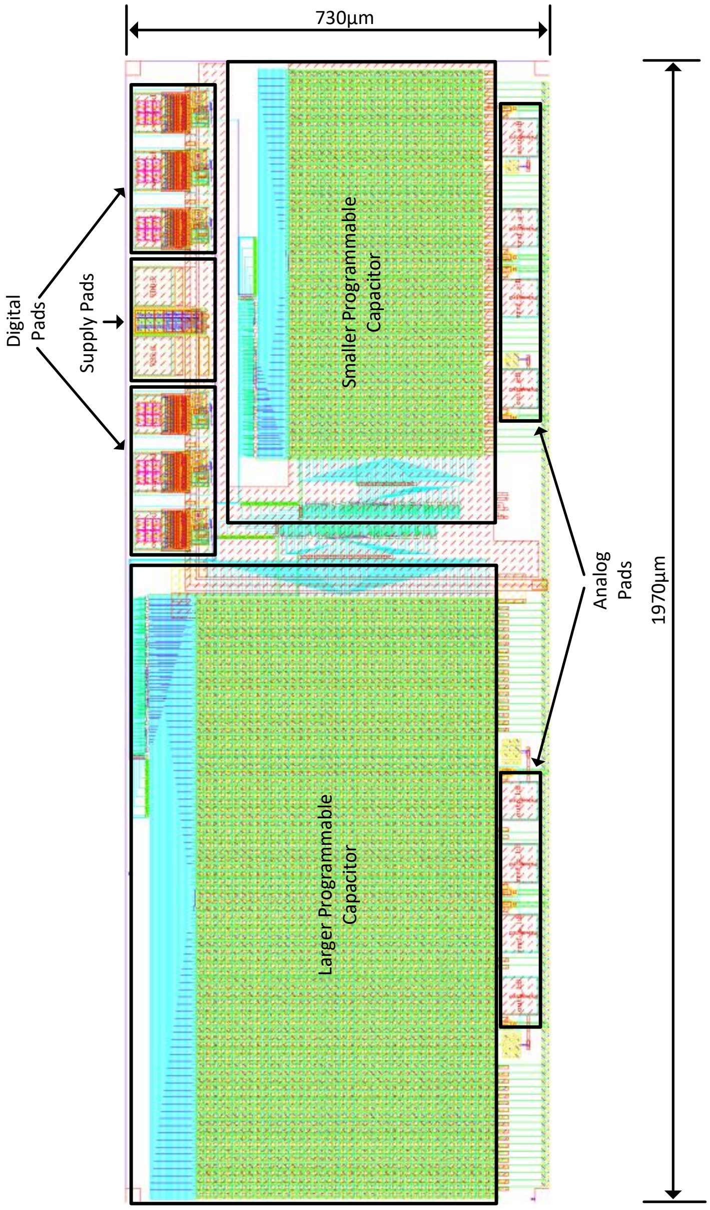


Figure A.10.: Layout of the implemented top-level programmable capacitor with 1600 CS-cells per branch.

Bibliography

[1] A. Pradeep, A. S, A. Unnikrishnan, A. S, S. P. R, and B. S. S D, "Material optimization for Capacitive pressure sensor- A COMSOL study," in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, IEEE, 2021, pp. 97–102, ISBN: 978-1-6654-2867-5. DOI: 10.1109/ICESC51422.2021.9532974.

[2] Q.-J. Xia, Y.-C. Zhang, Y. You, F. Zhou, and W.-g. Lu, "A 9.1µW Capacitance-to-Digital Converter for Pressure Sensor Systems," in *2022 IEEE 16th International Conference on Solid-State & Integrated Circuit Technology (ICSICT)*, 2022, pp. 1–3. DOI: 10.1109/ICSICT55466.2022.9963382.

[3] Y. Jung, S.-J. Kweon, H. Jeon, *et al.*, "A Sub-aF Super-High-Resolution Capacitance-to-Digital Converter with a Bandpass $\Delta\Sigma$ ADC," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2023, pp. 1–4. DOI: 10.1109/ISCAS46773.2023.10181452.

[4] J. Kim, J. .-. Plouchart, N. Zamdmer, *et al.*, "3-dimensional vertical parallel plate capacitors in an SOI CMOS technology for integrated RF circuits," in *2003 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.03CH37408)*, 2003, pp. 29–32. DOI: 10.1109/VLSIC.2003.1221153.

[5] L. Zhang, X. Cheng, T. Xiaodong, and X. Deng, "High voltage charge pump circuit using vertical parallel plate capacitors," in *2017 IEEE 12th International Conference on ASIC (ASICON)*, 2017, pp. 762–764. DOI: 10.1109/ASICON.2017.8252587.

[6] P. R. Gray, *Analysis and design of analog integrated circuits*, 5th ed. New York: Wiley, 2011, ISBN: 978-0-470-39877-7.

[7] H. Ghafarian, C. Moranz, M. Rajabzadeh, J. Leicht, and Y. Manoli, "A fully integrated charge pump using parasitics to increase the usable capacitance by 25 % and the efficiency by up to 18 % with poly-poly capacitors," in pp. 835–838. DOI: 10.1109/MWSCAS.2017.8053053.

[8] H. Omran, H. Alahmadi, and K. N. Salama, "Matching Properties of Femtofarad and Sub-Femtofarad MOM Capacitors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 6, pp. 763–772, 2016, ISSN: 1558-0806. DOI: 10.1109/TCSI.2016.2537824.

[9] N.-C. Chen, P.-Y. Chou, H. Graeb, and M. P.-H. Lin, "High-density MOM capacitor array with novel mortise-tenon structure for low-power SAR ADC," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, 2017, pp. 1757–1762. DOI: 10.23919/DATE.2017.7927277.

[10] M. Marin, S. Cremer, J.-C. Giraudin, and B. Martinet, "Modeling the Mismatch of High-k MIM Capacitors," in *2007 IEEE International Conference on Microelectronic Test Structures*, IEEE, 2007, pp. 115–119, ISBN: 1-4244-0780-X. DOI: 10.1109/ICMTS.2007.374466.

[11] E. Hourdakis, A. Travlos, and A. G. Nassiopoulou, "High-Performance MIM Capacitors With Nanomodulated Electrode Surface," *IEEE Transactions on Electron Devices*, vol. 62, no. 5, pp. 1568–1573, 2015, ISSN: 0018-9383. DOI: 10.1109/TED.2015.2411771.

[12] D. Xu, J. Chen, J. Koh, *et al.*, "Development of High-Density Metal-Insulator-Metal Capacitors with Top and Side Contacts," in *2023 34th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, IEEE, 2023, pp. 1–3, ISBN: 978-1-6654-5639-5. DOI: 10.1109/ASMC57536.2023.10121096.

[13] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, 1989, ISSN: 0018-9200. DOI: 10.1109/JSSC.1989.572629.

[14] J.-B. Shyu, G. C. Temes, and K. Yao, "Random errors in MOS capacitors," *IEEE Journal of Solid-State Circuits*, vol. 17, no. 6, pp. 1070–1076, 1982, ISSN: 0018-9200. DOI: 10.1109/JSSC.1982.1051862.

[15] H. Omran, R. T. ElAfandy, M. Arsalan, and K. N. Salama, "Direct Mismatch Characterization of Femtofarad Capacitors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 2, pp. 151–155, 2016, ISSN: 1558-3791. DOI: 10.1109/TCSII.2015.2468919.

[16] S. Haenzsche, S. Henker, and R. Schüffny, "Modelling of capacitor mismatch and non-linearity effects ini charge redistribution SAR ADCs," in *Proceedings of the 17th International Conference Mixed Design of Integrated Circuits and Systems - MIXDES 2010*, 2010, pp. 300–305.

[17] D. Bustamante, D. Janke, E. Swindlehurst, and S.-H. W. Chiang, "High-Precision, Mixed-Signal Mismatch Measurement of Metal–Oxide–Metal Capacitors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 11, pp. 1272–1276, 2017, ISSN: 1549-7747. DOI: 10.1109/TCSII.2016.2642820.

[18] V. Tripathi and B. Murmann, "Mismatch Characterization of Small Metal Fringe Capacitors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 8, pp. 2236–2242, 2014, ISSN: 1549-8328. DOI: 10.1109/TCSI.2014.2332264.

[19] A. Ahuja, K. Badami, C. Barbelenet, and S. Emery, "Comparison of Capacitive DAC Architectures for Power and Area Efficient SAR ADC Designs," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5, ISBN: 978-1-7281-9201-7. DOI: 10.1109/ISCAS51556.2021.9401768.

[20] M. P. .-. Lin, V. W. .-. Hsiao, C. .-. Lin, and N. .-. Chen, "Parasitic-Aware Common-Centroid Binary-Weighted Capacitor Layout Generation Integrating Placement, Routing, and Unit Capacitor Sizing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 8, pp. 1274–1286, 2017, ISSN: 0278-0070. DOI: 10.1109/TCAD.2017.2685598.

[21] C.-W. Lin, J.-M. Lin, Y.-C. Chiu, C.-P. Huang, and S.-J. Chang, "Common-centroid capacitor placement considering systematic and random mismatches in analog integrated circuits," in *2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2011, pp. 528–533.

[22] K.-P. Pun, L. Sun, and B. Li, "Unit capacitor array based SAR ADC," *Microelectronics Reliability*, vol. 53, no. 3, pp. 505–508, 2013, ISSN: 00262714. DOI: 10.1016/j.microrel.2012.09.012.

[23] S. Haenzsche and R. Schüffny, "Analysis of a charge redistribution SAR ADC with partially thermometer coded DAC," in *2013 European Conference on Circuit Theory and Design (ECCTD)*, IEEE, 2013, pp. 1–4, ISBN: 978-3-00-043785-4. DOI: 10.1109/ECCTD.2013.6662307.

[24] M. Saberi, R. Lotfi, K. Mafinezhad, and W. A. Serdijn, "Analysis of Power Consumption and Linearity in Capacitive Digital-to-Analog Converters Used in Successive Approximation ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 8, pp. 1736–1748, 2011, ISSN: 1549-8328. DOI: 10.1109/TCSI.2011.2107214.

[25] H. Ha, S. .-. Lee, B. Kim, H. .-. Park, and J. .-. Sim, "A 0.5-V, 1.47- µW 40-kS/s 13-bit SAR ADC With Capacitor Error Compensation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 840–844, 2014, ISSN: 1558-3791. DOI: 10.1109/TCSII.2014.2350378.

[26] S. Lei, D. Qinyuan, L. Chuangchuan, and Q. Gaoshuai, "Analysis on Capacitor Mismatch and Parasitic Capacitors Effect of Improved Segmented-Capacitor Array in SAR ADC," in *2009 Third International Symposium on Intelligent Information Technology Application*, IEEE, 2009, pp. 280–283, ISBN: 978-0-7695-3859-4. DOI: 10.1109/IITA.2009.193.

[27] L. Mendes, M. J. Rosario, and J. C. Vaz, "Performance of Si-Integrated Wide-Band Single-Ended Switched Capacitor Arrays," in *2006 13th IEEE International Conference on Electronics, Circuits and Systems*, 2006, pp. 482–485. DOI: 10.1109/ICECS.2006.379830.

[28] B.-K. Kim, T. Lee, D. Im, D.-K. Im, B. Kim, and K. Lee, "Design methodology of tunable impedance matching circuit with SOI CMOS tunable capacitor array for RF FEM," in *2013 Asia-Pacific Microwave Conference Proceedings (APMC)*, 2013, pp. 7–9. DOI: 10.1109/APMC.2013.6695173.

[29] R. R. Vallabhuni, J. Sravana, M. Saikumar, M. S. Sriharsha, and D. Rani, "An Advanced Computing Architecture for Binary to Thermometer Decoder using 18nm FinFET," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, 2020, pp. 510–515, ISBN: 978-1-7281-5821-1. DOI: `10.1109/ICSSIT48917.2020.9214105`.

[30] U. Tietze, C. Schenk, and E. Gamm, *Halbleiter-Schaltungstechnik*, 16., erweiterte und aktualisierte Auflage. Berlin, Germany: Springer Vieweg, 2019, ISBN: 978-3-662-48553-8.