Examensarbete

Design and Implementation of a Digitally Compensated N-Bit C-xC SAR ADC Model
Optimization of an Eight-Bit C-xC SAR ADC

Examensarbete utfört i Elektroniksystem
vid Tekniska högskolan vid Linköpings universitet
av
Claes Hallström (claha288@student.liu.se)

LiTH-ISY-EX--13/4679--SE
Linköping 2013
Design and Implementation of a Digitally Compensated N-Bit C-xC SAR ADC Model
Optimization of an Eight-Bit C-xC SAR ADC

Examensarbete utfört i Elektroniksystem vid Tekniska högskolan vid Linköpings universitet av
Claes Hallström (claha288@student.liu.se)
LiTH-ISY-EX--13/4679--SE

Supervisor: Dr. Rolf Sundblad
AnaCatum AB
Examiner: Dr. J. Jacob Wikner
ISY, Linköpings universitet

Linköping, 13 juni 2013
In this master’s thesis a model of a digitally compensated N-bit C-xC SAR ADC was developed. The architecture uses charge redistribution in a C-xC capacitor network to perform the conversion. Focus in the master’s thesis was set to understand how the charge is redistributed in the network during the conversion and calibration phase. Redundancy and parasitic capacitors is present in the system and rises the need for extra conversion steps as well as a calibration algorithm. The calibration algorithm, Bit Weight Estimation, calculates a weight corresponding to each bit which is used in the last conversion step to perform a digital weighting. The result of extensive calculations in different C-xC capacitor networks was a model in Python of an N-bit C-xC SAR ADC. That model was used to create a model of an eight-bit C-xC SAR ADC and finding suitable parameters for it through calculations and simulations. The parameters giving the best INL was chosen. With the best parameters the C-xC SAR ADC static and dynamic performance was tested and showed an INL of less than \( \pm 1 \text{LSB} \), SNR of 47.8 dB and ENOB of 7.6 bits.
Abstract

In this master’s thesis a model of a digitally compensated N-bit C-xC SAR ADC was developed. The architecture uses charge redistribution in a C-xC capacitor network to perform the conversion. Focus in the master’s thesis was set to understand how the charge is redistributed in the network during the conversion and calibration phase. Redundancy and parasitic capacitors is present in the system and rises the need for extra conversion steps as well as a calibration algorithm. The calibration algorithm, Bit Weight Estimation, calculates a weight corresponding to each bit which is used in the last conversion step to perform a digital weighting. The result of extensive calculations in different C-xC capacitor networks was a model in Python of an N-bit C-xC SAR ADC. That model was used to create a model of an eight-bit C-xC SAR ADC and finding suitable parameters for it through calculations and simulations. The parameters giving the best INL was chosen. With the best parameters the C-xC SAR ADC static and dynamic performance was tested and showed an INL of less than ±1LSB, SNR of 47.8 dB and ENOB of 7.6 bits.
Acknowledgments

First, I would like to thank my closest friends that I have studied with the past five years. For the weekly lunch with both the serious and funny discussions that we have had the past term, which was always the highlight of the week.

I would also like to thank to staff at AnaCatum AB and especially Christer Jansson for his help whenever I needed an opinion or something explained. Also a big thank to Björn Årleskog, whom I shared room with and that I have been randomly throwing questions at during this term.

Lastly, I would like to thank my examiner, Dr. J. Jacob Wikner and my supervisor, Dr. Rolf Sundblad for making this master thesis possible.

Linköping, June 2013
Claes Hallström
List of Figures

2.1 Transfer function of an ideal three-bit ADC. ............................................ 4
2.2 Quantization error of an ideal ADC. ........................................................... 5
2.3 Transfer function of an ideal ADC and ADC with positive offset error. .......... 5
2.4 Transfer function of an ideal ADC and ADC with positive gain error. ............. 6
2.5 Transfer function of an ideal ADC and ADC with DNL error. ....................... 7
2.6 Code centers of an ideal ADC and transfer function of an ADC with INL error. ..... 8
2.7 Transfer function of an ideal ADC and ADC with code ‘100’ missing. ............. 8

3.1 Example of the conversion algorithm for a three-bit SAR ADC. ................... 14
3.2 Layout of a SAR ADC showing the different building blocks. ....................... 15
3.3 Capacitor network of an N-bit binary weighted capacitor array. ................... 16
3.4 Capacitor network of an N-bit two stage weighted capacitor array. ............... 16
3.5 Capacitor network of an N-bit C-2C/C-xC capacitor array. ......................... 16

5.1 The architecture of the C-xC SAR ADC. ..................................................... 25
5.2 A C-xC link and its \( C_{imp} \). ................................................................. 28

6.1 Output around middle code range with uncalibrated ADC. ......................... 36
6.2 Output around middle code range with calibrated ADC. ........................... 36
6.3 Output when using a ramp as input from the uncalibrated ADC. ................. 37
6.4 Output when using a ramp as input from the calibrated ADC. ..................... 37
6.5 INL of converted ramp from the uncalibrated ADC. .................................. 38
6.6 INL of converted ramp from the calibrated ADC. .................................... 39
6.7 Comparison between output when using sine as input to an uncalibrated and a calibrated ADC. ................................................................. 39
6.8 Zoomed in on output when using sine as input to the uncalibrated ADC. ........ 40
6.9 Zoomed in on output when using sine as input to the calibrated ADC. .......... 40
6.10 Spectrum of converted sine from the uncalibrated ADC. ........................... 41
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.11</td>
<td>Spectrum of converted sine from the calibrated ADC</td>
<td>41</td>
</tr>
<tr>
<td>A.1</td>
<td>Structure of a two-bit C-xC capacitor array</td>
<td>48</td>
</tr>
<tr>
<td>A.2</td>
<td>Structure of a three-bit C-xC capacitor array</td>
<td>49</td>
</tr>
<tr>
<td>A.3</td>
<td>Structure of a four-bit C-xC capacitor array</td>
<td>51</td>
</tr>
<tr>
<td>A.4</td>
<td>Structure of a four-bit C-xC capacitor array with two directly weighted MSB</td>
<td>58</td>
</tr>
<tr>
<td>A.5</td>
<td>Structure of a four-bit C-xC capacitor array with three directly weighted MSB</td>
<td>60</td>
</tr>
</tbody>
</table>
# List of Tables

4.1 A summary of all the methods and techniques discussed in the chapter. .......................................................... 23

6.1 List and explanation of C-xC SAR ADC parameters. ............ 33
6.2 Search range for parameters in the parameter sweep. .......... 34
6.3 The parameters with the best INL from the parameter sweep. .. 34
6.4 List of C-xC SAR ADC parameters and their best values. ...... 35

7.1 List and explanation of C-xC SAR ADC parameters and their best values. ...................................................... 44
# Contents

List of Figures vii  
List of Tables ix  
Notation xiii  

## 1 Introduction 1  
1.1 Background ........................................... 1  
1.2 Aim ..................................................... 1  
1.3 Outline ............................................... 1  
1.4 Thesis Outline ....................................... 2  

## 2 Analog to Digital Converter 3  
2.1 Introduction ......................................... 3  
2.2 The ADC ............................................... 3  
2.2.1 Quantization ..................................... 3  
2.3 Static Performance Metrics 4  
2.3.1 Offset Error ................................... 4  
2.3.2 Gain Error ...................................... 6  
2.3.3 Full Scale Error ................................ 6  
2.3.4 Differential Non-Linearity .................... 6  
2.3.5 Integral Non-Linearity ......................... 7  
2.3.6 Missing Codes .................................. 7  
2.4 Dynamic Performance Metrics 7  
2.4.1 Signal to Noise Ratio ......................... 7  
2.4.2 Signal to Noise and Distortion Ratio ......... 9  
2.4.3 Effective Number of Bits ...................... 9  
2.4.4 Total Harmonic Distortion ..................... 9  
2.4.5 Spurious Free Dynamic Range ................. 10  
2.5 Architectures 10  
2.5.1 Flash ............................................ 10  
2.5.2 Pipelined ....................................... 10  
2.5.3 Sigma-Delta .................................... 10
7.2 Future Work ................................................. 44

A Capacitor Network Calculations .......................... 47
   A.1 Conversion ................................................. 47
      A.1.1 Two-Bit C-xC ......................................... 47
      A.1.2 Three-Bit C-xC ......................................... 49
      A.1.3 Four-Bit C-xC ......................................... 51
      A.1.4 N-Bit C-xC ........................................... 53
   A.2 Bit Weight Estimation .................................... 54
      A.2.1 Three-Bit C-xC ......................................... 54
      A.2.2 N-Bit C-xC ........................................... 58
   A.3 Directly Weighted MSBs ................................. 58
      A.3.1 Four-Bit C-xC ......................................... 58
      A.3.2 N-Bit C-xC with $n$ directly weighted MSB ...... 61

Bibliography ................................................. 63
## Notation

### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter - An ADC is a mixed-signal integrated circuit that convert an analog input level to a digital output code.</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital to Analog Converter - A DAC is a mixed-signal integrated circuit that convert a digital input code to an analog output level.</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current - A DC is an input signal with constant amplitude over time.</td>
</tr>
<tr>
<td>S/H</td>
<td>Sample and Hold - A S/H is a circuit that tracks a signal and then holds it for a specified period.</td>
</tr>
<tr>
<td>SAR</td>
<td>Successive Approximation Register - The SAR is the control block of a SAR ADC.</td>
</tr>
<tr>
<td>TI</td>
<td>Time-Interleaved - A number of connected ADCs that are interleaved in time is called a TI ADC.</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bit - The right-most bit in a binary number is called the LSB.</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Bit - The left-most bit in a binary number is called the MSB.</td>
</tr>
<tr>
<td>DNL</td>
<td>Differential Non-Linearity - The deviation of an output code width from one LSB is defined as DNL.</td>
</tr>
<tr>
<td>INL</td>
<td>Integral Non-Linearity - The deviation of an output code center from the ideal center is defined as INL.</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio - The ratio between the power of the signal and the power of the noise expressed in dB is called SNR.</td>
</tr>
<tr>
<td>SNDR</td>
<td>Signal to Noise and Distortion Ratio - The ratio between the power of the signal and the sum of the power of the noise and distortion expressed in dB is called SNDR.</td>
</tr>
<tr>
<td>ENOB</td>
<td>Effective Number Of Bits - Expressing SNDR in bits instead of dB is defined as ENOB.</td>
</tr>
<tr>
<td>THD</td>
<td>Total Harmonic Distortion - The ratio between the sum of the power of the harmonic components and the power the signal expressed in dB is called THD.</td>
</tr>
<tr>
<td>SFDR</td>
<td>Spurious Free Dynamic Range - The ratio between the power of the signal over the power of the largest peak of spurious expressed in dB is called SFDR.</td>
</tr>
</tbody>
</table>
1

Introduction

1.1 Background

Today AnaCatum AB uses a digitally compensated 12-bit TI SAR ADC for all their projects. Some of their projects require a resolution of 12-bit and some require lower resolution. By using a 12-bit ADC when only a resolution of eight bits is required the analog and digital domain of the ADC will be much more complex than necessary. Reducing this complexity in both the analog and digital domain will result in an ADC which requires less area and consumes less power.

1.2 Aim

To make a model of a digitally compensated eight-bit ADC. This model should be based on the existing Matlab model of AnaCatum’s digitally compensated 12-bit ADC. The designed ADC model should maintain the original system properties of the architecture, i.e. the 12-bit ADC. A reduction in complexity in both the analog and digital domain should be made for the model compared to the 12-bit ADC model. The designed eight-bit ADC model should be evaluated using both static (INL, DNL, etc.) and dynamic (SNR, ENOB, etc.) performance metrics.

1.3 Outline

In this master’s thesis a model of an N-bit C-xC SAR ADC with digital calibration based on AnaCatum’s 12-bit C-xC SAR ADC was developed. The model is
written in Python and built up using classes representing the different parts of the ADC. Extensive calculations of capacitor networks during both the conversion and calibration phase were made in order to develop the model. Currently the only noise or error that is modeled is noise and offset in the comparator, but this is not used in the simulation. The developed model was then used to create a model of an eight-bit C-xC SAR ADC. To find suitable parameters for the model of the eight-bit ADC was the next task. The parameters were for example the number of extra bits needed in both the conversion and calibration phase. To determine some of these parameters a sweep was made over a suitable range and the parameter set showing the best INL was chosen. Using the chosen parameters a few simulations were made to determine the performance of the ADC. The simulations aimed at confirming that the calibration algorithm worked and to evaluate the static and dynamic performance metrics. Comparing the uncalibrated and calibrated the ADC showed an improvement in INL from $-7.26/8.59$ LSB to $-0.99/0.93$ LSB, i.e. an improvement of more than 85%. For the dynamic performance metrics, SNR improved from 33.90dB to 47.85dB and ENOB from 4.66 bits to 7.65 bits.

### 1.4 Thesis Outline

**Chapter 2** covers the basics of ADCs. The definition of different performance metrics, both static and dynamic is presented as well as a few different ADC architectures.

**Chapter 3** discusses the SAR ADC architecture and its building blocks. Focus is set on different DAC architectures that can be used with the SAR ADC.

**Chapter 4** discusses correction and calibration of ADCs. Specific methods for the SAR ADC architecture from different articles are presented. The calibration method used for AnaCatum’s SAR ADC is also briefly explained.

**Chapter 5** presents the model that was developed. Both theory and calculations behind the model and its calibration algorithm as well as the code are discussed.

**Chapter 6** presents the simulations which were performed with the developed model. Both simulations to find the best parameters and performance evaluation with the chosen parameters were conducted.

**Chapter 7** summarizes the master’s thesis and presents suggestions for future improvements and development.

**Appendix A** presents the calculations on different capacitor networks which were done in order to develop the model.
2.1 Introduction

This chapter covers the basics of ADCs. The definition of different performance metrics, both static and dynamic is presented as well as a few different ADC architectures.

2.2 The ADC

An ADC, as the name suggests, converts an analog signal to a digital code. This process consists of a quantization in both time and amplitude. The quantization in time is done using a S/H circuit that samples the input at given time intervals according to a sampling frequency. Next is the quantization in amplitude which is performed differently for different ADC architectures, see Section 2.5. As there is quantization involved the analog signal and digital code will not match perfectly [Sundström, 2011]. Figure 2.1 shows the transfer function of an ideal three-bit ADC.

2.2.1 Quantization

The quantization in amplitude is closely related to the resolution of the ADC. An N-bit ADC has a resolution of N and $2^N$ codes with which the amplitude can be represented. Each code has a code width of one LSB which is defined according to Equation 2.1 [Sundström, 2011]

$$1\text{LSB} = \frac{V_{FS}}{2^N}$$ (2.1)
where $V_{FS} = V_{MAX} - V_{MIN}$, i.e. the voltage range and $N$ is the number of bits used in the ADC. The quantization will introduce an error in the conversion, the quantization error. For an ideal ADC the quantization error will be uniformly distributed on -0.5 LSB to 0.5 LSB [Sundström, 2011]. Figure 2.2 shows the quantization error of an ideal ADC.

2.3 Static Performance Metrics

The static performance metrics are parameters that evaluates the performance of an ADC with a DC or a very slow ramp as input signal [IEEE Std 1241, 2000]. The most common static performance metrics are described below.

2.3.1 Offset Error

The offset error of an ADC is defined as the deviation of the ADC’s transfer function from the ideal ADC’s transfer function at the first transition level, i.e. from output code ‘0’ to ‘1’. The error is measured in LSB and defined as positive if the transition occurs before 0.5 LSB and negative if the transition occurs after 0.5 LSB [Lundsberg, 2002]. Figure 2.3 shows the transfer function of an ADC with positive offset error compared to the transfer function of an ideal ADC.

An offset error will limit the range of the ADC. Positive offset error will give maximum output code before the input level reaches its maximum. For a negative offset error the ADC will output code ‘0’ for small input levels.
2.3 Static Performance Metrics

Figure 2.2: Quantization error of an ideal ADC.

Figure 2.3: Transfer function of an ideal ADC and ADC with positive offset error.
2.3.2 Gain Error

The gain error of an ADC is defined as the deviation of the last step's code center of the transfer function from the code center of an ideal ADC. This is measured after compensating for offset error and also in LSB [Lundsberg, 2002]. Figure 2.4 shows the transfer function of an ADC with positive gain error compared to the transfer function of an ideal ADC.

![Transfer function of an ideal ADC and ADC with positive gain error.](image)

A positive and negative gain error will limit the range of the ADC in the same way as a positive and negative offset error respectively.

2.3.3 Full Scale Error

The full scale error of an ADC is the same as the gain error without compensating for the offset error [Lundsberg, 2002]. Equation 2.2 shows the simple relation between full scale, gain and offset error.

\[
\text{Full Scale Error} = \text{Gain Error} + \text{Offset Error}
\]  

(2.2)

2.3.4 Differential Non-Linearity

The DNL of an ADC is defined as the deviation from the code width of an ideal ADC, i.e. the deviation from one LSB for each code width [Lundsberg, 2002]. The maximum and minimum DNL is often of most interest. Figure 2.5 shows the transfer function of an ADC with DNL error compared to the transfer function of an ideal ADC.
2.4 Dynamic Performance Metrics

The dynamic performance metrics are parameters that evaluates the performance of an ADC with a time varying signal as input, e.g. a sine [IEEE Std 1241, 2000]. The most common dynamic performance metrics are described below.

2.4.1 Signal to Noise Ratio

Signal to noise ratio is defined as the ratio of the power of the signal and the power of the noise. SNR can mathematically be calculated according to
Figure 2.6: Code centers of an ideal ADC and transfer function of an ADC with INL error.

Figure 2.7: Transfer function of an ideal ADC and ADC with code '100' missing.
Equation 2.3 [Atmel, 2011]

\[ \text{SNR} = 10 \cdot \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \] (2.3)

where \( P_{\text{signal}} \) and \( P_{\text{noise}} \) are the power of the signal and noise respectively. Using the mathematical definition in Equation 2.3 SNR will be expressed in dB. For an ideal ADC the SNR can be calculated according to Equation 2.4

\[ \text{SNR} = 6.02N + 1.76 \] (2.4)

which can be derived from Equation 2.3 from the fact that the only noise in an ideal ADC is the quantization noise.

### 2.4.2 Signal to Noise and Distortion Ratio

Signal to noise and distortion ratio can mathematically be calculated according to Equation 2.5 [Atmel, 2011]

\[ \text{SNDR} = 10 \cdot \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}} + P_{\text{distortion}}} \right) \] (2.5)

where \( P_{\text{signal}} \), \( P_{\text{noise}} \) and \( P_{\text{distortion}} \) are the power of the signal, noise and distortion respectively. SNDR is defined as the ratio of the first and the sum of the two other. Using the mathematical definition in Equation 2.5 SNDR will be expressed in dB.

### 2.4.3 Effective Number of Bits

Effective number of bits can mathematically be calculated according to Equation 2.6 [Atmel, 2011].

\[ \text{ENOB} = \frac{\text{SNDR} - 1.76}{6.02} \] (2.6)

Equation 2.6 is obtained from Equation 2.4. ENOB and SNDR represent the same quantity but expressed in bits and dB respectively.

### 2.4.4 Total Harmonic Distortion

Total harmonic distortion is defined as the ratio of the sum of the power of the harmonic components and the signal power. THD can mathematically be calculated according to Equation 2.7 [Atmel, 2011]

\[ \text{THD} = 10 \cdot \log_{10} \left( \frac{P_1 + P_2 + \cdots + P_n}{P_{\text{signal}}} \right) \] (2.7)

where \( P_{\text{signal}} \) is the power of the signal and \( P_i, i \in [1, n] \) is the power of the i:th harmonic.
2.4.5 Spurious Free Dynamic Range

Spurious free dynamic range is defined as the ratio of the input signal power over the power of the largest peak of spurious. SFDR can mathematically be calculated according to Equation 2.8 [Atmel, 2011]

\[
\text{SFDR} = 10 \cdot \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{spurious}}} \right)
\]  

(2.8)

where \(P_{\text{signal}}\) is the power of the signal and \(P_{\text{spurious}}\) is the power of the largest peak of spurious.

2.5 Architectures

Depending on what an ADC is intended to be used for there are many architectures to choose from. Some very important parameters are sampling rate, resolution, power consumption etc. While some architectures will provide high sampling rate others will provide a higher resolution but at a lower sampling rate. The most common ADC architectures are described below.

2.5.1 Flash

Flash ADCs have a high conversion rate but at the price of drastically increased power consumption for increased resolution. The high conversion rate in the flash ADC is because of its parallel structure. A resistance ladder of \(2^N\) resistances is used for an N-bit flash ADC to generate reference voltages. Each resistance is followed by a comparator. During the conversion the output from the S/H circuit is compared to all the reference voltages to determine the closest one, i.e. compared to all reference voltages in parallel [Elbornsson, 2003].

2.5.2 Pipelined

A pipelined ADC uses a pipeline of low resolution (two to three bits) flash ADCs. In the first stage of the pipeline the output from the S/H circuit is converted with a flash ADC to get the MSB. Next a DAC is used to subtract the converted part of the signal from the output from the S/H circuit. In the next stage the process is repeated but with the difference, the quantization error, from the first stage as input to the S/H circuit. This is repeated for the desired number of stages. A pipelined ADC can convert a new sample in each stage when the conversion is done since each stage has its own S/H circuit. For a pipelined ADC the conversion time and power consumption grows linearly with increased resolution [Elbornsson, 2003].

2.5.3 Sigma-Delta

The sigma-delta ADC uses a one-bit ADC with a feedback loop with a one-bit DAC. This together with a complex digital part consisting of a noise shaping
filter and decimation defines the sigma-delta ADC architecture. The sigma-delta ADC uses oversampling to achieve very high precision but at the cost of low conversion rate [Elbornsson, 2003].

2.5.4 SAR

A SAR ADC uses a binary search algorithm to convert the output from the S/H circuit. A resistance/capacitance ladder/network together with a comparator and digital logic is used to do the binary search. An N-bit SAR ADC needs N comparisons to convert the output from the S/H circuit. The benefit of a SAR ADC is the need for only one comparator and only more resistances/capacitors are needed for higher resolution which gives a low power consumption and low cost [Elbornsson, 2003]. The SAR ADC will be further discussed in Chapter 3.

2.6 Conclusion

In this chapter the fundamentals of ADCs have been discussed. Focus was set on both static and dynamic performance metrics. The most common static performance metrics such as gain and offset error as well as INL and DNL have been properly defined and explained with examples. The chapter also showed the mathematical definitions of dynamic performance metrics including SNR, SNDR, ENOB, THD and SFDR. Popular ADC architectures such as flash, pipeline, sigma-delta and SAR were also briefly discussed.
3

Successive Approximation Register
Analog to Digital Converter

3.1 Introduction

This chapter discusses the SAR ADC architecture and its building blocks. Focus is set on different DAC architectures that can be used with the SAR ADC.

3.2 The SAR ADC

The SAR ADC uses a binary search algorithm to convert the analog input level to a digital output code. There are two different architectures that are normally used with the SAR ADC. One which uses a separate DAC and S/H circuit and one which uses a charge redistribution architecture. In the charge redistribution architecture the DAC is also used to sample the input. The advantage of the charge redistribution architecture is that it consumes less power than the architecture with a separate DAC and S/H circuit [Sundström, 2011].

3.3 Algorithm

The binary search algorithm that is used in the SAR ADC is a simple process that is repeated for N cycles for an N-bit SAR ADC. When a new value is sampled the SAR logic sets the MSB to ‘1’ and it is sent to the DAC. The comparator compares the sampled value and the output of the DAC. If the output value of the DAC is greater than the sampled value then the MSB is set to ‘0’, otherwise kept as ‘1’. This procedure is repeated for MSB-1 all the way down to the LSB [Sundström, 2011]. An example of a conversion is showed in Figure 3.1.
Successive Approximation Register Analog to Digital Converter

Figure 3.1: Example of the conversion algorithm for a three-bit SAR ADC.

In the example in Figure 3.1 the input is first compared to the output of the DAC with digital code '100'. This results in that the MSB should be kept as '1'. Next the input is compared to the output given by code '110', now the output of the DAC is greater than the input resulting in that the bit is reset to '0'. In the final step the input is compared to code '101' and since the output of the DAC is smaller than the input the LSB is kept. The conversion results in the output code '101'.

3.4 Building Blocks

An N-bit SAR ADC consists of four blocks, one analog and three digital. A S/H circuit, N-bit DAC and SAR logic are the digital blocks and a comparator is the only analog block. These parts together with reference voltages and clocks will convert the input level to an output code in N clock cycles [A. Rodriguez-Perez and Medeiro, 2011]. The structure of the SAR ADC can be seen in Figure 3.2.

3.4.1 Sample and Hold

The S/H circuit basically consists of a switch and a capacitor. Two modes are used to sample the input, track mode and hold mode. When the sampling signal (control signal) goes high the S/H circuit goes into track mode and tracks the input. Then when the sampling signal goes low it switches to hold mode and outputs a constant voltage until the control signal once again goes high [Elbornsson, 2003].
3.4 Building Blocks

Figure 3.2: Layout of a SAR ADC showing the different building blocks.

3.4.2 Comparator

The comparator is the only analog block in the SAR ADC. This is where the actual conversion process takes place. The comparator compares the sampled input level with the output of the DAC and outputs a logic ‘0’ or ‘1’. The result of this comparison is sent to the SAR for further processing [Sundström, 2011].

3.4.3 Digital to Analog Converter

The DAC takes the digital code from the SAR and converts the code to an analog value which is sent to the comparator. A DAC can use resistors or capacitors and there exist different architectures which can be used. Four of these DAC architectures which uses capacitors are discussed below.

Binary Weighted

An N-bit binary weighted capacitor array consists of $N + 1$ capacitors in parallel. Figure 3.3 shows an N-bit binary weighted capacitor array. The ratio between two neighbouring capacitors is two for all except for the last two (LSB), which have a ratio of one. This will give capacitors of sizes $C_N = 2^{N-1} C_u$, $C_{N-1} = 2^{N-2} C_u$, ... , $C_2 = 2C_u$, $C_1 = C_u$, $C_0 = C_u$ where $C_u$ is the unit capacitor. This will give a total capacitance of $2^N C_u$. As can be seen in the previous expression the area and power consumption of the binary weighted capacitor array increases with the resolution. [A. Rodriguez-Perez and Medeiro, 2011].

Two Stage

A two stage weighted capacitor array consists of two binary weighted capacitor arrays with a coupling capacitor in series with them. Figure 3.4 shows an N-bit two stage weighted capacitor array. The maximum capacitance of each stage will be $2^{N/2-1} C_u$ and the coupling capacitor will be $C_{coupling} = 2^N/2^{N/2-1}$. This will reduce the total capacitance compared to the binary weighted capacitor array and thus reduce the area and power consumption. [A. Rodriguez-Perez and Medeiro, 2011].
Figure 3.3: Capacitor network of an N-bit binary weighted capacitor array.

Figure 3.4: Capacitor network of an N-bit two stage weighted capacitor array.

C-2C

The C-2C structure is a developed version of the two stage weighted capacitor array. Instead of only two stages, N-1 stages is used for an N-bit C-2C capacitor array. Figure 3.5 shows an N-bit C-2C capacitor array. Capacitors with odd index and index zero will have a capacitance of \( C_u \) and the capacitors with even index will have a capacitance of \( 2C_u \). This will reduce the total capacitance even further and thus consume less power and occupy less area. The big problem with this structure is the influence of parasitic capacitors which will cause a degradation of the linearity [Cong, 2001].

Figure 3.5: Capacitor network of an N-bit C-2C/C-xC capacitor array.
C-xC

The C-xC structure is a further developed version of the C-2C structure. As the name suggests, capacitors of a size not equal of two is used instead. The C-xC has the same structure as the C-2C which can be seen in Figure 3.5. The difference is that the capacitors with even index will have a capacitance of $xC_u$, where $x$ is larger than two. This structure will also have the drawback of parasitic capacitors. Redundancy will also be present in this architecture because $x$ is larger than two, thus there will be a need for extra approximation steps as well as a calibration algorithm [Jansson, 2012]. The use of this DAC structure in a SAR ADC will be discussed more in Section 4.3.1 and Chapter 5.

3.4.4 Successive Approximation Register

The SAR is the control unit of the SAR ADC and from where the architecture got its name. It contains the control logic which determines each bit in the SAR ADC using the binary search algorithm. The SAR logic performs the binary search algorithm by controlling the switches connected to the DAC and the information received from the comparator [A. Rodriguez-Perez and Medeiro, 2011].

3.5 Time-Interleaved

TI ADCs are used to increase the sampling rate of an ADC. The idea is to increase the sampling frequency of the ADC by using several slower ADCs and have them working together in time-multiplexed mode. The different ADCs are interleaved in time which increase the effective sampling frequency linearly with the number of ADCs used. This will however require that the sampling is fast enough to sample the signal [Elbornsson, 2003].

3.6 Conclusion

In this chapter the SAR ADC including its building block and the binary search algorithm it uses to convert an analog input have been presented. The functionality of each of the four building blocks was discussed but focus was set on SAR ADC’s using charge redistribution. SAR ADC using charge redistribution uses a capacitor network to perform both the sampling and conversion of the analog input. The three most common architectures of capacitor networks, binary weighted, two stage and C-2C were presented as well the special C-xC architecture.
4 Correction and Calibration

4.1 Introduction

This chapter discusses correction and calibration of ADCs. Specific methods for the SAR ADC architecture from different articles are presented. The calibration method used for AnaCatum’s SAR ADC is also briefly explained.

4.2 Analog Correction/Calibration

The first attempts at correction and calibration of ADCs were made in the analog domain. The disadvantages of this are many, for example increased size and power consumption as well as it requires extra clock cycles to complete the calibration.

4.3 Digital Correction/Calibration

With the disadvantages of performing calibration of ADCs in the analog domain, the correction and calibration were moved to the digital domain. There are two different types of digital calibration, foreground and background calibration. In digital foreground calibration the normal operation is interrupted and calibration is done, i.e. the conversion phase is interrupted and the calibration phase starts. When using digital background calibration the calibration is done during the normal operation, i.e. the conversion and calibration phase are both running at the same time.
4.3.1 Methods/Techniques

There are many methods and techniques developed to perform correction and calibration of ADCs. The different methods minimize or optimize different aspects of the ADC. A few correction/calibration methods are listed below.

[Hotta et al., 2010] suggests a method to improve the reliability and speed of a conventional binary search SAR ADC by using three redundant comparators and a DAC with three reference voltages. This structure can be seen as a block diagram in Figure 1 in [Hotta et al., 2010]. In the figure the sampled signal is sent to the three comparators, which is equivalent to a two-bit flash ADC, and the output of each comparator is used in an encoder before the signal reaches the SAR logic. Redundancy in the comparators allows the comparators to take the wrong decision since this can be accounted for later in the process. All of this together with an error correction algorithm showed that a ten-bit SAR ADC with three comparators improved the conversion frequency. At six MHz the corner frequency was improved from three MHz to five MHz.

[McNeill et al., 2011] suggests a split ADC architecture for the SAR ADC to calibrate for nonlinearity errors caused by capacitor mismatch. In the split ADC architecture a single ADC is split into two ADCs as shown in the block diagram in Figure 2 in the article. The analog part of two ADCs in the structure consists of a capacitive DAC together with a network of switches and a comparator. The DAC is divided into blocks representing four bits each. Redundancy is built in between the blocks to allow for calibration of the ADC in a later step. These two ADCs convert the same signal and the two results are then digitally corrected and averaged to get the correct output code. The correction is run independently in both ADCs using estimates of capacitor mismatch errors. A background calibration algorithm to estimate the correction parameters is used with the difference between the two ADCs as input. Behavioral simulations gave convergence within 200 000 samples for a 16-bit one Msps ADC. After convergence INL and DNL were both better than ±1 LSB.

[Oh and Murmann, 2006] suggests a digital background calibration technique for TI SAR ADC that aims to correct analog circuit imperfections. In Figure 7 in [Oh and Murmann, 2006] Oh and Murmann shows a block diagram of their proposed architecture. In the figure the TI ADCs can be seen and how they are each individually calibrated. During the calibration the output of the ADC, currently under calibration, is run through an fast Fourier transform which output is used to correct the ADC such that the offset of each frequency bin will be equal to zero. The technique cancels mismatch between the channels in the TI architecture using communication protocol redundancy. An improvement of SNDR from 20 to 37 dB was shown through simulations of a six-bit 500-MS/s ADC using this calibration technique.

[Arpaia et al., 2009] suggests a method to compensate for the non-ideality of the S/H circuit which causes dynamic nonlinearities. The modeling of the dynamic nonlinearity is done accordingly to Figure 5 in [Arpaia et al., 2009].
There the phase distortion have been modeled analytically to get the phase-plane transfer characteristic of the SAR ADC under ideal conditions, i.e. no static nonlinearity. The method to compensate for the error, discussed in the article, applies a compensation technique to this function that maximizes the SNDR. Both simulations and real measurements showed improvements of the dynamic nonlinearities and SNDR.

[Keskin and Chew, 2011] suggests an offset and two gain error correction techniques for a SAR ADC. The offset and one gain error correction technique uses bottom-plate sampling and the other gain error correction technique uses charge sharing between the capacitors. Using the information from the bottom-plate sampling of a correction capacitor the two methods either shift up or down the sampled voltage. The other correction technique uses the ratio between the S/H capacitor and a correction capacitor to scale the sampled voltage. Figure 1, 2 and 3 in [Keskin and Chew, 2011] shows the three circuits for the different techniques discussed by Keskin and Chew. All the methods improved offset and gain error respectively. Gain error was improved from 65.5 lsb to 0.38 lsb and offset error from 10.6 lsb to 0.56 lsb. The simplicity of the methods can clearly be seen in the figures.

[Tong et al., 2012] suggests a self-calibration technique for a SAR ADC by correcting for mismatch in each capacitor independently. The errors are found during power-up and the correction during conversion is done using individual calibration DACs. The block diagram in Figure 3 in [Tong et al., 2012] shows the structure of the proposed self-calibrating SAR. In the figure the extra calibration DACs and the extra switches used to control the calibration are visible. Theses extra calibration DACs are charged during the startup, calibration phase, and then used during the conversion phase to correct the mismatch in each capacitor. Simulations on a 12-bit SAR ADC improved SNDR and SFDR with approximately nine and 22 dB respectively. The simulations also showed that nonlinearity errors caused by capacitor mismatch were reduced.

[Cho et al., 2010] suggests a capacitor reduction technique to reduce the area of a SAR ADC. The ADC uses a binary weighted split capacitor array with a merged capacitor switching technique. For a nine-bit SAR ADC with the proposed technique the number of unit capacitor used is reduced by approximately 50% compared to a normal binary weighted capacitor array. Also offset cancellation in the comparator and digital calibration for error correction was used. Figure 1 in [Cho et al., 2010] shows the SAR ADC architecture proposed by Cho and his colleagues. The interesting part is the DAC which can be seen more in detail in Figure 2 in [Cho et al., 2010]. Here the merged capacitor switching technique with a binary weighted split capacitor array is shown and it is shown how it reduces the number of unit capacitors needed. Measurements on the nine-bit ADC showed an DNL of 0.37 lsb, INL of 0.40 lsb, SNDR of 50.71 dB, SFDR of 66.72 dB and ENOB of 8.13 bits.

[Gururaj et al., 2011] suggests a method to enhance the conversion speed of SAR ADC. This is done by decreasing the number of comparisons needed for
an N-bit SAR ADC from $N$ down to $N - 5$. To reduce the number of comparisons a structure combining a five-bit flash ADC and SAR ADC is used. A block diagram of the proposed architecture for an eight-bit SAR ADC can be seen in Figure 1 in [Gururaj et al., 2011]. The figure shows how the sampled signal is first sent through the five-bit code generator, this is the step that reduces the number of comparisons needed in the architecture. The output is then used in a three-bit SAR ADC to achieve a total resolution of eight bits. All of this is controlled by an eight-bit microcontroller. Simulations for an eight-bit SAR ADC with the proposed method improved the speed with 62.5% and showed a DNL of 0.47 LSB and an INL of 0.5 LSB.

**AnaCatum’s Calibration Algorithm**

This algorithm uses redundancy and calibrates the error caused by parasitic capacitors in a C-xC capacitor network used in a SAR ADC. The algorithm estimates each bit’s weight by measuring it using the LSBs. Using the acquired data a calibration algorithm is used to determine the weight of each bit. The algorithm and theory behind it will be further discussed in Chapter 5 and evaluated in Chapter 6.

### 4.4 Conclusion

In this chapter a few different methods and techniques developed during the past ten years to improve the performance of ADCs have been presented. The different methods aimed at improving the static or dynamic performance or the sampling frequency. All the methods showed good improvements towards their respective target. A summary of the methods can be found in Table 4.1.
Table 4.1: A summary of all the methods and techniques discussed in the chapter.

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Hotta et al., 2010]</td>
<td>Improves reliability and conversion speed using three redundant comparators.</td>
</tr>
<tr>
<td>[McNeill et al., 2011]</td>
<td>Uses a split ADC architecture to calibrate errors caused by capacitor mismatch.</td>
</tr>
<tr>
<td>[Oh and Murmann, 2006]</td>
<td>An algorithm that uses communication protocol redundancy to correct analog circuit imperfection.</td>
</tr>
<tr>
<td>[Arpaia et al., 2009]</td>
<td>Uses a compensation algorithm to compensate for the non-ideality of the S/H circuit.</td>
</tr>
<tr>
<td>[Keskin and Chew, 2011]</td>
<td>Methods using bottom-plate sampling or a correction capacitor to correct for offset and gain error.</td>
</tr>
<tr>
<td>[Tong et al., 2012]</td>
<td>A self-calibration technique that corrects for mismatch in each capacitor independently.</td>
</tr>
<tr>
<td>[Cho et al., 2010]</td>
<td>Using a split capacitor array with a merged capacitor switching technique the area of the ADC is reduced.</td>
</tr>
<tr>
<td>[Gururaj et al., 2011]</td>
<td>Increases the conversion speed by reducing the number of comparisons needed by combining a flash and SAR ADC.</td>
</tr>
</tbody>
</table>
5.1 Introduction

This chapter presents the model that was developed. Both theory and calculations behind the model and its calibration algorithm as well as the code are discussed.

5.2 C-xC Theory

The theory behind the C-xC SAR ADC architecture is based on [Jansson, 2012]. Figure 5.1 shows the C-xC SAR ADC architecture. The structure uses a combination of a C-xC capacitor network and directly weighted MSBs.

![Figure 5.1: The architecture of the C-xC SAR ADC.](image)
5.2.1 Parasitic Capacitors

The influence of parasitic capacitors, $C_p$, in the $xC$ capacitors is not well defined and thus the ratio $xC/C$ is difficult to determine. This will affect the real capacitor values $C'$ and the actual voltages $v'_i$

$$C = C' + C_p$$ (5.1)

$$v_i = \frac{C'}{C' + C_p} \cdot v'_i$$ (5.2)

where $C_p$ is the parasitic capacitor and $C, v$ describes how the parasitic capacitor affects the real capacitor value $C'$ and the actual voltage $v'_i$. As can be seen in Equation 5.1 and Equation 5.2 the parasitic capacitor changes the ratio between links with a big uncertainty. However, these ratios will be stable over time which makes it possible to use calibration to correct for the errors it causes.

5.2.2 Redundancy

Because $x$ is larger than two there will be redundancy in the capacitor network. Therefore it is necessary to use extra approximation steps to get the desired resolution. Defining the ratio between weights in a C-xC link as

$$r_{CxC} = \frac{w_i}{w_{i-1}}, i \in [0, n + m - n_{MSB}]$$ (5.3)

where $w_i, w_{i-1}$ are bit weights, $n$ the desired number of bits, $m$ the number of extra bits needed and $n_{MSB}$ the number of directly weighted MSB. At any point in time the redundancy can be calculated as the sum of the less significant weights of the weight of the bit under conversion minus the LSB weight, this gives

$$\text{redundancy} = \sum_{j=1}^{i} \frac{w_i}{r_{CxC}^j} - (w_i - w_0)$$ (5.4)

where $r_{CxC}^j$ is the ratio for weight $j$, it is also known that the ratio for weight $w_0$ is

$$w_0 = \frac{w_i}{r_{CxC}^i}$$ (5.5)

Using Equation 5.5 and calculating the sum in Equation 5.4 give the total redundancy

$$\text{redundancy} = \frac{2 - r_{CxC}}{r_{CxC} - 1} \cdot \left(1 - \frac{1}{r_{CxC}^i}\right) \cdot w_i$$ (5.6)

Calculating the redundancy with $r_{CxC} = 2$ gives zero as expected.
Maximum Ratio

Assuming that a redundancy of $\pm \epsilon$ ($\pm 0.05$ is a suitable choice) of the remaining conversion range is needed, i.e. a relative redundancy of $2\epsilon$. This together with Equation 5.6 give the condition

$$w_i \cdot 2 \cdot \epsilon \geq \frac{2 - r_{CxC}}{r_{CxC} - 1} \cdot \left(1 - \frac{1}{r_i^{CxC}}\right) \cdot w_i$$  (5.7)

Rearranging Equation 5.7 and assuming that the term $r_i^{CxC}$ is small give an upper limit for the ratio.

$$r_{CxC} \leq \frac{2 + 2\epsilon}{1 + 2\epsilon} = r_{max}$$  (5.8)

Minimum Ratio

The ratio between MSB and LSB in a binary weighted capacitor array is

$$\frac{MSB}{LSB} = 2^{n-1}$$  (5.9)

where $n$ is the number of bits. Defining the ratio between directly weighted MSB capacitors

$$r_{MSB} = \frac{w_i}{w_{i-1}}, \ i \in [m + n - 1, m + n - (n_{MSB} - 1)]$$  (5.10)

where $w_i, w_{i-1}$ are bit weights, $n$ the desired number of bits, $m$ the number of extra bits needed and $n_{MSB}$ the number of directly weighted MSB. The ratio between MSB and LSB in the C-xC capacitor network with $n_{MSB}$ directly weighted MSB.

$$\frac{MSB}{LSB} = r_{MSB}^{(n_{MSB}-1)} \cdot r_{CxC}^{(m+n-n_{MSB})}$$  (5.11)

Using Equation 5.9 as a lower limit in Equation 5.11 gives

$$2^{n-1} \leq r_{MSB}^{(n_{MSB}-1)} \cdot r_{CxC}^{(m+n-n_{MSB})}$$  (5.12)

Rearranging Equation 5.12 gives a lower limit for the ratio.

$$r_{CxC} \leq \left(2^{n-1} \cdot r_{MSB}^{(1-n_{MSB})}\right)^{1/(m+n-n_{MSB})} = r_{min}$$  (5.13)

Design Ratio

The nominal $r_{CxC}$ to design for is either the arithmetic (Equation 5.14) or the geometric (Equation 5.15) mean of Equation 5.8 and Equation 5.13.

$$r_{nom} = \frac{r_{min} + r_{max}}{2}$$  (5.14)
\[ r_{nom} = \sqrt{r_{\text{min}} \cdot r_{\text{max}}} \]  

(5.15)

### 5.2.3 Capacitor Ratios

Assuming that the impedance, \( C_{\text{imp}} \) is the same for all C-xC links and breaking down a link according to Figure 5.2 it is possible to calculate the ratio \( r_{\text{CxC}} \).

![Figure 5.2: A C-xC link and its \( C_{\text{imp}} \).](image)

\[ r_{\text{CxC}} = \frac{C + C_{\text{imp}}}{C_{\text{imp}}} \]  

(5.16)

where \( E_i, E_{i-1} \) and \( C_{\text{imp}} \) can be seen in Figure 5.2. \( C_{\text{imp}} \) is also the capacitance seen when looking into the link.

\[ C_{\text{imp}} = \frac{xC(C + C_{\text{imp}})}{xC + C + C_{\text{imp}}} \]  

(5.17)

Solving Equation 5.17 with respect to \( C_{\text{imp}} \) gives

\[ C_{\text{imp}} = \frac{1}{2}(\sqrt{4x + 1} - 1)C \]  

(5.18)

Using Equation 5.16 to solve for Equation 5.18 for \( x \) results in

\[ x = \frac{1}{4} \left( \left(1 + \frac{2}{r_{\text{CxC}}} \right)^2 - 1 \right) \]  

(5.19)

Equation 5.19 can be used to calculate the capacitor ratios using the ratio \( r_{\text{CxC}} \) which was derived in Section 5.2.2.
5.3 Bit Weight Estimation

The theory behind the bit weight estimation is based on [Jansson, 2012]. The bit weight estimation consists of two steps. In the first step data for each bit is acquired and in the second step this data is used to calculate the bit weights.

5.3.1 Bit Weights

Equation 5.20 defines the relation between two bit weights.

\[ w_i = w_{i-1} \cdot r_i \]  \hspace{1cm} (5.20)

where \( w_i \) relates to bit \( b_i \) and \( r_i \) are the ratio between bit weight \( w_i \) and \( w_{i-1} \). Sampling the analog bit weight and converting it with the remaining bit weight will yield the expression

\[ w_i = \sum_{j=0}^{i-1} a_{i,j} \cdot w_j \]  \hspace{1cm} (5.21)

where \( a_{i,j} \) is the average of the comparator decisions for bit \( j \) made in the measurement series for the sampled bit \( i \). \( w_0 \) is an undetermined gain parameter which will be determined in the last step of the bit weight estimation. Thus it is useful to define relative bit weights, \( \omega_i \).

\[ \omega_i = \frac{w_i}{w_0} \iff w_i = \omega_i \cdot w_0 \]  \hspace{1cm} (5.22)

Equation 5.20 and Equation 5.21 can then be rewritten using Equation 5.22.

\[ \omega_i = \omega_{i-1} \cdot r_i \]  \hspace{1cm} (5.23)

\[ \omega_i = \sum_{j=0}^{i-1} a_{i,j} \cdot \omega_j \]  \hspace{1cm} (5.24)

Measure

Without offset in the comparator it is simple to measure each bit weight. Applying a reference voltage to the desired bit and convert it with the normal SAR ADC conversion process using the LSBs. However, there is often offset in the comparator and how to handle this problem can be found in [Jansson, 2012].

5.3.2 Calibration Algorithm

The calibration algorithm, where the bit weights are calculated, is divided into three parts. One part where the bit weights for a specified number of the LSBs are calculated and a second part where the bit weights for the remaining MSBs
are calculated. In a last step the bit weights are normalized to fit the desired resolution.

**Estimating the L LSBs**

For the L LSBs there are no need to find the small individual spread between their ratios nor is the resolution high enough to allow this small spread to be detected. The ratio \( r \) for the L LSBs.

\[
\omega_i = r^i, \quad i \in [1, L-1]
\]  

(5.25)

Equation 5.23 and Equation 5.25 give that

\[
\omega_i = r^i, \quad i \in [1, L-1]
\]  

(5.26)

Using Equation 5.24 to calculate the relative bit weight for bit \( L-1 \) gives

\[
\omega_{L-1} = \sum_{j=0}^{L-2} a_{L-1,j} \cdot \omega_j
\]  

(5.27)

Combining Equation 5.27 and Equation 5.26 result in

\[
r_{L-1} = \sum_{j=0}^{L-2} a_{L-1,j} \cdot r^j
\]  

(5.28)

Solving Equation 5.28 will give the desired ratio, \( r \) between the L LSBs. This is done using iterative successive approximation where \( r \) is assumed to be in the range \( 1.5 < r \leq 2 \). With the calculated \( r \) the relative bit weights for the L LSBs can be calculated using Equation 5.26.

**Estimating the remaining MSBs**

For the remaining MSBs \( i \geq L \) it is trivial to find the relative bit weights. Equation 5.24 can be used to determine the relative bit weights since the L LSBs relative bit weights are already known.

**Scaling**

To get the bit weights from the relative bit weights Equation 5.22 is used. The gain parameter \( w_0 \) is calculated according to Equation 5.29, which simply scales the weights to cover the desired input range

\[
w_0 = \frac{2^n - 1}{\sum_{i=nCal}^{n+m} w_i}
\]  

(5.29)

where \( n \) is the desired number of bits, \( m \) is the number of extra bits needed because of redundancy and \( nCal \) is the extra bits used during calibration.
The programming language Python was chosen to develop the model of the C-xC SAR ADC. Python offers the same functionality as Matlab/Octave but a more natural way to define classes and inheritance, thus giving it a better overview and structure of the code. The model was developed with help of the theory in Section 5.2 and Section 5.3 but mostly using the calculations on capacitor networks in Appendix A.

5.4.1 Classes

Description and usage of the different classes that were developed for the model can found below. Suggestions on how to develop the classes and introduce new classes can be found in Section 7.2.

Comparator

The comparator class is a simple class with only one function, comparing two values. When creating an instance of this object it is possible to add offset and Gaussian noise.

C-xC

The C-xC class models the C-xC capacitor network using a matrix, which was obtained through calculations of C-xC capacitor networks which can be found in Appendix A. The class has one function which solves the matrix equation and returns the calculated voltage at the node connected to the comparator.

SAR ADC

The SAR ADC class models the complete C-xC SAR ADC using a comparator and C-xC instance and has two functions, convert and calibrate. Convert simply takes an input and converts it to a digital word using switches together with the comparator and C-xC. Calibrate performs a calibration of the SAR ADC, i.e. it calculates the weight for each bit using the theory in Section 5.3.

5.5 Conclusion

In this chapter a lot of theory behind the C-xC architecture was covered. Some time was spent on theory and calculations behind parasitic capacitors, redundancy and capacitor ratios. The fundamentals behind the calibration algorithm, bit weight estimation, was also briefly explained. The biggest part of this chapter, calculations during the conversion and calibration phase on capacitor networks was included as an appendix. These calculations were the base for the understanding of the C-xC capacitor network. All this theory were used to make a class based model in Python to be used for simulations of the C-xC SAR ADC.
6 Simulation Results

6.1 Introduction

This chapter presents the simulations which were performed with the developed model. Both simulations to find the best parameters and performance evaluation with the chosen parameters were conducted.

6.2 Parameters

When designing the eight-bit C-xC SAR ADC there are several parameters that must be chosen. The parameters can be found in Table 6.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>The number of extra bits needed for full resolution</td>
</tr>
<tr>
<td>( n_{\text{MSB}} )</td>
<td>The number of directly weighted MSB</td>
</tr>
<tr>
<td>( r_{\text{MSB}} )</td>
<td>The ratio between the directly weighted MSB</td>
</tr>
<tr>
<td>( x )</td>
<td>The capacitor value</td>
</tr>
<tr>
<td>( n_{\text{CAL}} )</td>
<td>The number of extra bits used during calibration</td>
</tr>
<tr>
<td>( L )</td>
<td>The number of LSBs with the same ratio</td>
</tr>
</tbody>
</table>

The parameter \( m \) is the only one that can be directly calculated. Using that the capacitor ratios are designed with a relative redundancy of ten percent it is simple to calculate the number of extra bits needed. When using eight bits one extra bit is needed since \( 1.10 \cdot 8 = 8.8 \), i.e. \( m = 1 \). The parameter \( r_{\text{MSB}} \) will
for simplicity be set to two, i.e. the \( n_{\text{MSB}} \) directly weighted MSB will be binary weighted. The rest of the parameters, \( n_{\text{MSB}}, x, n_{\text{CAL}} \) and \( L \) will be chosen using a parameters sweep, see Section 6.2.1.

### 6.2.1 Parameter Sweep

Using the Python model developed in Chapter 5 a parameter sweep over the parameters \( n_{\text{MSB}}, x, n_{\text{CAL}} \) and \( L \) was made. Looking at Equation 5.19 it can be seen that the value of \( x \) can be calculated knowing all the other parameters. Therefore it is only necessary to find suitable sweep ranges for the other three parameters. AnaCatum’s 12-bit C-xC SAR ADC uses four directly weighted MSB, three extra bits for calibration and the seven LSBs have the same ratio. Since this is an eight-bit C-xC SAR ADC, it is not good to have more than four directly weighted MSB since it will then behave more like a normal binary weighted capacitor array. It should also not require more extra calibration bits than the 12-bit ADC. The number of LSBs with the same ratio should include at least the extra calibration bits but it should not include the directly weighted MSB. The range for the sweep for each parameter can be found in Table 6.2.

**Table 6.2:** Search range for parameters in the parameter sweep.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_{\text{MSB}} )</td>
<td>1,...,4</td>
</tr>
<tr>
<td>( n_{\text{CAL}} )</td>
<td>1,...,3</td>
</tr>
<tr>
<td>( L )</td>
<td>( n_{\text{CAL}} + 1,...,7 + m + n_{\text{CAL}} - n_{\text{MSB}} )</td>
</tr>
</tbody>
</table>

For each set of parameters a ramp covering the whole conversion range (65536 values between zero and 255) was converted and INL was calculated according to the description in Section 6.3.2. Table 6.3 shows the sets of parameters that gave the best INL.

**Table 6.3:** The parameters with the best INL from the parameter sweep.

<table>
<thead>
<tr>
<th>( n_{\text{MSB}} )</th>
<th>( n_{\text{CAL}} )</th>
<th>( L )</th>
<th>( \text{INL}_{\text{min}} )</th>
<th>( \text{INL}_{\text{max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>-0.9948</td>
<td>0.9138</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>-0.9930</td>
<td>0.9264</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>-0.9717</td>
<td>0.8943</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>-0.9765</td>
<td>0.8998</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>6</td>
<td>-1.0358</td>
<td>0.9529</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>-0.9644</td>
<td>0.8897</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>8</td>
<td>-1.0157</td>
<td>0.9477</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>-0.9735</td>
<td>0.9321</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>6</td>
<td>-1.0195</td>
<td>0.9253</td>
</tr>
</tbody>
</table>

As can be seen in Table 6.3 there are a lot of parameters giving approximately
the same relative INL. Since AnaCatum’s 12-bit C-xC SAR ADC uses three extra calibration bits there should be enough to use at most two extra calibrations in an eight-bit C-xC SAR ADC. Using four directly weighted MSB does not either seem to be a proper choice. Because then half the ADC would be a normal binary weight capacitor array. Having a higher $L$ will speed up the calibration algorithm. Therefore the most suitable choice of parameters seems to be the second row in Table 6.3. Table 6.4 summarizes the result of the parameter sweep, i.e. all the chosen parameters.

Table 6.4: List of C-xC SAR ADC parameters and their best values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>1</td>
</tr>
<tr>
<td>$n_{MSB}$</td>
<td>1</td>
</tr>
<tr>
<td>$r_{MSB}$</td>
<td>2</td>
</tr>
<tr>
<td>$x$</td>
<td>2.464</td>
</tr>
<tr>
<td>$n_{CAL}$</td>
<td>2</td>
</tr>
<tr>
<td>$L$</td>
<td>5</td>
</tr>
</tbody>
</table>

6.3 Performance

Using the chosen parameters in Table 6.4 the static and dynamic performance were evaluated as well as a comparison between converting with and without a calibrated C-xC SAR ADC.

6.3.1 Calibration

To see if the calibration works, the same input was converted with and without running the calibration algorithm first. As input, a ramp was chosen and the results around the middle of the code range can be seen in Figure 6.1 and Figure 6.2.

As can be seen in Figure 6.1 there are steps in the output at the MSBs, i.e. there are missing codes. This comes from that the SAR ADC is not calibrated and thus binary weights are used, which are not ideal in this case when the capacitor ratios are not binary weighted. In Figure 6.2 it can be seen that the calibration removes all of these steps occurring at the MSBs. From that it seems like the calibration algorithm does what it is supposed to do.

6.3.2 Static

To evaluate the static performance a ramp is chosen as input to both an uncalibrated and a calibrated SAR ADC. Figure 6.3 shows the converted ramp when no calibration of the ADC has been done and Figure 6.4 shows the converted ramp from a calibrated ADC.
Figure 6.1: Output around middle code range with uncalibrated ADC.

Figure 6.2: Output around middle code range with calibrated ADC.
6.3 Performance

Figure 6.3: Output when using a ramp as input from the uncalibrated ADC.

Figure 6.4: Output when using a ramp as input from the calibrated ADC.
Comparing Figure 6.3 and Figure 6.4 it can be seen, just like in the comparison between Figure 6.1 and Figure 6.2, that the calibration smooths the ramp and that there are no missing codes after calibration. Since this is not a binary ADC the code widths and step heights will not all be the same, even without noise, offset or other errors. Thus it is not interesting to look at either the gain, offset or full scale error. Neither is the DNL worth looking at since the code widths will not be equal to one LSB. INL is still interesting but needs to be measured in a different way, which not uses the DNL. To measure INL the best line through all the points is estimated using the least squares method and subtracted from the data. This gives the result in Figure 6.5 for the uncalibrated ADC and in Figure 6.6 for the calibrated ADC.

\begin{figure}[ht]
\centering
\includegraphics[width=\textwidth]{inl_plot}
\caption{INL of converted ramp from the uncalibrated ADC.}
\end{figure}

Figure 6.5, the uncalibrated ADC, shows a minimum INL of $-7.26$ LSB and a maximum INL of $8.59$ LSB. This compared to Figure 6.6, the calibrated ADC, which shows a minimum INL of $-0.99$ LSB and a maximum INL of $0.93$ LSB. Thus the calibration algorithm improves the INL of the eight-bit C-xC SAR ADC with more than 85%.

### 6.3.3 Dynamic

To evaluate the dynamic performance a sine is chosen as input. First, a conversion is made with an uncalibrated ADC and secondly with a calibrated ADC. Looking at Figure 6.7a and Figure 6.7b much difference can not be seen between the uncalibrated and calibrated ADC. But by zooming in on one period of the sine the difference between the uncalibrated and calibrated ADC becomes more clear. Comparing Figure 6.8 and Figure 6.9 it can be seen, exactly
6.3 Performance

**Figure 6.6:** INL of converted ramp from the calibrated ADC.

like in the case with the ramp as input, that the calibration algorithm smooths the output.

**Figure 6.7:** Comparison between output when using sine as input to an uncalibrated and a calibrated ADC.

Calculating the spectrum of the output from the uncalibrated and calibrated ADC results in Figure 6.10 and Figure 6.11.
Figure 6.8: Zoomed in on output when using sine as input to the uncalibrated ADC.

Figure 6.9: Zoomed in on output when using sine as input to the calibrated ADC.
6.3 Performance

Figure 6.10: Spectrum of converted sine from the uncalibrated ADC.

Figure 6.11: Spectrum of converted sine from the calibrated ADC.
In Figure 6.11, the spectrum for the calibrated ADC, there is not any harmonics or spurious present. Thus it is only interesting to look at the SNR and ENOB for both the uncalibrated and calibrated ADC. Calculating these dynamic performance metrics for the uncalibrated ADC result in an SNR of 33.93 dB and ENOB of 4.66 bits. For the calibrated ADC the SNR is 47.85 dB and ENOB is 7.65 bits. The calibration algorithm improves the SNR with almost 14 dB which is an improvement of 40% and ENOB with almost three bits. Compared to an ideal ADC the calibrated eight-bit C-xC SAR ADC is not far away. This is not unexpected since the Python model does not model any noise or other errors in the ADC. In an ideal ADC the only noise present is the quantization noise. Calculating the SNR for ideal an eight-bit ADC according do Equation 2.4 gives an SNR of 49.92 dB. Because the ADC is ideal there will be no distortion and thus the SNDR will equal the SNR. This results in an ENOB of eight bits.

6.4 Conclusion

In this chapter the results of different simulations with the model were presented. The simulations were divided into two parts. In the first part a set of simulations were made to find the best choice of parameters for the eight-bit C-xC SAR ADC. In the second part these parameters were used to evaluate the static and dynamic performance of the ADC. Since the model does not model noise or other errors and because of the nature of the C-xC SAR ADC only INL, SNR and ENOB were of interest. The calibrated eight-bit C-xC SAR ADC showed an INL of less than ±1 LSB, SNR of 47.8 dB and ENOB of 7.6 bits. The use of calibration was a big impact on these numbers as it improved the INL with more than 85% and the SNR and ENOB with almost 14 dB and three bits respectively.
Conclusions

7.1 Summary

In this master’s thesis a model of an N-bit C-xC SAR ADC with digital calibration based on AnaCatum’s 12-bit C-xC SAR ADC was developed. The special C-xC SAR ADC architecture uses charge redistribution in a C-xC capacitor network to perform a normal SAR ADC conversion. A C-xC structure reminds a lot of the the C-2C structure where the binary relation between the capacitor is replaced with a non-binary relation, hence the x in the name instead of 2. Focus in this master’s thesis was set to understand how the charge is redistributed in the network when switches connected are switched between different reference voltages.

The use of a non-binary relation gives redundancy in the system, i.e. multiply output codes are possible for the same input. This results in that extra calibration cycles is needed for full resolution. The system is also affected by parasitic capacitors and therefore a calibration algorithm is needed. The calibration algorithm, which is called Bit Weight Estimation, starts by sampling each bit and then using the normal SAR conversion expressing that bit in its LSB. This data is used to calculate a weight for each bit, which is mainly done using a binary search algorithm. These weights is later used in the last step of the normal SAR conversion process to multiply each bit with its corresponding weight to get the final result of the conversion.

The model was written in Python and built up using classes representing the different parts of the C-xC SAR ADC. A simple class to model the comparator to allow for modeling of noise and offset was created, but during simulations both the noise and offset was set to zero. Using extensive calculations in C-
xC capacitor networks of different sizes a class representing the C-xC capacitor network could be developed. The calculations resulted in a matrix that modeled all the nodes in the capacitor network. During the conversion and calibration phase this matrix could be used to create a matrix equation which could be solved to yield the voltage at each node in the capacitor network. The last class put it all together by combining a comparator and C-xC class. It performs the calibration and conversion by controlling the switches connected to capacitor network and sending the appropriate voltages to the comparator and using the result of the comparison.

The developed model was then used to create a model of an eight-bit C-xC SAR ADC. To find suitable parameters for the model of the eight-bit ADC was the next task. A few of the parameters could be calculated mathematically but most of the could not be. These parameters were chosen by doing a sweep over a suitable range and the set of parameters showing the best INL was chosen. The best parameter values can be found in Table 7.1.

**Table 7.1: List and explanation of C-xC SAR ADC parameters and their best values.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>1</td>
<td>The number of extra bits needed for full resolution</td>
</tr>
<tr>
<td>n&lt;sub&gt;MSB&lt;/sub&gt;</td>
<td>1</td>
<td>The number of directly weighted MSB</td>
</tr>
<tr>
<td>r&lt;sub&gt;MSB&lt;/sub&gt;</td>
<td>2</td>
<td>The ratio between the directly weighted MSB</td>
</tr>
<tr>
<td>x</td>
<td>2.464</td>
<td>The capacitor value</td>
</tr>
<tr>
<td>n&lt;sub&gt;CAL&lt;/sub&gt;</td>
<td>2</td>
<td>The number of extra bits used during calibration</td>
</tr>
<tr>
<td>L</td>
<td>5</td>
<td>The number of LSBs with the same ratio</td>
</tr>
</tbody>
</table>

Using the chosen parameters a few simulations were made to determine the performance of the ADC. The simulations aimed at confirming that the calibration algorithm works and to evaluate the static, in this case only the INL, and dynamic, in this case the SNR and ENOB, performance metrics. Comparing the uncalibrated and calibrated the ADC showed an improvement in INL from $-7.26/8.59$ LSB to $-0.99/0.93$ LSB, i.e. an improvement of more than 80%. For the dynamic performance metrics, SNR improved from 33.90dB to 47.85dB, i.e. an improvement of more than 70%, and ENOB from 4.66 bits to 7.65 bits, i.e. an improvement of more than 60%.

### 7.2 Future Work

There is a lot of work that could be done in the future to further develop this model and take it to the next level. A first step would be to improve the
Python model. This could be done in numerous ways. One way is to further develop the existing classes and for example allow for non-binary directly weighted MSB and to handle offset in the comparator. A good way to improve the model would also be to extend with new classes, for example a switch, capacitor and source class. Improving the existing and extending with new classes would make the model more realistic. This would also allow for more errors to be easily introduced in the model. Some examples of errors are:

- Settling time in switches
- Mismatch in capacitors
- kTC noise

With all these new errors introduced, new simulations will have to be made to evaluate the performance. Also new simulations, probably a parameter sweep will have to be made to find the most suitable parameters. This because the newly introduced errors will have an effect on the performance and thus other parameters may be better. With all these new errors introduced the simulations and the evaluate of the performance would be more realistic. The second step in the modeling would be to construct a hardware model. Then use the same method as with the Python model to evaluate its performance. In other words, first create a model that is as ideal as possible and then further develop it and introduce all kinds of possible error sources. The last step would be to create and actual ADC to run real performance tests on.
This appendix presents the calculations on different capacitor networks which were done in order to develop the model.

A.1 Conversion

To develop a model for the conversion phase in an N-bit C-xC calculations is first made for lower resolutions and then generalized to N-bit.

A.1.1 Two-Bit C-xC

This example shows how to calculate the voltages at the two nodes in a two-bit C-xC capacitor array. Figure A.1 shows the structure of the two-bit C-xC capacitor array.

In Figure A.1 $V_0$ and $V_1$ are voltage nodes where $V_1$ is connected to the comparator. $S_0$ and $S_1$ are switches used by the SAR logic to determine the bits and $S_R$ and $S_S$ are switches used during the sample and hold phase.

During the sample phase $S_0$, $S_1$, and $S_S$ are connected to $V_{IN}$ and $S_R$ to ground. This causes $V_1 = 0$ and the charges $q_i$ can be calculated over each capacitor $C_i$.

\[
q_3 = (V_{IN} - V_1)C = V_{IN}C \quad (A.1a)
\]
\[
q_2 = (V_0 - V_1)xC = V_0xC \quad (A.1b)
\]
\[
q_1 = (V_{IN} - V_0)C \quad (A.1c)
\]
\[
q_0 = (V_{IN} - V_0)(y - 1)C \quad (A.1d)
\]
A Capacitor Network Calculations

Next during the hold phase $S_0$, $S_1$ and $S_S$ are connected to ground and $S_R$ is disconnected. After that is the bit set phase where $S_0$ and $S_1$ are used by the SAR logic to perform the binary search algorithm. During this phase the charges $\tilde{q}_i$ can be calculated over each capacitor $C_i$

$$\tilde{q}_3 = (b_1 - \tilde{V}_1)C$$  \hspace{1cm} (A.2a)
$$\tilde{q}_2 = (\tilde{V}_0 - \tilde{V}_1)xC$$  \hspace{1cm} (A.2b)
$$\tilde{q}_1 = (b_0 - \tilde{V}_0)C$$  \hspace{1cm} (A.2c)
$$\tilde{q}_0 = (0 - \tilde{V}_0)(y - 1)C = -\tilde{V}_0(y - 1)C$$  \hspace{1cm} (A.2d)

where $b_i$ equals zero or $V_{REF}$ depending on the corresponding switch $S_i$ and $\tilde{V}_i$ is the voltage at node $V_i$ during the bit set phase. Using Equations A.1 and Equations A.2 together with the principle of charge preservation, new equations can be derived for each node. In node $V_0$ charge preservation gives that $q_1 + q_0 - q_2 = \tilde{q}_1 + \tilde{q}_0 - \tilde{q}_2$ which results in Equation A.3

$$(V_0 - \tilde{V}_0)(-x - y) + (0 - \tilde{V}_1)x = b_0 - yV_{IN}$$  \hspace{1cm} (A.3)

and in node $V_1$ that $q_3 + q_2 = \tilde{q}_3 + \tilde{q}_2$ which results in Equation A.4

$$(V_0 - \tilde{V}_0)x + (0 - \tilde{V}_1)(-1 - x) = b_1 - V_{IN}$$  \hspace{1cm} (A.4)

Matrix Form

Using the notation $\Delta V_k = V_k - \tilde{V}_k$ and that $V_1 = 0$, Equation A.3 and Equation A.4 can be written in matrix form according to Equation A.5

$$\begin{pmatrix} -x - y & x \\ x & -1 - x \end{pmatrix} \begin{pmatrix} \Delta V_0 \\ \Delta V_1 \end{pmatrix} = \begin{pmatrix} b_0 - yV_{IN} \\ b_1 - V_{IN} \end{pmatrix}$$  \hspace{1cm} (A.5)
and in an even more compact form according to Equation A.6

$$\begin{pmatrix} -x - y \\ x \\ -1 - x \end{pmatrix} \Delta V = b - \begin{pmatrix} y \\ 1 \end{pmatrix} V_{IN}$$ (A.6)

where $\Delta V$ is the vector $\begin{pmatrix} \Delta V_0 \\ \Delta V_1 \end{pmatrix}^T$ and $b$ is the vector $\begin{pmatrix} b_0 \\ b_1 \end{pmatrix}^T$.

**Example**

Solving the matrix equation in Equation A.6 for $x = 2 \Rightarrow y = 2$ yields

$$\begin{pmatrix} -4 \\ 2 \\ -3 \end{pmatrix} \Delta V = b - \begin{pmatrix} 2 \\ 1 \end{pmatrix} V_{IN}$$ (A.7)

Solving Equation A.7 and looking at the interesting voltage at node $V_1$, which is connected to the comparator, give Equation A.8.

$$\Delta V_1 = V_{IN} - \frac{b_1}{2} - \frac{b_0}{4} \Leftrightarrow \tilde{V}_1 = \frac{b_1}{2} + \frac{b_0}{4} - V_{IN}$$ (A.8)

Equation A.8 is the expected result of a two-bit C-2C capacitor array, i.e. a C-xC capacitor array with $x$ equal to two.

**A.1.2 Three-Bit C-xC**

This example shows how to calculate the voltages at the three nodes in a three-bit C-xC capacitor array. Figure A.2 show the structure of the three-bit C-xC capacitor array.

![Figure A.2: Structure of a three-bit C-xC capacitor array.](image)

In Figure A.2 $V_0$, $V_1$ and $V_2$ are voltage nodes where $V_2$ is connected to the comparator. $S_0$, $S_1$ and $S_2$ are switches used by the SAR logic to determine the bits and $S_R$ and $S_S$ are switches used during the sample and hold phase.

During the sample phase $S_0$, $S_1$, $S_2$ and $S_S$ are connected to $V_{IN}$ and $S_R$ to ground. This causes $V_2 = 0$ and the charges $q_i$ can be calculated over each
A Capacitor Network Calculations

Capacitor $C_i$:

\[ q_5 = (V_{IN} - V_2)C = V_{IN}C \]  
\[ q_4 = (V_1 - V_2)C = V_1xC \] 
\[ q_3 = (V_{IN} - V_1)C \] 
\[ q_2 = (V_0 - V_1)xC \] 
\[ q_1 = (V_{IN} - V_0)C \] 
\[ q_0 = (V_{IN} - V_0)(y - 1)C \]  

Next during the hold phase $S_0$, $S_1$, $S_2$ and $S_5$ are connected to ground and $S_R$ is disconnected. After that is the bit set phase where $S_0$, $S_1$ and $S_2$ are used by the SAR logic to perform the binary search algorithm. During this phase the charges $\tilde{q}_i$ can be calculated over each capacitor $C_i$

\[ \tilde{q}_5 = (b_2 - \tilde{V}_2)C \]  
\[ \tilde{q}_4 = (\tilde{V}_1 - \tilde{V}_2)xC \] 
\[ \tilde{q}_3 = (b_1 - \tilde{V}_1)C \] 
\[ \tilde{q}_2 = (\tilde{V}_0 - \tilde{V}_1)xC \] 
\[ \tilde{q}_1 = (b_0 - \tilde{V}_0)C \] 
\[ \tilde{q}_0 = (0 - \tilde{V}_0)(y - 1)C = -\tilde{V}_0(y - 1)C \]

where $b_i$ equals zero or $V_{REF}$ depending on the corresponding switch $S_i$ and $\tilde{V}_i$ is the voltage at node $V_i$ during the bit set phase. Using Equations A.9 and Equations A.10 together with the principle of charge preservation, new equations can be derived for each node. In node $V_0$ charge preservation gives that $q_1 + q_0 - q_2 = \tilde{q}_1 + \tilde{q}_0 - \tilde{q}_2$ which results in Equation A.11

\[ (V_0 - \tilde{V}_0)(-x - y) + (V_1 - \tilde{V}_1)x = b_0 - yV_{IN} \]  

and in node $V_1$ that $q_3 + q_2 - q_4 = \tilde{q}_3 + \tilde{q}_2 - \tilde{q}_4$ which results in Equation A.12

\[ (V_0 - \tilde{V}_0)x + (V_1 - \tilde{V}_1)(-1 - 2x) + (0 - \tilde{V}_2)x = b_1 - V_{IN} \]

and in node $V_2$ that $q_5 + q_4 = \tilde{q}_5 + \tilde{q}_4$ which results in Equation A.13

\[ (V_1 - \tilde{V}_1)x + (0 - \tilde{V}_2)(-1 - x) = b_2 - V_{IN} \]

**Matrix Form**

Using the notation $\Delta V_k = V_k - \tilde{V}_k$ and that $V_2 = 0$, Equation A.11, Equation A.12 and Equation A.13 can be written in matrix form according to Equation A.14

\[
\begin{pmatrix}
-x - y & x & 0 \\
 x & -1 - 2x & x \\
 0 & x & -1 - x
\end{pmatrix}
\begin{pmatrix}
\Delta V_0 \\
 \Delta V_1 \\
 \Delta V_2
\end{pmatrix}
= 
\begin{pmatrix}
b_0 - yV_{IN} \\
b_1 - V_{IN} \\
b_2 - V_{IN}
\end{pmatrix}
\]
and in an even more compact form according to Equation A.15

\[
\begin{pmatrix}
-x - y & x & 0 \\
x & -1 - 2x & x \\
0 & x & -1 - x
\end{pmatrix}
\Delta V = b - \begin{pmatrix} y \\ 1 \\ 1 \end{pmatrix} V_{IN} \tag{A.15}
\]

where \( \Delta V \) is the vector \( (\Delta V_0 \ \Delta V_1 \ \Delta V_2)^T \) and \( b \) is the vector \( (b_0 \ b_1 \ b_2)^T \).

**Example**

Solving the matrix equation in Equation A.15 for \( x = 2 \Rightarrow y = 2 \) yields

\[
\begin{pmatrix}
-4 & 2 & 0 \\
2 & -5 & 2 \\
0 & 2 & -3
\end{pmatrix}
\Delta V = b - \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} V_{IN} \tag{A.16}
\]

Solving Equation A.16 and looking at the interesting voltage at node \( V_2 \), which is connected to the comparator, give Equation A.17.

\[
\Delta V_2 = V_{IN} - \frac{b_2}{2} - \frac{b_1}{4} - \frac{b_0}{8} \Leftrightarrow \tilde{V}_2 = \frac{b_2}{2} + \frac{b_1}{4} + \frac{b_0}{8} - V_{IN} \tag{A.17}
\]

Equation A.17 is the expected result of a three-bit C-2C capacitor array, i.e. a C-xC capacitor array with \( x \) equal to two.

**A.1.3 Four-Bit C-xC**

This example shows how to calculate the voltages at the four nodes in a four-bit C-xC capacitor array. Figure A.3 show the structure of the four-bit C-xC capacitor array.

![Figure A.3: Structure of a four-bit C-xC capacitor array.](image)

In Figure A.3 \( V_0, V_1, V_2 \) and \( V_3 \) are voltage nodes where \( V_3 \) is connected to the comparator. \( S_0, S_1, S_2 \) and \( S_3 \) are switches used by the SAR logic to determine the bits and \( S_R \) and \( S_S \) are switches used during the sample and hold phase.

During the sample phase \( S_0, S_1, S_2, S_3 \) and \( S_S \) are connected to \( V_{IN} \) and \( S_R \)
to ground. This causes $V_3 = 0$ and the charges $q_i$ can be calculated over each capacitor $C_i$.

\[
\begin{align*}
q_7 &= (V_{IN} - V_3)C = V_{IN}C \\ 
q_6 &= (V_2 - V_3)xC = V_2xC \\ 
q_5 &= (V_{IN} - V_2)C \\ 
q_4 &= (V_1 - V_2)xC \\ 
q_3 &= (V_{IN} - V_1)C \\ 
q_2 &= (V_0 - V_1)xC \\ 
q_1 &= (V_{IN} - V_0)C \\ 
q_0 &= (V_{IN} - V_0)(y - 1)C
\end{align*}
\]  

(A.18a)

(A.18b)

(A.18c)

(A.18d)

(A.18e)

(A.18f)

(A.18g)

(A.18h)

Next during the hold phase $S_0, S_1, S_2, S_3$ and $S_S$ are connected to ground and $S_R$ is disconnected. After that is the bit set phase where $S_0, S_1, S_2$ and $S_3$ are used by the SAR logic to perform the binary search algorithm. During this phase the charges $\tilde{q}_i$ can be calculated over each capacitor $C_i$

\[
\begin{align*}
\tilde{q}_7 &= (b_3 - \tilde{V}_3)C & (A.19a) \\ 
\tilde{q}_6 &= (\tilde{V}_2 - \tilde{V}_3)xC & (A.19b) \\ 
\tilde{q}_5 &= (b_2 - \tilde{V}_2)C & (A.19c) \\ 
\tilde{q}_4 &= (\tilde{V}_1 - \tilde{V}_2)xC & (A.19d) \\ 
\tilde{q}_3 &= (b_1 - \tilde{V}_1)C & (A.19e) \\ 
\tilde{q}_2 &= (\tilde{V}_0 - \tilde{V}_1)xC & (A.19f) \\ 
\tilde{q}_1 &= (b_0 - \tilde{V}_0)C & (A.19g) \\ 
\tilde{q}_0 &= (0 - \tilde{V}_0)(y - 1)C = -\tilde{V}_0(y - 1)C & (A.19h)
\end{align*}
\]

where $b_i$ equals zero or $V_{REF}$ depending on the corresponding switch $S_i$ and $\tilde{V}_i$ is the voltage at node $V_i$ during the bit set phase. Using Equations A.18 and Equations A.19 together with the principle of charge preservation, new equations can be derived for each node. In node $V_0$ charge preservation gives that $q_1 + q_0 - q_2 = \tilde{q}_1 + \tilde{q}_0 - \tilde{q}_2$ which results in Equation A.20

\[
(V_0 - \tilde{V}_0)(-x - y) + (V_1 - \tilde{V}_1)x = b_0 - yV_{IN}
\]  

(A.20)

and in node $V_1$ that $q_3 + q_2 - q_4 = \tilde{q}_3 + \tilde{q}_2 - \tilde{q}_4$ which results in Equation A.21

\[
(V_0 - \tilde{V}_0)x + (V_1 - \tilde{V}_1)(-1 - 2x) + (V_2 - \tilde{V}_2)x = b_1 - V_{IN}
\]  

(A.21)

and in node $V_2$ that $q_5 + q_4 - q_6 = \tilde{q}_5 + \tilde{q}_4 - \tilde{q}_6$ which results in Equation A.22

\[
(V_1 - \tilde{V}_1)x + (V_2 - \tilde{V}_2)(-1 - 2x) + (0 - \tilde{V}_3)x = b_2 - V_{IN}
\]  

(A.22)

and in node $V_3$ that $q_7 + q_6 = \tilde{q}_7 + \tilde{q}_6$ which results in Equation A.23

\[
(V_2 - \tilde{V}_2)x + (0 - \tilde{V}_3)(-1 - x) = b_3 - V_{IN}
\]  

(A.23)
Matrix Form

Using the notation $\Delta V_k = V_k - \tilde{V}_k$ and that $V_3 = 0$, Equation A.20, Equation A.21, Equation A.22 and Equation A.23 can be written in matrix form according to Equation A.24

$$
\begin{pmatrix}
-x - y & x & 0 & 0 \\
x & -1 - 2x & x & 0 \\
0 & x & -1 - 2x & x \\
0 & 0 & x & -1 - x
\end{pmatrix}
\begin{pmatrix}
\Delta V_0 \\
\Delta V_1 \\
\Delta V_2 \\
\Delta V_3
\end{pmatrix} =
\begin{pmatrix}
b_0 - yV_{IN} \\
b_1 - V_{IN} \\
b_2 - V_{IN} \\
b_3 - V_{IN}
\end{pmatrix}
$$

(A.24)

and in an even more compact form according to Equation A.25

$$
\begin{pmatrix}
-x - y & x & 0 & 0 \\
x & -1 - 2x & x & 0 \\
0 & x & -1 - 2x & x \\
0 & 0 & x & -1 - x
\end{pmatrix}
\Delta V =
\begin{pmatrix}
y \\
1 \\
1 \\
1
\end{pmatrix}
V_{IN}
$$

(A.25)

where $\Delta V$ is the vector $(\Delta V_0 \quad \Delta V_1 \quad \Delta V_2 \quad \Delta V_3)^T$ and $b$ is the vector $(b_0 \quad b_1 \quad b_2 \quad b_3)^T$.

Example

Solving the matrix equation in Equation A.25 for $x = 2 \Rightarrow y = 2$ yields

$$
\begin{pmatrix}
-4 & 2 & 0 & 0 \\
2 & -5 & 2 & 0 \\
0 & 2 & -5 & 2 \\
0 & 0 & 2 & -3
\end{pmatrix}
\Delta V =
\begin{pmatrix}
2 \\
1 \\
1 \\
1
\end{pmatrix}
V_{IN}
$$

(A.26)

Solving Equation A.26 and looking at the interesting voltage at node $V_3$, which is connected to the comparator, give Equation A.27.

$$
\Delta V_3 = V_{IN} - \frac{b_3}{2} - \frac{b_2}{4} - \frac{b_1}{8} - \frac{b_0}{16} \Leftrightarrow \tilde{V}_3 = \frac{b_3}{2} + \frac{b_2}{4} + \frac{b_1}{8} + \frac{b_0}{16} - V_{IN}
$$

(A.27)

Equation A.27 is the expected result of a four-bit C-2C capacitor array, i.e. a C-xC capacitor array with $x$ equal to two.

A.1.4 N-Bit C-xC

Using the analyses in Section A.1.1, Section A.1.2 and Section A.1.3 general equations for an N-bit C-xC capacitor array can be derived. Using the notation $\Delta V_k = V_k - \tilde{V}_k$ we can derive three simple expressions for each node. In node $V_0$

$$
\Delta V_0(-x - y) + \Delta V_1 x = b_0 - yV_{IN}
$$

(A.28)
and in node $V_k$, $0 < k < N - 1$

$$\Delta V_{k-1}x + \Delta V_k(-1 - 2x) + \Delta V_{k+1}x = b_k - V_{IN} \quad (A.29)$$

and in node $V_{N-1}$

$$\Delta V_{N-2}x + \Delta V_{N-1}(-1 - x) = b_{N-1} - V_{IN} \quad (A.30)$$

**Matrix Form**

Equation A.28, Equation A.29 and Equation A.30 can all be written together in a matrix form according to Equation A.31

$$\begin{pmatrix} -x - y & x & 0 & \ldots & \ldots & \ldots & 0 \\ x & -1 - 2x & x & 0 & \vdots \\ 0 & x & -1 - 2x & x & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & 0 & x & -1 - 2x & x & 0 \\ 0 & \ldots & \ldots & \ldots & 0 & x & -1 - x \end{pmatrix} \begin{pmatrix} y \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{N-1} \end{pmatrix} \quad (A.31)$$

where $\Delta V$ is the vector $(\Delta V_0 \; \Delta V_1 \; \ldots \; \Delta V_{N-1})^T$ and $b$ is the vector $(b_0 \; b_1 \; \ldots \; b_{N-1})^T$.

### A.2 Bit Weight Estimation

To develop a model for the bit weight estimation phase in an N-bit C-xC calculations are first made for a 3-bit C-xC and then generalized to N-bit.

#### A.2.1 Three-Bit C-xC

This example shows how the bit weight estimation is done in a three-bit C-xC capacitor array. Figure A.2 shows the structure of the three-bit C-xC capacitor array. The calculations are done in three steps, in each step one bit is determined.

**Determine bit two, MSB**

During the sample phase $S_0$, $S_1$, $S_5$ and $S_R$ are connected to ground and $S_2$ is connected to $V_{REF}$. This causes $V_2 = 0$ and the charges $q_i$ can be calculated...
A.2 Bit Weight Estimation

over each capacitor $C_i$.

$$q_5 = (V_{REF} - V_2)C = V_{REF}C \quad (A.32a)$$
$$q_4 = (V_1 - V_2)xC = V_1xC \quad (A.32b)$$
$$q_3 = (0 - V_1)C = -V_1C \quad (A.32c)$$
$$q_2 = (V_0 - V_1)xC \quad (A.32d)$$
$$q_1 = (0 - V_0)C = -V_0C \quad (A.32e)$$
$$q_0 = (0 - V_0)(y - 1)C = -V_0(y - 1)C \quad (A.32f)$$

Next during the hold phase $S_0, S_1, S_2$ and $S_S$ are connected to ground and $S_R$ is disconnected. After that is the bit set phase where $S_0$ and $S_1$ are used by the SAR logic to perform the binary search algorithm. During this phase the charges $q_i$ can be calculated over each capacitor $C_i$.

$$\tilde{q}_5 = (0 - \tilde{V}_2)C = -\tilde{V}_2C \quad (A.33a)$$
$$\tilde{q}_4 = (\tilde{V}_1 - \tilde{V}_2)xC \quad (A.33b)$$
$$\tilde{q}_3 = (b_1 - \tilde{V}_1)C \quad (A.33c)$$
$$\tilde{q}_2 = (\tilde{V}_0 - \tilde{V}_1)xC \quad (A.33d)$$
$$\tilde{q}_1 = (b_0 - \tilde{V}_0)C \quad (A.33e)$$
$$\tilde{q}_0 = (0 - \tilde{V}_0)(y - 1)C = -\tilde{V}_0(y - 1)C \quad (A.33f)$$

where $b_0$ and $b_1$ equals zero or $V_{REF}$ depending on the corresponding switch $S_i$ and $\tilde{V}_i$ is the voltage at node $V_i$ during the bit set phase. Using Equations A.32 and Equations A.33 together with the principle of charge preservation, new equations can be derived for each node. In node $V_0$ charge preservation gives that $q_1 + q_0 - q_2 = \tilde{q}_1 + \tilde{q}_0 - \tilde{q}_2$ which results in Equation A.34

$$\Delta V_0(-x - y) + \Delta V_1x = b_0 \quad (A.34)$$

and in node $V_1$ that $q_3 + q_2 - q_4 = \tilde{q}_3 + \tilde{q}_2 - \tilde{q}_4$ which results in Equation A.35

$$\Delta V_0x + \Delta V_1(-1 - 2x) + \Delta V_2x = b_1 \quad (A.35)$$

and in node $V_2$ that $q_5 + q_4 = \tilde{q}_5 + \tilde{q}_4$ which results in Equation A.36

$$\Delta V_1x + \Delta V_2(-1 - x) = -V_{REF} \quad (A.36)$$

Equation A.34, Equation A.35 and Equation A.36 can be written in matrix form according to Equation A.37

$$ \begin{pmatrix} -x - y & x & 0 \\ x & -1 - 2x & x \\ 0 & x & -1 - x \end{pmatrix} \begin{pmatrix} \Delta V_0 \\ \Delta V_1 \\ \Delta V_2 \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} V_{REF} \quad (A.37)$$
Determine bit one

During the sample phase $S_0$, $S_2$, $S_S$ and $S_R$ are connected to ground and $S_1$ is connected to $V_{REF}$. This causes $V_2 = 0$ and the charges $q_i$ can be calculated over each capacitor $C_i$.

\begin{align*}
q_5 &= (0 - V_2)C = 0 \quad (A.38a) \\
q_4 &= (V_1 - V_2)x_C = V_1x_C \quad (A.38b) \\
q_3 &= (V_{REF} - V_1)C \quad (A.38c) \\
q_2 &= (V_0 - V_1)x_C \quad (A.38d) \\
q_1 &= (0 - V_0)C = -V_0C \quad (A.38e) \\
q_0 &= (0 - V_0)(y - 1)C = -V_0(y - 1)C \quad (A.38f)
\end{align*}

Next during the hold phase $S_0$, $S_1$, $S_2$ and $S_S$ are connected to ground and $S_R$ is disconnected. After that is the bit set phase where $S_0$ is used by the SAR logic to perform the binary search algorithm. During this phase the charges $q_i$ can be calculated over each capacitor $C_i$.

\begin{align*}
\tilde{q}_5 &= (0 - \tilde{V}_2)C = -\tilde{V}_2C \quad (A.39a) \\
\tilde{q}_4 &= (\tilde{V}_1 - \tilde{V}_2)x_C \quad (A.39b) \\
\tilde{q}_3 &= (0 - \tilde{V}_1)C = -\tilde{V}_1C \quad (A.39c) \\
\tilde{q}_2 &= (\tilde{V}_0 - \tilde{V}_1)x_C \quad (A.39d) \\
\tilde{q}_1 &= (b_0 - \tilde{V}_0)C \quad (A.39e) \\
\tilde{q}_0 &= (0 - \tilde{V}_0)(y - 1)C = -\tilde{V}_0(y - 1)C \quad (A.39f)
\end{align*}

where $b_0$ equals zero or $V_{REF}$ depending on the corresponding switch $S_i$ and $\tilde{V}_i$ is the voltage at node $V_i$ during the bit set phase. Using Equations A.38 and Equations A.39 together with the principle of charge preservation, new equations can be derived for each node. In node $V_0$ charge preservation gives that $q_1 + q_0 - q_2 = \tilde{q}_1 + \tilde{q}_0 - \tilde{q}_2$ which results in Equation A.40

\[ \Delta V_0(-x - y) + \Delta V_1 x = b_0 \quad (A.40) \]

and in node $V_1$ that $q_3 + q_2 - q_4 = \tilde{q}_3 + \tilde{q}_2 - \tilde{q}_4$ which results in Equation A.41

\[ \Delta V_0 x + \Delta V_1(-1 - 2x) + \Delta V_2 x = -V_{REF} \quad (A.41) \]

and in node $V_2$ that $q_5 + q_4 = \tilde{q}_5 + \tilde{q}_4$ which results in Equation A.42

\[ \Delta V_1 x + \Delta V_2(-1 - x) = 0 \quad (A.42) \]

Equation A.40, Equation A.41 and Equation A.42 can be written in matrix form according to Equation A.49

\[
\begin{pmatrix}
-x - y & x & 0 \\
0 & -1 - 2x & x \\
0 & x & -1 - x
\end{pmatrix}
\begin{pmatrix}
\Delta V_0 \\
\Delta V_1 \\
\Delta V_2
\end{pmatrix}
= 
\begin{pmatrix}
b_0 \\
0 \\
0
\end{pmatrix}
- 
\begin{pmatrix}
0 \\
1 \\
0
\end{pmatrix}
V_{REF} 
\quad (A.43)
Determine bit zero, LSB

During the sample phase \( S_1, S_2, S_S \) and \( S_R \) are connected to ground and \( S_0 \) is connected to \( V_{REF} \). This causes \( V_2 = 0 \) and the charges \( q_i \) can be calculated over each capacitor \( C_i \).

\[
\begin{align*}
q_5 &= (0 - V_2)C = 0 \quad \text{(A.44a)} \\
q_4 &= (V_1 - V_2)xC = V_1xC \quad \text{(A.44b)} \\
q_3 &= (0 - V_1)C = -V_1C \quad \text{(A.44c)} \\
q_2 &= (V_0 - V_1)xC \quad \text{(A.44d)} \\
q_1 &= (V_{REF} - V_0)C \quad \text{(A.44e)} \\
q_0 &= (0 - V_0)(y - 1)C = -V_0(y - 1)C \quad \text{(A.44f)}
\end{align*}
\]

Next during the hold phase \( S_0, S_1, S_2 \) and \( S_S \) are connected to ground and \( S_R \) is disconnected. After that is the bit set phase where no binary search can be performed since the LSB is the last bit. During this phase the charges \( 	ilde{q}_i \) can be calculated over each capacitor \( C_i \).

\[
\begin{align*}
\tilde{q}_5 &= (0 - \tilde{V}_2)C = -\tilde{V}_2C \quad \text{(A.45a)} \\
\tilde{q}_4 &= (\tilde{V}_1 - \tilde{V}_2)xC \quad \text{(A.45b)} \\
\tilde{q}_3 &= (0 - \tilde{V}_1)C = -\tilde{V}_1C \quad \text{(A.45c)} \\
\tilde{q}_2 &= (\tilde{V}_0 - \tilde{V}_1)xC \quad \text{(A.45d)} \\
\tilde{q}_1 &= (0 - \tilde{V}_0)C = -\tilde{V}_0C \quad \text{(A.45e)} \\
\tilde{q}_0 &= (0 - \tilde{V}_0)(y - 1)C = -\tilde{V}_0(y - 1)C \quad \text{(A.45f)}
\end{align*}
\]

where \( \tilde{V}_i \) is the voltage at node \( V_i \) during the bit set phase. Using Equations A.44 and Equations A.45 together with the principle of charge preservation, new equations can be derived for each node. In node \( V_0 \) charge preservation gives that \( q_1 + q_0 - q_2 = \tilde{q}_1 + \tilde{q}_0 - \tilde{q}_2 \) which results in Equation A.46

\[
\Delta V_0(-x - y) + \Delta V_1x = -V_{REF} \quad \text{(A.46)}
\]

and in node \( V_1 \) that \( q_3 + q_2 - q_4 = \tilde{q}_3 + \tilde{q}_2 - \tilde{q}_4 \) which results in Equation A.47

\[
\Delta V_0x + \Delta V_1(-1 - 2x) + \Delta V_2x = 0 \quad \text{(A.47)}
\]

and in node \( V_2 \) that \( q_5 + q_4 = \tilde{q}_5 + \tilde{q}_4 \) which results in Equation A.48

\[
\Delta V_1x + \Delta V_2(-1 - x) = 0 \quad \text{(A.48)}
\]

Equation A.40, Equation A.41 and Equation A.42 can be written in matrix form according to Equation A.49

\[
\begin{pmatrix}
-x - y & x & 0 \\
x & -1 - 2x & x \\
0 & x & -1 - x
\end{pmatrix}
\begin{pmatrix}
\Delta V_0 \\
\Delta V_1 \\
\Delta V_2
\end{pmatrix} =
\begin{pmatrix}
1 \\
0 \\
0
\end{pmatrix} V_{REF} \quad \text{(A.49)}
\]
A.2.2 N-Bit C-xC

Using the analyses in Section ?? and Section A.2.1 it can be concluded that during the bit weight estimation phase the left hand side of the matrix equation that needs to be solved will be the same as during the conversion phase. But for the right hand side Equation A.50 is used

\[ b - v V_{REF} \]  

(A.50)

where \( b \) is an \( N \times 1 \) vector where \( b[i] = 0 \) for \( i \) larger than or equal to the bit that is currently being determined and the others are determined through the normal SAR ADC conversion process. \( v \) is also an \( N \times 1 \) vector with zeros everywhere except for \( v[i] = 1 \) where \( i \) is the bit currently being determined.

A.3 Directly Weighted MSBs

To further develop the model to include directly weighted MSBs, calculations are first made for a four-bit C-xC and then generalized to N-bit. The calculations are done for the conversion phase but will be applied to the bit weight estimation phase as well.

A.3.1 Four-Bit C-xC

This example shows how the bit conversion is done in a four-bit C-xC capacitor array with directly weighted MSBs. The calculations are done in four steps, in each step one more bit is set to be directly weighted.

One Directly Weighted MSB

With only one directly weighted bit the structure will be exactly the same as in Figure A.3. This will result in the same equation as in Equation A.25.

Two Directly Weighted MSB

This example shows how to calculate the voltages at the four nodes in a four-bit C-xC capacitor array with two directly weighted MSB. Figure A.4 show the structure of the four-bit C-xC capacitor array with two directly weighted MSB.

\[ \text{Figure A.4: Structure of a four-bit C-xC capacitor array with two directly weighted MSB.} \]
A.3 Directly Weighted MSBs

In Figure A.4 $V_0, V_1$ and $V_2$ are voltage nodes where $V_2$ is connected to the comparator. $S_0, S_1, S_2$ and $S_3$ are switches used by the SAR logic to determine the bits and $S_R$ and $S_S$ are switches used during the sample and hold phase.

During the sample phase $S_0, S_1, S_2, S_3$ and $S_S$ are connected to $V_{IN}$ and $S_R$ to ground. This causes $V_2 = 0$ and the charges $q_i$ can be calculated over each capacitor $C_i$.

\[
q_6 = (V_{IN} - V_2)4C = 2V_{IN}C \quad (A.51a)
\]
\[
q_5 = (V_{IN} - V_2)4C = V_{IN}C \quad (A.51b)
\]
\[
q_4 = (V_1 - V_2)xC = V_1xC \quad (A.51c)
\]
\[
q_3 = (V_{IN} - V_1)C \quad (A.51d)
\]
\[
q_2 = (V_0 - V_1)xC \quad (A.51e)
\]
\[
q_1 = (V_{IN} - V_0)C \quad (A.51f)
\]
\[
q_0 = (V_{IN} - V_0)(y - 1)C \quad (A.51g)
\]

Next during the hold phase $S_0, S_1, S_2, S_3$ and $S_S$ are connected to ground and $S_R$ is disconnected. After that is the bit set phase where $S_0, S_1, S_2$ and $S_3$ are used by the SAR logic to perform the binary search algorithm. During this phase the charges $\tilde{q}_i$ can be calculated over each capacitor $C_i$

\[
\tilde{q}_6 = (b_3 - \bar{V}_2)2C \quad (A.52a)
\]
\[
\tilde{q}_5 = (b_2 - \bar{V}_2)C \quad (A.52b)
\]
\[
\tilde{q}_4 = (\bar{V}_1 - \bar{V}_2)xC \quad (A.52c)
\]
\[
\tilde{q}_3 = (b_1 - \bar{V}_1)C \quad (A.52d)
\]
\[
\tilde{q}_2 = (\bar{V}_0 - \bar{V}_1)xC \quad (A.52e)
\]
\[
\tilde{q}_1 = (b_0 - \bar{V}_0)C \quad (A.52f)
\]
\[
\tilde{q}_0 = (0 - \bar{V}_0)(y - 1)C = -\bar{V}_0(y - 1)C \quad (A.52g)
\]

where $b_i$ equals zero or $V_{REF}$ depending on the corresponding switch $S_i$ and $\bar{V}_i$ is the voltage at node $V_i$ during the bit set phase. Using Equations A.51 and Equations A.52 together with the principle of charge preservation, new equations can be derived for each node. In node $V_0$ charge preservation gives that $q_1 + q_0 - q_2 = \tilde{q}_1 + \tilde{q}_0 - \tilde{q}_2$ which results in Equation A.53

\[
(V_0 - \bar{V}_0)(-x - y) + (0 - \bar{V}_1)x = b_0 - yV_{IN} \quad (A.53)
\]

and in node $V_1$ that $q_3 + q_2 - q_4 = \tilde{q}_3 + \tilde{q}_2 - \tilde{q}_4$ which results in Equation A.54

\[
(V_0 - \bar{V}_0)x + (V_1 - \bar{V}_1)(-1 - 2x) + (V_2 - \bar{V}_2)x = b_1 - V_{IN} \quad (A.54)
\]

and in node $V_2$ that $q_6 + q_5 + q_4 = \tilde{q}_6 + \tilde{q}_5 + \tilde{q}_4$ which results in Equation A.55

\[
(V_0 - \bar{V}_0)x + (0 - \bar{V}_1)(-3 - x) = 2b_3 + b_2 - 3V_{IN} \quad (A.55)
\]
Three Directly Weighted MSB

This example shows how to calculate the voltages at the four nodes in a four-bit C-xC capacitor array with three directly weighted MSB. Figure A.5 shows the structure of the four-bit C-xC capacitor array with three directly weighted MSB.

![Figure A.5: Structure of a four-bit C-xC capacitor array with three directly weighted MSB.](image)

In Figure A.5 $V_0$ and $V_1$ are voltage nodes where $V_1$ is connected to the comparator. $S_0$, $S_1$, $S_2$ and $S_3$ are switches used by the SAR logic to determine the bits and $S_R$ and $S_S$ are switches used during the sample and hold phase.

During the sample phase $S_0$, $S_1$, $S_2$, $S_3$ and $S_S$ are connected to $V_{IN}$ and $S_R$ to ground. This causes $V_1 = 0$ and the charges $q_i$ can be calculated over each capacitor $C_i$.

\[
q_5 = (V_{IN} - V_1)4C = 4V_{IN}C \tag{A.56a}
\]
\[
q_4 = (V_{IN} - V_1)xC = 2V_{IN}C \tag{A.56b}
\]
\[
q_3 = (V_{IN} - V_1)C = V_{IN}C \tag{A.56c}
\]
\[
q_2 = (V_0 - V_1)xC = xV_0 \tag{A.56d}
\]
\[
q_1 = (V_{IN} - V_0)C \tag{A.56e}
\]
\[
q_0 = (V_{IN} - V_0)(y - 1)C \tag{A.56f}
\]

Next during the hold phase $S_0$, $S_1$, $S_2$, $S_3$ and $S_S$ are connected to ground and $S_R$ is disconnected. After that is the bit set phase where $S_0$, $S_1$, $S_2$ and $S_3$ are used by the SAR logic to perform the binary search algorithm. During this
phase the charges $\tilde{q}_i$ can be calculated over each capacitor $C_i$

\[ \tilde{q}_5 = (b_3 - \tilde{V}_1)4C \quad (A.57a) \]
\[ \tilde{q}_4 = (b_2 - \tilde{V}_1)2C \quad (A.57b) \]
\[ \tilde{q}_3 = (b_1 - \tilde{V}_1)C \quad (A.57c) \]
\[ \tilde{q}_2 = (\tilde{V}_0 - \tilde{V}_1)xC \quad (A.57d) \]
\[ \tilde{q}_1 = (b_0 - \tilde{V}_0)C \quad (A.57e) \]
\[ \tilde{q}_0 = (0 - \tilde{V}_0)(y - 1)C = -\tilde{V}_0(y - 1)C \quad (A.57f) \]

where $b_i$ equals zero or $V_{REF}$ depending on the corresponding switch $S_i$ and $\tilde{V}_i$ is the voltage at node $V_i$ during the bit set phase. Using Equations A.56 and Equations A.57 together with the principle of charge preservation, new equations can be derived for each node. In node $V_0$ charge preservation gives that $q_1 + q_0 - q_2 = \tilde{q}_1 + \tilde{q}_0 - \tilde{q}_2$ which results in Equation A.58

\[ (V_0 - \tilde{V}_0)(-x - y) + (0 - \tilde{V}_1)x = b_0 - yV_{IN} \quad (A.58) \]

and in node $V_1$ that $q_5 + q_4 + q_3 + q_2 = \tilde{q}_5 + \tilde{q}_4 + \tilde{q}_3 + \tilde{q}_2$ which results in Equation A.59

\[ (V_0 - \tilde{V}_0)x + (0 - \tilde{V}_1)(-7 - x) = 4b_3 + 2b_2 + b_1 - 7V_{IN} \quad (A.59) \]

### Four Directly Weighted MSB

Using four directly weighted MSB will result in a binary weighted capacitor array which is of no interest here.

### A.3.2 N-Bit C-xC with $n$ directly weighted MSB

Using the analysis in Section A.3.1 it can be seen that using directly weighted MSBs will result in fewer voltage nodes but the same equations can be used as in Section A.1.4 except for the node connected to the comparator. With $n$ directly weighted MSB the following equations can be found for each node. In node $V_0$

\[ \Delta V_0(-x - y) + \Delta V_1x = b_0 - yV_{IN} \quad (A.60) \]

and in node $V_k, 0 < k < N - n$

\[ \Delta V_{k-1}x + \Delta V_k(-1 - 2x) + \Delta V_{k+1}x = b_k - V_{IN} \quad (A.61) \]

and in node $V_{N-n}$

\[ \Delta V_{N-n-1}x + \Delta V_{N-n}(-(2^n - 1) - x) = \sum_{i=0}^{n-1} b_{N-n+i} \cdot 2^i - (2^n - 1)V_{IN} \quad (A.62) \]

During the conversion phase Equation A.60, Equation A.61 and Equation A.62 can be used. But during the bit weight estimation phase the right hand side
of each equation will change accordingly to the analysis in Section A.2.


Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekomersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsmann i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© Claes Hallström (claha288@student.liu.se)