

# Design of power-efficient VCO-ADCs using coarse-fine readout structures

Simon Ooghe

Student number: 01809683

Supervisor: Prof. dr. ir. Pieter Rombouts

Counsellors: Ir. Brendan Saux, Ir. Jonas Borgmans

Master's dissertation submitted in order to obtain the academic degree of  
Master of Science in Electrical Engineering - main subject Electronic Circuits and Systems

Academic year 2022-2023



# Design of power-efficient VCO-ADCs using coarse-fine readout structures

Simon Ooghe

Student number: 01809683

Supervisor: Prof. dr. ir. Pieter Rombouts

Counsellors: Ir. Brendan Saux, Ir. Jonas Borgmans

Master's dissertation submitted in order to obtain the academic degree of  
Master of Science in Electrical Engineering - main subject Electronic Circuits and Systems

Academic year 2022-2023

# Preface

Throughout the past year, I have been able to discover the compelling subject of VCO-ADCs. I am very grateful to my supervisor Prof. dr. ir. Pieter Rombouts and counsellors Ir. Brendan Saux and Ir. Jonas Borgmans for introducing me to this subject and supporting me during this journey. Their interest in the topic and in my progress was an important motivation. A heartfelt thank you to Brendan Saux for assisting me with any problems and always offering helpful advice and relevant insights. I am also thankful to the other members of the CAS-team for their support and guidance, in particular Ir. Tobias Cromheecke for his explanation on the digital design. It was always a pleasure to talk and hear about the research progress in the CAS-group.

Thanks to the fellow students in the thesis room for creating a space where we could discuss results or have a laugh while working on our thesis. Finally, I could not have completed this thesis without the support of my parents, sisters, grandparents, and girlfriend, who have always encouraged me to follow my interests and motivated me to overcome challenges.

Simon OOGHE, May 2023

## **Admission to Loan**

The author gives permission to make this master dissertation available for consultation and to copy parts of this master dissertation for personal use. In the case of any other use, the copyright terms have to be respected, in particular with regard to the obligation to state expressly the source when quoting results from this master dissertation.

De auteur geeft de toelating deze masterproef voor consultatie beschikbaar te stellen en delen van de masterproef te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de bepalingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze masterproef.

Simon OOGHE, May 2023

## **Explanation regarding the master's thesis**

This master's dissertation is part of an exam. Any comments formulated by the assessment committee during the oral presentation of the master's dissertation are not included in this text.

# Design of power-efficient VCO-ADCs using coarse-fine readout structures

by

Simon OOGHE

Master's Dissertation submitted to obtain the academic degree of  
Master of Science in Electrical Engineering

Academic year 2022–2023

Supervisor: Prof. dr. ir. Pieter Rombouts  
Counsellors: Ir. Brendan Saux, Ir. Jonas Borgmans

Faculty of Engineering and Architecture  
Ghent University

## Abstract

Analog-to-digital converters (ADCs) are ubiquitous in modern electronics. Voltage controlled oscillator (VCO) based ADCs take advantage of the fast-switching advanced CMOS technologies to perform digital-friendly, efficient analog-to-digital conversion [1]. However, traditional single-bit quantized VCO-ADCs require the sampling frequency to be higher than the maximal VCO frequency. A coarse counter can be added to the VCO readout structure to reduce the necessary sampling frequency, and it was shown by Borgmans et al. [2] that this could lead to a more power-efficient design. In this thesis, an exploration of the specific challenges faced by the coarse-fine VCO-ADC is presented. Asynchrony between the coarse and fine counters is investigated in detail, and a double connected coarse counter is proposed which avoids the effect of asynchrony up to high VCO frequencies with a low power consumption. The other circuits which make up the VCO-ADC are discussed, with a focus on how these affect the noise performance and power consumption of the VCO-ADC. Several crucial parameters are selected and optimized for given specifications on the signal-to-noise ratio (SNR) and third harmonic distortion (HD3). A design algorithm which performs this optimization is developed, explained, and applied to a coarse-fine VCO-ADC design. This design is simulated on circuit level and achieves an SNR of 76.52 dB and a HD3 of  $-37.09$  dB at a bandwidth of 40 MHz. When this VCO-ADC is simulated in a pseudo-differential setup and the output is digitally calibrated, it reaches a signal-to-noise and distortion ratio (SNDR) of 74.86 dB for a power consumption of only 1.862 mW.

## Keywords

Analog-to-digital conversion, VCO, coarse-fine readout, coarse counter asynchrony, power-efficient

# Design of power-efficient VCO-ADCs using coarse-fine readout structures

Simon Ooghe

Supervisor: Prof. dr. ir. Pieter Rombouts

Counsellors: Ir. Brendan Saux, Ir. Jonas Borgmans

**Abstract**—This work presents a method to design coarse-fine VCO-ADCs operating at a high VCO frequency. A design sized using this algorithm was simulated, leading to a VCO-ADC with an SNR of 79.80 dB and an SNDR of 74.86 dB for a power consumption of 1.862 mW. The coarse-fine VCO-ADC is first discussed on a system level with a focus on how the asynchrony between the coarse and fine counter reduces the performance. A connected double coarse counter design is proposed with improved timing constraints and a reduced power consumption through the use of gated NAND-latches. Other building blocks which make up the VCO-ADC are also discussed with a focus on how to design and size these components for a minimal power consumption. By co-optimizing several parameters, a design algorithm for a given SNR and HD3 at a certain bandwidth is obtained, extending the algorithm presented in [1].

**Index Terms**—analog-to-digital conversion, voltage-controlled oscillator, coarse-fine readout, coarse counter asynchrony, power-efficient

## I. INTRODUCTION

To understand and communicate with the world around us, sensors and receivers measure analog signals of physical quantities. An analog-to-digital converter (ADC) is necessary to convert these signals to the digital domain. As with all electronics, there is a desire to process more information at a faster rate. Technology scaling has allowed CMOS digital circuits to achieve the required faster switching speeds. However, the voltage headroom in advanced CMOS technologies is limited, making it a challenging environment to design operational amplifiers and comparators [2]. A more digital-friendly ADC can be obtained using a voltage controlled oscillator (VCO). VCO-ADCs have been designed both for sensors [3], wireless receivers [4], and other applications.

It was suggested in [1] that a VCO-ADC using a coarse-fine readout structure could be optimal to achieve demanding specifications at a minimal power consumption. This work presents a circuit-level design of a coarse-fine VCO-ADC. It deals with the challenge of asynchrony between the coarse and fine counters by proposing a novel coarse counter. An algorithm to combine and size different circuits for the VCO-ADC is also presented and a resulting circuit-level design is simulated.

## II. SYSTEM-LEVEL CONSIDERATIONS

### A. Coarse-Fine VCO-ADC Model

A coarse-fine VCO-ADC is shown in figure 1. The signal is quantized by sampling the location of an edge in a ring oscillator with  $N_\phi$  delay cells. The coarse counter counts up to  $N_c - 1$  full cycles of the ring oscillator, while the fine counters determine the position of the edge within a cycle. The addition of the

coarse counter gives the designer the freedom to determine the sampling frequency  $f_s$  separately from the VCO frequency  $f_{VCO}$ : the condition  $f_s > f_{VCO}$  is relaxed to  $f_s > f_{VCO}/N_c$ . By selecting a suitable value of  $N_c$ ,  $f_s$  can be reduced significantly leading to a decreased power consumption in the sampling and digital stages of the VCO-ADC.

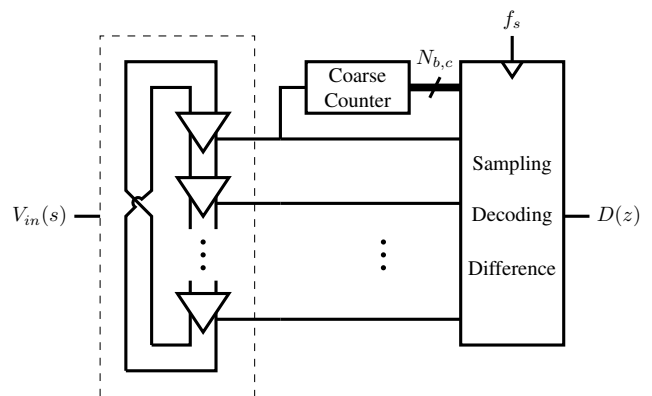


Figure 1: Coarse-fine VCO-ADC

Figure 2 shows a block diagram of the VCO-ADC. The input voltage modulates the VCO frequency, and this is integrated to give the phase, which is quantized in steps of  $\pi/N_\phi$ . The quantized phase is sampled and a difference operation is applied.

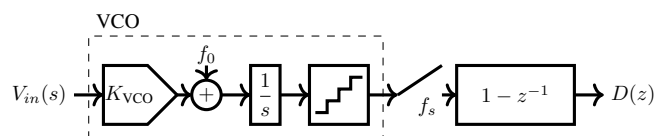


Figure 2: Block diagram of the VCO-ADC

Quantization is typically modelled as discrete additive noise, allowing us to extract equation 1 for the output  $D(z)$  from the block diagram. Based on this transfer function, equation 2 for the signal-to-quantization noise ratio (SQNR) is proposed in [5].

$$D(z) = \left[ \frac{K_{VCO} V_{in}(s) + f_0}{s} \right]^* (1 - z^{-1}) + Q(z) (1 - z^{-1}) \quad (1)$$

$$\text{SQNR} = 20 \log_{10} \left( \frac{2N_\phi f_{tune}}{f_s} \right) + 30 \log_{10} \left( \frac{f_s}{2f_{BW}} \right) - 3.41 \quad (2)$$

### B. Coarse Counter Asynchrony

One factor which can significantly affect the performance of the coarse-fine VCO-ADC is asynchrony between the coarse and fine counters. The coarse counter block in figure 1 will require a certain time to determine its next value. The waveform will therefore be delayed compared to the fine counter, but is sampled at the same time. Figure 3 shows the effect of asynchrony: the coarse counter only takes its next value a time  $\tau_c$  after the falling edge of the reference phase  $V_{\phi,0}$ . On the falling clock edge, the sampled value of the coarse counter is 1 lower than expected, leading to an error of  $2N_\phi$  in the output of the VCO-ADC.

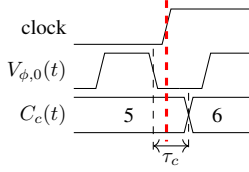


Figure 3: Waveforms showing asynchrony

This asynchrony has a dramatic effect of the quality of the signal. The ratio of the signal to the combined noise due to asynchrony and quantization  $\text{SNR}_{Q,\tau_c}$  is shown in figure 4 for a system-level simulation in Simulink. For the single counter, this is shown in blue and immediately drops steeply from the SQNR when a small  $\tau_c$  is introduced. This can be solved using a double counter, as discussed in [3].

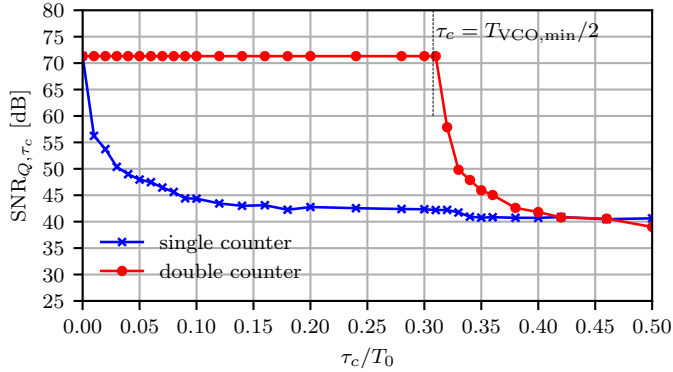


Figure 4: Effect of asynchrony on  $\text{SNR}_{Q,\tau_c}$

The concept of a double counter is shown in figure 5. The value of coarse counter A increases at the rising edge of the phase and counter B increments at the falling edge. If the fine counter value is low, the edge on which coarse counter A reacts is least recent, and therefore its value is selected, and vice versa if the fine counter value is high. The double coarse counter is also simulated in Simulink, and the results are shown as the red dots in figure 4. As visible, the  $\text{SNR}_{Q,\tau_c}$  remains equal to the signal-to-noise ratio (SNR) without asynchrony until the  $\tau_c$  becomes larger than  $T_{\text{VCO,min}}/2$ , half of the minimal VCO period. Longer delays mean that the correct value is still not determined when this coarse counter is again selected. The condition in equation 3 should be met to avoid effects of asynchrony.

$$\tau_c < \frac{T_{\text{VCO,min}}}{2} \quad (3)$$

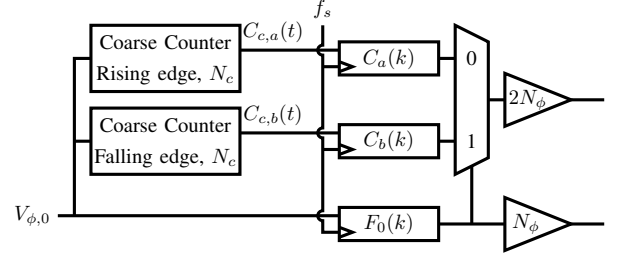


Figure 5: Readout using the double coarse counter.

### C. Double Connected Coarse Counter

Meeting the condition in 3 is difficult with a traditional synchronous or asynchronous counter. In a synchronous counter, the calculation of the next state needs to happen extremely fast and consumes a significant amount of power.

In an asynchronous counter, the calculation ripples through the different flipflops storing  $N_{b,c}$  different bits. If the propagation time of each of the flipflops is written as  $\tau_b$ , the maximal total propagation time is  $\tau_c = N_{b,c}\tau_b$ . This leads to a stricter constraint on  $\tau_b$  to meet the presented condition, especially if a coarse counter with a higher number of bits is required.

The proposed solution is to start from two asynchronous counters but change the connections between flipflops to achieve faster propagation and low power consumption. These connections are shown in figure 6. In this figure, coarse counter A and B each consist of  $N_{b,c}$  gated latches.

The red, dotted connection from the Q output of latch  $na$  to the D input of latch  $nb$  guarantees that the count in coarse counter B will follow the count in counter A correctly. The difference operation can therefore be applied with samples from different counters. This was proposed by Perez et al. in [3].

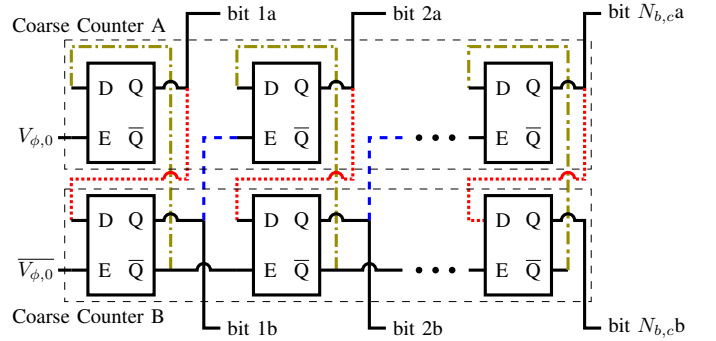


Figure 6: Connected asynchronous binary counters

The blue, dashed connection in 6 is a novel idea presented in this work to feed the value at the data output Q of counter  $nb$  through to the enable input E of counter  $(n+1)a$ . The start of the transition in bit  $(n+1)a$  is now brought forward by half a cycle of the previous bit. This is shown for the first two bits in figure 7. At the falling VCO edge, the rising edge of bit 1b occurs and this will immediately start the transition on bit 2a. The transitions of bit  $na$  and bit  $nb$  will not overlap as long as the condition in



equation 4 is fulfilled. The timing constraint is most strict for the first bit, for which it becomes the condition in equation 3.

$$N\tau_b < 2^{N-1} \frac{T_{VCO,\min}}{2} \text{ for all } N = 1, \dots, N_b \quad (4)$$

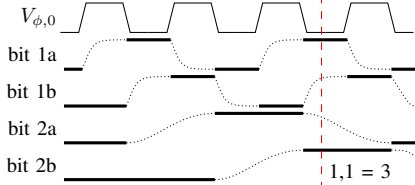


Figure 7: Waveforms in the connected binary counter

The correct bit at each level can be selected by implementing the multiplexing operation in series: as figure 7 shows, the value of bit 2a is reliable if the value of bit 1 is 0 and bit 2b is reliable if the value of bit 1 is 1. The output of the first multiplexer therefore becomes the selection bit for the second multiplexer, and this continues until the  $N_{b,c}$  bits are selected.

The final connection is implemented as the olive-coloured, dash-dotted line in figure 6. Due to this connection, the data input of bit  $na$  only changes when the value of bit  $nb$  changes. The transitions on counter a and counter b happen half a period of the previous signal apart, therefore the data input will not change during the half period after the edge used as clock edge. The counter therefore does not require a flipflop but can be implemented using gated latches. These latches are significantly simpler to design and therefore reduce the power consumption in the coarse counter.

### III. CIRCUIT DESIGN

#### A. Ring Oscillator and Tuning Circuit

The different building blocks which make up the VCO-ADC in figure 1 have to be designed as a transistor-level circuit. In this design, some parameters have a fixed sizing, especially when it is expected that a minimal sizing is optimal for both speed and power consumption, taking advantage of the digital-friendly nature of the used technology, which is 28 nm CMOS. All transistors are implemented in ultra low threshold voltage (ulvt) flavor, and the supply voltage  $V_{dd}$  equals 900 mV.

Some parameters which are crucial to the performance are not yet sized in this section. The sizing of these parameters is done by the algorithm described in section IV.

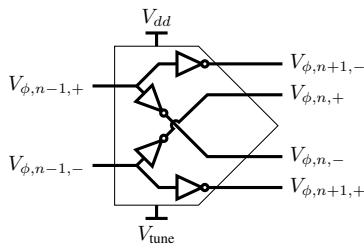


Figure 8: Feed-forward delay cell

The ring oscillator consists of the delay cells shown in figure 8. These delay cells, described in [6] and implemented in a VCO-ADC in [7] allow the ring oscillator to reach higher frequencies than in standard cross-coupled delay cells since the signal is fed forward through a single inverter path from stage  $n-1$  to stage  $n+1$ . The NMOS transistors are sized with minimal length 30 nm. Their width is denoted as  $W_n$  as this is crucial to the trade-off between input-referred thermal noise and current consumption in the ring through impedance scaling. The PMOS transistor width equals to  $2W_n$  and length is 30 nm. The number of delay cells  $N_\phi$  in the ring will also be optimized in the algorithm.

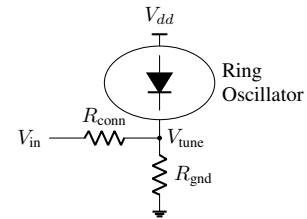


Figure 9: VCO tuning circuit

The ring will be tuned by applying a voltage  $V_{tune}$  to the bottom of the ring, which is set by the circuit in figure 9, described in [8]. It was shown in [9] that the ring oscillator characteristic is similar to a diode characteristic. The two resistors  $R_{conn}$  and  $R_{gnd}$  apply a suitable load line to this diode characteristic, given by equation 5. Due to the tuning voltage the bottom voltage of the square wave is not rail-to-rail anymore. This makes the operation of the coarse counter and sampling in the fine counters more difficult.

$$I_{ring} = V_{tune} \left( \frac{1}{R_{conn}} + \frac{1}{R_{gnd}} \right) - \frac{V_{in}}{R_{conn}} \quad (5)$$

#### B. Coarse Counter and Buffer

The buffer circuit of figure 10 is used to provide a larger amplitude of the square wave to the input of the coarse counter. All transistors in this buffer will be sized minimally with length 30 nm and width 100 nm to minimize power consumption at a high speed and therefore avoid that the condition in equation 4 limits the VCO frequency too strongly.

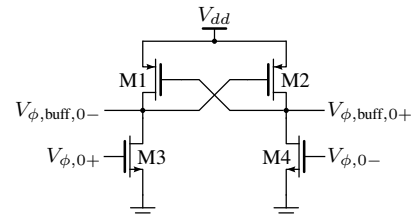


Figure 10: Coarse counter buffer

The output of the buffer circuit  $V_{\phi,buff,0+}$  is used as the positive input  $V_{\phi,0}$  of the connected counters in figure 6 and  $V_{\phi,buff,0-}$  as  $\overline{V_{\phi,0}}$ . The gated latches in this figure are implemented using four NAND-gates as in figure 11. Each of the NAND-gates itself is implemented as in figure 12. Once again, the transistors will be sized minimally for high speed at low power consumption.

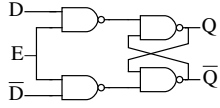


Figure 11: Gated NAND-latch

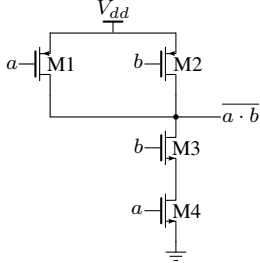


Figure 12: NAND gate

Thanks to the buffer circuit, the NAND-latch operates correctly up to a  $V_{\text{tune}}$  of 650 mV, allowing us to significantly reduce the voltage over the ring oscillator and therefore also its current consumption. The performance of the NAND-gates together with the buffer circuit is shown in figure 13. The rise and fall times of the NAND-latch are below 55 ps for a tuning voltage up to 500 mV. Since the highest VCO frequencies are obtained at low  $V_{\text{tune}}$ , a maximal VCO frequency  $f_{\text{VCO,max}}$  up to 8 GHz can be achieved under condition 4. As visible in figure 13d, the gated latches require only a limited transient charge to switch to a new output. The buffer consumes a significant leakage current as seen in figure 13c, but the higher achievable  $V_{\text{tune}}$  allows a larger reduction in the VCO current.

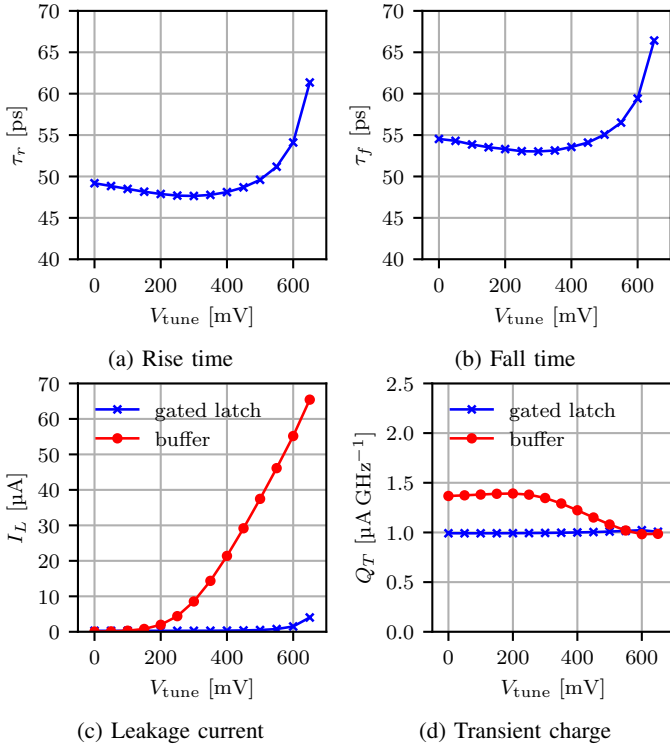


Figure 13: Performance of the buffered NAND-latch

### C. Sense Amplifier

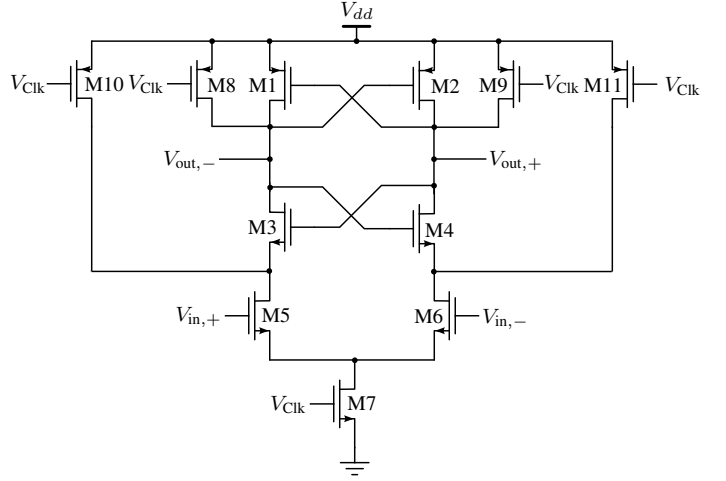


Figure 14: StrongARM sense amplifier

To sample the output of the ring oscillator very quickly and correctly at high  $V_{\text{tune}}$ , a StrongARM sense amplifier [10] [11] is used. This circuit is shown in figure 14. The StrongARM outputs rail-to-rail values of  $V_{\text{out,+}}$  and  $V_{\text{out,-}}$ , forcing a low value at the side where the current drawn by M5 or M6 is strongest. All transistors are sized minimally, except for M7 which is sized with a width of 200 nm and a length of 30 nm. The regeneration time constant of cross-coupled transistors M1-M4 determines the time it takes to reach the rail-to-rail output required by the digital circuit. After a time of 262 ps, a loose upper bound for the probability that the output is metastable is in the order of  $10^{-15}$ , low enough to not affect the SNR of the VCO-ADC. This value will therefore be applied as the input delay when synthesizing the digital circuit.

### D. Digital Circuit

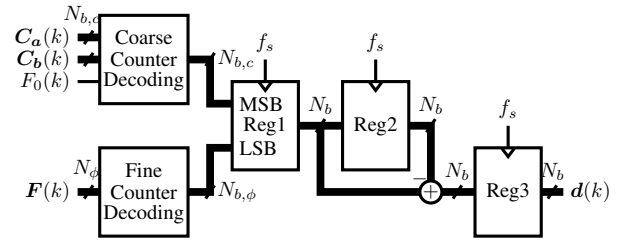


Figure 15: Digital circuit

The digital chain which processes the sampled outputs is shown in figure 15 and consists of a decoder for the coarse and fine counters and three registers and a subtraction which are used to perform the difference operation shown in figure 2. The coarse counter decoder is implemented as a multiplexing operation over the different coarse bits. The value of the fine counter is determined using a lookup table: it is the position and direction of the rising or falling edge which determines the value of the fine counter, and therefore only  $2^{\lceil \log_2 N_\phi \rceil + 1}$  different codes

are expected at the input of this decoder. The other codes can lead to any value out the output, implemented by a `default` case with a ‘don’t care’ ( $\times$ ) output. The synthesized digital circuit for  $N_\phi = 16$  and  $N_{b,c} = 3$  has a longest-path delay of 484 ns. The circuit can therefore be synthesized without the need to pipeline any operations up to a sampling frequency of approximately 1.5 GHz. A design without pipelining saves a lot of power as extra registers and other calculation blocks are not needed. This is reflected in the limited power consumption of the digital circuit of only  $168 \mu\text{W}$  at  $f_s = 1 \text{ GHz}$ .

#### IV. OPTIMIZING THE DESIGN

The circuits are sized to achieve a certain SNR and third harmonic distortion (HD3) for a given bandwidth at minimal power consumption. Using empirical data obtained from simulating the different building blocks in the VCO-ADC circuit and taking into account the effect of different parameters on the SNR and power consumption, an optimal design can be obtained. The remaining sizing parameters are the resistors  $R_{\text{gnd}}$  and  $R_{\text{conn}}$  of the tuning circuit, the width of the NMOS transistors  $W_n$  of the delay cells, the number of delay cells  $N_\phi$  and the sampling frequency  $f_s$ .

These variables provide us with a large design space, which can be reduced by considering several bounds on the variables.  $N_\phi$  is limited to integer powers of two as this will allow us to decode the fine and coarse counters separately and therefore more efficiently. Together with the maximal values of  $f_s$  and  $f_{\text{VCO,max}}$ , this allows us to define a set of combinations for  $N_\phi$ ,  $f_{\text{VCO,max}}$  and  $f_{\text{VCO,min}}$  for which the SQNR is larger than the desired SNR. For the maximal and minimal VCO frequency, the points on curve of  $I_{\text{ring}}$  in function of  $V_{\text{tune}}$  for the original width  $W_{n,0}$  allow us to determine  $R_{\text{conn},0}$  and  $R_{\text{gnd},0}$  using equation 5. These values will be resized further together with  $W_n$  using impedance scaling.

Based on  $R_{\text{conn},0}$  and  $R_{\text{gnd},0}$ , a curve of  $V_{\text{in}}$  against  $f_{\text{VCO}}$  is obtained. This curve is fitted with a third order polynomial approximation of the input-output relation of the VCO with its tuning circuit, allowing us to determine whether the requirement on HD3 is met. This requirement is invariant under impedance scaling.

If the desired HD3 is achieved, the input-referred thermal noise can be calculated based on the analysis in [9] and [12] for the width  $W_{n,0}$ . The ratio of the power in the input signal to the power of the maximal noise voltage will be written as  $\text{SNR}_{\text{in},T,0}$ . The actual signal-to-noise ratio after impedance scaling can be calculated using its proportionality to  $W_n$ . Similarly, the maximal SQNR at the maximal sampling frequency  $f_{s,\text{max}}$  is determined and this value is rescaled by a factor  $f_s/f_{s,\text{max}}$ . To obtain the desired SNR, the values of  $f_s$  and  $W_n$  are therefore related through equation 6.

$$\frac{f_{s,\text{max}}}{\text{SQNR}_{\text{max}} f_s} = \frac{1}{\text{SNR}} - \frac{W_{n,0}}{\text{SNR}_{\text{in},T,0} W_n} \quad (6)$$

Using equation 6, all variables are now expressed in function of  $W_n$  for the considered iteration point. A lower bound for  $W_n$  can be obtained by solving this equation for  $f_s = f_{s,\text{max}}$ , and another lower bound will be set as 800 nm, as the width of the ring oscillator transistors also affects the performance under mismatch, described in [13] and the VCO frequency will be more strongly

affected by the capacitive load of the sense amplifier when the width is smaller. The maximum of these bounds will be used.

An upper bound for  $W_n$  is also necessary. To make sure that the difference operation and integration cancel each other for the signal term in 1,  $f_s$  has to be larger than approximately  $10f_{BW}$ . Depending on the relative values in equation 6, this can place an upper bound on  $W_n$ . If the SQNR at the minimal sampling frequency is lower than the required SNR, the value of  $f_s$  moves asymptotically to a minimal value. The number of bits in the coarse counter, given by  $N_{b,c,\text{max}} = \lceil \log_2(f_{\text{VCO,max}}/f_{s,\infty}) \rceil$ , can be determined at the minimal  $f_s$ . This allows us to express the power consumption as a rational function of  $W_n$  and therefore determine an upper bound for the point where it reaches its minimal value.

Based on these two lower bounds and one or two upper bounds, the points where  $W_n$  is a multiple of 100 nm in the desired range are selected. The estimated power consumption for each of the circuits can be calculated in function of the different parameters and the simulated data. Determining this estimation for the selected points of  $W_n$  and then storing the points and parameters achieving minimal power consumption allows us to optimize the consumed power over the different iteration points.

#### V. RESULTS

The algorithm is applied to a coarse-fine VCO-ADC design with a maximal bandwidth of 40 MHz, SNR of 76 dB, and HD3 of  $-40 \text{ dB}$ . These specifications and the resulting design parameters are shown in table I. The predicted VCO characteristics and power estimations, are shown on the first data row in table II.

$f_{BW}$	$f_{BW,\text{min}}$	$\text{SNR}_{\text{target}}$	$\text{HD3}_{\text{target}}$
40 MHz	100 kHz	76 dB	$-40 \text{ dB}$

$f_s$	$W_n$	$R_{\text{conn}}$	$R_{\text{gnd}}$	$N_\phi$	$N_{b,c}$
1.387 GHz	800 nm	495 $\Omega$	765 $\Omega$	32	2

Table I: Specifications and design parameters

	$f_{\text{VCO,max}}$	$f_{\text{VCO,min}}$	SQNR	$\text{SNR}_{\text{in},T}$
Alg	4.360 GHz	0.536 GHz	78.69 dB	79.35 dB
Sim	4.314 GHz	0.543 GHz	78.34 dB	81.17 dB

	$P_{\text{VCO}}$	$P_{\text{CC}}$	$P_{\text{SA}}$	$P_{\text{dig}}$	$P_{\text{tot}}$
Alg	544 $\mu\text{W}$	45 $\mu\text{W}$	85 $\mu\text{W}$	204 $\mu\text{W}$	888 $\mu\text{W}$
Sim	519 $\mu\text{W}$	38 $\mu\text{W}$	99 $\mu\text{W}$	232 $\mu\text{W}$	888 $\mu\text{W}$

Table II: Predicted and simulated performance

The VCO-ADC is simulated in Cadence, with all the parameters as described above or in the table I. Figure 16 shows the obtained spectrum for a single-ended implementation of the VCO-ADC. The slope of 20 dB/decade is clearly visible. At low frequencies, there is a noise floor due to white noise. The SNR is slightly higher than predicted. Based on the noise floor and on a transient simulation without noise, the SQNR was found to equal 78.34 dB and  $\text{SNR}_{\text{in},T}$  equals 81.17 dB. Peaks in the spectrum at the harmonics of the input frequency are clearly visible. The HD3 is lower than predicted and therefore does not exactly meet the specification. Some relevant performance values are shown in the second data row of table II. Although

the predicted power consumption of the different blocks do not always match perfectly,  $P_{\text{tot}}$  is equal to  $888\mu\text{W}$  in both the estimated and simulated results. This limited power consumption shows that the design is indeed very power-efficient.

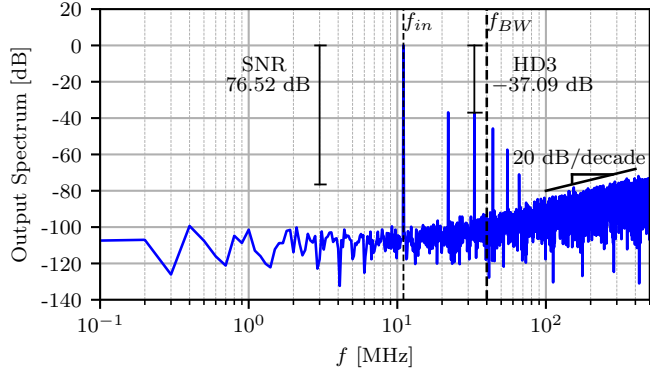


Figure 16: Single-ended output spectrum

Due to the harmonic distortion which affects the output spectrum of the digital output, the SNDR of the VCO-ADC is only 33.95 dB. To reduce the harmonic distortion, a pseudo-differential operation using two VCO-ADCs and digital calibration are used. Figure 17 shows how this is implemented.

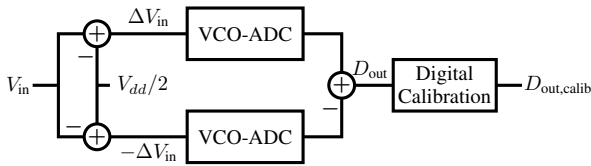


Figure 17: Differential operation and digital calibration

Figure 18 shows the output spectrum of the FFT after calibration and in the pseudo-differential configuration. The pseudo-differential operation removes the even harmonics. It also increases the SNR as the signal amplitude doubles while the noise amplitude increases by approximately 3 dB. The uneven harmonics decrease significantly by calibration. An SNDR of 74.86 dB is obtained. The power consumption is doubled and an extra difference operation and register are added. The total power consumption is therefore  $1.862\text{ mW}$  leading to a Schreier figure-of-merit ( $\text{FOM}_S$ ) of 178.18 dB.

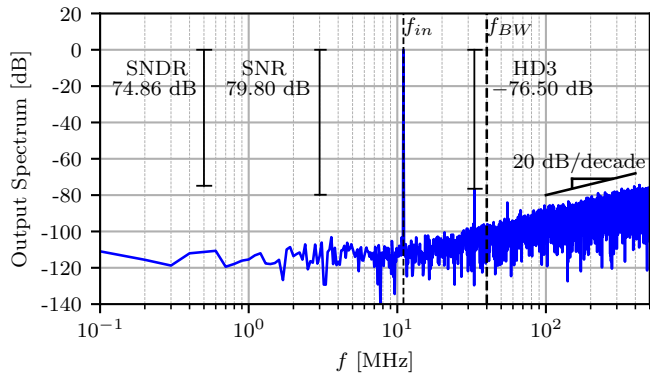


Figure 18: Output spectrum after calibration

## VI. CONCLUSION

This work presented two developments: a configuration for a double coarse counter which allows us to implement this a power-efficient coarse counter up to high VCO frequencies, and a method to combine different circuits based on a set of given specifications into a power-efficient coarse-fine VCO-ADC. These developments, together with the different other circuits presented in earlier papers [7] [8] [11], made it possible to implement a coarse-fine VCO-ADC with an SNR of 76.52 dB in a single-ended configuration, or an SNDR of 74.86 dB when it is implemented differentially and the output is digitally calibrated.

Improvements to the algorithm can be made by examining the HD3 more carefully to obtain a better estimation and including a condition for the effect of mismatch on the VCO performance. As the SQNR performance of the coarse-fine VCO-ADC is limited by the maximal tuning and sampling frequency, faster circuits would be necessary to obtain a substantial improvement. Another option would be to investigate how the coarse-fine VCO-ADC can be combined with higher order noise shaping to obtain a lower SQNR.

## REFERENCES

- [1] J. Borgmans, E. Sacco, P. Rombouts, and G. Gielen, "Methodology for Readout and Ring Oscillator Optimization Toward Energy-Efficient VCO-Based ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 3, pp. 985–998, 2022.
- [2] J. Borgmans and P. Rombouts, "Toward 'digital' analogue-to-digital converters," *Electronics Letters*, vol. 55, no. 10, pp. 568–569, 2019.
- [3] C. Perez, R. Garvi, G. Lopez, A. Quintero, F. Leger, P. Amaral, A. Wiesbauer, and L. Hernandez, "A VCO-based ADC with direct connection to a microphone MEMS, 80dB peak SNDR and  $438\mu\text{W}$  power consumption," *IEEE Sensors Journal*, pp. 1–1, 2023.
- [4] T.-F. Wu and M. S.-W. Chen, "A 40MHz-BW 76.2dB/78.0dB SNDR/DR Noise-Shaping Nonuniform Sampling ADC with Single Phase-Domain Level Crossing and Embedded Nonuniform Digital Signal Processor in 28nm CMOS," in *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 262–264, 2020.
- [5] J. Kim, T.-K. Jang, Y.-G. Yoon, and S. Cho, "Analysis and Design of Voltage-Controlled Oscillator Based Analog-to-Digital Converter," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 18–30, 2010.
- [6] I. Kovacs and M. Neag, "New dual-loop topology for ring VCOs based on latched delay cells," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2018.
- [7] J. Borgmans and P. Rombouts, "Enhanced circuit for linear ring VCO-ADCs," *Electronics Letters*, vol. 55, no. 10, pp. 583–585, 2019.
- [8] Babaie Fishani, Amir and Rombouts, Pieter, "Highly linear VCO for use in VCO-ADCs," *Electronics Letters*, vol. 52, no. 4, pp. 268–269, 2016.
- [9] J. Borgmans, R. Riem, and P. Rombouts, "The Analog Behavior of Pseudo Digital Ring Oscillators Used in VCO ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2827–2840, 2021.
- [10] B. Razavi, "The StrongARM Latch [A Circuit for All Seasons]," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12–17, 2015.
- [11] H. Xu and A. A. Abidi, "Analysis and Design of Regenerative Comparators for Low Offset and Noise," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 8, pp. 2817–2830, 2019.
- [12] J. Borgmans and P. Rombouts, "Noise Optimization of a Resistively-Driven Ring Oscillator for VCO-Based ADCs," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 775–779, 2022.
- [13] J. Borgmans and P. Rombouts, "The Mismatch Performance of Pseudo Digital Ring Oscillators Used in VCO ADCs: PSRR and CMRR," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 2, pp. 579–592, 2023.

# Contents

<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>v</b>
<b>List of Abbreviations</b>	<b>vi</b>
<b>1 Modelling the VCO-ADC</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Quantization and Sampling . . . . .	2
1.3 VCO-ADC block diagrams . . . . .	3
1.4 Ring Oscillator and VCO phases . . . . .	5
1.5 Signal-to-Quantization-Noise Ratio . . . . .	6
1.6 The VCO as a Pulse Frequency Modulator . . . . .	8
1.7 Coarse-Fine VCO-ADC . . . . .	9
1.8 Goal and Organization of this Thesis . . . . .	10
<b>2 System-Level Considerations</b>	<b>12</b>
2.1 System-Level Model . . . . .	12
2.2 SQNR of the Coarse-Fine VCO-ADC Model . . . . .	14
2.3 Asynchrony of the Coarse Counter . . . . .	17
2.4 Double Coarse Counter . . . . .	21
2.5 Metastability and Mismatch in the Double Counter . . . . .	24
<b>3 Design of the Coarse Counter</b>	<b>25</b>
3.1 Synchronous Counters . . . . .	25
3.2 Asynchronous Binary Counters . . . . .	26
3.3 Double Connected Binary Counter . . . . .	28
3.4 Gated Latch Design . . . . .	31
3.4.1 Gated NAND-latch . . . . .	31
3.4.2 Gated NOT-latch . . . . .	33
3.4.3 Comparison . . . . .	35

---

<b>4</b>	<b>Design of the Other Circuit Elements</b>	<b>38</b>
4.1	Overview . . . . .	38
4.2	Delay cells . . . . .	39
4.3	VCO Tuning Circuit . . . . .	42
4.4	Coarse Counter Buffer . . . . .	44
4.5	StrongARM Sense Amplifier . . . . .	47
4.6	Digital Design . . . . .	51
4.7	Combining the Circuits . . . . .	53
4.7.1	Design Space and Iteration Variables . . . . .	53
4.7.2	VCO Frequency Characteristic and Distortion . . . . .	54
4.7.3	Impedance Scaling and SNR . . . . .	55
4.7.4	Power calculation . . . . .	57
<b>5</b>	<b>Results</b>	<b>60</b>
5.1	Single-Ended VCO-ADC performance . . . . .	60
5.2	Pseudo-Differential Operation and Calibration . . . . .	63
5.3	Effect of VCO Layout . . . . .	65
<b>6</b>	<b>Conclusion</b>	<b>68</b>
6.1	Conclusion . . . . .	68
6.2	Future Work . . . . .	69
<b>A</b>	<b>Simulink Model</b>	<b>70</b>
<b>B</b>	<b>Digital Verilog Code</b>	<b>74</b>
B.1	Pseudo-Differential Implementation . . . . .	74
B.2	Decoding and Difference Block . . . . .	75
B.3	Fine Counter Decoder . . . . .	76
B.4	Coarse Counter Decoder . . . . .	78
B.5	Register and Difference . . . . .	79
<b>C</b>	<b>Design Algorithm in Python</b>	<b>81</b>
	<b>Bibliography</b>	<b>86</b>

## List of Figures

1.1	Quantization and sampling of an analog input signal . . . . .	2
1.2	Block diagram of the VCO-ADC . . . . .	4
1.3	Block diagram of the VCO-ADC: phase model . . . . .	4
1.4	Block diagram of the VCO-ADC: additive quantization noise . . . . .	4
1.5	Ring Oscillator . . . . .	5
1.6	Block diagram of the VCO-ADC: amplification by the ring oscillator . . . . .	5
1.7	Relative magnitude of the frequency response . . . . .	6
1.8	Noise shaping and filtering of quantization noise . . . . .	7
1.9	Block diagram of the VCO-ADC: PFM model . . . . .	8
1.10	Conceptual representation of the coarse-fine VCO-ADC . . . . .	9
2.1	Schematic of the decoder of the coarse-fine counter values . . . . .	13
2.2	Waveform of the output of the ideal ADC . . . . .	14
2.3	Output spectrum of the ideal VCO-ADC . . . . .	15
2.4	SQNR of the ADC output at different input frequencies. . . . .	16
2.5	Readout of ring oscillator phase $V_{\phi,0}$ . . . . .	17
2.6	Effects of asynchrony on signals in the readout circuit . . . . .	18
2.7	Waveform of the output of the ADC affected by asynchrony . . . . .	19
2.8	Effect of asynchrony on the SNR . . . . .	20
2.9	Readout of ring oscillator phase $V_{\phi,0}$ using the double coarse counter. . . . .	22
2.10	Effect of asynchrony in double counter . . . . .	22
2.11	Effect of asynchrony on the SNR for the double counter . . . . .	23
2.12	Errors due to asynchrony in the double counter readout circuit . . . . .	23
2.13	Margins around transition to withstand mismatch . . . . .	24
3.1	Synchronous counter . . . . .	25
3.2	Asynchronous binary counter . . . . .	26
3.3	Waveforms in the asynchronous binary counter . . . . .	27
3.4	Connected asynchronous binary counters . . . . .	28
3.5	Double connected binary counter . . . . .	29
3.6	Waveforms in the double connected binary counter . . . . .	29

3.7	Double connected binary counters: gated latch implementation . . . . .	30
3.8	Gated NAND-latch . . . . .	31
3.9	NAND gate transistor-level design . . . . .	32
3.10	Transient response for the gated NAND-latch . . . . .	33
3.11	Gated NOT-latch transistor level design . . . . .	33
3.12	Transient response for the gated NOT-latch . . . . .	34
3.13	Relevant performance metrics for the gated latches . . . . .	36
4.1	Different blocks which remain to be designed . . . . .	38
4.2	Design of the feed-forward delay cell . . . . .	39
4.3	Modulation of the VCO output by the tuning voltage . . . . .	40
4.4	Input-output characteristics of the ring oscillator VCO. . . . .	41
4.5	Input-referred thermal voltage noise of the ring oscillator . . . . .	42
4.6	Design of the VCO tuning circuit . . . . .	43
4.7	Plot of the operation of the VCO tuning circuit . . . . .	43
4.8	Coarse counter buffer circuit . . . . .	45
4.9	Relevant performance metrics for the buffered NAND-latch . . . . .	46
4.10	StrongARM sense amplifier . . . . .	47
4.11	Transition waveforms of a StrongARM sense amplifier . . . . .	49
4.12	Digital blocks to determine the ADC output . . . . .	51
5.1	Output spectrum of the single-ended VCO-ADC without thermal noise . . . . .	61
5.2	Output spectrum of the single-ended VCO-ADC with thermal noise . . . . .	62
5.3	Pseudo-differential operation and digital calibration of the VCO-ADC . . . . .	63
5.4	Output spectrum of the pseudo-differentially operated VCO-ADC . . . . .	64
5.5	Output spectrum of the VCO-ADC after calibration . . . . .	65
5.6	Layout of the delay cell . . . . .	66
5.7	Output spectrum of the single-ended VCO-ADC, noiseless, layout of delay cell . . . . .	67
A.1	Top level/testbench of the Simulink model . . . . .	70
A.2	VCO-ADC in Simulink model . . . . .	71
A.3	VCO part 1 in Simulink model . . . . .	72
A.4	VCO part 2 in Simulink model . . . . .	72
A.5	VCO phase 0 readout in Simulink model . . . . .	72
A.6	Other VCO phases readout in Simulink model . . . . .	73



## List of Tables

3.1	Transistor sizing of the NAND gate . . . . .	32
3.2	Transistor sizing of NOT-gate . . . . .	34
4.1	Transistor sizing of the inverters in the feedforward delay cell . . . . .	39
4.2	Transistor sizing of the buffer circuit . . . . .	45
4.3	Updated transistor sizing of the NAND gate . . . . .	45
4.4	Transistor sizing of the strongARM sense amplifier . . . . .	48
4.5	Transistor sizing of the NAND gate in the latch following the sense amplifier . . . . .	48
4.6	Specifications and resulting optimal design parameters . . . . .	58
4.7	Specifications and resulting estimated variables . . . . .	58
5.1	Relevant results of running the algorithm with the given specifications. . . . .	60
5.2	Relevant simulation results with given specifications and sizing . . . . .	62
5.3	Comparison with the state-of-the-art VCO-ADCs and a recent delta-sigma modulator . . . . .	65
5.4	Ring oscillator characteristics for delay cells after layout . . . . .	66

## List of Abbreviations

ADC	analog-to-digital converter
FFT	fast Fourier transform
FOM <sub>S</sub>	Schreier figure-of-merit
iid	independent and identically distributed
HD3	third harmonic distortion
NTF	noise transfer function
OSR	oversampling ratio
PFM	pulse-frequency modulation
SNDR	signal-to-noise-and-distortion ratio
SNR	signal-to-noise ratio
SQNR	signal-to-quantization-noise ratio
THD	total harmonic distortion
VCO	voltage-controlled oscillator
VCO-ADC	voltage-controlled-oscillator-based analog-to-digital converter

# Modelling the VCO-ADC

## 1.1 Introduction

We are constantly surrounded by a multitude of signals. These signals originate from physical processes, such as the temperature in the room or the strength of a magnetic field, or they can be generated by humans to communicate information. Sensors and receivers intercept and measure these signals, producing a continuously changing and infinitely precise analog value. An analog signal contains information about a physical quantity. To process and store this information the signal should be represented as samples consisting of a finite number of bits. Therefore, the analog signal needs to be converted to a digital signal by an analog-to-digital converter (ADC).

To extract faster signals of a higher bandwidth or more precisely determine the signal value by using a higher number of bits, the sampling rate of the ADC can be increased. Thanks to technology scaling, these faster sampling rates can be achieved in digital circuits. However, these advanced technologies only allow for a small voltage headroom which causes difficulties when designing traditional analog components such as operational amplifiers [1]. A voltage-controlled oscillator (VCO) takes advantage of this evolution: the increased switching speed allows us to generate a signal in a wide frequency range. The VCO converts the analog input voltage to a sine wave with a frequency proportional to the voltage. This frequency now represents the input signal, allowing us to more precisely determine its value.

This VCO-based ADC (VCO-ADC) can be implemented using a digital-friendly ring oscillator, allowing us to sample the different phases of this ring oscillator and reach a high VCO frequency. Several research examples [3] [4] present a VCO-ADC with single bit, or ‘fine’ quantization on each phase of the ring oscillator. This requires the sampling frequency to be equal to or double the VCO frequency, leading to high digital power consumption. In this thesis, a VCO-ADC which includes a multi-bit, or ‘coarse’ counter on one of the phases combined with fine counters on the other phases will be discussed. This leads to a power-efficient VCO-ADC design as the digital power consumption is greatly reduced by decreasing the sampling frequency.

## 1.2 Quantization and Sampling

A well-designed ADC generates a digital output which accurately and precisely represents the analog input signal  $v_{\text{in}}$  provided to it. The analog input signal measured by a sensor or receiver can take any real value in a certain range and can change at any time. To process this signal in a digital system, the signal must be represented by a limited number of bits at certain time instants. The range of real values must be divided such that each part of this range is represented by an integer output value. This is called quantization and is sketched in figure 1.1a. Samples of the analog signal are taken at certain time instants, separated by the sampling period  $T_s$ . Sampling is illustrated in figure 1.1b.

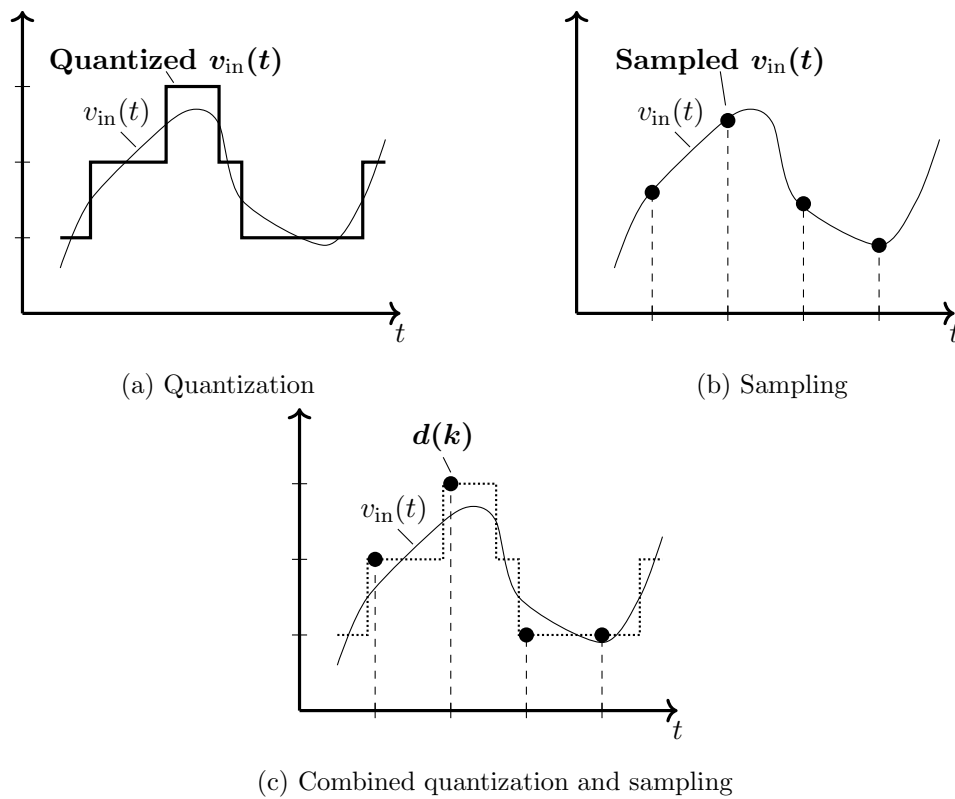


Figure 1.1: Quantization and sampling of an analog input signal

The digital output signal  $d(k)$  is obtained by combining quantization and sampling, visible in figure 1.1c. Ideally, the digital output values are linearly related to the analog input value at the sampling time instant  $kT_s$ . However, in figure 1.1c it can be seen that quantization adds a noise term  $q(kT_s)$  to the digital signal. The digital output signal is shown in equation 1.1, where  $K_{\text{ADC}}$  and  $C$  describe the ideal linear behaviour of the ADC. The equation shows that sampling and quantization can be interchanged without affecting the output, which is also visible in figure 1.1c.

$$d(k) = (K_{\text{ADC}}v_{\text{in}}(kT_s) + C) + q(kT_s) \quad (1.1)$$

The digital signal will therefore always be affected by quantization noise. Sampling does not necessarily cause an error: the Nyquist-Shannon sampling theorem [5] states that a signal can be reconstructed from samples taken at a sampling frequency  $f_s$  when its bandwidth  $f_{BW}$  is smaller than half of this sampling frequency. This condition is shown in equation 1.2.

$$f_{BW} < \frac{f_s}{2} \quad (1.2)$$

The frequency  $f_s/2$  is known as the Nyquist frequency. All frequency components at absolute frequencies higher than the Nyquist frequency are folded into the band  $[-f_s/2, f_s/2)$  by sampling. This process is called aliasing and will change the spectrum in this band. The spectrum of the input signal will not be aliased if the sampling is sufficiently high compared to the bandwidth.

Besides quantization, the digital output is also affected by other imperfections. Transistors and resistors in the ADC-circuit will add thermal and  $1/f$ -noise to the signal, represented by the input-referred noise  $n_{in}$ . Throughout the signal chain, nonlinearities introduce harmonics, which cause a distortion  $e_D(v_{in}(t))$ . Equation 1.3 includes these imperfections.

$$d(k) = (K_{ADC}(v_{in}(kT_s) + n_{in}(kT_s)) + C) + q(kT_s) + e_D(v_{in}(kT_s)) \quad (1.3)$$

### 1.3 VCO-ADC block diagrams

Equation 1.1 represents an ideal ADC, only affected by quantization noise. The transfer function of an ideal VCO-ADC should be similar to this equation. To determine this transfer function, the VCO-ADC is modelled in a block diagram. The input-output relation can then be expressed as a Z-domain transfer function. The Z-domain representation of equation 1.1 is given in equation 1.4. The function  $D(z)$  represents the digital output and  $Q(z)$  the quantization noise. The function  $v_{in}$  is continuous and will be represented in the Laplace domain as  $V_{in}(s)$ , on which sampling with aliasing is applied to bring this to the Z-domain. Sampling with aliasing is represented by the star operator  $[\cdot]^*$ , defined in [6]. This is a linear operator, and the transform  $z = e^{sT_s}$  can be used to bring a function of  $z$  inside the star operator.

$$D(z) = [K_{ADC}V_{in}(s) + C]^* + Q(z) \quad (1.4)$$

A suitable block diagram representation for the VCO-ADC is found by identifying the different transformations which affect the signal. In an ideal VCO-ADC, the VCO produces a square wave output signal  $V_\phi$  of which the frequency is proportional to the input voltage [7]. Since this signal is a square wave, it is already quantized. Sampling the signal, with sampling frequency  $f_s$ , creates a digital signal. A difference operation is applied, subtracting the previous sample from this sample to obtain the desired signal  $D(z)$ . This description, in which all of the signal values are determined by their voltage, is shown in the block diagram of figure 1.2.

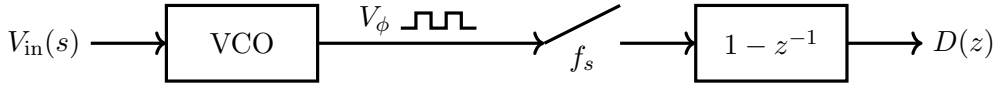


Figure 1.2: Block diagram of the VCO-ADC

A different view of the VCO should be taken to understand how the output signal  $D(z)$  is related to the input voltage and why the difference operation is necessary. The VCO produces a signal with an instantaneous frequency  $f_{\text{VCO}}$ , given by equation 1.5. In this equation,  $K_{\text{VCO}}$  is the VCO gain and  $f_0$  is the free-running frequency. The instantaneous phase  $\phi$  of the VCO output is found by integrating the angular frequency  $2\pi f_{\text{VCO}}$ , shown in the Laplace domain in equation 1.6. This phase is quantized due to the square wave  $V_\phi$ . The quantized phase can be viewed as the output of the VCO and is sampled at frequency  $f_s$ . Figure 1.3 shows a block diagram for this phase model of the VCO-ADC.

$$f_{\text{VCO}}(s) = K_{\text{VCO}}V_{\text{in}}(s) + f_0 \quad (1.5)$$

$$\phi(s) = \frac{2\pi K_{\text{VCO}}V_{\text{in}}(s) + 2\pi f_0}{s} \quad (1.6)$$

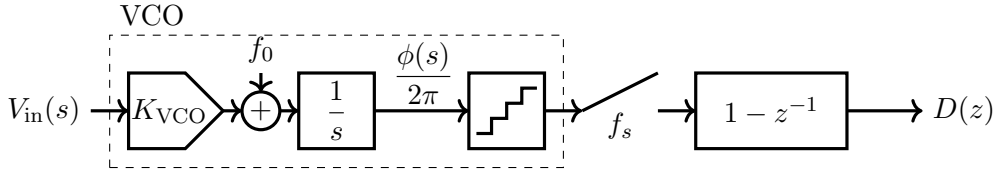


Figure 1.3: Block diagram of the VCO-ADC: phase model

In the phase model, the effect of quantization is still unresolved, making it difficult to express a transfer function. To estimate the effect of quantization, the position of sampling and quantization are switched as allowed according to section 1.2. It is assumed that the samples of quantization noise are unrelated to the phase  $\phi(s)$ , uniformly distributed over the range  $[0, 1)$ , and independent and identically distributed (iid). The effect of quantization is therefore modelled as additive white noise. This quantization noise  $Q(z)$  is added after sampling in figure 1.4.

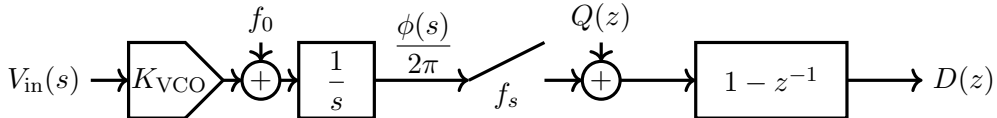


Figure 1.4: Block diagram of the VCO-ADC: additive quantization noise

In the resulting block diagram in figure 1.4, the phase in equation 1.6 is sampled directly. The quantization noise is added to the sampled signal and the difference with the previous sample is used to obtain the digital ADC output, expressed in equation 1.7.

$$D(z) = \left[ \frac{K_{\text{VCO}}V_{\text{in}}(s) + f_0}{s} \right]^* (1 - z^{-1}) + Q(z) (1 - z^{-1}) \quad (1.7)$$

## 1.4 Ring Oscillator and VCO phases

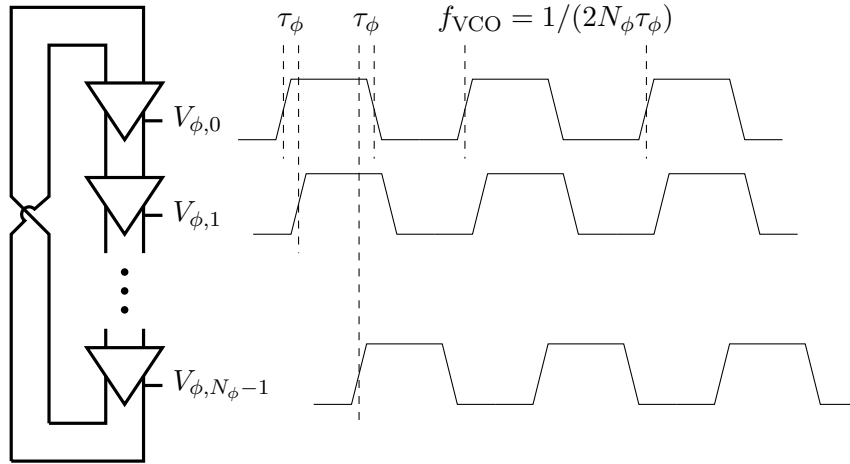


Figure 1.5: Ring Oscillator

The VCO output in the previous section was assumed to be a square wave with frequency proportional to the input voltage. This VCO can be implemented as a ring oscillator [7] [8]. A ring oscillator consists of a ring of delay cells. Each cell introduces a delay  $\tau_\phi$ . Therefore, a ring with  $N_\phi$  cells produces a wave with frequency  $1/(2N_\phi\tau_\phi)$ . Figure 1.5 shows the operation of this ring oscillator. The outputs of each delay cell are called the phases of the VCO, denoted as  $V_{\phi,n}$ . To determine the output of the VCO, both edges of all  $N_\phi$  phases are considered. The phase is quantized up to  $\pi/(2N_\phi)$ . Alternatively, this can be viewed as an amplification of the phase  $\phi$  in the block diagram by  $2N_\phi$ , setting the quantization step to 1. This becomes clear when it is considered that all the VCO phases are added together after sampling, which can also be done before sampling in the block diagram due to linearity of the star operator. Hence, the precision of the VCO is improved using the ring oscillator.

Figure 1.6 is an adapted version of figure 1.4, where the factor  $2N_\phi$  is included after the integrator block. This allows us to express  $D(z)$  in equation 1.8, which shows that the amplitude of the digital signal is multiplied by a factor equal to the number of phases.

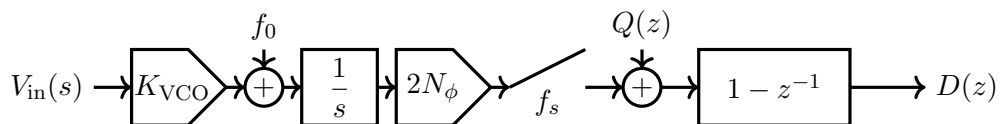


Figure 1.6: Block diagram of the VCO-ADC: amplification by the ring oscillator

$$D(z) = \left[ \frac{2N_\phi K_{VCO} V_{in}(s) + 2N_\phi f_0}{s} \right]^* (1 - z^{-1}) + Q(z) (1 - z^{-1}) \quad (1.8)$$

## 1.5 Signal-to-Quantization-Noise Ratio

In equation 1.8, two terms are clearly visible: a signal term relating  $D(z)$  to  $V_{\text{in}}(s)$ , and an error term with noise  $Q(z)$  and noise transfer function (NTF)  $(1 - z^{-1})$ . The signal term can be modified by placing the difference operation inside the sampling with aliasing block using the transformation  $z = e^{sT_s}$ . The resulting signal inside the star operator is a linear function of the input voltage multiplied by a phase shift and a sinc function [9]. This is shown in equation 1.9.

$$D_{\text{signal}}(z) = \left[ \frac{(2N_\phi K_{\text{VCO}} V_{\text{in}}(j2\pi f) + 2N_\phi f_0) e^{-j\pi f T_s} \text{sinc } f T_s}{f_s} \right]^* \quad (1.9)$$

Inside the sampling an aliasing block, the input signal is now affected by the frequency response  $H(f)$ , shown in equation 1.10. The behaviour of the frequency response has two important effects. Consider a VCO-ADC with a bandwidth for the analog input limited to a bandwidth  $f_{\text{BW}}$ . If  $f_{\text{BW}} \ll f_s$ , the sinc function in the frequency response can be approximated by 1, leading to the linear relation between the analog input and the digital output that is desired. This low frequency approximation of the frequency response is visible in figure 1.7. The signal term of the digital output in equation 1.11 is therefore a linear function of the input voltage affected by a delay due to the phase shift, which is then sampled and aliased. The maximal amplitude of the digital signal depends on the tuning range  $f_{\text{tune}}$  of the VCO as  $N_\phi f_{\text{tune}}/f_s$ .

$$H(f) = \frac{2N_\phi K_{\text{VCO}} e^{-j\pi f T_s} \text{sinc } f T_s}{f_s} \quad (1.10)$$

$$D_{\text{signal}}(z) = \left[ \frac{2N_\phi K_{\text{VCO}} V_{\text{in}}(j2\pi f) + 2N_\phi f_0 e^{-j\pi f T_s}}{f_s} \right]^* \text{ for } f_{\text{BW}} \ll f_s \quad (1.11)$$

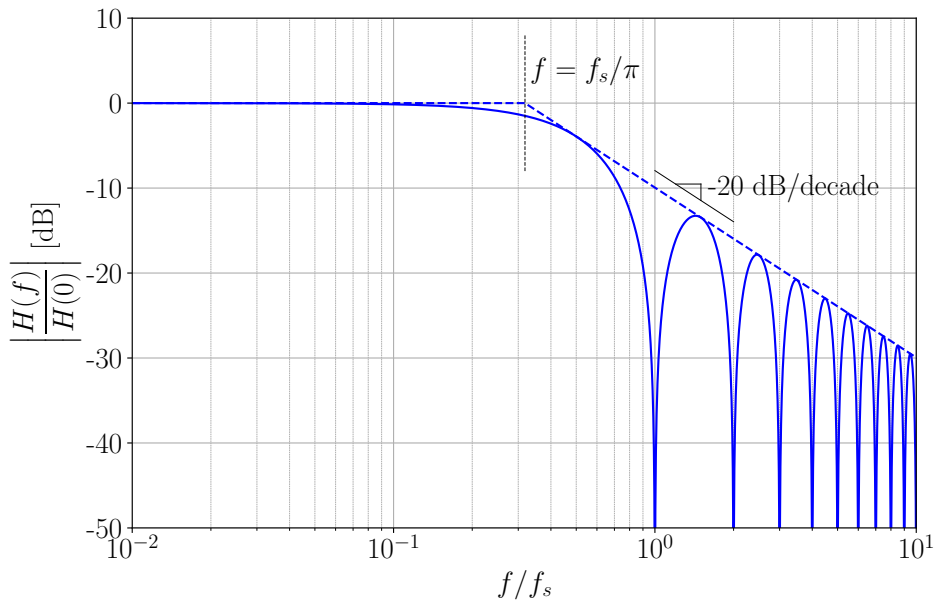


Figure 1.7: Relative magnitude of the frequency response



The magnitude of the transfer function is plotted in figure 1.7. The sinc function acts as a low-pass filter, with the dashed line showing an approximation of a first-order low-pass filter. This is strengthened by the zeroes at multiples of  $f_s$ . The sinc function acts as an inherent anti-aliasing filter for input-referred thermal noise of the VCO-ADC. The design of an explicit anti-aliasing filter is therefore unnecessary for this work.

The second term in equation 1.11 describe the effect of quantization noise on the digital output signal. Quantization noise was modelled as a sequence of iid samples  $q(n)$ , uniformly distributed over  $[0, 1)$ , leading to white noise  $Q(z)$  with variance  $\sigma_Q^2 = 1/12$ . This is multiplied by the NTF, leading to the spectral density in equation 1.12.

$$S_Q(f) = \frac{\sigma_Q^2}{f_s/2} |1 - e^{-2j\pi \frac{f}{f_s}}|^2 = \frac{\sigma_Q^2}{f_s} 8 \sin^2 \pi \frac{f}{f_s} \quad (1.12)$$

The noise at the output of the ADC is not white due to the NTF. A large portion of the noise power is shifted to higher frequencies. This effect is called noise shaping. For low frequencies, the noise spectral density is approximated as a quadratic function of frequency in equation 1.13.

$$S_Q(f) = \sigma_Q^2 \frac{8\pi^2 f^2}{f_s^3} \text{ when } f \ll f_s \quad (1.13)$$

When the VCO is defined for an upper bandwidth of the input signal  $f_{BW}$ , all of the components in the output spectrum above this bandwidth are noise and can be removed. Removing this noise is done using digital filtering techniques. A perfect filter removes all of the power in the spectrum in the frequency range higher than  $f_{BW}$  and does not affect the spectrum below  $f_{BW}$ . The effect of noise shaping and filtering on the noise spectral density is shown schematically in figure 1.8. The noise power, represented as the area under the spectral density curve, is greatly reduced by filtering. The remaining quantization noise power is found by integrating the noise spectral density of equation 1.13 over the frequency range from 0 until  $f_{BW}$ , expressed in equation 1.14. The oversampling ratio (OSR) is defined as the ratio  $f_s/(2f_{BW})$ . A high OSR reduces the quantization noise significantly when noise shaping is applied, leading to the factor  $\text{OSR}^{-3}$ .

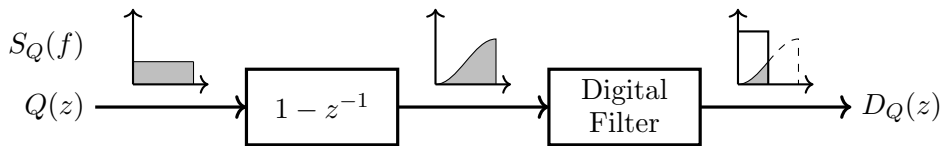


Figure 1.8: Noise shaping and filtering of quantization noise

$$P_Q = \int_0^{f_{BW}} \sigma_Q^2 \frac{8\pi^2 f^2}{f_s^3} df = \frac{1}{36} \pi^2 \left( \frac{2f_{BW}}{f_s} \right)^3 = \frac{1}{36} \pi^2 (\text{OSR})^{-3} \quad (1.14)$$

Combining the expression for the amplitude of the signal with the quantization noise power allows us to calculate the signal-to-quantization-noise ratio (SQNR) of the VCO-ADC in equation 1.15. This expression is described by Kim et al. in [9].

$$\text{SQNR}_{\text{dB}} = 20 \log_{10} \left( \frac{2N_{\phi} f_{\text{tune}}}{f_s} \right) + 30 \log_{10} \left( \frac{f_s}{2f_{BW}} \right) - 3.41 \quad (1.15)$$

A VCO-ADC is designed based on a target signal-to-noise ratio SNR and a given bandwidth  $f_{BW}$ . The number of phases, VCO frequency, and sampling frequency have to be designed so that the SQNR exceeds the target SNR, for a minimal power consumption. Equation 1.15 will therefore be essential to a successful VCO-ADC design.

The first term in equation 1.15 is related to the signal power, and the second term to the inverse of the noise power. The tuning frequency range  $f_{\text{tune}}$  and the number of phases  $N_{\phi}$  affect the SQNR in an identical way: both cause an increase in the signal of 20 dB per decade. This makes sense as we have identified the  $f_{\text{VCO}}$  is inversely proportional to  $N_{\phi}$  in section 1.4. The effect of these parameters on the thermal noise, low frequency noise, and power consumption will be different, which is considered in the design of the VCO-ADC in chapter 4.

The sampling frequency affects both the signal power and the quantization noise power. The quantization noise decreases by 30 dB per decade as the sampling frequency increases. This is partially offset by the decreases in signal power by 20 dB per decade increase in  $f_s$ . The SQNR therefore increases by 10 dB per decade.

## 1.6 The VCO as a Pulse Frequency Modulator

The model in figure 1.4 allows us to obtain an approximation for the SQNR in equation 1.15. Due to the assumption that the noise is independent of the input signal, this approximation is slightly inaccurate. A more exact model was developed by Gutierrez et al. [10]: the ring oscillator VCO is viewed as a pulse-frequency modulator (PFM). This PFM produces a Dirac delta pulse in the time domain every time the phase crosses a multiple of  $\pi/N_{\phi}$ . This PFM signal is then multiplied by the same frequency response  $H(f)$  as identified earlier, and sampled with aliasing. The full system is seen in figure 1.9.

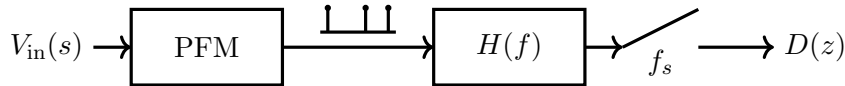


Figure 1.9: Block diagram of the VCO-ADC: PFM model

The PFM spectrum can be calculated analytically for a sinusoidal input signal. It consists of a pulse at the signal frequency, and at multiples of the signal frequency mixed with multiples of the VCO free-running frequency. This leads to a spectrum of Dirac delta pulses, described in [10]. The noise is given by the pulses due to the mixing products, multiplied by the frequency response and then aliased to the baseband. An analytical expression based on which the VCO-ADC can be designed is not available using the PFM interpretation, as opposed to expression 1.15 obtained when the quantization noise is assumed to be white. However, the PFM interpretation is useful when discussing the simulated SQNR for the system model in chapter 2.

## 1.7 Coarse-Fine VCO-ADC

Until this point, it was assumed that the sampling frequency  $f_s$  and the VCO frequency can be chosen independently. If a single bit is used to sample each phase of the ring oscillator, the output of the difference operation is ambiguous by a multiple of  $2N_\phi$ , as it is not known how many full cycles have been completed. To avoid this ambiguity, the sampling frequency must be higher than the VCO frequency, as expressed in equation 1.16.

$$f_s > f_{\text{VCO,max}} \quad (1.16)$$

This requirement leads to a minimal sampling frequency which can be used in the design of a VCO-ADC. Although equation 1.15 shows that a lower  $f_s$  decreases the SQNR, it could also significantly reduce the power consumption of the VCO-ADC. The power consumed by the samplers and the digital circuit is proportional to  $f_s$ . A lower  $f_s$  can also simplify the digital design as less registers are necessary to perform calculations over multiple periods, reducing the need to pipeline operations. Therefore, a power-efficient design for given requirements might benefit from the possibility to decrease the sampling frequency below the limit of equation 1.16.

To decrease the minimal sampling frequency, the VCO-ADC requires a counter which indicates the number of full cycles which have passed. It is sufficient to have this coarse counter at one of the ring oscillator phases, as the other phases are all in the same cycle. The single-bit counters on the other phases are called fine counters, leading to a coarse-fine VCO-ADC design. This is shown in figure 1.10.

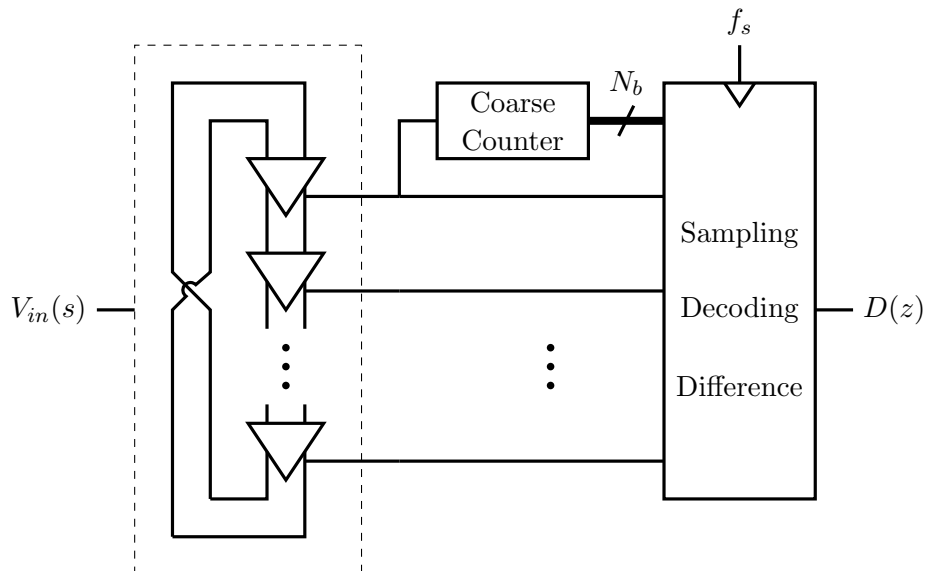


Figure 1.10: Conceptual representation of the coarse-fine VCO-ADC

The output of the coarse counter in figure 1.10 consists of  $N_{b,c}$  bits. This means that the counter can count up to a number of full cycles equal to  $N_c = 2^{N_{b,c}} - 1$ . The value  $N_c$  places a lower limit on  $f_s$ . If the number of full VCO cycles between two samples could be more than  $N_c$ , the resulting digital value would again be ambiguous. The time between samples should therefore now be less than  $N_c T_{VCO}$ . This limit is expressed in terms of the maximal VCO frequency in equation 1.17.

$$f_s > \frac{f_{VCO, \max}}{N_c} \quad (1.17)$$

Designing a coarse counter which can count up to a suitable value  $N_c$  therefore allows us to choose  $f_s$  significantly lower than when no coarse counter is used. The sampling frequency  $f_s$  also has a lower limit due to the shape of the frequency response  $H(f)$ . Equation 1.11 assumes that the sampling frequency is much greater than the bandwidth. Otherwise, the input signal will be affected by the filtering of  $H(f)$ . There is no theoretical upper limit on the sampling frequency, but practical limits may arise when designing the circuits in chapter 4. It should be noted that the coarse counter is not necessary when the sampling frequency is larger than the VCO frequency. In this case, the value of the VCO-ADC can be read out using single-bit quantisation on all phases.

## 1.8 Goal and Organization of this Thesis

The goal of this thesis is to create a circuit-level design for a power-efficient coarse-fine VCO-ADC. As discussed in the previous section, moving from a fine to a coarse-fine VCO-ADC can lead to a reduced power consumption at given specifications. However, adding a coarse counter to the readout circuit also presents its own challenges. The performance and imperfections of a coarse-fine VCO-ADC are investigated first on a system level, before the different blocks are designed as a transistor circuit. By thoroughly describing the relevant characteristics of the circuits, a general design method is developed and described as an algorithm. This design algorithm is inspired by the work of Borgmans et al. [2], which suggests that an optimal power consumption under demanding specifications might be obtained by a coarse-fine VCO-ADC.

To this purpose, the different building blocks of the VCO-ADC and their non-idealities are identified through a system model in chapter 2. This model will focus on the quantization noise, identifying the performance of the ideal coarse-fine VCO-ADC through simulation. The effect of asynchrony between the coarse and fine counter on this performance is then discussed and modelled, and a solution to avoid a decreased performance is presented.

Chapter 3 focuses on the design of the coarse counter based on the requirements identified in the previous chapter. The design of the coarse counter is emphasized in this thesis as this building block is unique to the coarse-fine VCO-ADC. By elegantly linking flip-flops or gated latches, a design is achieved with more relaxed timing constraints. The gated latches are then designed as a transistor-level circuit, and the performance of this circuit is discussed.

---

A transistor-based circuit is designed for the other building blocks in chapter 4. The focus is on designing the circuits to consume a minimal amount of power while still meeting certain specifications. Hence, some important sizing parameters will be identified and the power consumption and performance of building blocks will be expressed in function of these parameters. For other circuits, minimal sizing will be applied, taking advantage of the digital-friendly properties of the 28 nm technology used for this design. To conclude this chapter, an algorithm will be written and explained to optimize the sizing of the previously identified parameters for a minimal power consumption under given specifications.

This algorithm is applied to design a coarse-fine VCO-ADC for a bandwidth of 40 MHz, an SNR of 76 dB, and a third harmonic distortion (HD3) of  $-40$  dB. The performance of the resulting design is presented in chapter 5, and the distortion is reduced by employing the VCO-ADC in a differential operation and by calibration of the resulting digital output signal. Finally, an exploration of the effect of the layout on the VCO performance is presented. The thesis is then concluded with an overview of the achieved results and a suggestion of further research possibilities.

## System-Level Considerations

### 2.1 System-Level Model

The performance of coarse-fine VCO-ADCs and the effect of imperfections on their design can be studied using a system-level model. This allows us to consider possible issues separately, in a controlled environment, and test their effects. The results can then be described and explained theoretically. The goal is to identify some important guidelines that will be taken into account when designing the VCO-ADC on circuit level and interpreting its results.

The coarse-fine VCO-ADC is modelled in Simulink. The full, hierarchical Simulink model is shown in appendix A. The model consists of an ideal VCO, a coarse counter, and flipflops which sample the phases and the coarse counter output value on the rising sampling clock edge. The VCO produces  $N_\phi$  parallel square wave signals which all have an identical frequency. This frequency is a linear function of  $v_{\text{in}}(t)$ , as in equation 1.5. This perfectly linear relation is not possible in an actual ring oscillator, but is a useful assumption in the system level model. The phases of two adjacent square waves differ by  $\pi/N_p$ . The instantaneous phase of  $\phi_n(t)$  of the VCO output phase  $V_{\phi,n}$  can therefore be expressed as in equation 2.1.

$$\phi_n(t) = \int (2\pi K_{\text{VCO}} v_{\text{in}}(\tau) + 2\pi f_0) d\tau - \frac{n}{N_p} \pi \quad (2.1)$$

The coarse counter is attached to VCO phase  $V_{\phi,0}$ . It will only count full cycles of this phase, counting up to a maximal value of  $N_c$ . The model of the coarse-fine VCO-ADC therefore structurally agrees with 1.10. As shown, the behaviour of this coarse-fine VCO-ADC can be described by the block diagram of figure 1.6 or more precisely by the PFM model block diagram of figure 1.9. This requires us to decode the sampled coarse counter values and VCO phases to obtain the phase information. A simple function is written in Matlab which performs this decoding and the difference operation.

The different blocks in the system-level model mimic the different parts of the VCO-ADC circuit: the ideal VCO replaces a ring oscillator with a tuning circuit, and a transistor circuit implements for the flipflops and coarse counter. After the flipflops, the discrete and quantized samples are processed by a digital block, replaced by the Matlab function in the system model.

Equation 2.1 shows that each signal  $V_{\phi,n}$  produced by the VCO has a different instantaneous phase, and therefore the time at which a transition occurs in this signal also differs. The coarse counter is attached to phase  $V_{\phi,0}$  so this phase is used as a reference. At the sampling time, all phases up to a certain phase  $V_{\phi,i}$  will already have undergone the most recent transition and will be identical to the reference phase. These phases should be counted, and determine the position within a half-cycle, while the reference phase determines the current half of the cycle and the coarse counter determines the number of full cycles. The reference phase value should therefore be multiplied by  $N_\phi$  and the counter value by  $2N_\phi$ . The calculation of the ADC value can be schematically represented as in figure 2.1, where XNOR ports are used to compare the other VCO phase values to the reference phase. The calculated value will be noted as  $s(k)$ .

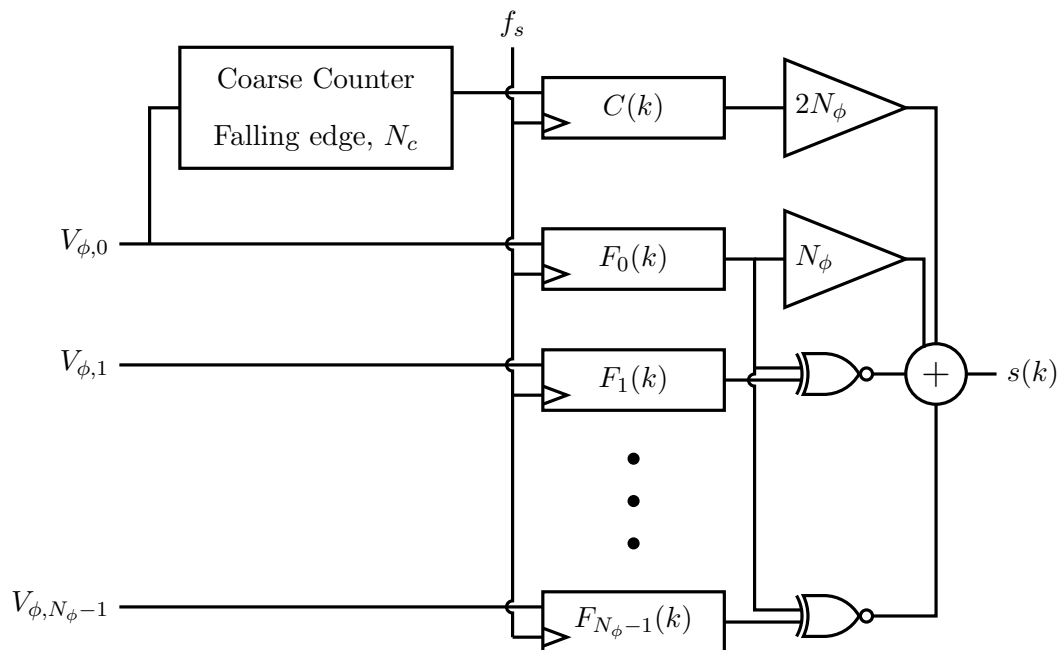


Figure 2.1: Schematic of the decoder of the coarse-fine counter values

The output of this circuit  $s(k)$  represents the total phase change of the VCO since the start of the measurement, quantized and sampled. In figure 1.6 and figure 1.9, the signal  $s(k)$  is found right before the difference operator. The value of  $s(k)$  is not reset between samples, but this does not pose any problems as long as the condition in equation 1.17 is met. The coarse counter should restart at 0 when it overflows, and the negative difference can be corrected for by adding  $2N_cN_\phi$ , the maximal value of  $s(k)$ , to the digital output. This is done automatically in a digital circuit which takes the difference of two unsigned integers. Applying the difference operation on  $s(k)$  leads to the ADC output found in section 1.4 and expressed in equation 1.8.

## 2.2 SQNR of the Coarse-Fine VCO-ADC Model

The system-level model of the coarse-fine VCO-ADC presented above can be simulated by applying a sinusoidal wave at the output. This simulation verifies the SQNR performance of the model and allows us to compare this result to the white quantization noise model and to the PFM model. The SQNR can be calculated using 1.15, repeated below as 2.2, and depends on the VCO tuning range  $f_{\text{tune}}$ , the number of phase  $N_\phi$ , and the sampling frequency  $f_s$ . In the system model, both  $N_\phi$  and  $f_{\text{tune}}$  can be arbitrarily chosen. The sampling frequency is restricted by the condition in equation 1.17.

$$\text{SQNR}_{\text{dB}} = 20 \log_{10} \left( \frac{2N_\phi f_{\text{tune}}}{f_s} \right) + 30 \log_{10} \left( \frac{f_s}{2f_{\text{BW}}} \right) - 3.41 \quad (2.2)$$

In the system-level model, the sampling frequency  $f_s$  was set equal to 1 GHz, the VCO free-running frequency  $f_0$  to 5 GHz and the VCO gain  $K_{\text{VCO}}$  to 2.5 GHz. The VCO has 32 phases. As input, sine waves with an amplitude of 1 V and a range of frequencies between 10 MHz and 39 MHz were applied. This leads to a VCO tuning range  $f_{\text{tune}} = 5$  GHz. A waveform of the digital output samples of the ADC for a frequency of 11 MHz is shown in figure 2.2.

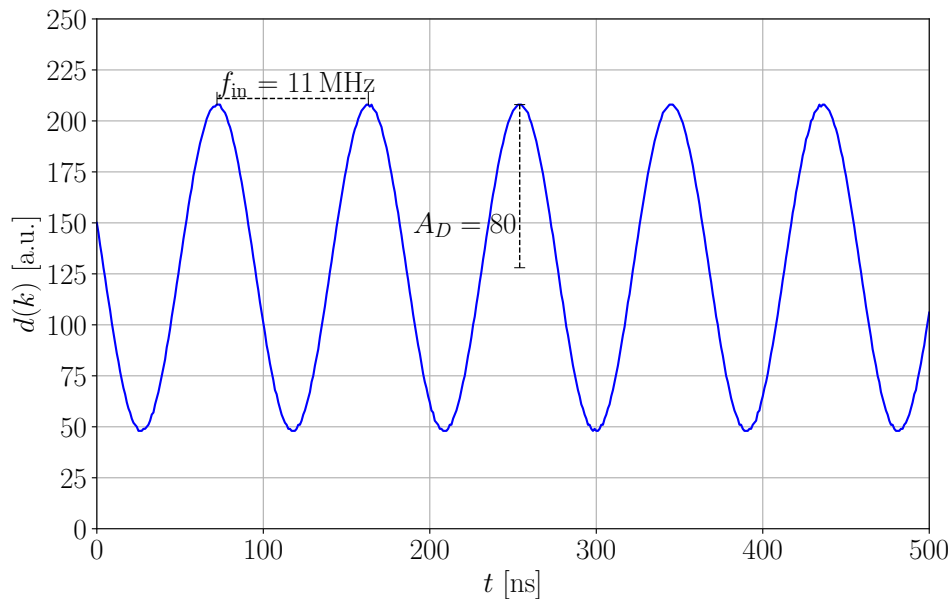


Figure 2.2: Waveform of the output of the ideal ADC

The waveform shows an amplitude which is 80 times bigger than the quantization step, which is the expected amplitude  $N_\phi f_{\text{tune}}/f_s$  given the parameters mentioned above. The signal is affected by noise, which is clearly visible in the region around the top and bottom of the sine wave. Note that noise-shaping is already applied by the difference operation, but the spectrum is not yet filtered.



Several methods are available to process the data and filter the quantization noise outside of the signal band. The most perfect filtering is obtained by performing a fast Fourier transform (FFT) over a number of samples which are periodically repeating. All of the frequencies used in these simulations are multiples of 1 MHz. The frequencies of the noise components are also multiples of 1 MHz according to the PFM model, as explained in section 1.6. This means that the resulting waveforms are periodic over 1000 samples.

Due to this periodicity, the FFT can be taken without any windowing and should reflect the behaviour described by the PFM model. The FFT will map all noise components to their exact frequency, as if the input signal is an infinitely long sinusoidal signal. This FFT is shown in figure 2.3. The spectrum shows a clear peak at the input frequency  $f_{in}$  of 11 MHz. Noise shaping is also visible: the noise increases by 20 dB/decade as marked on figure 2.3. Due to this noise shaping, the values of noise below the upper limit of the frequency band  $f_{BW}$  are in general lower than those outside the frequency band.

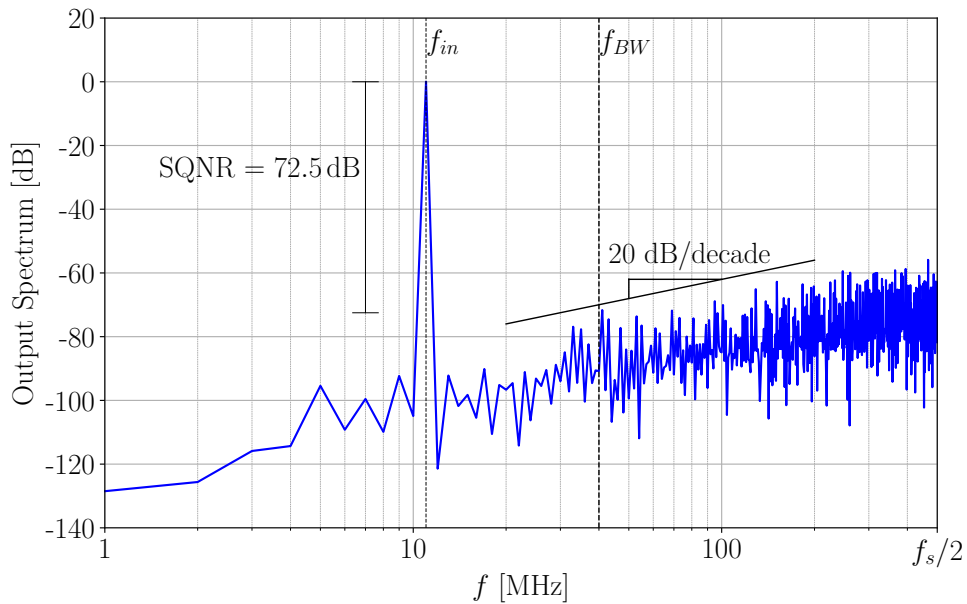


Figure 2.3: Output spectrum of the ideal VCO-ADC

Based on this spectrum, the noise in the frequency band can be calculated. The signal and DC-component can be identified and removed from the FFT. All components at frequencies higher than the bandwidth can also be removed, only leaving the components in the baseband different from zero. The total noise power can then be calculated as the power in the remaining frequency bins below the bandwidth, and the SQNR from the noise power and the amplitude of the signal. The SQNR calculated using this method will further be denoted as  $SQNR_{FFT}$ .

Filtering the signal in the frequency domain using an FFT will not be possible in a practical application, as the filter should be applied on the actual output values of the VCO-ADC. Therefore, practical decimation happens in the time domain using a 30-point Finite Impulse Response (FIR) filter. Decimation using an FIR filter is also applied to the output in the Simulink model, where decimation by factors 2, 2, and 3 are applied to get a total decimation by a factor of 12, as well as a factor 8 by applying a decimation with factor 2 three times. A sinusoidal fit is then applied to the decimated output, allowing us to calculate the amplitude of the fitted sine wave and noise power as the sum of the squares of the error terms in the time domain. The SQNR calculated using this method will further be denoted as  $SQNR_{dec8}$  and  $SQNR_{dec12}$ .

The SQNR values, expressed in dB, have been plotted in figure 2.4 for the three processing methods described above. The theoretical SQNR is calculated based on equation 2.2, and based on the PFM interpretation of the VCO-ADC. Both of these calculated results are plotted as dashed lines.

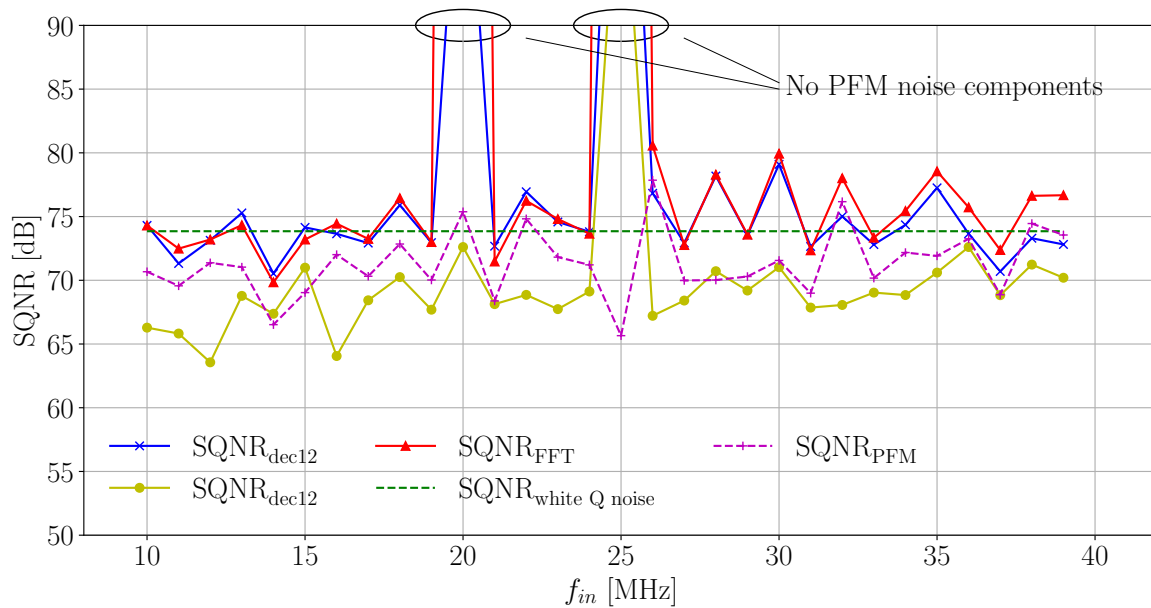


Figure 2.4: SQNR of the ADC output at different input frequencies.

The SQNR values in figure 2.4 obtained using the different methods can be compared to the theoretical values. The SQNR calculated using equation 2.2 is equal to 73.85 dB. The order of magnitude of this value agrees with most of the simulated values processed by a decimation with a factor 12 or using an FFT. However, the simulated values vary significantly with the input frequency. When they are processed by taking an FFT of the periodic input signal, the SQNR varies between 69.84 dB at 14 MHz and far over 90 dB at 20 MHz and 25 MHz. All the values agree very well with the expected values from the PFM view, confirming that this interpretation of the VCO-ADC allows for a more exact SQNR calculation.

In the PFM model the noise components only appear at discrete frequencies  $qf_0 + rf_{\text{in}}$ . Due to aliasing, these components appear in the spectrum of the discrete signal at frequencies of  $(qf_0 + rf_{\text{in}} \bmod f_s)$ . For some values of  $f_{\text{in}}$ , the number of discrete frequencies where the components appear is very limited. This can lead to extremely high SQNR values, as is the case with the input frequency of 20 MHz and 25 MHz where no components are aliased in the baseband. The simulation slightly discretizes all signals for calculations and the precision of the values in calculations is also imperfect. The values are similar enough to assume that the PFM model precisely describes the behaviour of the Simulink model.

The SQNR for decimated signals is also included in figure 2.4. For decimation with a factor 12, the SQNR takes a very similar value to the SQNR using the FFT for frequencies up to 28 MHz. At higher frequencies, the amplitude of the signal is reduced by the filter, as it is not a perfect brick-wall filter. The difference between the SQNR calculated using decimation and the SQNR using FFT becomes larger due to this reduced amplitude. A lower decimation factor, for example the factor 8 used in this example, moves this filtering effect out of the baseband. The drawback is that the noise is now increased, as the OSR is reduced. Theoretically, a change from an OSR of 12.5, as defined by the bandwidth  $f_{BW}$ , to an OSR of 8 decreases the SQNR by 5.81 dB. The decrease visible in figure 2.4 in simulations varies around this value. The filter design is important for the application of the VCO-ADC, but is not the main focus of this work. Therefore, this problem will not be discussed in more depth, and further calculations of the SQNR will be done using an FFT and an input frequency of 11 MHz, which has noise components at many frequencies according to the PFM model.

## 2.3 Asynchrony of the Coarse Counter

The performance of the ideal coarse-fine VCO-ADC is verified by the system model, allowing us to introduce nonidealities. One factor which will affect the VCO-ADC performance is asynchrony, which can be identified by looking at the readout of the reference phase. The schematic of the readout circuit, repeated in figure 2.5, shows that  $V_{\phi,0}$  affects two paths which are sampled. This phase is sampled directly, but will also be used as an input to determine the value of the coarse counter. The continuous-time signal of the coarse counter is noted here as  $C_c(t)$ . This signal is then also sampled as  $C(k)$ .

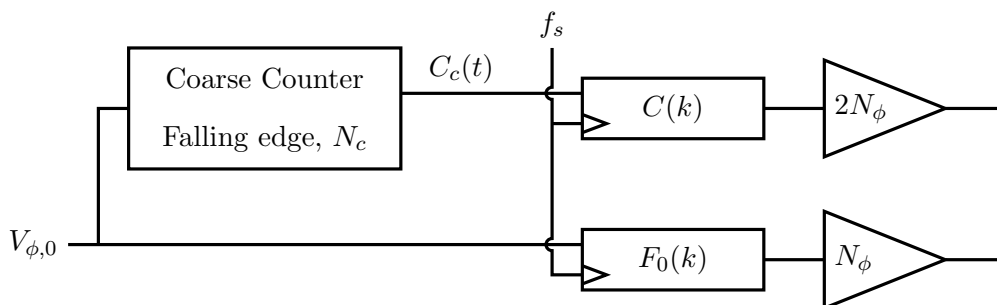


Figure 2.5: Readout of ring oscillator phase  $V_{\phi,0}$

The output value of the coarse counter is not determined instantly. The signal  $C_c(t)$  is only reliable and available for sampling after a certain delay  $\tau_c$ . This contrasts with the values of the fine counters, which do not need to pass through another circuit block and are available for sampling almost instantly. The coarse counter and the fine counters react asynchronously to the transition of ring oscillator phase  $V_{\phi,0}$ . This problem is therefore called the asynchrony of the coarse counter.

Asynchrony does not necessary cause errors. If the time between the last transition of the VCO phase and the sampling clock edge is larger than the time required to calculate the coarse counter value, no error will be visible. The sampled values of coarse and fine counters will then be correct and the effect of asynchrony will not be visible in the ADC output. However, the sampling clock edge can also happen at a time when the coarse counter value has not transitioned yet since the last clock edge. Figure 2.6a shows this situation. The sampling clock edge is the rising clock edge in this example, as it will be in all further places where a clock is marked. The time instant at which sampling occurs is marked by the thicker dashed red line. The values of  $V_{\phi,0}(t)$  and  $C_c(t)$  are sampled as 0 and 5 respectively.

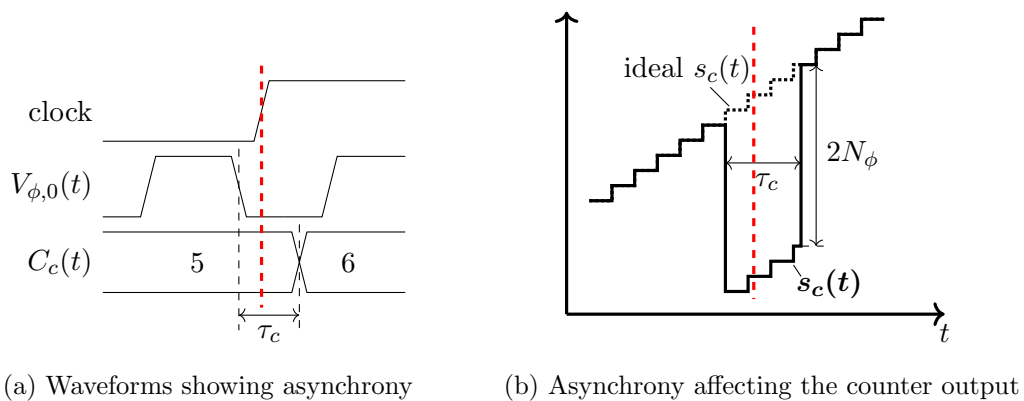


Figure 2.6: Effects of asynchrony on signals in the readout circuit

The waveforms in figure 2.6 show the main problem with asynchrony. As seen in figure 2.1, an increase of 1 in the value of the coarse counter is multiplied by  $2N_\phi$  and added to the value of the fine counters to obtain  $s(k)$ . The falling edge of phase  $V_{\phi,0}$  resets the fine counters from  $2N_\phi - 1$  to 0. The reset of the fine counters should therefore be countered by an increase of  $2N_\phi$  in the value of the coarse counter, but this does not happen at the same time as the reset due to asynchrony. The continuous-time signal  $s_c(t)$  can be considered as the value of  $s(k)$  before sampling. The signal  $s_c(t)$  is also the quantized version of the phase and therefore expected to be a non-decreasing function. The evolution of  $s_c(t)$  in function of time is shown in figure 2.6b. As shown there, if sampling happens during the time frame  $\tau_c$  after the transition when the coarse counter has not increased yet, the sampled value will have an error of  $-2N_\phi$ .

The effect of this asynchrony on the ADC output and the SQNR value is simulated. A delay block is added between the ideal coarse counter and the sampler in the system model. The combination of the ideal coarse counter and the delay gives a coarse counter which behaves as sketched in the waveforms in figure 2.6a. The value of the counter does not change until a time  $\tau_c$  after the transition. An identical delay is used for all coarse counter transitions. The simulation is done with identical parameters as the previous simulations in section 2.2, at an input frequency of 11 MHz. The delay will be expressed in function of the VCO free running period  $T_0 = 200$  ps as this parameter is important to interpret its effect on the SQNR. Figure 2.7 shows a waveform of the ADC output  $d(k)$  before filtering when it is affected by asynchrony modelled by a delay  $\tau_c = T_0/20$ .

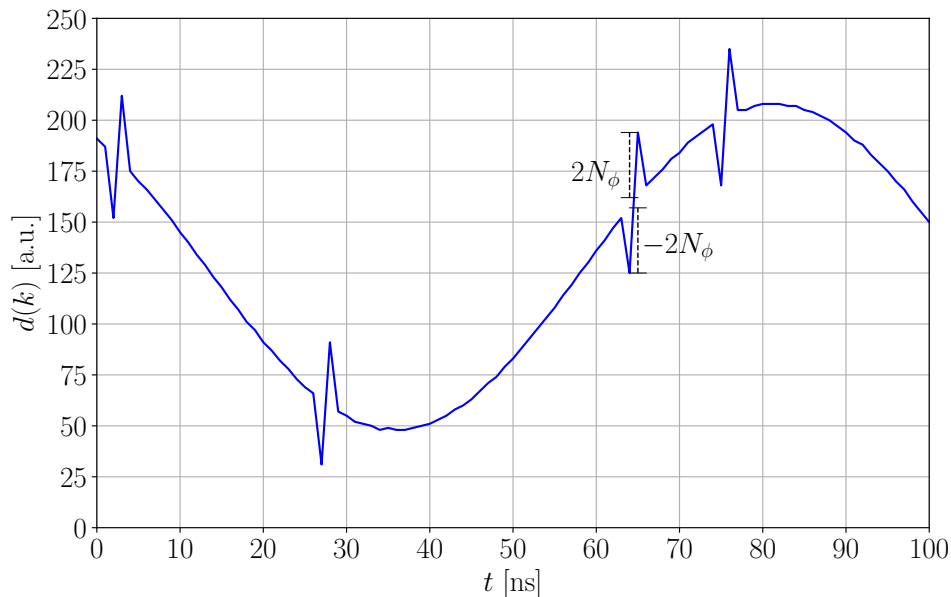


Figure 2.7: Waveform of the output of the ADC affected by asynchrony

Comparing this waveform to the example waveform without asynchrony in figure 2.2, it can be seen that the waveform in figure 2.7 shows errors. These errors are visible as pairs of a negative peak of 64 lower than the expected value followed by a positive peak of 64 in the next sample. This is due to the difference operation: the value of  $s(k)$  is only affected during a single sample, but this also affects the value of  $d(k)$  of the next sample. The error due to asynchrony can be modelled as noise  $n_{\tau_c}(k)$  added to the ideal  $s(k)$ .

To further develop the model, a description of the noise is needed. This noise affecting  $s(k)$  will take a value of either 0 or  $-2N_\phi$ . During each period of the VCO  $T_{VCO}$ , the noise takes the value  $-2N_\phi$  for a time  $\tau_c$ . It is assumed that the time instant when sampling happens is independent of the time instant when transitions on VCO phase 0 happen. The difference between the time of sampling and the time of the last VCO falling edge is therefore uniformly distributed over the period of the VCO. The probability of an error of  $-2N_\phi$  is then equal to the relative amount

of time at which there would be an error,  $\tau_c/T_{VCO}$ . The delay  $\tau_c$  is equal for all samples in the system model for asynchrony. The VCO period  $T_{VCO}$  is different for each sample and depends on the input value  $v_{in}(t)$ . For the sinusoidal input signal, it can be assumed that the noise behaves as if the period is always equal to the average period  $T_0$ , which is the inverse of the free-running frequency  $f_0$ . With these two assumptions, the noise is iid for all samples, just like the quantization noise.  $n_\tau(k)$  can therefore be modelled as a Bernoulli process, which takes value 0 with probability  $1 - \tau_c/T_0$  and value  $-2N_\phi$  with probability  $\tau_c/T_0$ . The noise variance is that of a Bernoulli variable, expressed in equation 2.3.

$$\sigma_{\tau_c}^2 = (2N_\phi)^2 \frac{\tau_c}{T_0} \left(1 - \frac{\tau_c}{T_0}\right) \quad (2.3)$$

This noise  $n_\tau(k)$  is white noise, due to the assumption that the noise is iid for all samples. This means that the spectral density will be constant over the discrete band  $f_s$ . Just as with the quantization noise described in figure 1.8, the noise is shaped by the NTF which consists of the difference operator, and then filtered by the digital filter. The power of the noise which remains in the frequency band of interest is given by the variance multiplied by a factor proportional to the third power of the oversampling rate. This asynchrony noise power is shown in equation 2.4.

$$P_{\tau_c} = \frac{1}{3} \pi^2 (2N_\phi)^2 \frac{\tau_c}{T_0} \left(1 - \frac{\tau_c}{T_0}\right) \text{OSR}^{-3} \quad (2.4)$$

Based on the noise power due to delay expressed above and the quantization noise power in equation 1.14,  $\text{SNR}_{Q,\tau_c}$  can be calculated as the SNR under quantisation noise and asynchrony. This theoretical SNR is plotted in function of the delay  $\tau_c$  in figure 2.8, as a dashed line. The system model with asynchrony was also simulated for different values of  $\tau_c$ . The resulting SNR of these simulations is also plotted as the full line in figure 2.8.

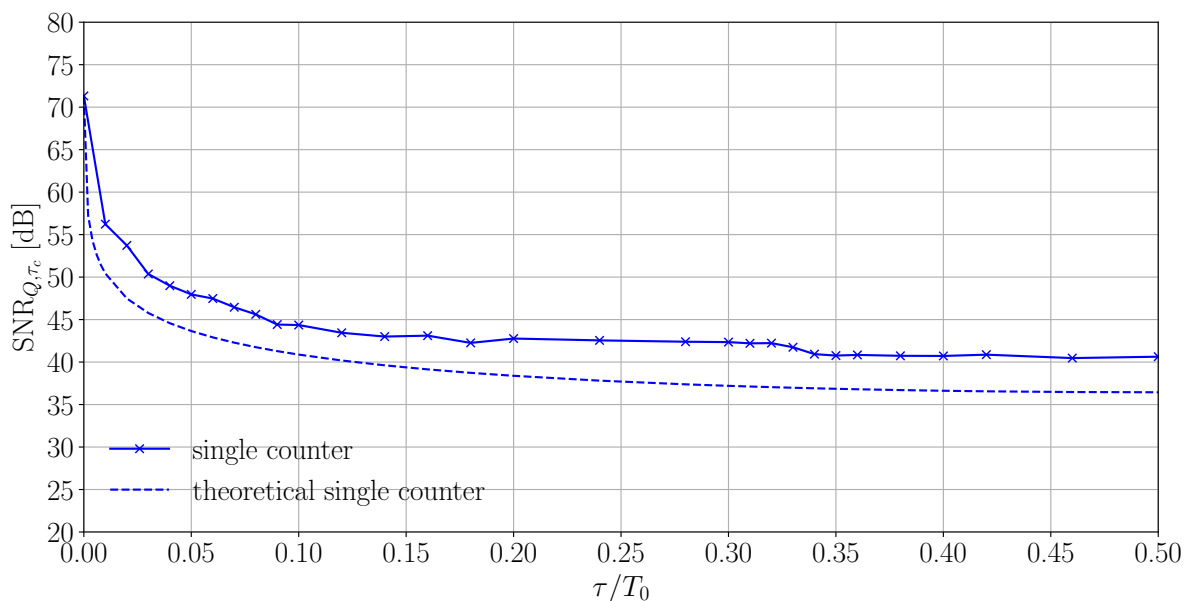


Figure 2.8: Effect of asynchrony on the SNR

The SNR becomes severely lower due to the effect of even a small delay, as is visible in figure 2.8. The theoretical and simulated values of  $\text{SNR}_{Q,\tau_c}$  both decay significantly, although there is some difference between the values on the plot. This may be due to the assumptions on the independence of the VCO and clock outputs and the effect of the average VCO frequency. In the theoretical model, the power of the noise due to asynchrony is proportional to  $\tau_c/T_0$  when for  $\tau_c \ll T_0$ . When the value of  $P_{\tau_c}$  is larger than  $P_Q$ , the power of the quantization noise, the SNR value will decrease by  $-10 \log(\tau_c/T_0)$  dB as the delay increases. This explains the rapid decay of  $\text{SNR}_{Q,\tau_c}$  in figure 2.8.

In reality, the effect of asynchrony is not modelled perfectly by this system model. The coarse counter does not need to have the previous value as output during the calculation of the new value. It is possible that the output transitions between different values before reaching the desired value. If the coarse counter is sampled in one of these transition states, the value might be wrong by more than 1 and a larger absolute error can affect the value of  $s(k)$ . Another possibility is that the value of the coarse counter is changing exactly during the rising clock edge, when this value gets sampled. The sampled value may be unresolved between a logic ‘1’ or ‘0’, and this unresolved value can propagate to the digital circuit which calculates  $s(k)$ . This is called metastability [11] and may affect both the sampling of the coarse and fine counter outputs, but is mentioned here already as the timing for the coarse counter transitions may cause metastable states even during the transition  $\tau_c$ .

The relatively simple model of equation 2.4 indicates the importance of dealing with asynchrony in a coarse-fine VCO-ADC. In this model, the power of the asynchrony noise becomes dominant over the quantization noise power at a delay of  $T_0/(12(2N_\phi)^2)$ . Only a very small asynchrony can be tolerated to reach the desired SNR if the design of the VCO-ADC is limited by the SQNR. It is impossible to design a coarse counter which calculates the correct value in such a short time. To obtain a reliable coarse counter output at much larger delays, a different readout circuit for  $V_{\phi,0}$  must be designed.

## 2.4 Double Coarse Counter

The problem of asynchrony is solved always a reliable value of the coarse counter available for sampling. To this end, a readout circuit should be designed for which the next coarse counter value has already been calculated when the transition of the fine counter occurs. Two coarse values need to be available for this: one value which can be sampled before the transition, and another value which can be sampled after the edge on  $V_{\phi,0}$ . Both of these counters require some time to calculate the new value and therefore should start their transition at different time instants. One will count on the falling edges of VCO phase 0, as the counter in figure 2.5 already did. The other counter will count on the rising edges. Figure 2.9 shows the proposed readout circuit. Perez et al. [12] have demonstrated such a double coarse counter for a VCO-ADC connected to a microphone MEMS.

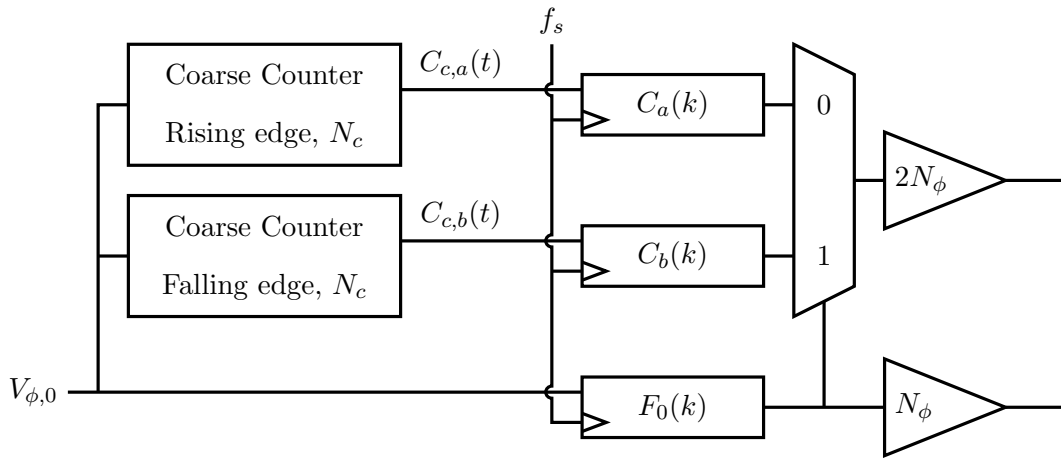


Figure 2.9: Readout of ring oscillator phase  $V_{\phi,0}$  using the double coarse counter.

Counting on alternating edges of VCO phase 0 has a dual advantage. The edges are separated by half a period of the VCO, giving both coarse counters an equal time to calculate the new value. Additionally, the fine counter sample  $F_0(k)$  can be used to select which counter is reliable. If the fine counter has digital value ‘0’, the last falling edge occurred more recently than the last rising edge. Therefore, the counter which counts on the rising edge,  $C_a(k)$ , is be more reliable, and vice versa when the fine counter value is ‘1’. In figure 2.9, the multiplexer selects the value used for the calculation of  $s(k)$ . In the example of figure 2.10,  $s(k)$  is correctly determined by selecting the value of coarse counter A, which has already increased.

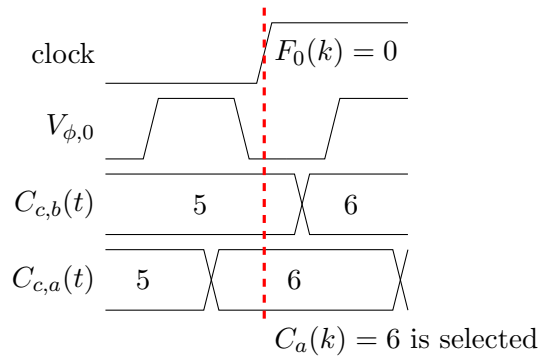


Figure 2.10: Effect of asynchrony in double counter

Figure 2.10 shows that the double coarse counter effectively deals with asynchrony between the coarse and the fine counters. As long as the coarse counters guarantee a reliable output a delay  $\tau_c$  after the transition, the correct counter value is sampled and  $s(k)$  can be calculated without errors. To design the coarse counter, it is required to determine the maximal delay that can be tolerated. The system model with the double coarse counter is therefore simulated. Both counters have the same delay  $\tau_c$ , and the decoding function is adapted to select the correct output similarly to the multiplexer in figure 2.9. The resulting  $\text{SNR}_{Q,\tau_c}$  value is plotted in figure 2.11, where the SNR for the single counter from figure 2.8 is also included as a reference.



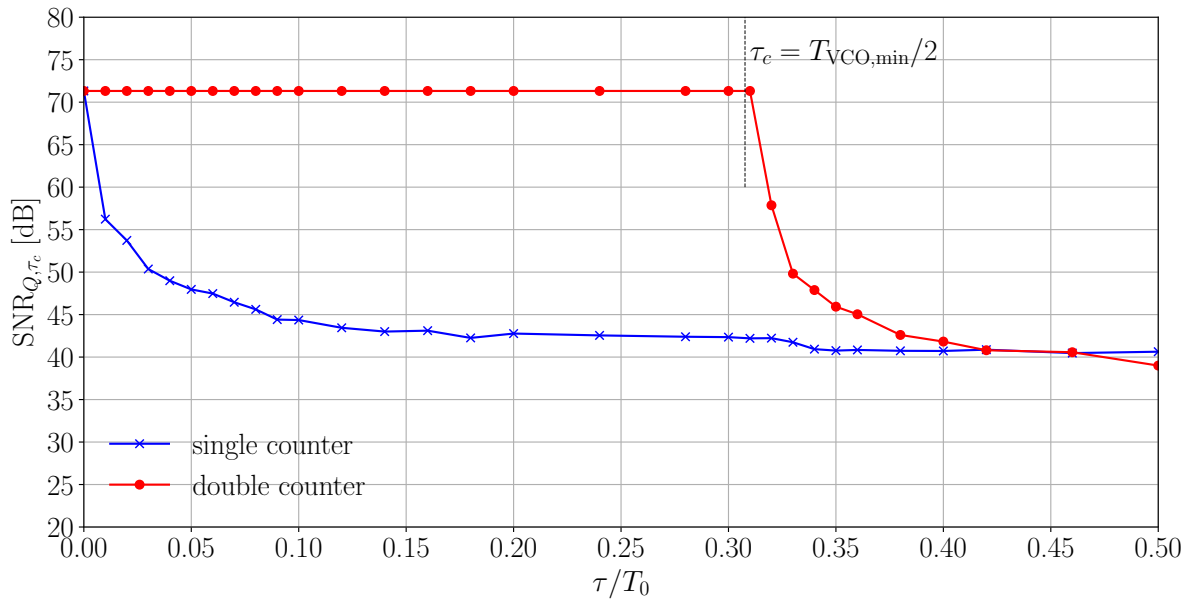
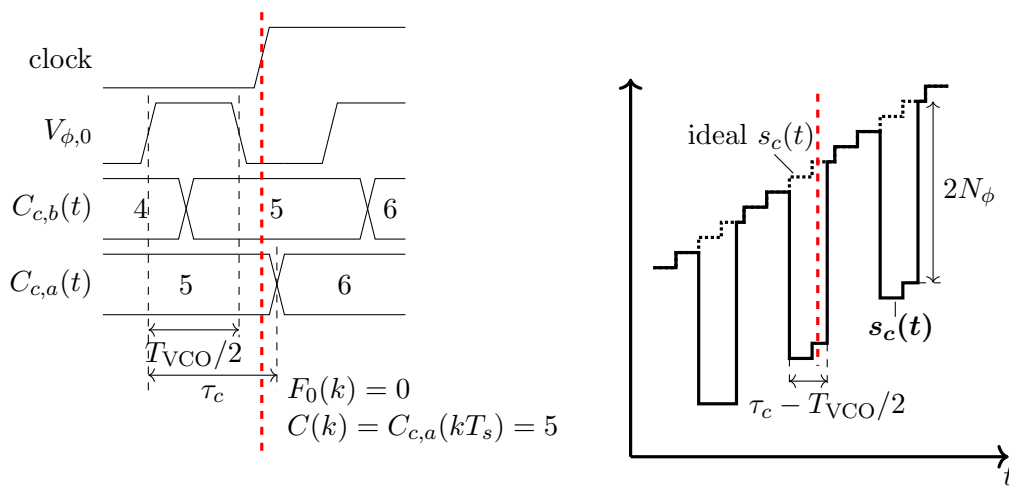


Figure 2.11: Effect of asynchrony on the SNR for the double counter

The values of  $SNR_{Q,\tau_c}$  for the double counter in figure 2.11 remain unaffected by the delay for low values of  $\tau_c$ .  $SNR_{Q,\tau_c}$  starts decreasing rapidly when the delay is larger than half of the minimal VCO period  $T_{VCO,min}$ , as marked by the dashed line in figure 2.11. To avoid that asynchrony affects the SNR of the VCO-ADC, the condition on the delay of the coarse counter given in equation 2.5 should be met.

$$\tau_c < \frac{T_{VCO,min}}{2} \tag{2.5}$$



(a) Asynchrony waveforms

(b) Asynchrony affecting the counter output

Figure 2.12: Errors due to asynchrony in the double counter readout circuit

Figure 2.12a shows the waveforms for the situation when the coarse counter introduces a delay larger than  $T_{VCO}/2$ . The value of  $s_c(t)$  is again affected errors of  $-2N_\phi$  during a short time after each transition on VCO phase 0. These errors will be visible in the ADC output when it is sampled during this time. These errors in  $s_c(t)$  only take place when the delay time is longer than half of the minimal VCO period, leading to the decreased value of  $SNR_{Q,\tau_c}$  in figure 2.11.

## 2.5 Metastability and Mismatch in the Double Counter

The double coarse counter has an additional advantage over the single coarse counter. Consider the situation in which the condition in equation 2.5 coarse counter is met. One of the coarse counter values has certainly completed its last transition. It provides a reliable value for sampling, which is selected by the multiplexer. The coarse counter value can therefore not be metastable, which is important as the bits of the coarse counter will be the most significant bits in the value of  $s(k)$ . Metastability can still occur in the fine counters. This will be further discussed when designing the samplers of the fine counters in chapter 4.

A final interesting aspect of the double coarse counter is what happens when the clock edge approximately coincides with the fine counter edge. Due to mismatch in the circuits or slightly different arrival times of the clock signal, it is possible that the transition or sampling of the coarse counter occurs at a slightly different time than the fine counter. For a short time around the VCO edge, the double coarse counter should be able to give the correct value whether the fine counter is sampled as ‘0’ or ‘1’. To avoid errors, the coarse counter should therefore hold its value for a short time  $\epsilon$  after the transition and also finish its transition a time  $\epsilon$  before the necessary VCO edge. This is indicated in figure 2.13.

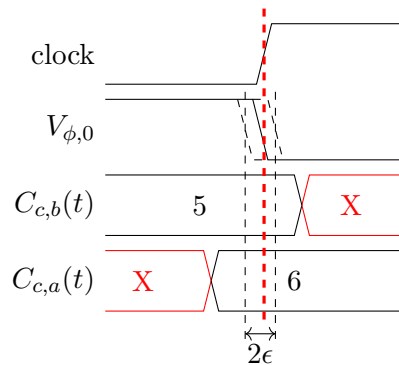


Figure 2.13: Margins around transition to withstand mismatch

## Design of the Coarse Counter

The double coarse counter consists of two identical counter circuits. These counters need to be designed carefully and are unique to the coarse-fine VCO-ADC. Counting can be implemented using different methods and different codes can be used to represent the counted value. This code is always shown as a number of bits  $N_{b,c}$  which can be sampled and used to maximally distinguish  $2^{N_{b,c}}$  different counter values. A counter which counts  $N_c$  different values therefore requires at least  $\lceil \log_2 N_c \rceil$  bits.

### 3.1 Synchronous Counters

A synchronous counter is a first possible implementation for a counting circuit. In this circuit, all of the bits representing the counted value are clocked into a flipflop simultaneously at the time of the relevant VCO edge. The next bits are then calculated using a digital circuit consisting of logic gates. Figure 3.1 shows this situation. The bits are sampled by rising-edge triggered D-flipflops. The implementation of the digital circuit which calculates the next value varies for different codes and implementations, and is therefore represented by the generic ‘calc’ block. Note that this logic can be different to determine the new input for each bit. The rest of the circuit is identical for all  $N_{b,c}$  bits, which are sampled by flipflops and provide the inputs for the calculation of the next bits.

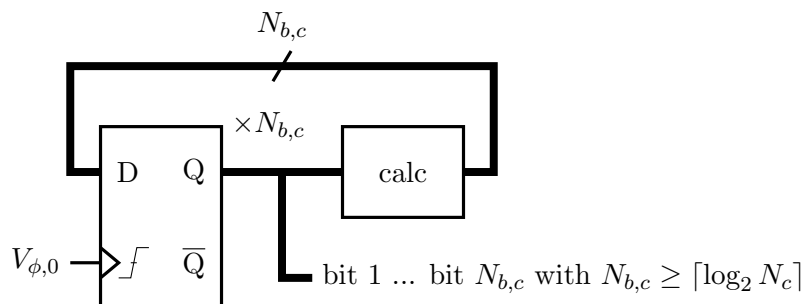


Figure 3.1: Synchronous counter

Several designs in literature use a synchronous counter. Daniëls et al. [13] have designed a VCO-ADC with a synchronous binary counter. Quintero et al. [14] demonstrated a design based on a Gray synchronous counter.

The design of a coarse counter for the VCO-ADC presented in this work will mainly take into account the expected power consumption of the coarse counter and the difficulty of achieving the timing constraint for the double coarse counter presented in section 2.4. In the synchronous coarse counter, the bits need to propagate through the flipflops in less than half of the minimal VCO period. The time for the calculation to make the new bits available is also less than  $T_{VCO,min}/2$ . Both of these timing constraints are acceptable, although the logic required to determine the bits becomes more complex if the number of bits increases. For a coarse counter which counts many cycles, an extensive calculation may be necessary to determine the next bit which is difficult to implement this within the required timing constraints. Other codes, such as the maximum length sequence, make it easier to calculate the next bit but the calculation after sampling is more complex.

The problem with the synchronous counter is therefore that the calculation of the next value consumes a significant amount of power, due to the limited time for this calculation. The increased complexity when the number of bits increases also makes this type of counter less attractive to have a scalable counter which can be used in many applications. A second drawback is that all bits are registered by the flipflops during each VCO cycle. This is more often than necessary as not all bits will change each cycle. The flipflops therefore consume more power than necessary. These flipflops also have to be designed carefully, as the VCO output signal is not a rail-to-rail square wave due to the tuning of the VCO. It is therefore argued that a more power-efficient coarse counter than the synchronous binary counter can be designed for the VCO-ADC.

### 3.2 Asynchronous Binary Counters

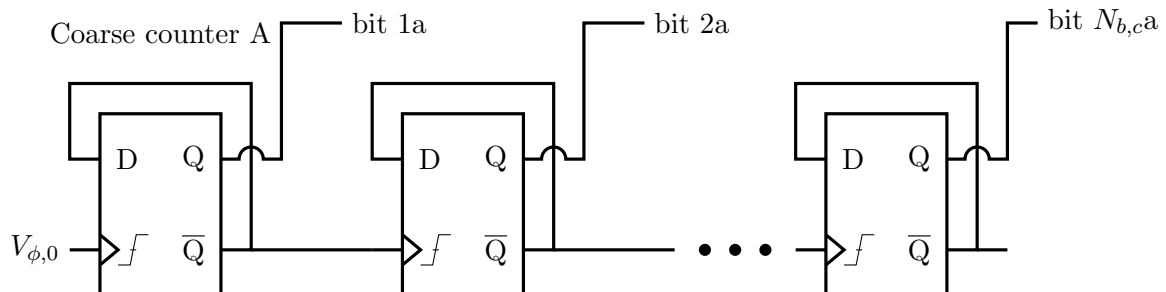


Figure 3.2: Asynchronous binary counter

An asynchronous binary counter, shown in figure 3.2, is expected to be a more power-efficient design for the coarse counter. In this counter design, the inverse output of the counter is used both as the new data input and as the clock signal of the next counter. The bit of the counter

therefore changes its value every time the rising edge appears at the clock input of the flipflop. This means that the waveform of the bit will have half of the frequency of the waveform of the previous bit. Figure 3.3 shows that this creates a binary counter when the values of the different bits are combined.

Note that the asynchronous binary counter is a power-efficient design: no calculations are necessary to determine the next value for the flipflop, as the inverse signal will always be available. Also, the values of the flipflops will be clocked less often due to the lower frequency of the signal of the previous flipflop. The number of times the values are clocked is the minimal number of times needed for a binary counter, as the bits change every time. Since the counter output is binary, the sampled bits selected by the multiplexer can be used directly in calculations.

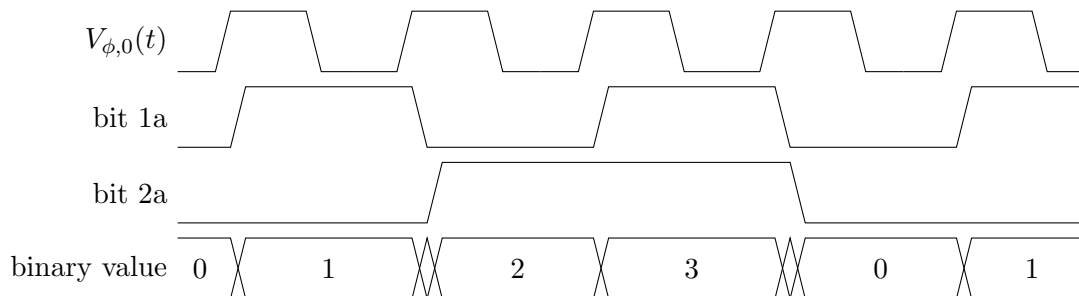


Figure 3.3: Waveforms in the asynchronous binary counter

The waveforms in figure 3.3 show how the binary signal is generated. The code increments on every rising edge of  $V_{\phi,0}$ . This causes bit 1 to change its value. When bit 1 transitions from '1' to '0', bit 2 flips. The falling edge can ripple through the counter until a certain bit becomes high and the binary value is increased. If all of the bits fall, the counter is reset automatically as desired in the system model in section 2.1.

When the asynchronous counter is implemented in a double coarse counter design, it has to be guaranteed that the value of coarse counter B follows the value of coarse counter A. Otherwise, mistakes can be made when applying the difference operation to successive samples. Therefore, Perez et al. [12] have proposed the coarse counter in figure 3.4. The red, dotted lines connect the output of counter A with the data input of counter B. Due to these connections, the flipflops in counter B always sample the value of the corresponding bit of coarse counter A. On the rising edge of  $V_{\phi,0}$ , counter A increases. Counter B then takes the same value as counter A on the falling edge, avoiding any issues between successive samples.

Looking in more detail at the sketched waveforms in figure 3.3, it can be seen that the transition of each flipflop occurs slightly delayed compared to the relevant edge of its clock signal. These delays are the main issue with the asynchronous binary counter. Due to propagation time and rise time, the output of the flipflop may only react to the clock signal after a delay  $\tau_b$ . For some transitions, all the bits have to flip sequentially. The time until the coarse counter output is reliable then equals the sum of the delays of  $N_{b,c}$  flipflops. Using this worst-case delay as  $\tau_c$ , the

condition in 2.5 can be expressed in equation 3.1. The condition for the delay of a bit becomes worse as the number of coarse counter bits increases. No general, scalable coarse counter design is obtained with the asynchronous binary counter.

$$\tau_c = N_{b,c}\tau_b < \frac{T_{VCO,min}}{2} \quad (3.1)$$

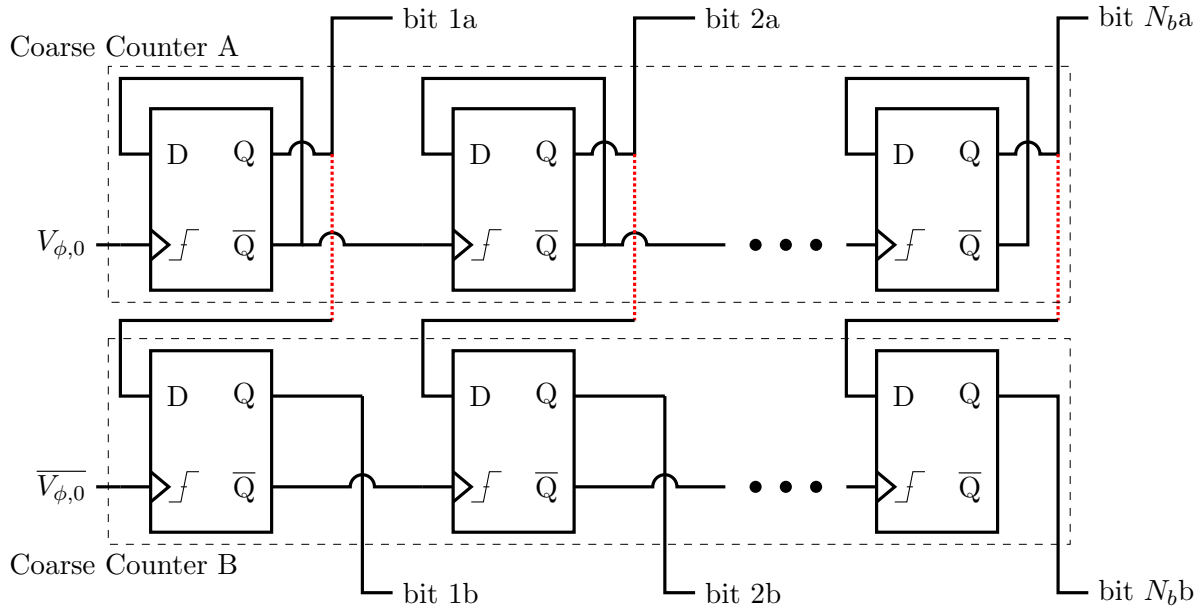


Figure 3.4: Connected asynchronous binary counters

### 3.3 Double Connected Binary Counter

The main issue with the asynchronous binary counter is that the flipflops react sequentially as a transition ripples through the counter. The condition in equation 3.1 can therefore significantly limit the maximal VCO frequency. A high VCO frequency is necessary to achieve a high SQNR as described by equation 1.15. An improvement to the connected asynchronous binary counter of figure 3.2 is required to relax the timing constraint on the minimal VCO period.

To solve this issue, the double connected binary counter in figure 3.5 is proposed. In this counter, the blue, dashed lines connect the data output of bit 1b to the clock input of bit 2a. Therefore, the transition of bit 2a starts when bit 1b rises, which is a time  $T_{VCO}/2$  earlier than in the connected asynchronous binary counter. This is visible in figure 3.6, which shows the waveforms in the double connected binary counter.

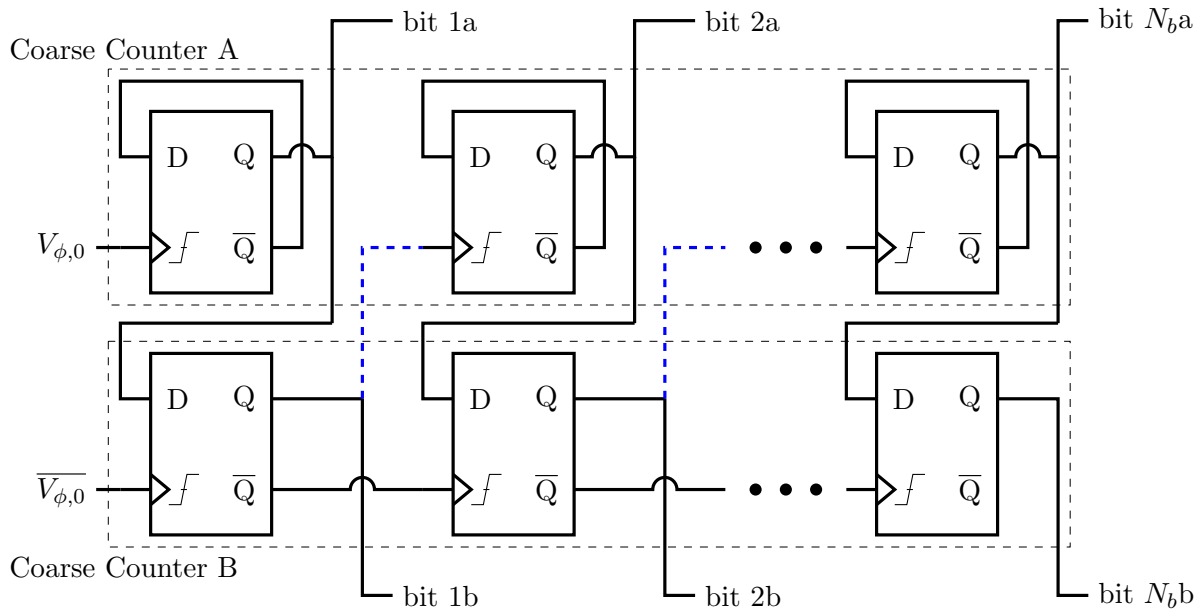


Figure 3.5: Double connected binary counter

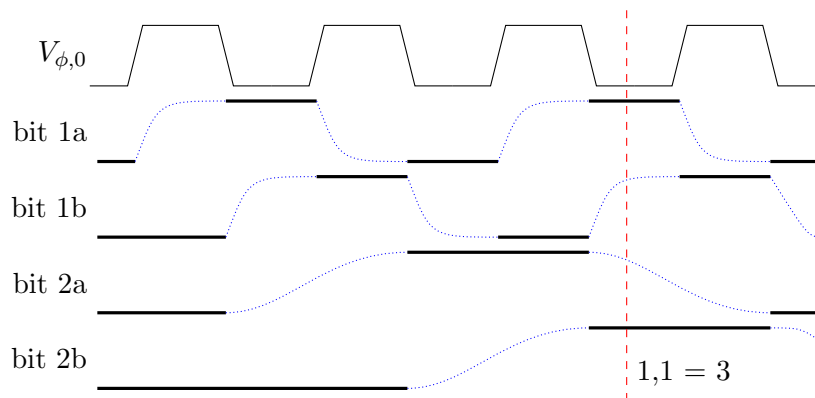


Figure 3.6: Waveforms in the double connected binary counter

Figure 3.6 shows the advantage of the new method to connect the double binary counter. Bit 2a and bit 2b react to opposite edges of the same bit, 1b. Therefore, the transitions of the bits, shown by the blue, dotted lines in the waveform, are maximally spread out. It is required that one of the bits 2a and 2b is reliable at any time. The reliable bit is marked by the full black line in the waveform in figure 3.6. By maximally spreading the transitions, the time for the edge to ripple through the first and second flipflop is also maximal. Bit 1b still requires its transition to be completed in a time  $T_{VCO,min}/2$ . However, the condition on bit 2b is strongly relaxed: the time between the edge of  $V_{\phi,0}$  to which it reacts and the moment it needs to be reliable is now  $T_{VCO,min}$ , double the requirement in equation 3.1. The double connected coarse counter therefore allows us to use a significantly higher VCO frequency.

The double connected coarse counter requires us to select each bit independently from the correct counter. Bit 2b is always reliable when the selected value of bit 1 is high, and bit 2a is reliable when this value is low. This is also visible in figure 3.6. Similarly to the selection of the first bit by the fine counter output, the second bit can now be selected by the first bit of the coarse counter. The multiplexers selecting the different bits must now be placed in series, as the output of each multiplexer selects the next bit. This increases the complexity of determining the coarse counter value after sampling, a drawback of this type of counter. However, it is expected that the simpler counter circuit at the relatively high VCO frequency will outweigh the increased complexity of the digital circuit, which works at the lower sampling frequency. In figure 3.6, bit 1a is selected due to the ‘0’ value of  $V_{\phi,0}$  and bit 2b due to the ‘1’ value of bit 1a. The coarse counter has correctly determined that 3 rising edges have happened since the beginning of the waveforms.

This method of connecting the coarse counters can easily be expanded to a higher number of bits. Bit  $na$  always has to transition at the rising edge of bit  $(n-1)b$  and bit  $nb$  on the falling edge. The time until the transition on bit  $n$  has to be completed is always half of the period of bit  $n-1$ . Therefore, the time for the edge to ripple through the counter to a certain bit doubles at each bit. This can be expressed as in equation 3.2. Comparing this condition to 3.1 shows that the timing constraint in the double connected binary counter is less strict.

$$N\tau_b < 2^{N-1} \frac{T_{VCO,\min}}{2} \text{ for all } N = 1, \dots, N_b \quad (3.2)$$

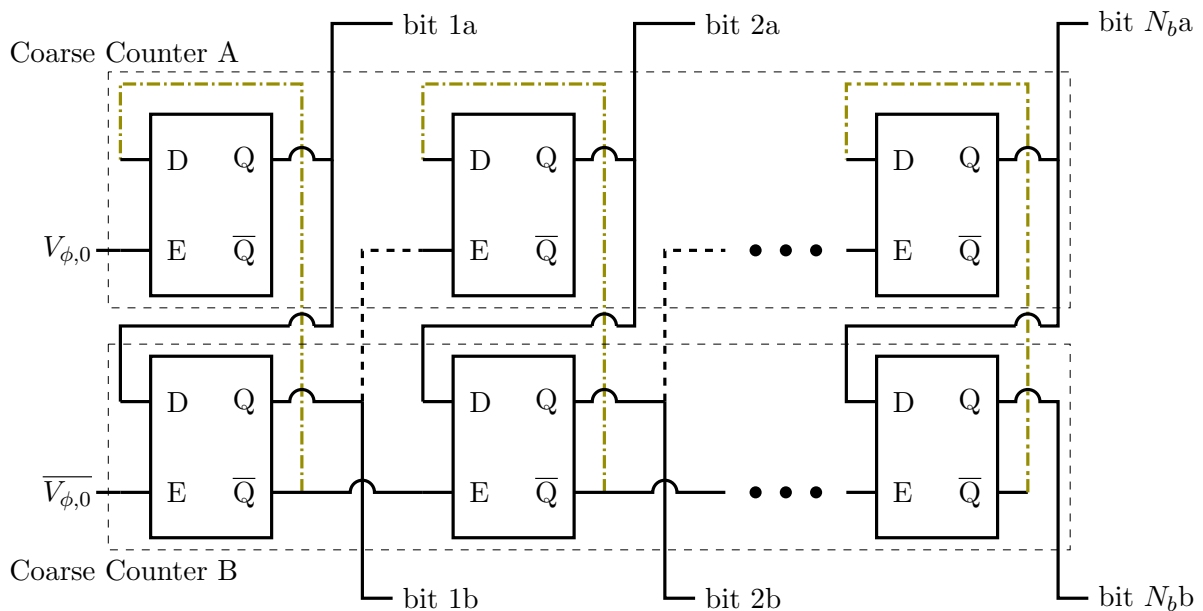


Figure 3.7: Double connected binary counters: gated latch implementation



A further simplification for the connected asynchronous binary counters is shown in figure 3.7. In this design, gated latches are used instead of flipflops, as shown by the enable input noted ‘E’ instead of the clock input of the flipflops. This does not change the functionality of the counter because the inverse of the output of bit 1b is now connected to the data input of bit 1a. When the VCO phase is high, the gated latch of bit 1a is enabled and takes the inverse value of bit 1b. When it is low, the latch of bit 1b is enabled and takes the value of bit 1b. Hence, one bit only reacts to the change in the other bit at the next clock edge, as desired. The dash-dotted lines show the altered connection compared to figure 3.5.

The advantage of this new design is that a gated latch is easier to design than a flipflop, which usually consists of two gated latches in a master-slave configuration. Since a gated latch is a simpler circuit than a flipflop, less parasitic capacitances need to be charged when a transition occurs. Therefore, the power consumption is expected to be lower.

## 3.4 Gated Latch Design

### 3.4.1 Gated NAND-latch

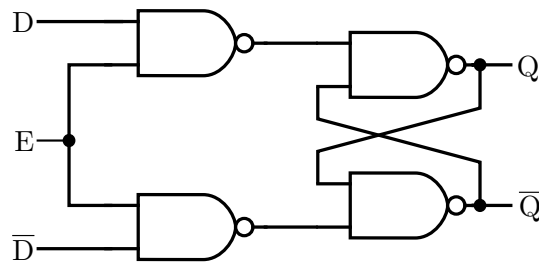


Figure 3.8: Gated NAND-latch

To complete the design of the coarse counter, a circuit-level implementation of the gated latches is necessary. A logic gate design for the gated latches used in the circuit of figure 3.7 is shown in figure 3.8. The design consists of 4 NAND gates. The last two gates are cross-coupled to form a NAND-latch. The first two gates are used to mask the input of this latch when the enable signal is low.

The transistor-level design of the NAND-gate itself can be seen in figure 3.9. All circuit-level designs are implemented in 28 nm CMOS, using ultra low threshold voltage (ulvt) flavor for the transistors. In this design, the NMOS transistors are sized minimally, with a width of 100 nm and a length of 30 nm. The PMOS transistors are sized with the same length but their width is doubled to 200 nm, to have a similar  $g_m/I_d$  value as the NMOS transistor and therefore place the transition level of the circuit at approximately  $V_{dd}/2$ . This small sizing is used to minimize the parasitic capacitances on the nodes of the circuit. Small parasitic capacitances are expected to increase the speed of the circuit, and reduce its power consumption. The sizing of the NAND-gate is repeated in table 3.1.

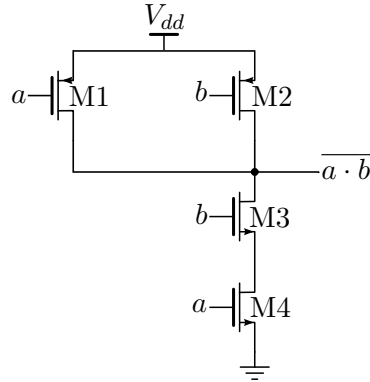


Figure 3.9: NAND gate transistor-level design

	M1	M2	M3	M4
width [nm]	200	200	100	100
length [nm]	30	30	30	30

Table 3.1: Transistor sizing of the NAND gate

The circuit of figure 3.8 with figure 3.9 as NAND-gate is implemented in cadence. As visible in figure 3.9, the inputs of the gate are not symmetrical. Input  $b$  is used for the enable signal and for the outputs, while the data inputs and internal signals are applied to input  $a$  of the gates. The transient response of the circuit on the rising edge of the enable signal is simulated and plotted in figure 3.10. The enable signal itself is shown as a dotted line, of which the rising edge starts at 80 ps and has a rise time of 20 ps. The situation when  $D$  is high and the output  $Q$  rises is shown in figure 3.10a. Figure 3.10b shows the opposite situation when  $D$  is low the output  $Q$  falls. Due to symmetry, the transient responses of  $\bar{Q}$  are identical to the plotted responses, but react oppositely to the value of the data input.

The VCO output  $V_{\phi,0}$  is applied as the enable signal for the first coarse counter. As it will be explained in section 4.3, this signal is not rail-to-rail. The low value of the square wave of  $V_{\phi,0}$  can become significantly higher than 0 mV due to the tuning of the VCO. Therefore, the transient responses in figure 3.10 are plotted for different values of this tuning voltage  $V_{\text{tune}}$ . The simulated responses are obtained with eleven different  $V_{\text{tune}}$  values between 0 mV and 500 mV, separated by 50 mV. When  $V_{\text{tune}}$  is increased, the system reacts faster to the rising edge because the voltage at which the PMOS transistors switch off and the NMOS transistors switch on is reached earlier. However, when  $V_{\text{tune}}$  becomes too high, the next data value partially or fully leaks through. This means that the output does not have a desired value close to 0 mV at the start of the simulations when  $V_{\text{tune}}$  is equal to 450 mV or 500 mV, as visible in figure 3.10a. Increasing  $V_{\text{tune}}$  also increases the leakage current as the NMOS transistor of the enable signal lets more current pass through. For this reason, the enable signal is applied as signal  $b$  in the NAND-gate of figure 3.9, which has a slightly higher source voltage due to the drain-source voltage of transistor M4.

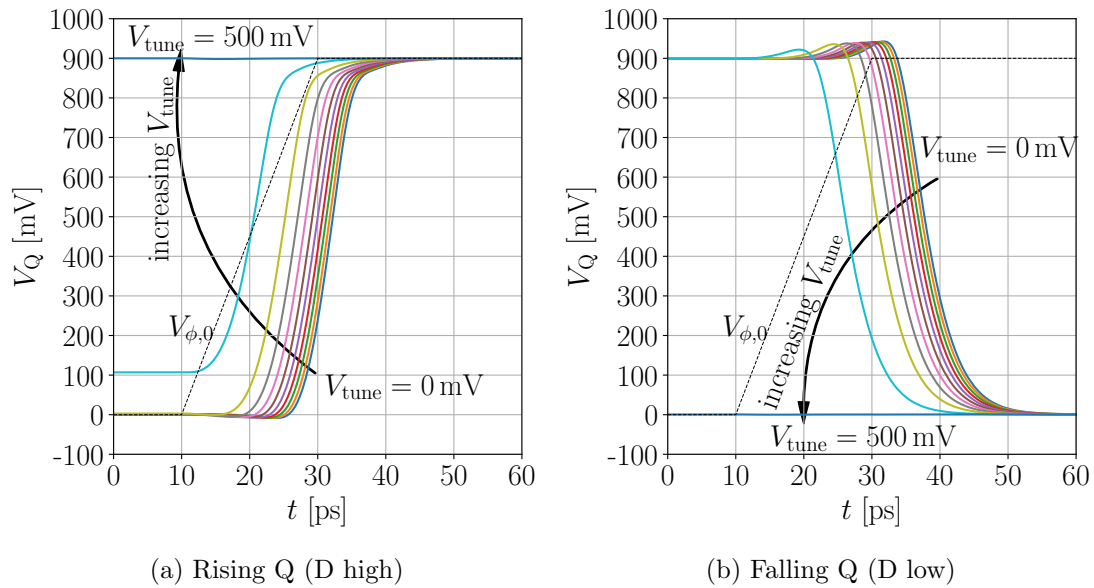


Figure 3.10: Transient response for the gated NAND-latch

The gated NAND-latch is expected to work well at values of  $V_{\text{tune}}$  up to 400 mV. At these values, the rising edge has finished its transition less than 40 ps after the start of the rising edge on the enable signal, as seen in figure 3.10a. The falling edge takes slightly longer, up to 51 ps in figure 3.10b. Detailed results of the rise and fall times, as well as the power consumption, are shown in figure 3.13.

### 3.4.2 Gated NOT-latch

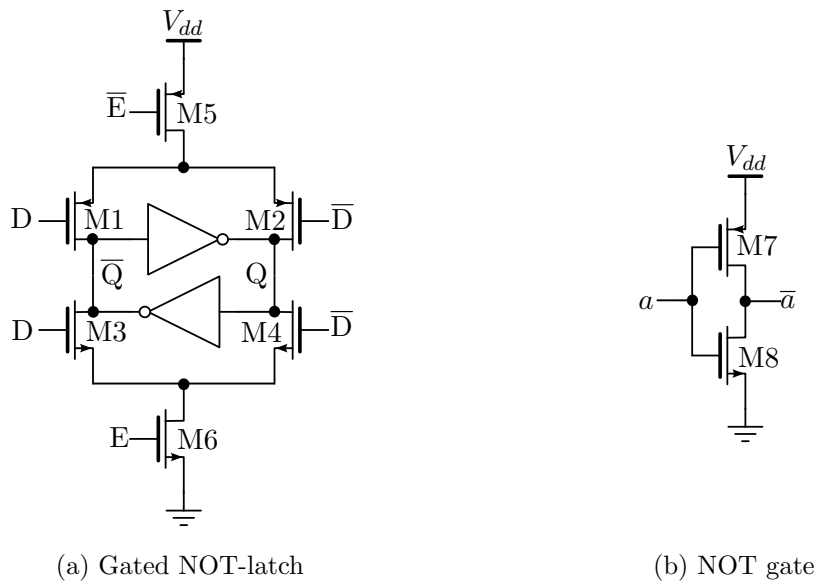


Figure 3.11: Gated NOT-latch transistor level design

	M1	M2	M3	M4	M5	M6	M7	M8
width [nm]	200	200	100	100	200	100	200	100
length [nm]	30	30	30	30	30	30	30	30

Table 3.2: Transistor sizing of NOT-gate

Another possible design for the gated latch is described by Baert and Dehaene in [15] and shown in figure 3.11. This design uses cross-coupled NOT-gates to store the value of the signal. It has been used in a coarse counter by placing the latches in a ring with four elements. This performs a divide-by-four operation and counts two bits in Gray code in each divide-by-four block. The circuit performs exactly as a gated latch and can therefore also be used in the configuration of figure 3.7, which removes the need to decode the 2 bits of the Gray counter. As seen in figure 3.11, both the enable signal and its inverse are necessary, but these are available for the output of the VCO or the output of the previous latch. This design contains 5 PMOS and 5 NMOS transistors, which are 6 transistors less than the 8 PMOS and 8 NMOS transistors required by the gated NAND-latch. Table 3.2 shows the sizing of the transistors in the NOT-gate.

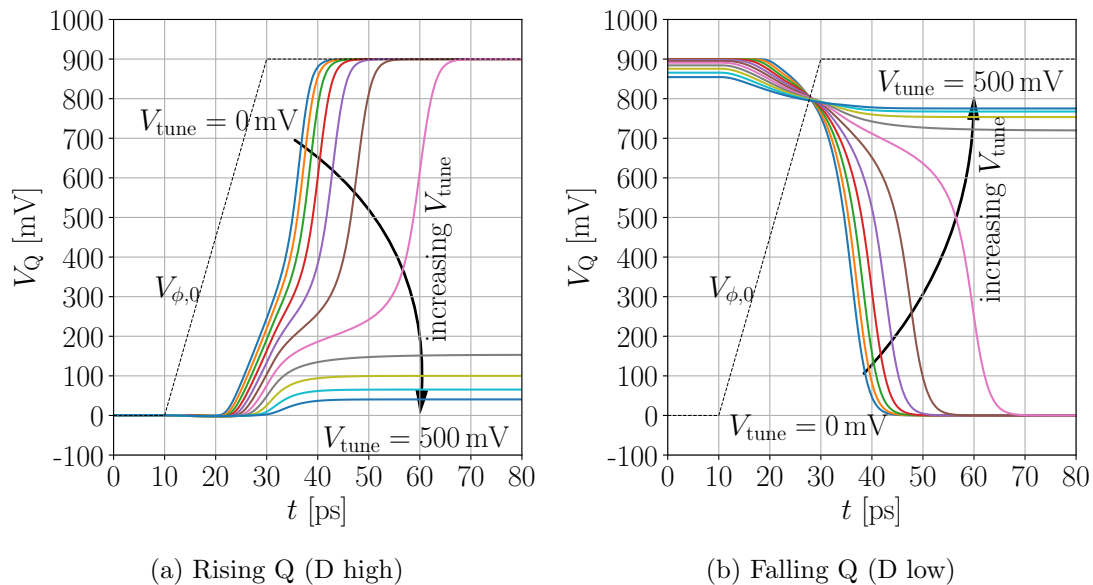


Figure 3.12: Transient response for the gated NOT-latch

The transient response of the rising and falling output has also been simulated for the gated NOT-latch. This results in the plots in figure 3.12. The simulation has been done with the same set of values for  $V_{\text{tune}}$ . It can be seen that the system reacts quite differently to increasing  $V_{\text{tune}}$  compared to the NAND-latch. In figure 3.12, increasing the tuning voltage causes the value of the NOT-latch to only partially rise or fall to the correct value, compared to rising or falling too early for the NAND-latch in figure 3.10. Both of these situations are of course undesired. The NOT-latch reacts faster at lower  $V_{\text{tune}}$  and slower as  $V_{\text{tune}}$  is increased, which is opposite of the behaviour of the NAND-latch.

### 3.4.3 Comparison

The NOT-latch only works up to  $V_{\text{tune}}$  of 300 mV, making it more likely that the system needs a buffer between the VCO and the coarse counter than with the NAND-latch. Together with the maximal value of  $V_{\text{tune}}$  for which the gated latch operates successfully, the following four other metrics can be defined to compare the performance of the NAND-latch and the NOT-latch:

- Rise time  $\tau_r$ : time difference between the beginning of the rising edge on the enable signal and the time instant when the value of the rising output reaches 899 mV.
- Fall time  $\tau_f$ : time difference between the beginning of the falling edge on the enable signal and the time instant when the value of the falling output reaches 1 mV.
- Leakage current  $I_L$ : current consumption of the gated latch when the data input and output are opposite, measured 5 ps before the rising edge of the enable signal.
- Transient charge  $Q_T$ : the integral of the current consumption from the beginning of the rising edge of the enable signal until 100 ps later.

The last two metrics allow us to estimate the total power consumption of the coarse counter. For each bit of the coarse counter, either gated latch a or b is driven by the data signal opposite to its output signal and will therefore leak current. The current due to the transient response of one gated latch in the counter is given by the transient charge multiplied by twice the frequency of the counter output waveform,  $2Q_T f_n$ , for both the rising and falling edge of the counter. The frequency of the first counter is half of the VCO frequency and this frequency halves again for each counter. Counting together the transient powers for all gated latches leads to an estimation of the total average power consumption given in equation 3.3 for the coarse counter.

$$P_{CC} = V_{dd} (N_b I_L + 2Q_T f_{VCO}) \quad (3.3)$$

These four performance indicators for the gated latch are calculated from simulations and plotted in figure 3.13. The plots contain both the results for the gated latch based on the cross-coupled NAND-gates, with the blue crosses, and the cross-coupled NOT-gates, with the red dots. The points have been plotted for different values of  $V_{\text{tune}}$ . Only the values of  $V_{\text{tune}}$  at which the latches function properly, as seen in figures 3.10 and 3.12, are included in the plots.

The rise times are plotted in figure 3.13a. The rise times of the NAND-latch remain fairly constant in function of the tuning voltage, decreasing slightly as  $V_{\text{tune}}$  increases. In comparison, the rise times of the NOT-latch increase with  $V_{\text{tune}}$ . The NAND-gate has a higher  $\tau_r$  than the NOT-gate at low values, but a lower  $\tau_r$  at high values. Looking at the fall times in figure 3.13b, a similar behaviour can be observed. The values of  $\tau_r$  and  $\tau_f$  for the NOT-gate are similar, but the value of  $\tau_f$  for the NAND-gate is about 10 ps lower than the value of  $\tau_r$ . Taking into account that the minimal VCO period  $T_{VCO,\text{min}}$  has to be higher than  $2\tau_r$  and than  $2\tau_f$ , the plotted values allow a maximal VCO frequency of 8 GHz.

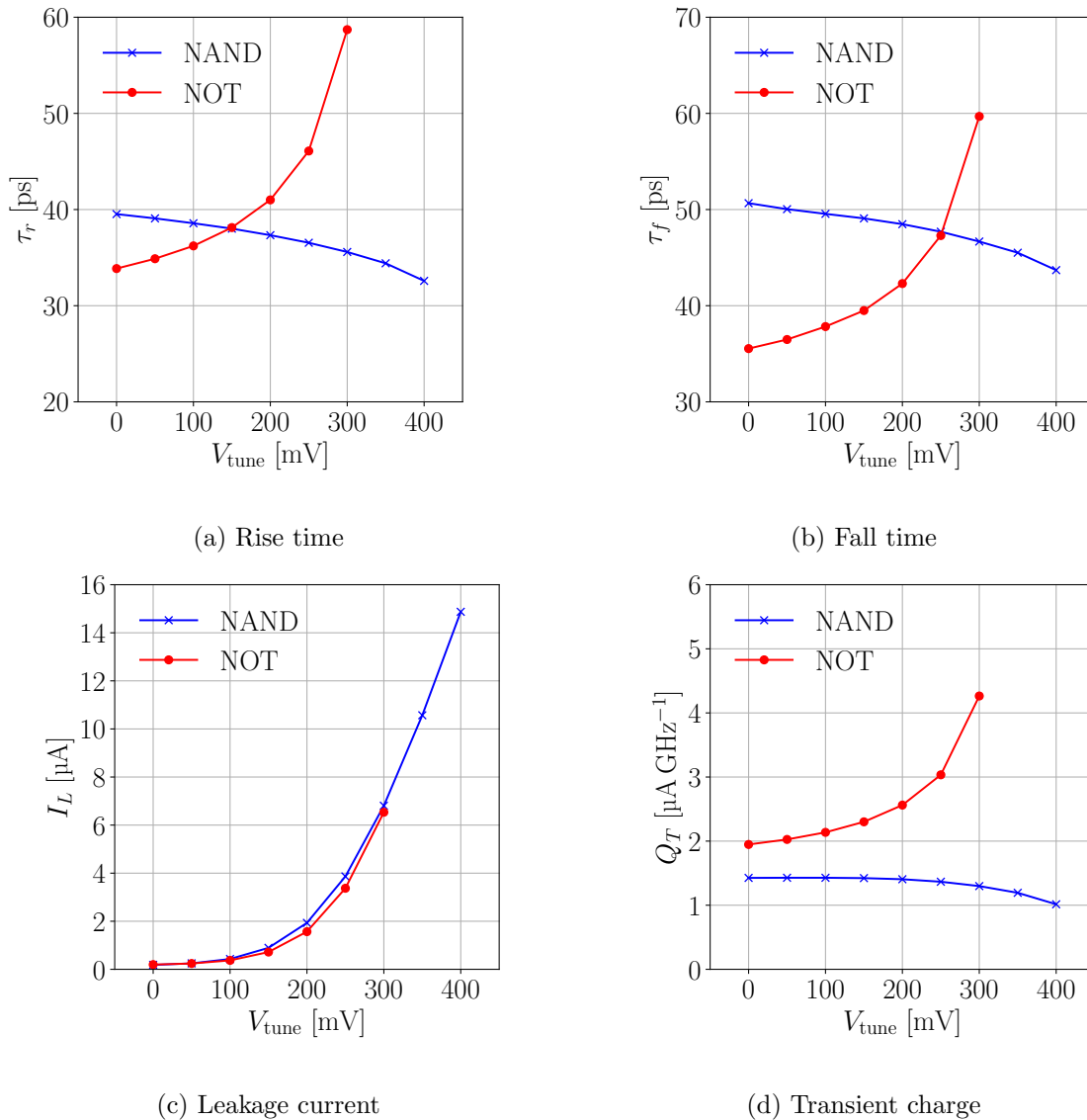


Figure 3.13: Relevant performance metrics for the gated latches

Looking at the current consumption of the two gated latch designs, it can be seen that the leakage current is very similar in the NAND and the NOT latch in figure 3.13c. Both of the leakage currents strongly increase as  $V_{\text{tune}}$  increases, since this increases the subthreshold current in transistors which are expected to be in cutoff.

The transient charge is plotted in figure 3.13d and is expressed in  $\mu\text{A GHz}^{-1}$ , to be able to compare this value to the value of the leakage current. On this plot, a clear difference can be seen between the latches: the NAND-latch consumes significantly less power to transition from one stage to another stage than the NOT-latch. It is expected that the VCO will operate at several GHz, meaning that the current due to the transition will be similar in magnitude or slightly higher than the leakage current.

A more accurate estimation of the total power consumption in the NAND-latch can now be made: the leakage current of the first counter  $I_{L,C_1}$  is dominant over the other leakage currents, so only this counter is included in the first term of equation 3.3, which is rewritten as equation 3.4.

$$P_{CC} = V_{dd}(I_{L,C_1} + 2Q_T f_{VCO}) \quad (3.4)$$

The NAND-latch therefore has two significant advantages to the NOT-latch: a lower power consumption and a wider range of values of the tuning voltage that can be used. Implementing the coarse counter as a double connected binary counter, shown in figure 3.7, and using gated NAND-latches for this design therefore allowed us to design a more power-efficient coarse counter. This double counter will also be resistant to asynchrony as discussed in section 2.4. The rest of the circuit of the coarse-fine VCO-ADC will be designed around this coarse counter.

## Design of the Other Circuit Elements

## 4.1 Overview

The previous chapters provide us with a thorough understanding of the Coarse-Fine VCO-ADC on a system level. This understanding makes it possible to split the design into smaller parts, and identify some important building blocks and the connections between these building blocks. A suitable design for each block is now required. The blocks will then be linked in a full schematic, an important step towards a chip design. The schematic is simulated to produce results which more accurately reflect the true performance of the coarse-fine VCO-ADC. As mentioned in the previous section, the circuits are designed for a 28 nm CMOS technology and all transistors use ulvt flavor.

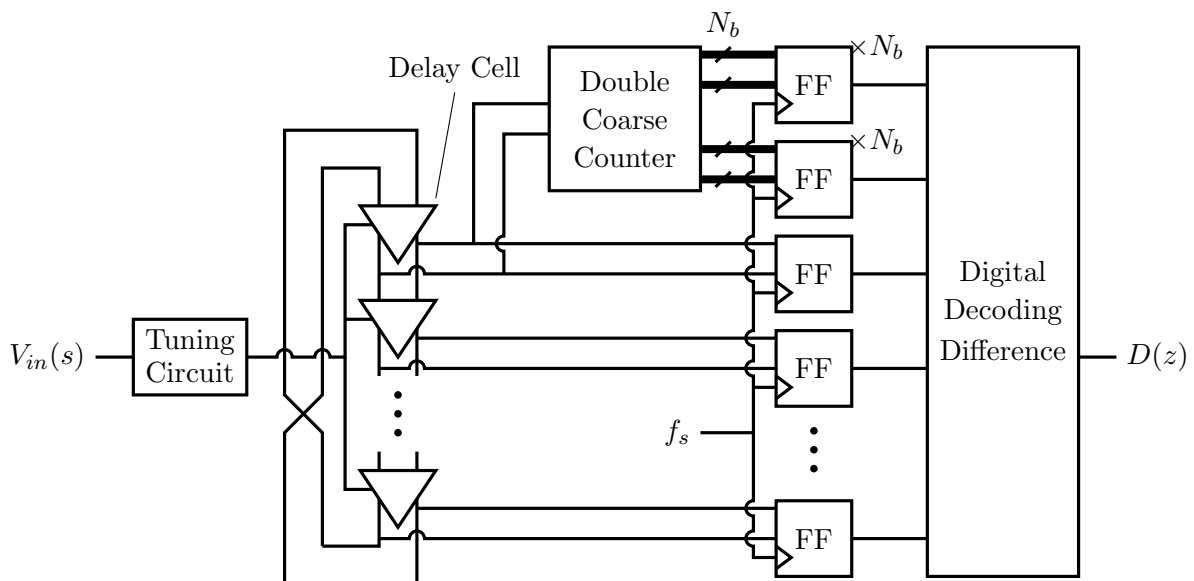


Figure 4.1: Different blocks which remain to be designed



Figure 4.1 therefore gives an overview of the full system. The different named blocks in the figure are the circuits which need to be designed. The double coarse counter was already designed in chapter 3. If necessary, a buffer might be added to this. The other blocks which require a circuit are described in this chapter. The ring oscillator consists of delay cells, controlled by the signal from the tuning circuit. The flipflops sample the input signal at frequency  $f_s$ . The digital circuit, which processes these samples, is described in the hardware description language Verilog and performs the decoding and difference operation which produces the output signal  $D(z)$ .

The performance of each of the designed components will be tested in simulations. For each of the components, the power consumption is an important characteristic. Other important aspects are the speed of the components as seen for example by the settling time, the input and output voltage range, and the spectral density for the thermal and  $1/f$ -noise. Knowing the relevant performance characteristics will allow us to optimize the VCO-ADC successfully for a certain bandwidth and SNR requirement. Throughout this chapter, some important parameters will be identified which will be sized in section 4.7 to minimize the power consumption.

## 4.2 Delay cells

In the expression for the signal-to-quantization-noise ratio derived in section 1.5, it was shown that the SQNR increases by 20 dB/decade with the product of the number of cells and the VCO tuning range,  $N_\phi f_{\text{tune}}$ . This product is inversely proportional to the delay of a single cell. Therefore, designing a fast delay cell is important for the performance of the full VCO-ADC.

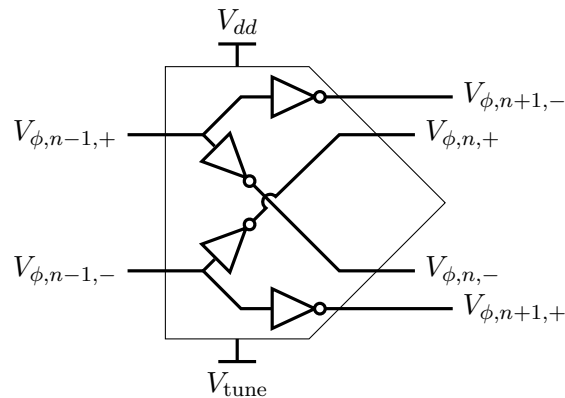


Figure 4.2: Design of the feed-forward delay cell

	NMOS	PMOS
width [nm]	$W_n$	$2W_n$
length [nm]	30	30

Table 4.1: Transistor sizing of the inverters in the feedforward delay cell

In [16], a delay cell which uses feed-forward cross-coupled inverters is described. These delay

cells achieve a low delay and therefore a high frequency without reducing the number of phases, and were applied in a VCO-ADC in [17]. This design is shown in figure 4.2. The two central inverters are crossed to form the main path of the delay cell. To make sure that the ring oscillator operates differentially as a ring of  $N_\phi$  delay cells, auxiliary inverters are required. These are the two outer inverters in figure 4.2. The auxiliary inverters of cell  $n$  are connected to the output of delay cell  $n + 1$ . As derived in detail in [8], the feedforward inverters increase the frequency of the ring. The capacitive load at a cell precharges slightly and is then charged by current from both the auxiliary and the main inverters, causing a lower delay. By placing the delay cells in a ring and correctly crossing both the main signal path and the auxiliary path when the end and the beginning of the loop are connected, the ring oscillator is created.

The delay caused by the delay cell will be tuned by applying the voltage  $V_{\text{tune}}$  to the bottom of the inverters in the delay cell. Over the entire ring, a voltage  $V_{\text{ring}} = V_{dd} - V_{\text{tune}}$  and a related current  $I_{\text{ring}}$  will be observed. At a higher  $I_{\text{ring}}$ , the load capacitance will charge faster and the frequency of the VCO increases. Since this  $V_{\text{tune}}$  is the bottom voltage of the inverters in the delay cell, the low value of the square wave will be equal to  $V_{\text{tune}}$ . Therefore, the amplitude of the square output wave is modulated by the bottom voltage of the delay cell, as shown in figure 4.3. This modulation complicates the design of the flipflops and of the first gated latch in the coarse counter.

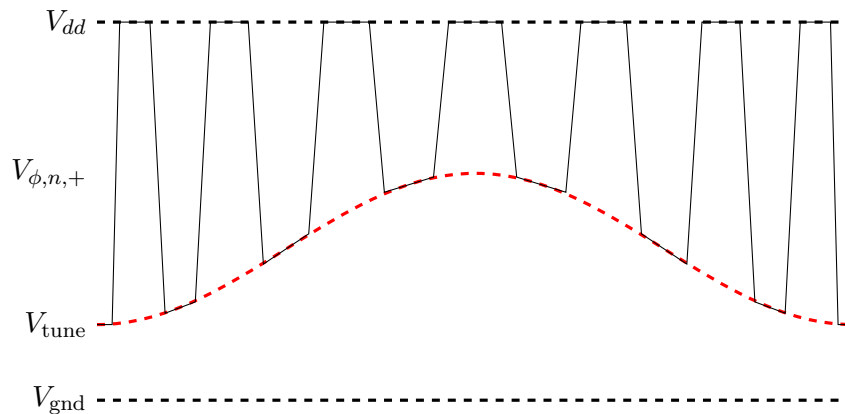


Figure 4.3: Modulation of the VCO output by the tuning voltage

Two other factors also define the ring oscillator VCO: the number of delay cells  $N_\phi$  and the sizing of the inverters. The PMOS transistors in the inverter are again sized to have double the width of the NMOS transistors,  $W_p = 2W_n$ . The lengths of these transistors are minimal at 30 nm. Therefore, the two parameters defining the ring oscillator are  $N_\phi$  and  $W_n$ , which will be optimized in section 4.7. The sizing of these parameters determines the VCO frequency and power consumption. It also determines the input-referred thermal noise and  $1/f$ -noise. The sizing is repeated in table 4.1, in function of  $W_n$ .

The diode model developed by Borgmans et al. [8] is useful to reason on the relation between the different parameters and the ring oscillator characteristics. This model is based on the observation that the relation between the current through the ring oscillator  $I_{\text{ring}}$  and voltage over the ring oscillator  $V_{\text{ring}}$  is similar to the relation for a diode. Figure 4.4a shows this relation for a ring with 16 phases ( $N_\phi = 16$ ) and a  $W_n$  of 3200 nm. Due to the low current up to 300 mV and the increasing slope after this point, the plot is similar to a diode characteristic. Next to this plot, figure 4.4b shows the frequency of this ring oscillator plotted as function of  $V_{\text{ring}}$ . In a simple model, the current increases is independent of the number of phases and increases linearly with the transistor width. As the load capacitance is also linearly proportional to the transistor width, the delay of the cell and therefore the VCO frequency is independent of the transistor width. As identified earlier, the frequency is inversely proportional to the number of delay cells. These relatively simple relations will be used to size the parameters.

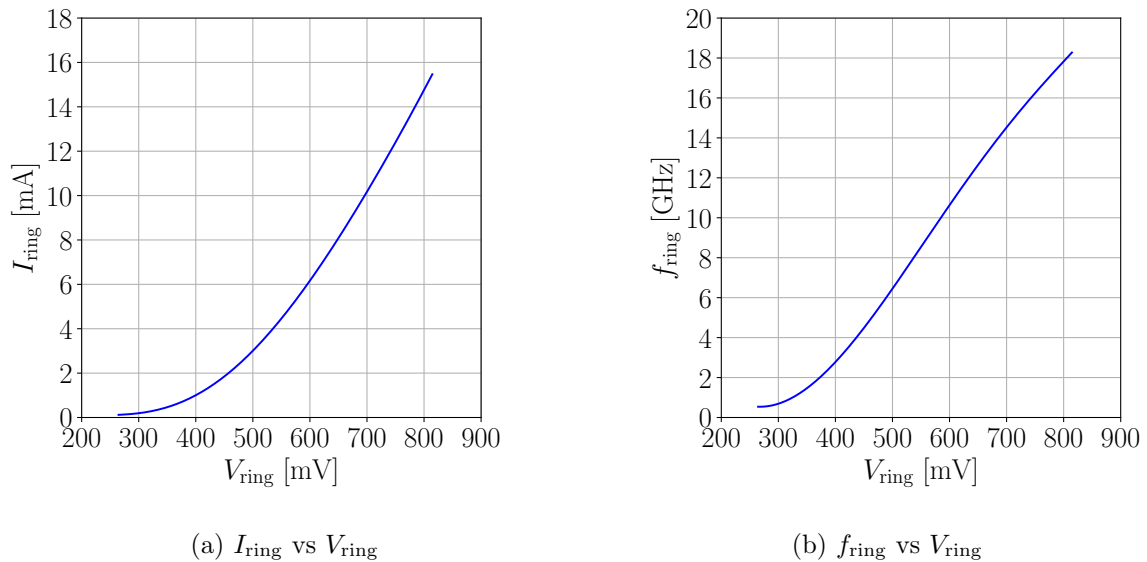


Figure 4.4: Input-output characteristics of the ring oscillator VCO.

The similarity to the behaviour of a diode goes further than the  $I$ - $V$ -characteristic shown in figure 4.4a. Equation 4.1 gives an approximate expression for the spectral density of the white noise of the ring oscillator. The noise is expressed as an equivalent voltage source in series with the ring oscillator. The first part of this expression,  $kT/g_{\text{ring}}$  is similar to the expression for the white noise voltage of a diode. The conductance  $g_{\text{ring}}$  is the derivative of the noise characteristic in figure 4.4a. The factor  $\Gamma_Z$  describes the effect of the impedance of the tuning circuit and takes a value between 0.5 and 1.

$$S_{V_{\text{ring}},w}(f) = \frac{kT}{g_{\text{ring}}} \Gamma_Z \quad (4.1)$$

The  $1/f$ -noise is also described in [8]. This noise decreases with the total channel area of the ring oscillator, and is therefore inversely proportional to the number of phases  $N_\phi$  and to the width of the NMOS transistors  $W_n$ . The expression for the  $1/f$ -noise is given in equation 4.2. The parameters  $K_{fn}$  and  $K_{fp}$  are technology-dependent, as is the oxide capacitance  $C_{ox}$ .

$$S_{V_{\text{ring}},1/f}(f) = \frac{1}{32N_\phi C_{ox}} \left( \frac{K_{fn}}{W_n L_n} + \frac{K_{fp}}{W_p L_p} \right) \frac{1}{f} \Gamma_Z \quad (4.2)$$

The relation of the current noise to  $W_n$  and  $N_\phi$  is not straightforward and depends on the tuning circuit, as changing these parameters changes the input-output characteristics of the VCO. This affects the operating point of the ring oscillator, and therefore also the value of  $g_{\text{ring}}$ . The input-referred thermal noise and its effect on the total SNR of the VCO-ADC will therefore be estimated when the different circuit blocks are combined in section 4.7. A plot of the input-referred voltage noise of the ring oscillator is shown in figure 4.5. The same ring oscillator as for the plots figure 4.4 is used, and an ideal voltage source of 400 mV is applied to tune the ring oscillator. The flat spectrum of the white noise and  $-10$  dB/decade slope of the  $1/f$ -noise are marked as the dashed line and visible in the spectrum, with a smooth transition between these regions.

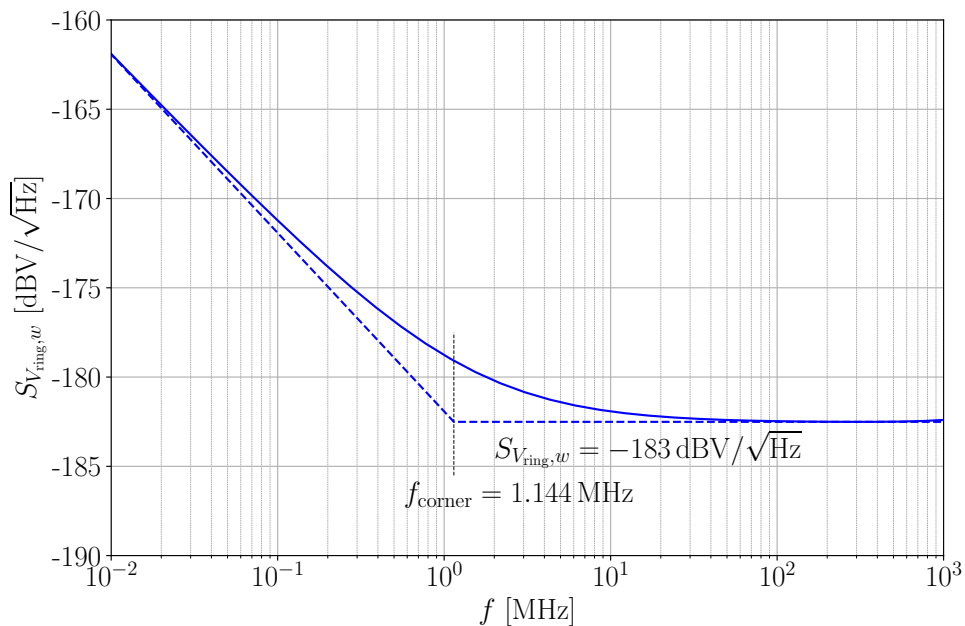


Figure 4.5: Input-referred thermal voltage noise of the ring oscillator

### 4.3 VCO Tuning Circuit

The curve of the frequency  $f_{\text{ring}}$  in function of the voltage  $V_{\text{ring}}$  in figure 4.4b shows clear nonlinear behaviour. To counter this nonlinearity, the tuning circuit shown in figure 4.6 was developed by Babaie-Fishani and Rombouts [18]. This circuit controls the value of  $V_{\text{ring}}$ , and therefore the frequency, by setting  $V_{\text{tune}} = V_{dd} - V_{\text{ring}}$  as a function of the input voltage  $V_{\text{in}}$ .

The relation between  $V_{in}$  and  $V_{tune}$  can be expressed in equation 4.3. It is a function of the ring oscillator characteristic shown in figure 4.4a and the values of the resistors  $R_{conn}$  and  $R_{gnd}$ . Choosing these resistors carefully for the given  $I$ - $V$  characteristic of the ring oscillator enables us to trade off a certain VCO tuning range  $f_{tune}$ , important for the signal strength of the VCO-ADC output, against the linearity and power consumption of the VCO.  $R_{conn}$  and  $R_{gnd}$  are therefore sized by the algorithm in section 4.7.

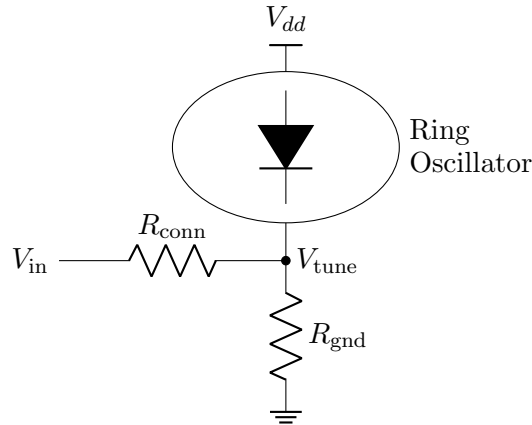


Figure 4.6: Design of the VCO tuning circuit

$$I_{ring} = V_{tune} \left( \frac{1}{R_{conn}} + \frac{1}{R_{gnd}} \right) - \frac{V_{in}}{R_{conn}} \quad (4.3)$$

Equation 4.3 can also be represented graphically, as in figure 4.7. The straight lines are the load lines applied by the tuning circuit at  $V_{in} = 0$  and at  $V_{in} = V_{dd}$ , while the curved line is a sketch of the  $I$ - $V$  characteristic mirrored to express this as a function of  $V_{tune}$ . The input range of  $V_{in}$  is therefore mapped on a smaller range of  $V_{tune}$ , and this range then defines the values of  $f_{ring}$  which can be obtained.

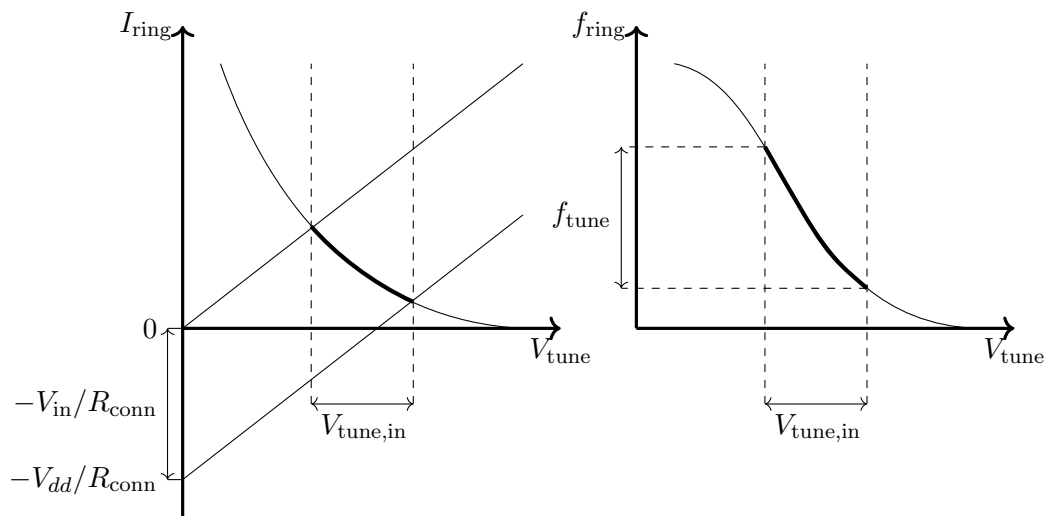


Figure 4.7: Plot of the operation of the VCO tuning circuit

Figure 4.7 sketches how the tuning circuit works. It can be seen that a curve of  $V_{\text{tune}}$  as function of  $V_{\text{in}}$  will be a convex upwards function, as  $V_{\text{tune}}$  increases faster for higher  $V_{\text{in}}$ . The tuning frequency range of interest is mainly at lower frequencies, to limit the power consumption. At these frequencies, the curve of  $f_{\text{ring}}$  against  $V_{\text{tune}}$  is convex downwards. When  $f_{\text{ring}}$  is plotted in function of  $V_{\text{in}}$ , these convex functions will partially cancel, leading to the desired characteristic with higher linearity.

To express the input-referred thermal noise for the VCO-ADC, the thermal noise of the ring oscillator should be transformed by the tuning circuit. The resistors of the tuning circuit also add noise. A detailed analysis of the effect of the tuning circuit on the input-referred thermal noise was done in [19]. In short, the noise transformation can be explained as follows:  $V_{\text{in}}$  contributes to the  $V_{\text{ring}}$  through a voltage divider circuit consisting of  $R_{\text{conn}}$  and  $R_{\text{gnd}}$ , shown in figure 4.6. Transforming this to its Thévenin equivalent places the ring oscillator, its noise, the series resistance and the transformed  $V_{\text{in}}$  all in series. Through the reverse Thévenin operation, a multiplication with  $(R_{\text{conn}} + R_{\text{gnd}})/R_{\text{gnd}}$ , the noise voltage source can then be placed in series with the original input voltage source. Equation 4.4 therefore presents the input-referred thermal noise density for the VCO and its tuning circuit.

$$S_{V_{\text{in,ring}}}(f) = S_{V_{\text{ring}}}(f) \left( \frac{R_{\text{conn}} + R_{\text{gnd}}}{R_{\text{gnd}}} \right)^2 \quad (4.4)$$

The resistors of the tuning circuit also contribute noise to the circuit. Since  $R_{\text{conn}}$  is in series with  $V_{\text{in}}$ , its thermal noise can be directly added to the input voltage. The noise of the resistor  $R_{\text{gnd}}$  can be modelled as a current source and then also transformed to its Thévenin equivalent. The total noise due to the resistors is shown in 4.5. Only the thermal noise by the tuning circuit and the ring oscillator is considered to affect the operation of the VCO-ADC. It is expected that the amplification by the VCO, as seen in the block diagrams in section 1.3 will cause this noise to be dominant over other noise sources.

$$S_{V_{\text{in,R}}}(f) = 4kT \left( R_{\text{conn}} + \frac{R_{\text{conn}}^2}{R_{\text{gnd}}} \right) \quad (4.5)$$

## 4.4 Coarse Counter Buffer

As shown by figure 4.3, the VCO tuning circuit modulates the lower level of the VCO output signal and this signal is therefore not rail-to-rail. A low frequency and a low current can be achieved by decreasing the voltage over the ring oscillator, as visible in figure 4.4. This means that a higher  $V_{\text{tune}}$  should be applied, up to approximately 630 mV. However, in section 3.4 it was shown that the gated latch of figure 3.8 only works up to a tuning voltage of 400 mV. To achieve a greater design freedom when combining and sizing all the circuits, a buffer circuit is designed. This buffer increases the voltage swing of the VCO output phase and applies a square wave with larger amplitude to the coarse counter.

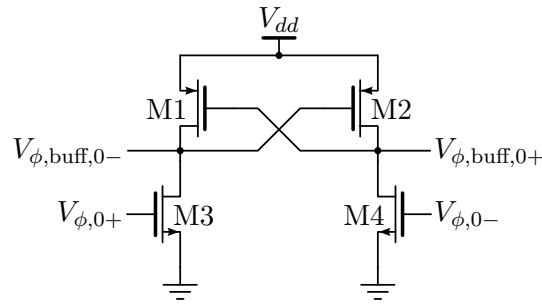


Figure 4.8: Coarse counter buffer circuit

The buffer circuit is shown in figure 4.8. It consists of a common source NMOS pair M3-M4 as input stage and a cross-coupled PMOS pair M1-M2 as output to which the buffered VCO phases are connected. The highest voltage at the input pair will force the voltage at its drain to zero, turning on the PMOS transistor at the opposite side. Since both transistors on one side, for example M2 and M3, are active, a leakage current will flow and the current at the buffer output will be forced to take a higher value. While this might not yet be a rail-to-rail output square wave, the difference between the high and low value will be larger which allows an easier operation of the gated NAND-latch.

Both the NMOS and PMOS transistors are sized minimally at a length of 30 nm and a width of 100 nm to allow the circuit to operate as fast as possible by reducing the parasitic capacitances. The asymmetry of the VCO output signal is now flipped: the low voltage reaches a level close to ground, while the high voltage does not reach its maximal value. This allows us to make a small adaptation to the NAND gates of figure 3.9: the PMOS transistor is also sized minimally now, reducing its width from 200 nm to 100 nm. The speed of the circuit is thereby further increased, while the transition level is lowered. The lower transition level is desired as it is placed more central between the high and low level of the buffer outputs  $V_{\phi,buff,0}$ , making the reduced higher voltage level easily distinguishable. Table 4.2 repeats the sizing of the buffer circuit, while table 4.3 shows the updated sizing of the NAND gate of figure 3.9.

	M1	M2	M3	M4
width [nm]	100	100	100	100
length [nm]	30	30	30	30

Table 4.2: Transistor sizing of the buffer circuit

	M1	M2	M3	M4
width [nm]	100	100	100	100
length [nm]	30	30	30	30

Table 4.3: Updated transistor sizing of the NAND gate

The same metrics for the coarse counter are now applied for a gated latch with its buffer circuit. The first conclusion is that the circuit works between a tuning voltage of 0 mV and 650 mV. The full range of the curve of figure 4.4b can therefore be used. The rise time and fall time increase slightly at lower values of  $V_{\text{tune}}$  and significantly at higher values. The timing constraints are now more strict at the higher  $V_{\text{tune}}$ , which is an advantage since the frequency is higher at a low tuning voltage. Looking at these curves in figure 4.9a and figure 4.9b, a maximal frequency up to 8 GHz can still be achieved without violating the timing constraint  $\tau_b < T_{\text{VCO,min}}/2$  of the first gated latch in equation 3.2.

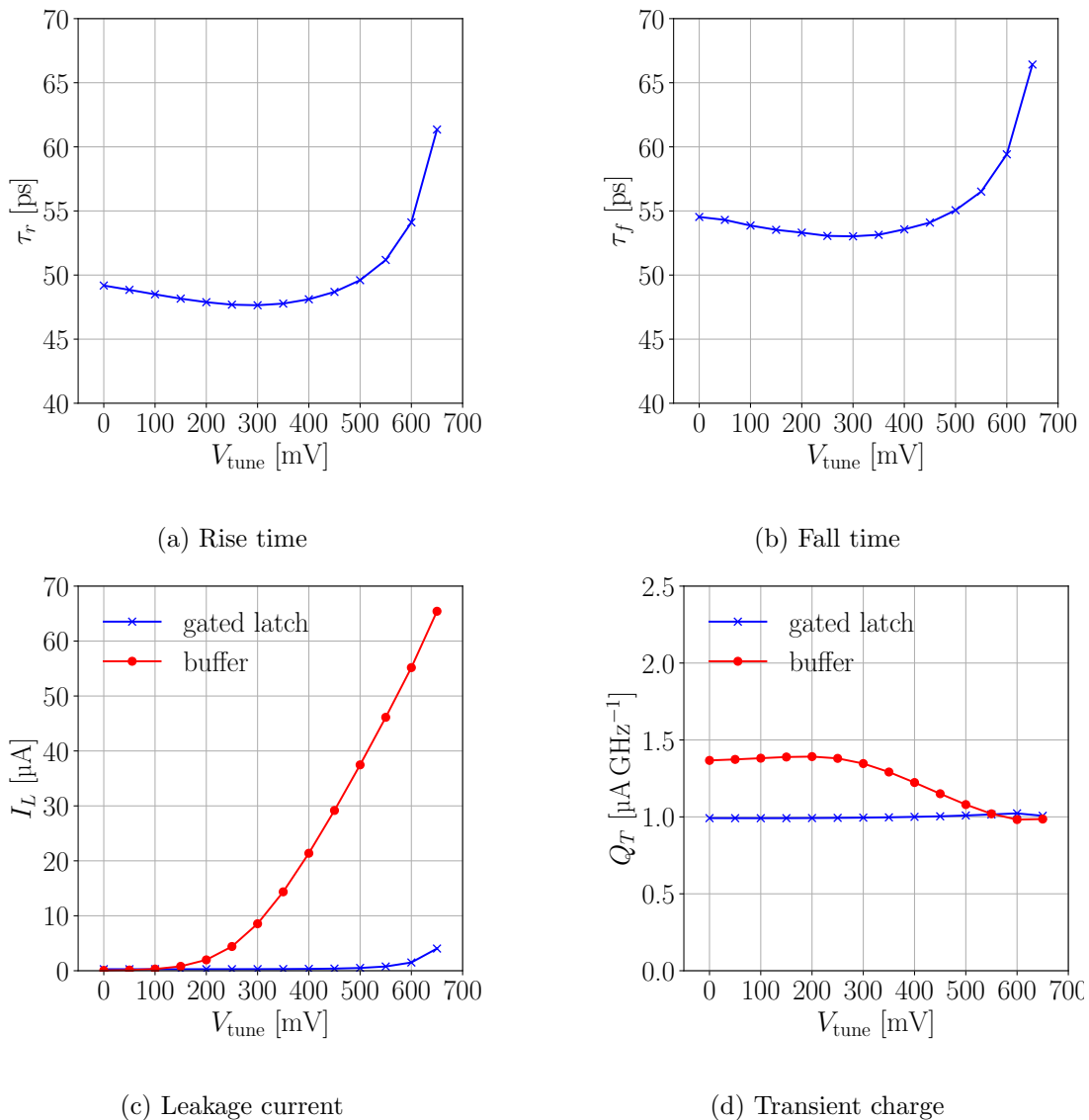


Figure 4.9: Relevant performance metrics for the buffered NAND-latch

The buffer circuit consumes a significant leakage current, up to approximately 66  $\mu\text{A}$ , visible in figure 4.9c. However, several observations show that this leakage current is acceptable in the design. Firstly, it can be noted that the leakage current of the gated latch is significantly reduced





	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
width [nm]	100	100	100	100	100	100	200	100	100	100	100
length [nm]	30	30	30	30	30	30	30	30	30	30	30

Table 4.4: Transistor sizing of the strongARM sense amplifier

	M1	M2	M3	M4
width [nm]	200	200	100	100
length [nm]	30	30	30	30

Table 4.5: Transistor sizing of the NAND gate in the latch following the sense amplifier

The circuit describing the StrongARM sense amplifier is shown in figure 4.10. The circuit operation is explained by considering the roles of the different transistors, as done in [21]. Transistors M1-M4 form a latch out of two cross-coupled inverters. Transistors M8 and M9 reset the inner node of these inverters, and M10 and M11 reset the bottom voltage of the inverters, all to  $V_{dd}$ . This reset phase is done when the clock signal is low, and guarantees that the next sample is not affected by an offset due to the last sample. When the clock signal becomes high, transistors M8-M11 are in cutoff and transistor M7 is active. The differential input pair is also activated now. The output is then generated from the input differential signal in three more phases: sampling, propagation, and regeneration, as defined by Xu and Abidi in [20].

During the sampling phase, transistors M1-M4 are in cutoff. The parasitic capacitors at the drain of the differential pair are fully charged by the reset phase. Due to the input signal, the current through M5 and M6 causes these capacitors to discharge. This discharge happens at slightly different rates due to the differential component of the input signal. After a time  $\tau_s$ , the voltages at the sources of M3 and M4 have fallen and the gate-source voltage exceeds the threshold voltage. These transistors are turned on, which starts the propagation phase. Transistor M3 and M4 are no longer in cutoff. Therefore, the parasitic capacitors at  $V_{out+}$  and  $V_{out-}$  also start discharging. This discharge is initially slow but increases as the gate-source voltage of transistors M3 and M4 also increases. Again, the falling voltage causes the next transistors, PMOS pair M1 and M2, to become active after a time  $\tau_p$  and the final phase begins.

In the regeneration phase, the cross-coupled inverters act on the difference between  $V_{out-}$  and  $V_{out+}$ . This difference has been amplified in the sampling phase and then propagated to the output in the propagation phase. The transfer of the differential voltage to the output stage can be modelled as an exponential charging of the load capacitor  $C_L$ , with time constant  $\tau_{reg} = C_L/G_M$ .  $G_M$  is the transconductance in the cross-coupled inverters. Its value of changes over time: initially, only the PMOS transistor pair M1-M2 contribute significantly to the regeneration. M3 and M4 later increase  $G_M$ , degenerated by M5 and M6 acting in the linear region. The differential output voltage rises exponentially, until this exponential is limited by the ground and supply voltage and the output signals settle to a low and a high digital output signal.

The StrongARM sense amplifier mainly consumes power by the charging of the load capacitances during the reset phase. Therefore, to size it for low power consumption, the parasitic capacitances are minimized. Transistors M8-M11 are used as digital switches and should have enough time to charge the necessary nodes so are sized minimally at a length of 30 nm and a width of 100 nm. Transistors M1-M4 are also sized minimally, to minimize the time constant of regeneration by having a high transconductance for a low capacitive load. The minimal sizing is also used for the input differential pair M5 and M6, again limiting the parasitic capacitance, the load on the VCO-ADC, and possible kickback due to the activation of M7 on the rising clock edge. Finally, the width of M7 is double that of M5-M6, 200 nm, to make sure that the current through the differential pair is not limited by the tail current. Table 4.4 summarizes this sizing.

A simulated transition of the output of the StrongARM sense amplifier is shown in figure 4.11. In this figure, the output and the rising edge of the clock partially overlap. The inputs  $V_{in,+}$  and  $V_{in,-}$  are also realistically modelled as outputs of the ring oscillator, with a bottom voltage at 550 mV and an exponential edge. These factors complicate the operation of the StrongARM and slow down the transition. The outputs  $V_{out,+}$  and  $V_{out,-}$  initially both fall, but a difference between the outputs is already forming. This differential voltage increases significantly once the PMOS transistors are activated, causing the outputs to take a high and a low value. The transition in the strongARM is approximately completed less than 80 ps after the start of the rising clock edge.

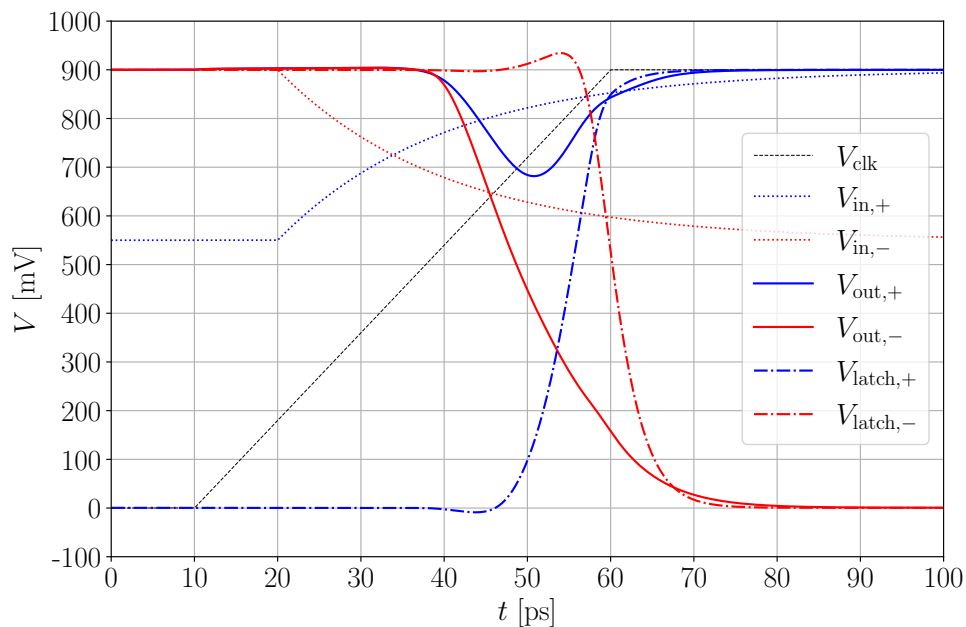


Figure 4.11: Transition waveforms of a StrongARM sense amplifier

A simulated transition of the output of the StrongARM sense amplifier is shown in figure 4.11. In this figure, the output and the rising edge of the clock partially overlap. The inputs  $V_{in,+}$  and  $V_{in,-}$  are also realistically modelled as outputs of the ring oscillator, with a bottom voltage at 550 mV and an exponential edge. These factors complicate the operation of the StrongARM and slow down the transition. The outputs  $V_{out,+}$  and  $V_{out,-}$  initially both fall, but a difference between the outputs is already forming. This differential voltage increases significantly once the PMOS transistors are activated, causing the outputs to take a high and a low value. The transition in the strongARM is approximately completed less than 80 ps after the start of the rising clock edge.

The digital output of the strongARM sense amplifier needs to be latched between samples, to make sure it is available during the reset phase. A NAND-latch is used, since this latch retains its value when both of its inputs are high. This NAND-latch is similar to the latch used in the coarse counter, but it is not gated. On figure 4.11, the transition of the output of the latch is also shown as the waveforms of  $V_{latch,+}$  and  $V_{latch,-}$ . It can be seen that this reacts very fast once the voltage  $V_{out,-}$  has decreased significantly, even settling faster than the output of the sense amplifier. The sizing of the NAND gates in this latch is shown in table 4.5

The sense amplifier does not consume a significant static current. Therefore, the average current consumption depends on the clock frequency and will be expressed in  $\mu\text{A GHz}^{-1}$ . Three situations can be distinguished in simulations. On the falling clock edge, the nodes recharge, requiring a charge of  $1.2 \mu\text{A GHz}^{-1}$ . On the rising clock edge, the current depends on whether the latched value has to switch or not. A charge of approximately  $1 \mu\text{A GHz}^{-1}$  is observed when the latch switches and approximately  $400 \text{ nA GHz}^{-1}$  when the output of the latch does not change. Assuming that the latch has an equal probability of switching or not switching at any sample, the current consumption of the sense amplifier and latch can be estimated as  $1.9 \mu\text{A GHz}^{-1} f_s$ .

Ideally, the sense amplifier would always give a certain, reliable output voltage before a certain transition time  $\tau_{SA}$ . However, the differential voltage at the start of the regeneration phase can be arbitrarily close to zero. If it is too close to zero, the output can be metastable, and this metastability can propagate through the latch into the digital circuit. The noise due to metastability will be well below the target SNR values of the coarse-fine VCO-ADC if the probability of a metastable output is below  $10^{-15}$  at  $\tau_{SA}$ .

The transition time will be given by  $\tau_{SA} = \tau_s + \tau_p + n_{reg}\tau_{reg}$ . The important parameter for metastability is the number of time constant  $n_{reg}$ . This value needs to be larger than 35 for the desired metastability probability. The other values can be obtained from simulations. A very loose upper bound for the time constant which defines the regeneration phase can be found as  $\tau_{reg} = C_L/g_{m1,2}$ , taking the rough assumption that only the pair M1-M2 contributes to regeneration. Depending on the direction of the transition, either  $g_{m1}$  or  $g_{m2}$  will reach a maximum, and this maximum is used to determine  $\tau_{reg}$ , while the time of this maximum determines  $\tau_s + \tau_p$ . A voltage of 450 mV is applied to both inputs of the sense amplifier, sized as described above.

This gives a  $g_m$  of  $70.9 \mu\text{S}$  after  $60 \text{ ps}$  and a load capacitance of  $408 \text{ aF}$ . A time constant  $\tau_{reg}$  of  $5.76 \text{ ps}$  is obtained, and the required transition time  $\tau_{SA}$  is therefore  $262 \text{ ps}$ . The NAND-latch will not add a significant additional delay to this transition time, as this will react very quickly to a variation in the output of the sense amplifier, as seen in figure 4.11.

The value of  $\tau_{SA}$  is important as this will reduce the available propagation time in the first part of the digital circuit. The actual value is expected to be significantly lower, as the actual  $g_m$  will be higher due to the effect of the NMOS pair. However, this gives us a necessary margin to use as an input when synthesizing the digital circuit.

## 4.6 Digital Design

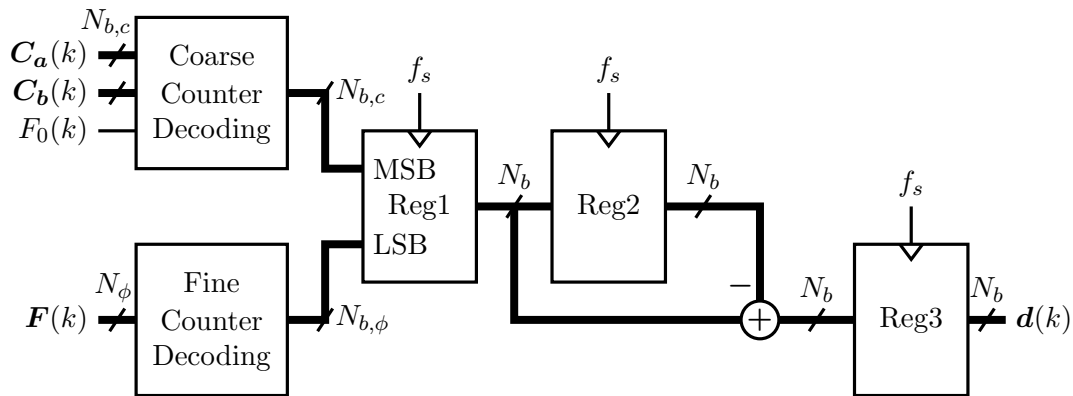


Figure 4.12: Digital blocks to determine the ADC output

The digital values at the output of the flipflops do not directly provide the ADC output. The fine and coarse counter output need to be combined to a binary value, and the difference between the value of the previous sample needs to be subtracted from the current sample. The digital chain used to calculate the ADC output is shown in figure 4.12. The fine and coarse counter are decoded separately, and three registers are used to apply the difference operation and store the output value. The different blocks in figure 4.12 are implemented separately as Verilog code and the different modules are then combined in a single top-level design. The full Verilog description which implements these different modules as well as the top module is listed in appendix B.

The decoder of the coarse counter has the  $2N_{b,c}$  coarse counter bits of both halves of the connected double counter from section 3.3 as inputs, as well as the fine counter bit sampled from  $V_{\phi,0}$ . The decoder for the coarse counter is implemented as a multiplexing operation using `case`-statements. As described in section 3.3, the fine counter bit selects the first bit of the coarse counter output, this first bit selects the second bit, and this process continues until all bits are selected. This leads to an output of  $N_{b,c}$  bits of the coarse counter decoder, which become the most significant bits (MSB) stored in the first register.

The decoder of the fine counter is implemented using a lookup table (LUT). The XNOR-scheme in figure 2.1 is not implemented directly as freedom is given to the compiler to find a more efficient implementation. The  $N_\phi$  input bits can only take  $2N_\phi$  different values: either a falling edge or a falling edge will propagate through the ring oscillator. The former will be sampled as a series of ones followed by zeros, and the latter as a series of zeros followed by ones. These  $2N_\phi$  different input bit vectors therefore result in a fine counter value from 0 to  $2N_\phi$ , and are implemented in the lookup table. All the other possible input bit vectors cannot occur when the ring oscillator is sampled properly and a default case with ‘don’t care’ (symbol  $\mathbf{x}$ ) as output takes care of these inputs. The fine counter has an output of  $N_{b,\phi} = 1 + \log_2 N_\phi$  bits. The number of phases  $N_\phi$  is always chosen as a power of 2. Otherwise, the decoding of the coarse and fine counter into a binary value needs to be combined and becomes more complicated. The fine counter output is stored as the least significant bits (LSB) in the first register.

The three registers and the difference operation are implemented as simple as possible and operate on values represented by  $N_b = N_{b,c} + N_{b,\phi}$ . The first register is not strictly necessary but splitting the decoding and the difference operation relaxes the timing constraint of the digital circuit. The second register is necessary to have the previous value available for the difference operation, and the third register stores the value for the ADC output.

The digital circuit is synthesized for a 16-phase VCO-ADC, with 3 coarse bits. The timing and power consumption of this circuit are of great interest, as these measures will be essential to successfully combine the circuits. The predicted maximal time for a combinatorial path in the circuit is found to be 348 ps, obtained for the calculation of the fine counter value. A large part of this delay is the 262 ps found as the delay of the sense amplifier in 4.5. Adding an uncertainty of 120 ps and setup time of 16 ps to this results in a total longest path delay of 484 ns, including the input delay. This suggests that the digital circuit can be designed up to a sampling frequency of 1.5 GHz without having to pipeline calculations by adding registers in the decoding of the coarse or fine counter. This is an important result: when designing the coarse-fine VCO-ADC, the clock frequency can be chosen independently from the VCO frequency. Therefore, the power consumption can be greatly reduced by choosing a sampling frequency which is low enough to avoid pipelining in the calculations. Pipelining is power-hungry because it requires additional registers to store intermediate results. The complexity of the digital design also increases. For a 1 GHz sampling frequency, the digital circuit consumes an estimated power consumption of only 168  $\mu\text{W}$ , which is significantly lower than the predicted power consumption in the VCO when looking at figure 4.4a. The power consumption is expected to be proportional to  $f_s$  and this factor also affects the power consumption in the sense amplifiers. Therefore,  $f_s$  is another parameter which can be optimized when connecting the different circuits.

## 4.7 Combining the Circuits

### 4.7.1 Design Space and Iteration Variables

Throughout the extensive exploration of the different blocks of the circuit, the power consumption of the circuits and their effect on other important design factors such as noise or timing is discussed and determined. This allows us to define a method and write an algorithm to size the parameters. The circuits designed in this chapter are connected as described in section 4.1 and figure 4.1 to obtain a full VCO-ADC design. This VCO-ADC has to meet certain specifications. A bandwidth  $f_{BW}$  is defined for the analog input signal. This is necessary to determine the SQNR. Combining the SQNR with the input-referred thermal noise  $SNR_{in,T}$  allows us to determine the SNR, which is also used as a specification for the VCO-ADC. Finally, the linearity of the VCO-ADC is considered. Linearity can be expressed by the total harmonic distortion (THD), which determines the signal-to-noise and distortion ratio (SNDR) together with the SNR. In this design, the THD is taken into account by designing the VCO-ADC for a certain third order harmonic distortion (HD3). The VCO-ADC will be applied in a pseudo-differential configuration in section 5.2, which will remove the even harmonics from the spectrum. Of the remaining harmonics, we expect the third harmonic to be dominant. Hence we will specify a certain target HD3. The goal of this section is to find a design method for the VCO-ADC which achieves these specifications using a minimal power consumption. The resulting algorithm is summarized in algorithm 1, and a full implementation in Python is shown in appendix C.

To achieve the desired design, several parameters should be set to an appropriate value. The resistors  $R_{conn}$  and  $R_{gnd}$  of the tuning circuit are not determined yet. The number of delay cells  $N_\phi$  and NMOS width  $W_n$  are required to fully define the ring oscillator. Finally, the clock frequency  $f_s$  is not fixed yet, and this also defines the number of coarse counter bits  $N_{b,c}$  through equation 1.17.

These five parameters give us a very large design space. However, it immediately becomes clear that the design space is limited by several restrictions. Figure 3.10b in section 3.3 shows that the maximal fall time of the coarse counter is slightly more than 50 ps, limiting the maximal VCO frequency to  $f_{max} = 8$  GHz. In the previous section, it was shown that the sampling frequency is also limited to a maximum of  $f_{s,max} = 1.5$  GHz, and due to the decay of the transfer function shown in figure 1.7, a lower limit  $f_{s,min} = 10f_{BW}$  is also applied. To be able to practically simulate the ring oscillator and to restrict the VCO area, the number of delay cells  $N_\phi$  is limited to 32. These restrictions already places an important maximum on the achievable SQNR: based on equation 1.15, the SQNR for a 40 MHz bandwidth is limited to 85.4 dB.

The optimization of the power consumption is based on a combination of iterations and calculations, using the simulated results presented in earlier sections. The values presented in graphs and data are extrapolated through their derived relations to the different parameters. At the same time, care is taken to avoid iterating over some irrelevant points, leading to a fast result. The first step is therefore to use the presented limit on the SQNR: this allows us to identify

the combinations of  $N_\phi$  and  $f_{\text{tune}}$  for which the SQNR exceeds the desired SNR, when  $f_s$  is set to its maximal value.  $N_\phi$  now only has a small number of possible values left and becomes the first iteration variable. For each  $N_\phi$ , the minimal value of  $f_{\text{tune}}$  makes it possible to iterate over  $f_{\text{VCO,min}}$  up to a value of  $f_{\text{max}} - f_{\text{tune}}$  and  $f_{\text{VCO,max}}$  which is greater than  $f_{\text{VCO,min}} + f_{\text{tune}}$ . This step is described as line 2 in algorithm 1.

---

**Algorithm 1** Optimization of power consumption based on data and given specifications

---

**Specifications:**  $\text{SNR}_{\text{target}}, \text{HD3}_{\text{target}}, f_{\text{BW}}$

```

1:  $f_{\text{ring}}, I_{\text{ring}}, V_{\text{tune}}, S_{V_{\text{ring}}}(f), f_c, P_{\text{dig},0}, \dots \leftarrow \text{data}$ 
2: for  $N_\phi, f_{\text{VCO,max}}, f_{\text{VCO,min}} : \text{SQNR}(N_\phi, f_{\text{VCO,max}}, f_{\text{VCO,min}}, f_s, \text{max}) > \text{SNR}_{\text{target}}$  do
3:   determine  $R_{\text{conn},0}, R_{\text{gnd},0}$  ▷ Equation 4.3
4:   determine curve of  $V_{\text{in}}$  against  $f_{\text{ring}}$  ▷ Equation 4.3 and figure 4.4
5:   determine HD3 ▷ Equations 4.8 and 4.9
6:   if  $\text{HD3} < \text{HD3}_{\text{target}}$  then
7:     determine  $V_{n,\text{in},T,0}^2$  ▷ Equations 4.11-4.13
8:     determine  $W_{n,\text{min}}$  and  $W_{n,\text{max}}$  ▷ Equations 4.14 - 4.20
9:     for  $W_n$  multiple of 100 nm in  $[W_{n,\text{min}}, W_{n,\text{max}}]$  do ▷ vectorized calculation
10:      determine  $f_s$  ▷ Equation 4.14
11:      determine  $P_{\text{tot}}$  ▷ Equation 4.21-4.24
12:      if  $P_{\text{tot}} < P_{\text{tot,min}}$  then
13:         $P_{\text{tot,min}} \leftarrow P_{\text{tot}}$ 
14:        store all variables
15:      end if
16:    end for
17:  end if
18: end for

```

---

### 4.7.2 VCO Frequency Characteristic and Distortion

At this point, the previously presented data are needed. The curves in figure 4.4 give a set of points  $(f_{\text{ring}}, V_{\text{tune}}, I_{\text{ring}})$  for  $N_{\phi,0} = 16$  and  $W_{n,0} = 3200$  nm. The parameterized sets of points  $(f_{\text{ring}}N_{\phi,0}/N_\phi, V_{\text{tune}}, I_{\text{ring}}W_n/W_{n,0})$  are therefore obtained. Using these points for  $f_{\text{VCO,min}}$  and  $f_{\text{VCO,max}}$  gives us the corresponding value of  $V_{\text{tune}}$  and  $I_{\text{ring}}$  at the maximum and minimum values of  $V_{\text{in}}$ . Plugging these into equation 4.3,  $R_{\text{conn},0}$  and  $R_{\text{gnd},0}$  are obtained at a width of  $W_{n,0} = 3200$  nm. These quantities will be inversely proportional to the width of the NMOS transistor, as can be seen from equation 4.3. Expressing all quantities at the original width  $W_{n,0}$  allows us to apply impedance scaling when the noise is calculated, to trade-off  $\text{SNR}_{\text{in},T}$  against the SQNR. Using the linear section between the endpoints of the tuning range on the curve of  $I_{\text{ring}}$  vs.  $V_{\text{ring}}$ , a good estimation for the value of  $g_{\text{ring},0}$  is obtained.

Using  $R_{\text{conn},0}$  and  $R_{\text{gnd},0}$  then allows us to determine the value of  $V_{\text{in}}$  at every point of the used section of the  $I$ - $V$  curve of the ring, marked in figure 4.7, by expressing equation 4.3 in function



of  $V_{\text{in}}$ . It can be seen that all dependencies on  $W_n$  drop out of this equation: we derive a curve of  $V_{\text{in}}$  against  $f_{\text{VCO}} = f_{\text{ring}}$ , independent of  $W_n$ . The notation  $f_{\text{VCO}}$  indicates the section of the curve of  $f_{\text{ring}}$  in function of  $V_{\text{in}}$  which can be accessed by tuning. This input-output curve of the VCO with its tuning circuit will determine the linearity of the VCO-ADC.

The linearity of the VCO-ADC is important as it determines the HD3, one of the specifications. The value of HD3 is estimated by assuming that  $f_{\text{VCO}}$  can be expressed as a polynomial function of  $V_{\text{in}}$  around its midpoint, so in function of  $\Delta V_{\text{in}} = V_{\text{in}} - V_{dd}/2$ . This is shown in equation 4.7, with the terms higher than fourth order grouped in the last term.

$$f_{\text{VCO}} = \alpha_0 + \alpha_1 \Delta V_{\text{in}} + \alpha_2 \Delta V_{\text{in}}^2 + \alpha_3 \Delta V_{\text{in}}^3 + \alpha_4 \Delta V_{\text{in}}^4 + \mathcal{O}(\Delta V_{\text{in}}^5) \quad (4.7)$$

We are only interested in the values of  $\alpha_1$  and  $\alpha_3$ , odd terms of the polynomial  $f_{\text{VCO}}$ . To estimate these values, the first and third order Legendre polynomial  $L_1$  and  $L_3$  are used. The function  $f_{\text{VCO}}$  is rescaled to the domain  $[-1, 1]$ , multiplied by the polynomial, and integrated over its domain. This gives the coefficients  $a_1$  and  $a_3$ , which can be used to express an estimation of the polynomial function,  $\hat{f}_{\text{VCO,odd}}$ , shown in equation 4.8.

$$\begin{aligned} \hat{f}_{\text{VCO,odd}} &= a_1 L_1 \left( \frac{2\Delta V_{\text{in}}}{V_{dd}} \right) + a_3 L_3 \left( \frac{2\Delta V_{\text{in}}}{V_{dd}} \right) \\ &= a_1 \frac{2\Delta V_{\text{in}}}{V_{dd}} - a_3 \frac{3}{2} \frac{2\Delta V_{\text{in}}}{V_{dd}} + a_3 \frac{5}{2} \left( \frac{2\Delta V_{\text{in}}}{V_{dd}} \right)^3 \\ &= \hat{\alpha}_1 \Delta V_{\text{in}} + \hat{\alpha}_3 \Delta V_{\text{in}}^3 \end{aligned} \quad (4.8)$$

It is clear by observing the second and third line of equation 4.8 that  $\hat{\alpha}_1$  and  $\hat{\alpha}_3$  can be calculated from the values of  $a_1$  and  $a_3$ . The response to a sine wave input of the VCO,  $\Delta V_{\text{in}} = V_{dd}/2 \sin \omega t$  in  $\hat{f}_{\text{VCO,odd}}$ , can be used to determine the ratio of the magnitude of the third harmonic to the fundamental frequency. This is calculated as HD3 in equation 4.9. When the HD3 is higher than the desired value, the current selection of iteration variables cannot lead to a successful design and the next iteration point is considered, as shown by line 5 and 6 in algorithm 1.

$$\text{HD3} = \left| \frac{\frac{1}{4}(V_{dd}/2)^2 \hat{\alpha}_3}{\hat{\alpha}_1 + \frac{3}{4}(V_{dd}/2)^2 \hat{\alpha}_3} \right| \quad (4.9)$$

### 4.7.3 Impedance Scaling and SNR

Besides a specification on the maximal HD3, a minimal value for SNR is also specified. As equation 4.10 shows, the SNR increases if either the input-referred thermal SNR or the SQNR increases. By impedance scaling, the value of  $\text{SNR}_{\text{in},T}$  is proportional to  $W_n$ . The SQNR is proportional to  $f_s$ , as shown in section 1.5. These two design variables which are yet undetermined can therefore be linked.  $W_n$  can only take discrete values as it is sized as a multiple of the minimal finger width 100 nm. Therefore,  $f_s$  can be expressed in function of  $W_n$  based on the desired SNR. The power will be minimized for a vector of values for  $W_n$  within an upper

and lower bound  $W_{n,\min}$  and  $W_{n,\max}$ . To determine these bounds, the thermal noise will first be calculated for the reference case of  $W_{n,0}$ .

$$\frac{1}{\text{SNR}} = \frac{1}{\text{SQNR}} + \frac{1}{\text{SNR}_{\text{in},T}} \quad (4.10)$$

In section 4.3, the thermal noise at the input of the VCO-ADC was derived. Equation 4.4 expresses how the noise density from the ring oscillator is transformed by the tuning circuit, referring to equation 4.1 and 4.2. This allows us to determine the following proportionalities for the white noise density from the ring (equation 4.11),  $1/f$ -noise density (equation 4.12), and white noise density from the resistors (equation 4.13).

$$S_{V_{\text{in},\text{ring},w}}(f) \propto \frac{1}{g_{\text{ring}}} \left( \frac{R_{\text{conn}} + R_{\text{gnd}}}{R_{\text{gnd}}} \right)^2 \quad (4.11)$$

$$S_{V_{\text{in},1/f,w}}(f) \propto \frac{1}{W_n N_\phi} \left( \frac{R_{\text{conn}} + R_{\text{gnd}}}{R_{\text{gnd}}} \right)^2 \frac{1}{f} \quad (4.12)$$

$$S_{V_{\text{in},R,w}}(f) \propto R_{\text{conn}} + \frac{R_{\text{conn}}^2}{R_{\text{gnd}}} \quad (4.13)$$

The known values in 4.5 and the value of the Boltzmann constant allow us to determine these noise spectral densities  $S_0(f)$  at our obtained values of  $R_{\text{conn},0}$ ,  $R_{\text{gnd},0}$ ,  $g_{\text{ring},0}$  and  $N_\phi$ . To integrate this over the bandwidth, a minimal bandwidth value  $f_{\text{BW},\min}$  is required for the  $1/f$ -noise as an input of the algorithm.  $f_{\text{BW},\min}$  is set to 10 kHz for all further calculations. Integrating and adding the different noise sources gives us the square of thermal noise voltage, denoted as  $V_{n,\text{in},T,0}^2$ . The ratios  $f_s/f_{s,\max}$  and  $W_n/W_{n,0}$  can be related through equation 4.14 using the specified  $\text{SNR}_{\text{target}}$ .

$$\frac{f_{s,\max}}{\text{SQNR}_{\text{max}} f_s} = \frac{1}{\text{SNR}_{\text{target}}} - \frac{V_{n,\text{in},T,0}^2 W_{n,0}}{\frac{1}{2}(V_{dd}/2)^2 W_n} \quad (4.14)$$

Setting  $f_s = f_{s,\max}$  gives a lower bound  $W_{n,L}$  on the transistor width in equation 4.15. This lower bound is always valid as the values of  $f_{\text{VCO},\min}$ ,  $f_{\text{VCO},\max}$ , and  $N_\phi$  in the current iteration are set to have  $\text{SQNR}_{\text{max}} > \text{SNR}_{\text{target}}$ . The width of the NMOS transistors also affects the mismatch performance, as discussed in [22]. Therefore, a minimal width of 800 nm is required. The highest of these lower bounds will be set as lower bound on  $W_n$ , as shown in 4.16.

$$W_{n,L} = \frac{\text{SNR}_{\text{target}} \text{SQNR}_{\text{max}}}{\text{SQNR}_{\text{max}} - \text{SNR}_{\text{target}}} \frac{V_{n,\text{in},T,0}^2 W_{n,0}}{\frac{1}{2}(V_{dd}/2)^2} \quad (4.15)$$

$$W_{n,\min} = \max \left( 800 \text{ nm}, 100 \text{ nm} \left\lceil \frac{W_{n,L}}{100 \text{ nm}} \right\rceil \right) \quad (4.16)$$

To determine an upper bound on  $W_n$ , two cases can be distinguished. If the minimal sampling frequency  $f_{s,\min}$  achieves an SQNR larger than the required SNR, an upper bound  $W_{n,U1}$  is calculated from  $f_{s,\min}$  using equation 4.17. A different upper bound is found by considering that the value of  $W_n$  can go to infinity while  $f_s$  decreases asymptotically to  $f_{s,\infty} =$

$f_{s,\max}\text{SNR}_{\text{target}}/\text{SQNR}_{\text{max}}$ . With this value, a maximal  $N_{b,c,\max}$  can be associated, calculated as  $N_{b,c,\max} = \lceil \log_2(f_{\text{VCO},\max}/f_{s,\infty}) \rceil$ , based on equation 1.17. When the total power is written down and  $N_{b,c}$  is replaced by  $N_{b,c,\max}$ , this is a linear function of  $f_s$  and  $W_n$  and can be expressed as in equation 4.18. Equation 4.14 can be used to replace  $f_s$  by a rational function of  $W_n$  as in the second part of equation 4.18. For this equation, the minimum can be analytically expressed. This entire upper bound on the power calculation is now expressed as a function of  $W_n$ .

$$W_{n,U1} = \frac{\text{SNR}_{\text{target}}\text{SQNR}_{\text{max}}f_{s,\min}}{\text{SQNR}_{\text{max}}f_{s,\min} - \text{SNR}_{\text{target}}f_{s,\max}} \frac{V_{n,\text{in},T,0}^2 W_{n,0}}{\frac{1}{2}(V_{dd}/2)^2} \quad (4.17)$$

$$\begin{aligned} P_{\text{tot}} &= P_s f_s + P_W W_n + P_0 \\ &= P_s \frac{\frac{1}{2}(V_{dd}/2)^2 f_{s,\max} \text{SNR} W_n}{\frac{1}{2}(V_{dd}/2)^2 \text{SQNR}_{\text{max}} W_n - V_{n,\text{in},T,0}^2 W_{n,0} \text{SQNR}_{\text{max}} \text{SNR}} + P_W W_n + P_0 \end{aligned} \quad (4.18)$$

The value of  $W_n$  at which this function reaches a minimum is found by differentiation and expressed as  $W_{n,\text{upper}}$  in equation 4.19. The actual optimal value  $W_{n,\text{opt}}$  is always less than or equal to this value since the value of  $P_s$  is lower for other  $N_{b,c}$ . Beyond the upper bound in 4.19, the power will always increase. Therefore, the first multiple of 100 nm larger than the bound is the final point of  $W_n$  considered in the calculation. If this is lower than the lower bound given by  $f_{s,\max}$ , only the value of the lower bound is considered, as summarized in 4.20. In algorithm 1, the calculations above are mentioned in line 8 and 9.

$$W_{n,\text{opt}} \leq W_{n,U2} = \text{SNR} \sqrt{\frac{P_s f_{s,\max} V_{n,\text{in},T,0}^2 W_{n,0}}{P_W \text{SQNR}_{\text{max}} \frac{1}{2}(V_{dd}/2)^2}} + \text{SNR} \frac{V_{n,\text{in},T,0}^2 W_{n,0}}{\frac{1}{2}(V_{dd}/2)^2} \quad (4.19)$$

$$W_{n,\max} = \begin{cases} \max \left( W_{n,\min}, \min \left( 100 \text{ nm} \left\lfloor \frac{W_{n,U1}}{100 \text{ nm}} \right\rfloor, 100 \text{ nm} \left\lceil \frac{W_{n,U2}}{100 \text{ nm}} \right\rceil \right) \right) & \text{if } W_{n,U1} > 0 \text{ nm} \\ \max \left( W_{n,\min}, 100 \text{ nm} \left\lceil \frac{W_{n,U2}}{100 \text{ nm}} \right\rceil \right) & \text{if } W_{n,U1} < 0 \text{ nm} \end{cases} \quad (4.20)$$

#### 4.7.4 Power calculation

For the selected values of  $W_n$ , the values of  $f_s$ ,  $I_{\max}$ ,  $I_{\min}$ , and  $N_{b,c}$  can be calculated. The total power consumption can then be calculated in function of these and the other identified parameters, by adding the power consumption of each block together. The power of each block is estimated by the equations shown below, which also lead to the values of  $P_s$  and  $P_W$  shown above. Equation 4.21 shows the power consumption of the ring oscillator, which is related to its average current. The power of the coarse counter is related to  $V_{\text{tune},\max}$ ,  $f_{\text{VCO},\max}$  and  $f_{\text{VCO},\max}$  as shown in equation 4.6, repeated below as equation 4.22 for both the low and high tuning voltage case. The power consumption of the sense amplifiers is proportional to the sampling frequency and to the number of sense amplifiers,  $N_\phi + 2N_{b,c}$ , as seen in 4.23. Finally, the digital circuit is estimated in 4.24 to consume a power which is proportional to  $f_s$  as well as  $N_{b,c} + N_{b,\phi}$ .

$$P_{\text{ring}} = V_{dd} \frac{I_{\max} + I_{\min}}{2} \quad (4.21)$$

$$P_{CC} = \begin{cases} V_{dd} \left( G_{L,CC} V_{\text{tune,max}} + 2Q_{T,CC} \frac{f_{VCO,\text{max}} + f_{VCO,\text{min}}}{2} \right) & \text{if } V_{\text{tune,max}} \leq 400 \text{ mV} \\ V_{dd} \left( G_{L,\text{buff}} V_{\text{tune,max}} + (Q_{T,\text{buff}} + 2Q_{T,CC,\text{buff}}) \frac{f_{VCO,\text{max}} + f_{VCO,\text{min}}}{2} \right) & \text{if } V_{\text{tune,max}} > 400 \text{ mV} \end{cases} \quad (4.22)$$

$$P_{SA} = V_{dd} Q_{T,SA} (N_{\phi} + 2N_{b,c}) f_s \quad (4.23)$$

$$P_{\text{dig}} = V_{dd} P_{\text{dig},0} \frac{f_s}{f_{s,0}} \frac{N_{b,c} + N_{b,\phi}}{N_{b,0}} \quad (4.24)$$

The power is calculated over the vector of possible values for  $W_n$  and resulting parameters. The point with the lowest power is selected and compared to the overall lowest power. If the power consumption at the current iteration is lower than at all earlier iterations, the design variables and results are stored as stated on line 14 of algorithm 1. After iterating over all combinations of  $N_{\phi}$ ,  $f_{VCO,\text{max}}$ , and  $f_{VCO,\text{min}}$ , the algorithm has determined the minimal power consumption. A selection of input specifications and design parameters is shown in table 4.6, and a selection of relevant estimated variables in table 4.7.

$f_{BW}$ [MHz]	SNR [dB]	HD3 [dB]	$W_n$ [nm]	$R_{\text{conn}}$ [ $\Omega$ ]	$R_{\text{gnd}}$ [ $\Omega$ ]	$N_{\phi}$	$f_s$ [GHz]
40	76	-40	800	495	765	32	1.387
40	76	-60	1700	206	140	32	1.49
40	76	-30	800	514	1167	16	1.43
40	80	-40	1100	81	82	32	1.49
40	80	-30	800	144	273	32	1.49
40	60	-40	800	3969	5569	16	0.32
40	60	-60	800	5455	5473	8	0.56
40	60	-30	800	3748	6312	16	0.31

Table 4.6: Specifications and resulting optimal design parameters

$f_{BW}$ [MHz]	SNR [dB]	HD3 [dB]	$P_{\text{tot}}$ [mW]	$f_{VCO,\text{max}}$ [GHz]	$f_{VCO,\text{min}}$ [GHz]	$N_{b,c}$
40	76	-40	0.888	4.360	0.536	2
40	76	-60	2.366	5.53	2.28	2
40	76	-30	0.789	8.00	0.54	3
40	80	-40	2.426	7.88	1.82	3
40	80	-30	1.472	6.90	0.37	3
40	60	-40	0.221	2.49	0.58	3
40	60	-60	0.241	4.43	1.48	3
40	60	-30	0.216	2.47	0.54	3

Table 4.7: Specifications and resulting estimated variables

The examples in table 4.6 show that the sizing is optimized carefully by considering changes in all of the parameters selecting their values to minimize the power consumption. Typically, the lowest possible width to meet certain specifications is chosen, as the power consumption of the VCO is larger than the power consumption of the other blocks. As expected, the power consumption rises when the requirements become more stringent. A higher requirement for the SNR requires a wider VCO tuning range and therefore the resistor values are decreased. A stricter specification for HD3 forces the VCO frequency to higher values, where the curve of  $V_{in}$  against  $f_{VCO}$  is more linear. Sometimes, the specifications cannot be met: this was the case when it was attempted to design a VCO with an SNR of 80 dB and a HD3 of  $-60$  dB. Despite the theoretical maximal SQNR of 85.4 dB, the requirement for HD3 reduces the range of  $f_{tune}$  and therefore also the maximal SQNR.

The algorithm presented above extends the algorithm of [2], but only for the case of the coarse-fine VCO-ADC. Whereas the different types of noise are treated separately in the original paper, algorithm 1 combines these to meet the SNR for optimal power consumption, using the trade-off between  $W_n$  and  $f_s$ . The linearity of the ring oscillator VCO and its tuning circuit are also taken into account. Note that the power consumption presented here is calculated based on the data about different circuits. This can make the resulting estimated power more accurate but the conclusions strongly depend on the design of the different circuits and may therefore change as circuits improve. It is attempted to provide freedom to the designer by implementing the values extracted from simulations as parameters in the algorithm, allowing an adaptability to different implementations of the coarse-fine VCO-ADCs building blocks.

### 5.1 Single-Ended VCO-ADC performance

The algorithm developed in section 4.7 and our understanding of the different circuit allows us to simulate a transistor level design for the VCO-ADC. It can then be checked whether the estimated performance by the algorithm is correct. The design is done for a VCO-ADC with a bandwidth of 40 GHz. The minimal SNR of this VCO is 76 dB and the third harmonic has to be more than 40 dB below the carrier. Table 5.1 shows an extensive selection of results obtained from running the design algorithm with these specifications. The results are grouped, showing the specifications, design variables, ring characteristics, the specifications, and the power consumption. The last three tables are therefore estimations of the actual characteristics, and the values obtained in simulations will be compared to this.

$f_{BW}$	$f_{BW,\min}$	$\text{SNR}_{\text{target}}$	$\text{HD3}_{\text{target}}$
40 MHz	100 kHz	76 dB	-40 dB

$f_s$	$W_n$	$R_{\text{conn}}$	$R_{\text{gnd}}$	$N_\phi$	$N_{b,c}$
1.387 GHz	800 nm	495 $\Omega$	765 $\Omega$	32	2

$f_{\text{VCO,max}}$	$f_{\text{VCO,min}}$	$V_{\text{tune,max}}$	$V_{\text{tune,min}}$	$I_{\text{VCO,max}}$	$I_{\text{VCO,min}}$
4.360 GHz	0.536 GHz	571 mV	346 mV	1.149 mA	0.081 mA

SQNR	$\text{SNR}_{\text{in},T}$	SNR	HD3
78.69 dB	79.35 dB	76.00 dB	-40.00 dB

$P_{\text{VCO}}$	$P_{\text{CC}}$	$P_{\text{SA}}$	$P_{\text{dig}}$	$P_{\text{tot}}$
554 $\mu\text{W}$	45 $\mu\text{W}$	85 $\mu\text{W}$	204 $\mu\text{W}$	888 $\mu\text{W}$

Table 5.1: Relevant results of running the algorithm with the given specifications.

The circuit is simulated in a transient simulation both with and without thermal noise. This allows us to first check the specifications on the SQNR and the HD3, and then determine whether the thermal noise constraints and overall SNR are met. Figure 5.1 shows a plot of the frequency spectrum calculated by FFT for the simulation without thermal noise. In this case, the single-ended VCO-ADC is simulated to check whether the simulations agree with estimated values by the algorithm. Since it will be applied in a pseudo-differential configuration, we are interested in the value of HD3 and the second harmonic is therefore ignored.

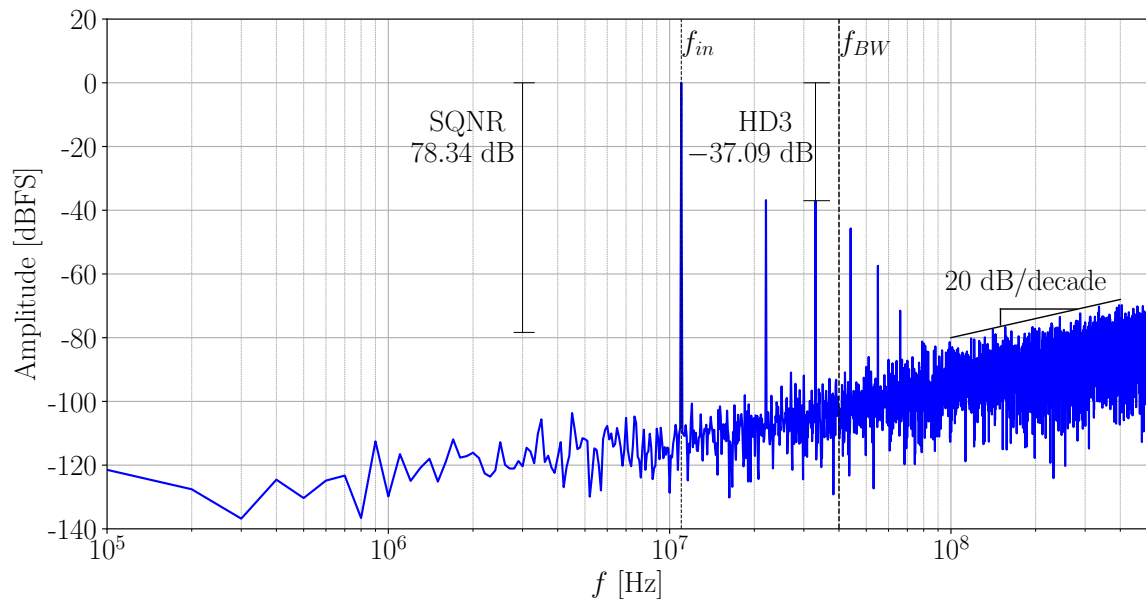


Figure 5.1: Output spectrum of the single-ended VCO-ADC without thermal noise

The resulting spectrum in figure 5.1 shows the expected properties of a VCO-ADC. The noise is shaped with a slope of 20 dB/decade, and since there is no thermal noise this shaping continues until the lowest frequency bin of 100 kHz. The total noise which is in the band below 40 MHz leads to an SQNR of 78.34 dB, which is only marginally lower than expected from the calculation in the algorithm. Besides the noise, the FFT also shows a non-ideality in the form of peaks at the harmonics of  $f_{in}$ , due to the non-linearity of the VCO-ADC. Both the second and third harmonic are in the bandwidth of the signal and will therefore cause distortion. The third harmonic distortion was found to equal  $-37.09$  dB, almost 3 dB higher and therefore worse than predicted by the algorithm.

When this transient simulation is adapted and thermal noise is added, the spectrum of figure 5.2 is obtained. A clear difference between this spectrum and the spectrum of figure 5.1 can be observed: at a frequency of 10 MHz, the noise level in the plot remains approximately constant. This is due to the white input-referred thermal noise, which unlike the quantization noise is not affected by noise shaping. The addition of thermal noise increases the total noise and therefore reduces the SNR to 76.52 dB. The SNR and the SQNR determined in figure 5.1 can be used to

determine the value of the ratio of the signal to the input-referred thermal noise  $\text{SNR}_{\text{in},T}$  based on equation 4.10. The noise density of the white thermal noise  $S_{V_{\text{in},T}}$  is obtained from this value. The  $\text{SNR}_{\text{in},T}$  is higher than expected at 81.17 dB, which is good as this counteracts the lower-than-expected SQNR. This may be due to the factor  $\Gamma_Z$  in equation 4.1 which was neglected in calculations but takes a value between 0.5 and 1, and therefore would reduce the white noise density. The value of  $S_{V_{\text{in},T}}$  observed in these simulations is indeed lower than predicted at  $-167.13 \text{ dBV}/\sqrt{\text{Hz}}$ , which leads to a noise power per bin of  $-107.19 \text{ dB}$ , in accordance with the noise floor in figure 5.2. The corner frequency is predicted to be at 415 MHz, but a longer simulation would be necessary to determine the corner frequency since the  $1/f$ -noise is not visible in this figure.

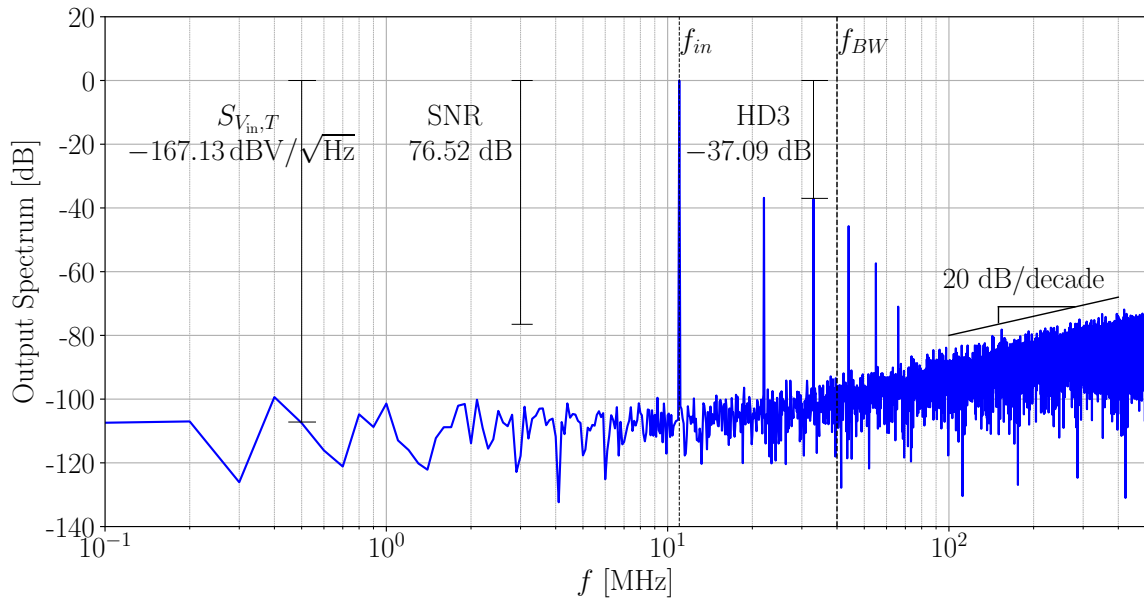


Figure 5.2: Output spectrum of the single-ended VCO-ADC with thermal noise

$f_{\text{VCO,max}}$	$f_{\text{VCO,min}}$	$V_{\text{tune,max}}$	$V_{\text{tune,min}}$	$I_{\text{VCO,max}}$	$I_{\text{VCO,min}}$
4.314 GHz	0.543 GHz	572 mV	346 mV	1.160 mA	0.078 mA

SQNR	$\text{SNR}_{\text{in},T}$	SNR	HD3
78.34 dB	81.17 dB	76.52 dB	-37.09 dB

$P_{\text{VCO}}$	$P_{\text{CC}}$	$P_{\text{SA}}$	$P_{\text{dig}}$	$P_{\text{tot}}$
519 $\mu\text{W}$	38 $\mu\text{W}$	99 $\mu\text{W}$	232 $\mu\text{W}$	888 $\mu\text{W}$

Table 5.2: Relevant simulation results with given specifications and sizing



The results of both simulations have been summarized in table 5.2. The table shows only the values extracted from simulations, the design variables and specifications remain the same as in table 5.1. In general, the results agree well with the predicted values from simulation, with the most significant difference being the HD3. The power consumption of the different blocks is also shown in this table. There are some differences between the predicted and simulated power consumption in the different blocks, but the total power consumption matches the predictions exceptionally well.

## 5.2 Pseudo-Differential Operation and Calibration

While the performance of the SNR is clearly met, the in-band second and third harmonic cause a significant distortion which affects the SNDR of the VCO-ADC. This SNDR is only 33.95 dB. Two steps are taken to lower the distortion level: a pseudo-differential configuration for the VCO-ADC removes the even-order terms, and the digital output is calibrated to significantly reduce the power in the odd harmonics. Both of these methods are common to linearize the VCO-ADC as described in [23]. Figure 5.3 shows how the pseudo-differential operation and calibration are implemented using two designed VCO-ADCs.

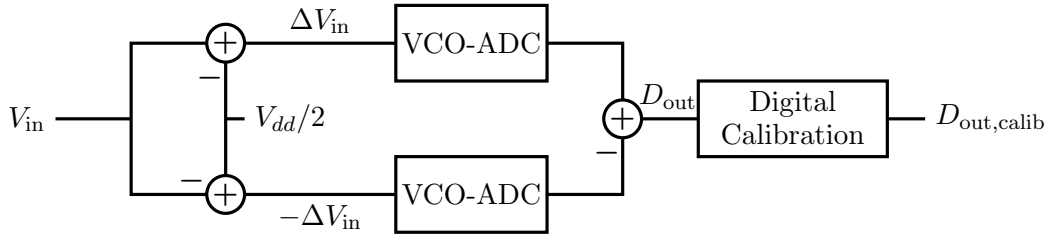


Figure 5.3: Pseudo-differential operation and digital calibration of the VCO-ADC

The effect of the pseudo-differential operation can be understood by writing the input of the calibration block in function of the input signal  $V_{in}$ . Using the non-linear expression  $f_{VCO}(\Delta V_{in})$  from section 4.7, the value of  $D_{out}$  can be written as in equation 5.1.

$$D_{out} = \frac{2N_{\phi}(f_{VCO}(\Delta V_{in}) - f_{VCO}(-\Delta V_{in}))}{f_s} + N_+ - N_- \quad (5.1)$$

The terms  $N_+$  and  $N_-$  indicate the noise on this digital signal. Using the polynomial expression of equation 4.7, it can immediately be seen that the even terms of the digital output signal are removed from this equation. The amplitude of the odd signal terms is doubled. This is expected to increase the SNR by 3 dB, as the noise is assumed to be uncorrelated and therefore its expected amplitude increases by a factor  $\sqrt{2}$ . Note that this increase in SNR also comes at a cost of doubling the power consumption.

Due to the pseudo-differential operation, the peak in the spectrum at the second, fourth, and other even harmonics is reduced to below the noise level, as visible in figure 5.4. The resulting SNR increases slightly more than expected, to 80.34 dB. HD3 remains the same at 37.09 dB.

As the third harmonic is the only remaining in-band harmonic and distortion remains dominant over the noise power, the resulting SNDR equals the value of HD3.

The third order harmonic still adds a significant distortion to the signal. To remove this distortion, the output is digitally calibrated. A seventh-order polynomial fit is applied to describe the distortion in function of the linear value, removing both the third, fifth, and seventh harmonic peak. The distortion is calculated for each input value in the time domain and subtracted from the original value. This operation can be written as in 5.2, where the distortion is estimated by the polynomial function  $f_D$ .

$$D_{\text{out,calib}} = D_{\text{out}} - f_D(D_{\text{out}}) \quad (5.2)$$

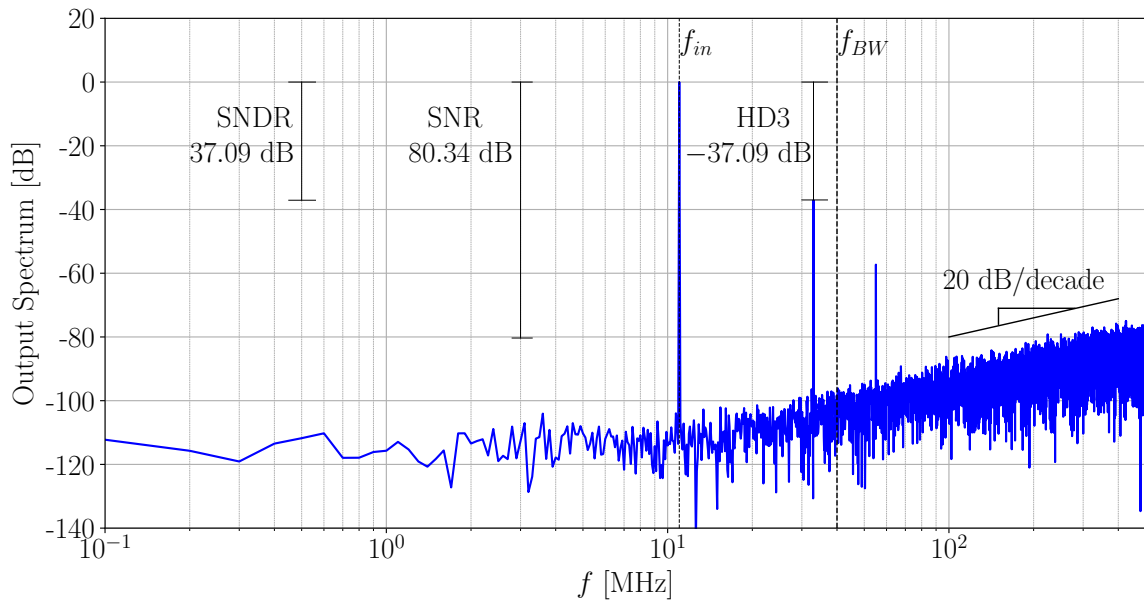


Figure 5.4: Output spectrum of the pseudo-differentially operated VCO-ADC

Calibration has a significant effect on the SNDR: the third harmonic distortion is improved to  $-76.50$  dB, which increases the SNDR to  $74.86$  dB. The SNR is slightly decreased to  $79.80$  dB, due to the stretching of some quantization values by applying calibration. The total power consumption is slightly higher than double the power consumption of the single VCO-ADC, as a difference block and a register are added, leading to a power consumption of  $1.862$  mW. The achieved SNDR of  $74.86$  dB for a design which consumes a power of  $1.862$  mW and has a  $40$  MHz bandwidth would lead to a Schreier figure-of-merit,  $\text{FOM}_S$ , of  $178.18$  dB, calculated as in equation 5.3. A comparison with the state-of-the-art in table 5.3 shows that this design can be very competitive, even if the performance is expected to decrease slightly for an on-chip implementation.

$$\text{FOM}_S = \text{SNDR} + 10 \log \frac{BW}{P} \quad (5.3)$$

	<b>This work</b>	[24]	[25]	[26]	[27]	[28]
Technology [nm]	28	65	28	65	65	28
Digital friendly	yes	yes	yes	yes	yes	no
Samp. Rate [GS/s]	1.39	4	2	0.3	1.6	6
BW [MHz]	40	200	40	30	10	120
Power [mW]	1.86	19.7**	17.5**	11.4	3.7	108.8**
SNR [dB]	79.8	62	-	-	66.2	-
SNDR [dB]	74.9	60.1	76.2	64	65.7	72.3
FOM <sub>S</sub> [dB]	178.2	162.2	169.8	158.2	160.0	162.7

<sup>a</sup> \*\*: not including digital core power consumption

Table 5.3: Comparison with the state-of-the-art VCO-ADCs and a recent delta-sigma modulator

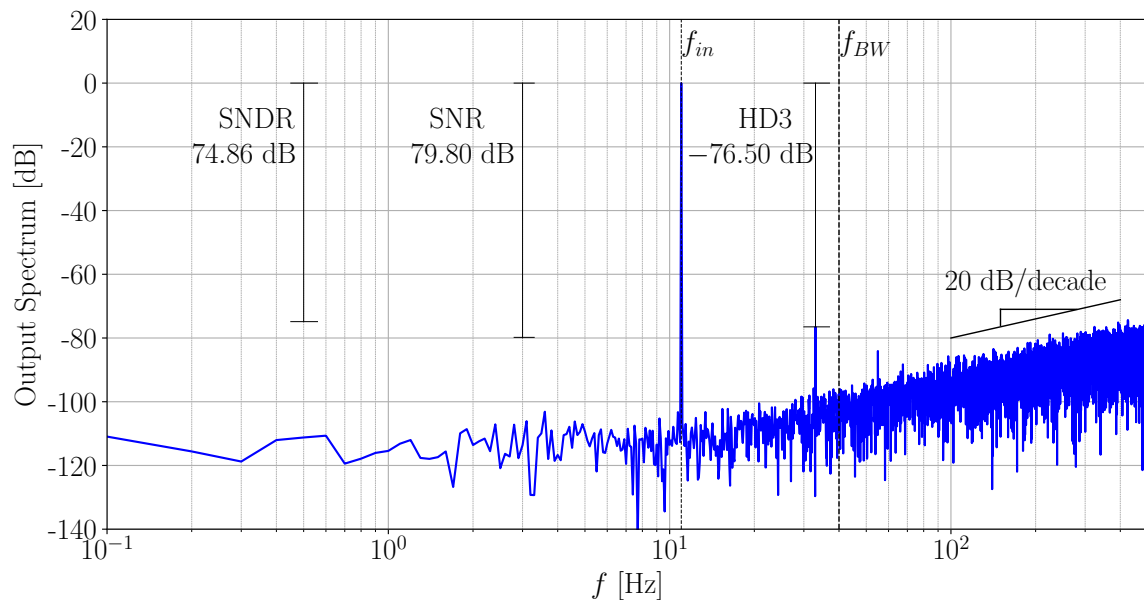


Figure 5.5: Output spectrum of the VCO-ADC after calibration

### 5.3 Effect of VCO Layout

When exploring the different circuits in chapter 4, it was observed that the performance of the VCO is crucial. The ring oscillator both consumes the largest power of all the different building blocks, and the speed of the delay cells is also crucial to the SQNR as this is proportional to  $(N_\phi f_{\text{tune}})^2$ . Finally, the noise of the VCO and its tuning circuit dominates the input-referred thermal noise.

To produce the VCO-ADC as an integrated circuit, a layout for the designed circuits should also be made. Since the VCO is crucial to the performance, a layout for a delay cell of another VCO was adapted to the sizing of this delay cell. The layout is shown for the poly and lower

metal layers in figure 5.6. This layout has been used to test some parameters which were crucial in the sizing algorithm. A noiseless transient simulation of the single-ended VCO-ADC is performed, using the full parasitic extraction (resistance and all capacitances) of the layout instead of the transistor-level schematic for the delay cell. Table 5.4 shows the resulting relevant VCO parameters, performance, and power consumption.

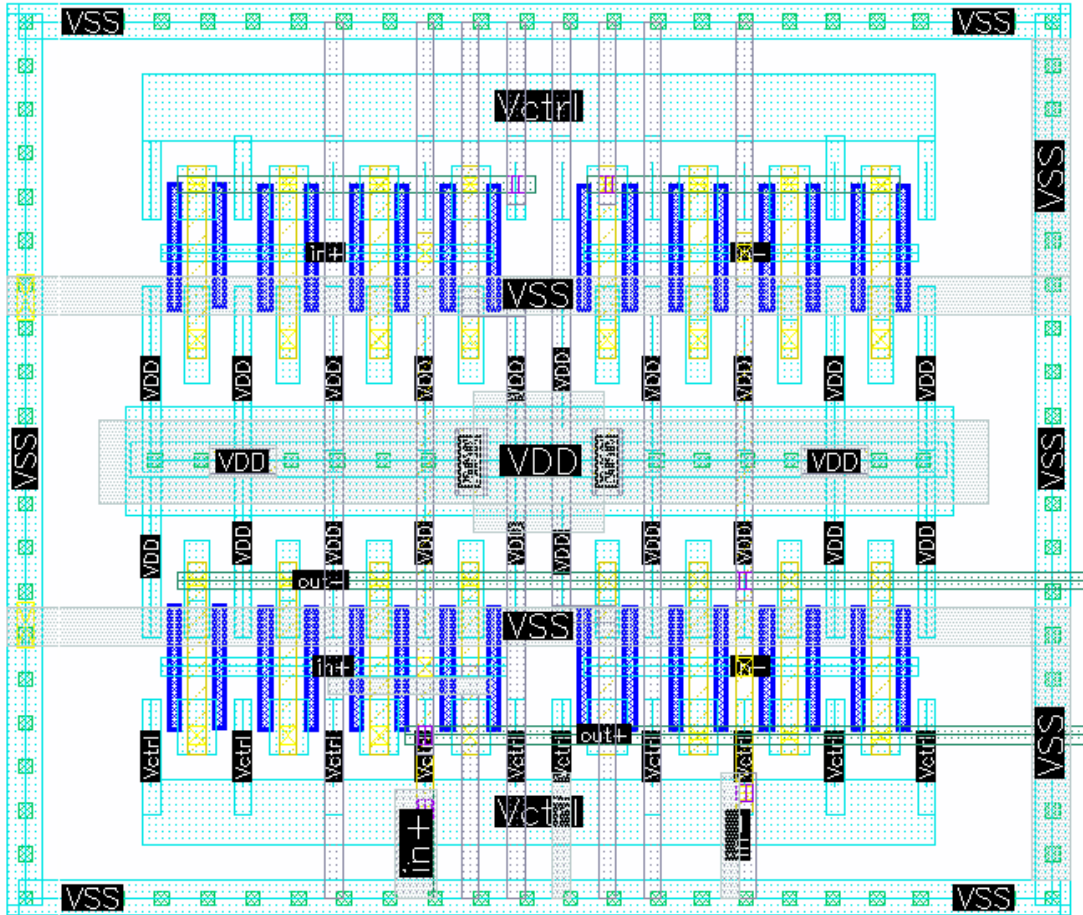


Figure 5.6: Layout of the delay cell

$f_{VCO,max}$	$f_{VCO,min}$	$V_{tune,max}$	$V_{tune,min}$	$I_{VCO,max}$	$I_{VCO,min}$
2.116 GHz	0.248 GHz	569 mV	328 mV	1.109 mA	0.067 mA

SQNR	HD3
71.65 dB	-36.10 dB

$P_{VCO}$	$P_{CC}$	$P_{SA}$	$P_{dig}$	$P_{tot}$
489 $\mu$ W	25 $\mu$ W	100 $\mu$ W	232 $\mu$ W	846 $\mu$ W

Table 5.4: Ring oscillator characteristics for delay cells after layout

The effect of the layout is immediately clear: due to the parasitic effects, the VCO frequency is more than halved. The current consumption in the VCO remains approximately the same, as does the tuning voltage. This means that the SQNR decreases by almost 6 dB compared to the SQNR of the single-ended VCO in figure 5.1, as the amplitude of the ADC output approximately halves. Since the current consumption does not change, the total power consumption also remains approximately the same. Only  $P_{CC}$  changes significantly due to the lower VCO frequency, but this is only a small part of the total power consumption. The SQNR and the HD3 are also shown in the spectrum in figure 5.7. The VCO-ADC is simulated over a shorter time, leading to less frequency bins in the FFT. The decrease in SQNR is clearly marked in this figure. The HD3 has also worsened slightly, probably due to a small shift in the curve of  $f_{VCO}$  in function of  $V_{in}$ .

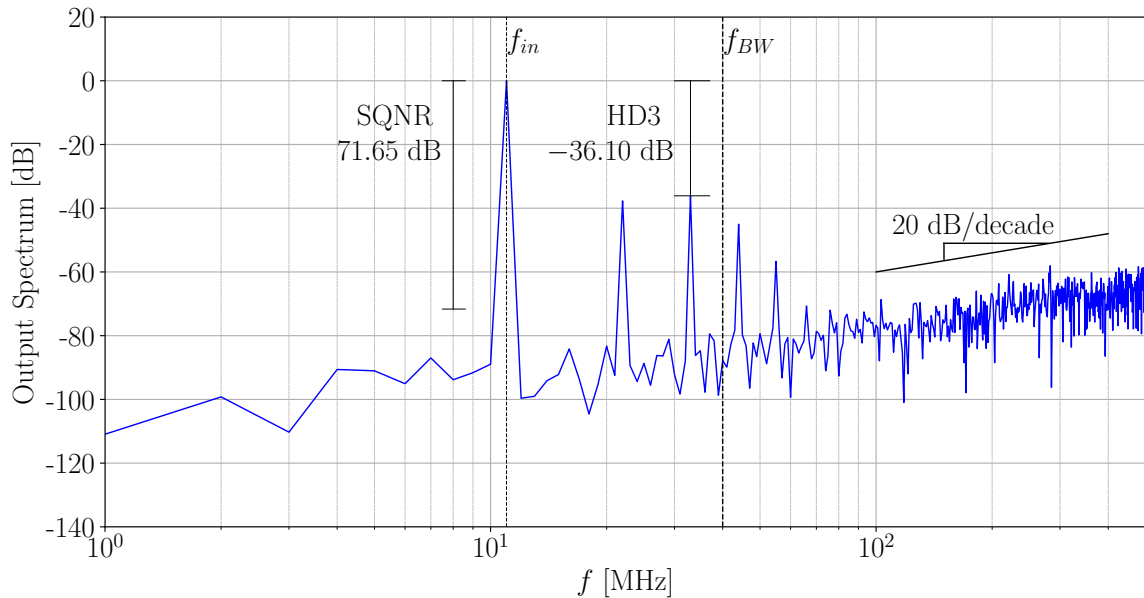


Figure 5.7: Output spectrum of the single-ended VCO-ADC, noiseless, layout of delay cell

It would be interesting if the effect of layout on the parameters could be included in the algorithm of section 4.7. It was derived in [8] that the frequency of the ring oscillator is inversely proportional to the load capacitance on the nodes after each delay cell. If we assume that the parasitic effects are mainly capacitive, an estimation of the load capacitance relative to the parasitic capacitors in the schematic for a certain width can be done. A factor which decreases the frequency in function of the width can therefore be determined. Further research should determine how this factor depends approximately on  $W_n$  and how this can be included in the algorithm.

## CHAPTER 6

### Conclusion

#### 6.1 Conclusion

This thesis presents a successful design of a coarse-fine VCO-ADC based on a given set of specifications, and an algorithm to achieve these specifications at a minimal power consumption. The algorithm is based on an extrapolation of important simulated characteristics of the individual building blocks that define the circuit in function of well-chosen sizing parameters. It therefore extends the power estimation and optimization presented in [2] for a coarse-fine VCO-ADC design while also taking input-referred noise and non-linearity into account. The resulting design is power-efficient, consuming only 1.862 mW for an SNDR of 74.86 dB and a bandwidth of 40 MHz. This results in a  $FOM_S$  of 178.18 dB, comparing favorably to the state-of-the-art.

To obtain the necessary building blocks and characteristics of the different circuits, a top-down approach was taken. Initially, a system-level model was constructed in Simulink and the predicted SQNR using different models and its simulated value was compared. This also verified the correct operation of a simple decoding algorithm to obtain the correct ADC output value from the different bits of the coarse and fine counters. Asynchrony between the coarse and fine counters was identified as a crucial imperfection in a circuit compared to the ideal that could lead to a strongly reduced performance of the VCO-ADC, as was shown by a simple model and through simulations.

The double coarse counter was suggested as a solution to mitigate the issue of asynchrony. While counting on both edges of the first fine counter phase indeed provides the coarse counter with some time to calculate the next value, the timing constraints introduced by the simple double coarse counter limit the available VCO frequency. A solution is obtained by carefully connecting the different flipflops that make up a double asynchronous counter, thereby designing a double connected asynchronous counter which feeds the ripple through to one of the flipflops which determine the next bit in the counter. The identification of the connected double coarse counter also allows us to design a more power-efficient coarse counter compared to previous designs, using gated NAND-latches.

Besides the coarse counter, the other building blocks are also closely examined. The behaviour of the ring oscillator VCO is thoroughly described in [8] and the tuning circuit in [19]. A simple buffer circuit was designed to be able to operate the coarse counter at a high  $V_{\text{tune}}$ , at which the VCO consumes a lower power. The StrongARM sense amplifier and an efficient digital circuit were implemented to sample and decode the coarse and fine counter values. Most of the circuits were already sized, often minimally to take advantage of the digital-friendly technology and reduce power consumption. The remaining five design variables are sized to achieve the specifications at a minimal power consumption, by iterating over design points within carefully selected bounds. The algorithm therefore allows us to set specifications and provide data about the different circuits which is then used to determine a design with a minimal power consumption.

The algorithm was applied to design a coarse-fine VCO-ADC with a bandwidth of 40 MHz for an SNR of 76 dB and HD3 of  $-40$  dB. The resulting design met the specification for SNR but did not exactly reach the desired value for HD3, having a linearity of  $-37.09$  dB. Using pseudo-differential operation of the VCO-ADC and digital calibration, an SNDR of 74.86 dB was obtained leading to a FOM<sub>S</sub> of 178.18 dB.

## 6.2 Future Work

The current design of the coarse-fine VCO-ADC implements an coarse-fine VCO-ADC up to circuit level. This is an important step towards a finished chip design, but there are some important further steps. A short exploration of the effect of implementing the VCO on a layout level was done in section 5.3. It is clear from this that the VCO layout will have a significant impact on the overall design. Preferably, the algorithm should take the effect of the VCO layout into account, immediately producing an estimation for both the performance before and after layout. Additionally, a finished chip requires the other circuits to also be transferred from a schematic to a layout. It will be interesting to see if this has any impact on the performance of the VCO-ADC.

The algorithm predicts the obtained SNR very well but the linearity is not as good as predicted, leading to a higher HD3 in simulation. The designer could take this into account by setting a margin on the desired HD3 value, but it would be preferred if the calculation could be done more accurately or a smart margin decided by the algorithm. Another aspect that could be tested for the current design and implemented in the algorithm is the effect of mismatch on the performance. This may place another significant lower bound on the NMOS width  $W_n$  when optimizing the power consumption, increasing the required current in the ring oscillator.

Finally, it must be noted that the performance of the current design is always limited by the maximal achievable SQNR of 85.4 dB, and the requirements on the HD3 even make this upper limit slightly lower. To improve the performance, a faster circuit which can achieve a higher  $f_{\text{VCO,max}}$  or  $f_s$  can be designed. Another interesting option is to implement a coarse-fine VCO-ADC with a higher noise-shaping order, thereby decreasing the in-band quantization noise.

# APPENDIX A

## Simulink Model

The simulink model which simulates the VCO on a system level is shown hierarchically below. The top level in figure A.1 shows the input and clock signal which control the VCO. Figure A.2 contains the full actual VCO-ADC. Figure A.3 and figure A.4 implement the VCO. The phase is calculated in figure A.3, the difference between the phases is added between these two parts and the square wave is then generated in figure A.4. The readout of phase 0 is shown in figure A.5, including both the coarse and the fine counter. The other phases are read out by the fine counter implemented in figure A.6.

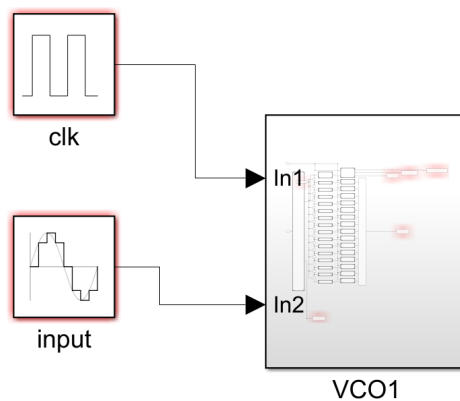


Figure A.1: Top level/testbench of the Simulink model



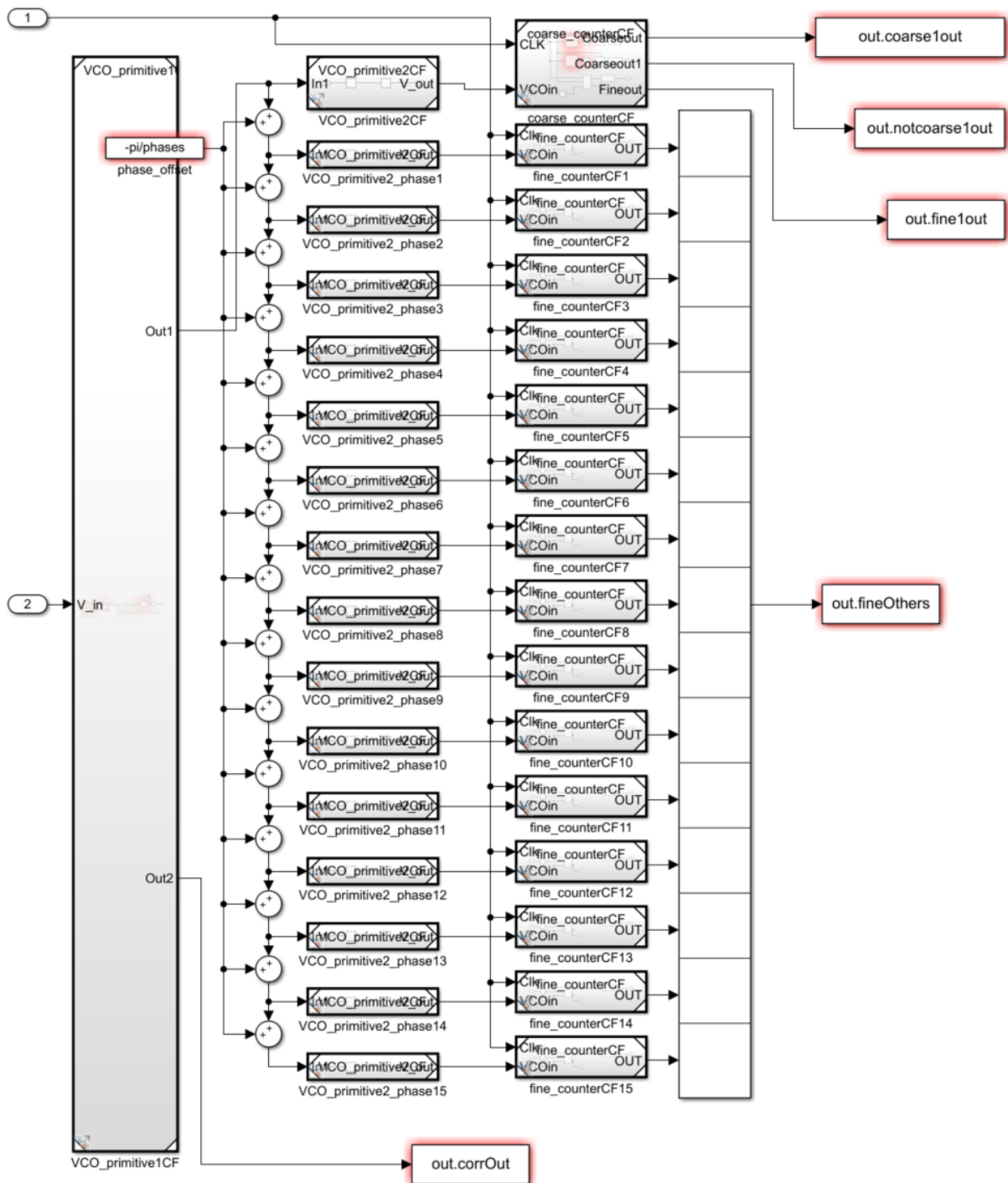


Figure A.2: VCO-ADC in Simulink model

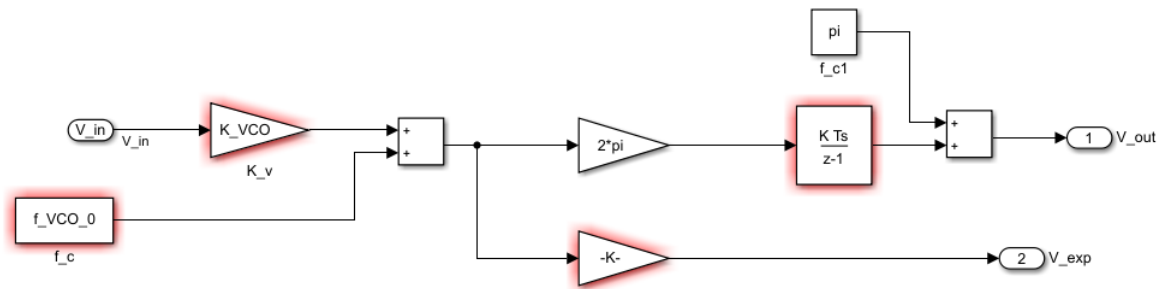


Figure A.3: VCO part 1 in Simulink model



Figure A.4: VCO part 2 in Simulink model

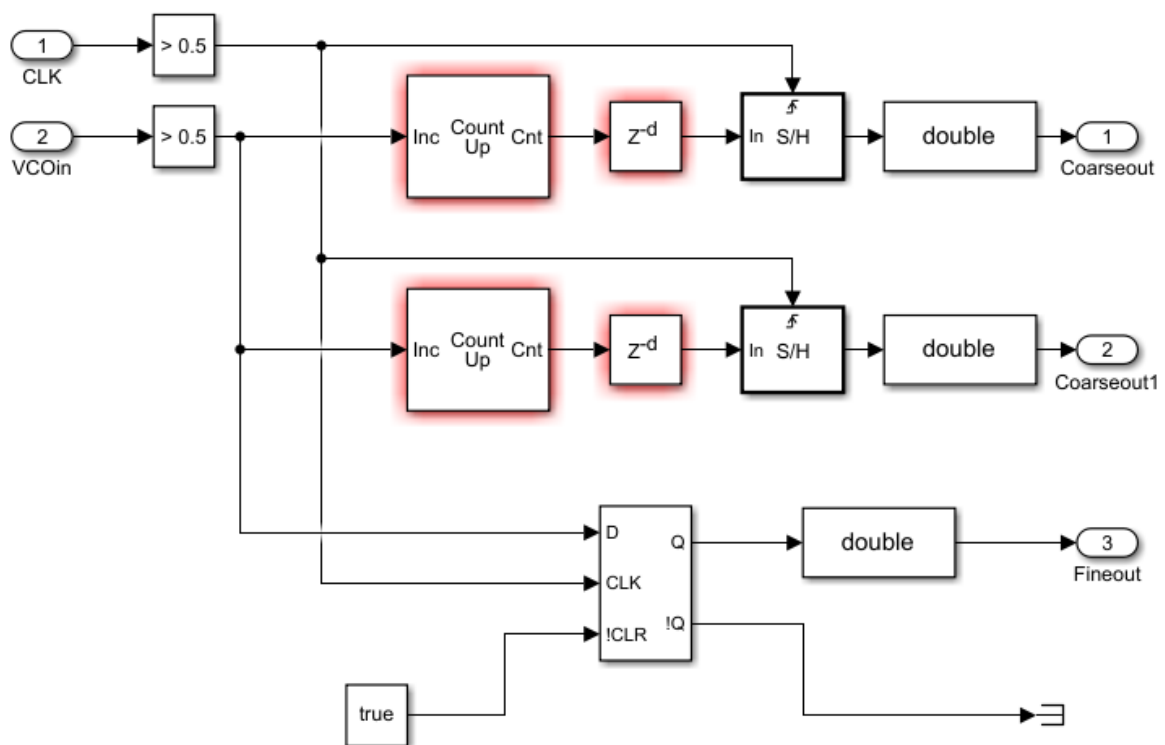


Figure A.5: VCO phase 0 readout in Simulink model

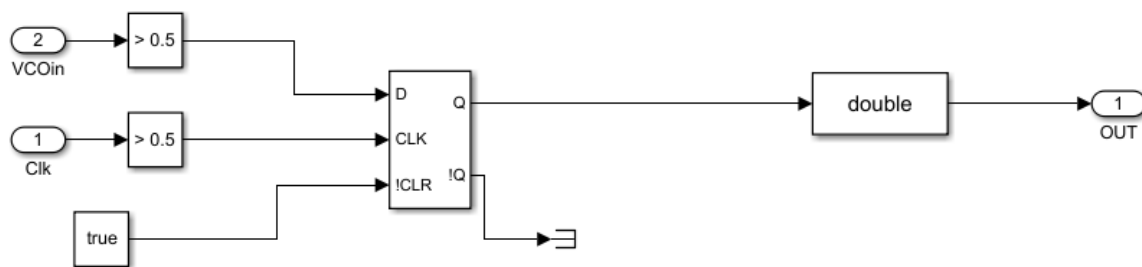


Figure A.6: Other VCO phases readout in Simulink model

## APPENDIX B

### Digital Verilog Code

#### B.1 Pseudo-Differential Implementation

The code below implements the digital circuit discussed in section 4.6 for a VCO-ADC with 32 phases and 2 coarse bits. It is also implemented as a hierarchy of the different blocks. The top level combines the block of figure 4.12 twice, together with a difference operation and a register. This achieves the desired digital circuit for the pseudo-differential implementation.

```
1 //Verilog HDL for "CoarseFineQSD_VCOADC",
2 // "Double_decoding_difference_block_32_2" "functional"
3
4
5 module Double_decoding_difference_block_32_2
6     // Parameter definitions
7     #( parameter N_coarse = 2, // Number of register bits
8       parameter N_fine = 6,
9       parameter N_phi = 32
10    )
11    // Port definition
12    ( // Inputs
13      input [0:N_phi-1]    phasesplus, // VCO output phases
14      input [0:N_coarse-1] CoarseAplus, // Coarse counter A bits
15      input [0:N_coarse-1] CoarseBplus, // Coarse counter B bits
16      input [0:N_phi-1]    phasesminus, // VCO output phases
17      input [0:N_coarse-1] CoarseAminus, // Coarse counter A bits
18      input [0:N_coarse-1] CoarseBminus, // Coarse counter B bits
19      input clk, rst,
20      // Outputs
21      output reg [N_coarse+N_fine-1:0] reg4out
22    );
23    wire [N_coarse+N_fine-1:0] reg3outplus;
24    wire [N_coarse+N_fine-1:0] reg3outminus;
25    wire [N_coarse+N_fine-1:0] reg4in;
26
27    Decoding_Difference_block_32_2 Decdiffplus(
28        .phases (phasesplus),
```

```

29         .CoarseA (CoarseAplus),
30         .CoarseB (CoarseBplus),
31         .clk (clk),
32         .rst (rst),
33         .reg3out (reg3outplus));
34
35     Decoding_Difference_block_32_2 Decdiffminus(
36         .phases (phasesminus),
37         .CoarseA (CoarseAminus),
38         .CoarseB (CoarseBminus),
39         .clk (clk),
40         .rst (rst),
41         .reg3out (reg3outminus));
42
43     Difference diff(
44         .minuend (reg3outplus),
45         .subtrahend (reg3outminus),
46         .difference (reg4in));
47
48     Register reg4(
49         .reg_in (reg4in),
50         .clk (clk),
51         .rst (rst),
52         .reg_out (reg4out));
53 endmodule

```

## B.2 Decoding and Difference Block

The decoding and difference block implements the circuit of figure 4.12, and consists of the same blocks as shown in this figure.

```

1 //Verilog HDL for "CoarseFineQSD_VCOADC",
2 // "Decoding_Difference_block" "functional"
3
4
5 module Decoding_Difference_block_32_2
6     // Parameter definitions
7     #(
8     parameter N_coarse = 2, // Number of register bits
9     parameter N_fine = 6,
10    parameter N_phi = 32
11    )
12    // Port definition
13    (
14    // Inputs
15        input [0:N_phi-1]    phases, // VCO output phases
16        input [0:N_coarse-1] CoarseA, // Coarse counter A bits
17        input [0:N_coarse-1] CoarseB, // Coarse counter B bits
18        input clk, rst,
19    // Outputs
20        output reg [N_coarse+N_fine-1:0] reg3out

```

```

21     );
22
23     wire [N_fine-1:0] reg1in_fine;
24     wire [N_coarse-1:0] reg1in_coarse;
25     wire [N_coarse+N_fine-1:0] reg1out;
26     wire [N_coarse+N_fine-1:0] reg2out;
27     wire [N_coarse+N_fine-1:0] reg3in;
28
29     J2B_LUT_32 J2B32(
30         .phases (phases),
31         .bits_fine (reg1in_fine));
32
33     CoarseCount_dec_2 CCdec2(
34         .Fine0 (phases[0]),
35         .CoarseA (CoarseA),
36         .CoarseB (CoarseB),
37         .bits_coarse (reg1in_coarse));
38
39     Register reg1(
40         .reg_in ({reg1in_coarse,reg1in_fine}),
41         .clk (clk),
42         .rst (rst),
43         .reg_out (reg1out));
44
45     Register reg2(
46         .reg_in (reg1out),
47         .clk (clk),
48         .rst (rst),
49         .reg_out (reg2out));
50
51     Difference diff(
52         .minuend (reg1out),
53         .subtrahend (reg2out),
54         .difference (reg3in));
55
56     Register reg3(
57         .reg_in (reg3in),
58         .clk (clk),
59         .rst (rst),
60         .reg_out (reg3out));
61 endmodule

```

## B.3 Fine Counter Decoder

The sampled phases are decoded using a lookup table which includes a default case followed by the 64 relevant cases which can occur in the fine counter.

```

1 //Verilog HDL for "CoarseFineQSD_VCOADC", "J2B_LUT_32" "functional"
2
3 module J2B_LUT_32
4     // Parameter definitions

```



```

57     32'b1000000000000000000000000000000000000000000000000000000 : bits_fine = 6'd32;
58     32'b1100000000000000000000000000000000000000000000000000000 : bits_fine = 6'd33;
59     32'b111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd34;
60     32'b1111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd35;
61     32'b11111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd36;
62     32'b111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd37;
63     32'b1111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd38;
64     32'b11111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd39;
65     32'b111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd40;
66     32'b1111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd41;
67     32'b11111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd42;
68     32'b111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd43;
69     32'b1111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd44;
70     32'b11111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd45;
71     32'b111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd46;
72     32'b1111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd47;
73     32'b11111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd48;
74     32'b111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd49;
75     32'b1111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd50;
76     32'b11111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd51;
77     32'b111111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd52;
78     32'b1111111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd53;
79     32'b11111111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd54;
80     32'b111111111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd55;
81     32'b1111111111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd56;
82     32'b11111111111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd57;
83     32'b111111111111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd58;
84     32'b1111111111111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd59;
85     32'b11111111111111111111111111111000000000000000000000000000000000000000000000000000 : bits_fine = 6'd60;
86     32'b1111111111111111111111111111110000000000000000000000000000000000000000000000000000 : bits_fine = 6'd61;
87     32'b11111111111111111111111111111110000000000000000000000000000000000000000000000000000 : bits_fine = 6'd62;
88     32'b111111111111111111111111111111110000000000000000000000000000000000000000000000000000 : bits_fine = 6'd63;
89     endcase
90     end
91 endmodule

```

## B.4 Coarse Counter Decoder

Multiplexing in the coarse counter is implemented as individual `case` statements for each multiplexer.

```

1 //Verilog HDL for "CoarseFineQSD_VCOADC",
2 // "CoarseCount_dec" "functional"
3
4
5 module CoarseCount_dec_2 // Parameter definitions
6     #(
7     parameter N_coarse = 2 // Number of coarse count bits
8     )
9     // Port definition
10    (

```



```

11 // Inputs
12 input [0:N_coarse-1] CoarseA, // CoarseCountA output
13 input [0:N_coarse-1] CoarseB, // CoarseCountB output
14 input Fine0, // Fine counter phase 0
15 // Outputs
16 output reg [N_coarse-1:0] bits_coarse
17 );
18
19 reg bitszero = 0;
20 reg bitsone = 0;
21
22 always @ (Fine0 or CoarseA or CoarseB) begin
23 case (Fine0)
24 default : bitszero = 0;
25 0 : bitszero = CoarseA[0];
26 1 : bitszero = CoarseB[0];
27 endcase
28 case (bitszero)
29 default : bitsone = 0;
30 0 : bitsone = CoarseA[1];
31 1 : bitsone = CoarseB[1];
32 endcase
33 bits_coarse = {bitsone, bitszero};
34 end
35 endmodule

```

## B.5 Register and Difference

The registers are implemented using a synchronous reset. The difference block is a straightforward subtraction.

```

1 //Verilog HDL for "CoarseFineQSD_VCOADC", "Register" "functional"
2
3 module Register
4 // Parameter definitions
5 #(
6     parameter N_bits = 8 // Number of register bits
7 )
8 // Port definition
9 (
10 // Inputs
11     input [N_bits-1:0] reg_in, // VCO output phases
12     input clk, rst,
13 // Outputs
14     output reg [N_bits-1:0] reg_out
15 );
16
17 always @(posedge clk) begin
18     if (rst)
19         reg_out <= 0;
20     else

```

```
21     reg_out <= reg_in;
22 end
23 endmodule

1 //Verilog HDL for "CoarseFineQSD_VCDADC", "Difference" "functional"
2
3
4 module Difference
5 // Parameter definitions
6     #(
7         parameter    N_bits = 8 // Number of bits
8     )
9 // Port definition
10    (
11 // Inputs
12     input [N_bits-1:0]    minuend, // bits from which the subtraction is
    done
13     input [N_bits-1:0]    subtrahend, // bits which are subtracted (minuend
- subtrahend)
14 // Outputs
15     output reg [N_bits-1:0] difference
16 );
17
18     always @ (minuend or subtrahend) begin
19         difference = minuend - subtrahend;
20     end
21
22 endmodule
```

## APPENDIX C

### Design Algorithm in Python

The full code for the algorithm discussed in section 4.7 and summarized in algorithm 1 is displayed below.

```
1 import numpy as np
2
3 #helper functions
4 def diffextrap(x, x_array, y_array):
5     x1 = x_array[x_array <= x][-1]
6     x2 = x_array[x_array > x][0]
7     y1 = y_array[x_array <= x][-1]
8     y2 = y_array[x_array > x][0]
9     y = (y2-y1)/(x2-x1)
10    return y
11
12 def linextrap(x,x1,y1,x2,y2):
13    y = y1+(x-x1)*(y2-y1)/(x2-x1)
14    return y
15
16 def linextrap_array(x, x_array, y_array):
17    x1 = x_array[x_array <= x][-1]
18    x2 = x_array[x_array > x][0]
19    y1 = y_array[x_array <= x][-1]
20    y2 = y_array[x_array > x][0]
21    y = linextrap(x,x1,y1,x2,y2)
22    return y
23
24
25 #data technology
26 wNstepmin = 100 #nm
27 Vdd = 900 #mV
28
29 # data VCD
30 [fring0, Iring0, Vtune0] = np.loadtxt("results.csv",delimiter=",") #Hz,A,V
31 fring0 = fring0/10**6 #MHz
32 Iring0 = Iring0*10**3 #mA
33 Vtune0 = Vtune0*10**3 #mV
```

---

```

34 Nphi0 = 16
35 Wn0 = 3200 #nm
36
37 #data noise
38 whitenoise0_dB = -183 #dBV/rootHz
39 fc0 = 1.144 #MHz
40 Vnoise0 = 400 #mV
41 gring_nref = -diffextrap(Vnoise0,Vtune0,Iring0) #S
42 whitenoise0 = 10**(whitenoise0_dB/10) #V^2/Hz
43
44 #constants
45 k_boltzmann = 1.38*10**(-23) # W^2 K^-1 Hz^-1
46 T = 300 #K
47
48 # data power consumption
49 g_L_CCO = 1/40/10**3 #mA/mV
50 QTCCO = 1.5/10**6 #mA/MHz
51 g_L_buff0 = 3/4*1/10/10**3 #S
52 QTbuff0 = 1/10**6 #mA/MHz
53 QTCCbuff0 = 1/10**6
54 QTFF0 = 1.9/10**6 #mA/MHz
55 Pdig0 = 0.168 #mW
56 fsdig0 = 1000 #MHz
57 Nb0 = 8
58
59 #data specification targets
60 SNR_min_dB = 76 #dB
61 HD3_max_dB = -40 #dB
62 BW = 40 #MHz
63 BWmin = 0.1 #MHz (for 1/f noise integration)
64 HD3_max = 10**(HD3_max_dB/20)
65 SNR_min = 10**(SNR_min_dB/10)
66
67 #data limitations on variables
68 Wnmin0 = 800 #nm
69 Wnstepmin = 100 #nm
70 f_VCO_max_max = 8000 #MHz
71 fs_max0 = 1500 #MHz
72 fs_min0 = 10*BW #MHz
73 Nphilist0 = np.array([4, 8, 16, 32])
74
75 #initial var
76 min_P_tot = 0
77
78 #calc Nphi for SQNR > target
79 Nphimin = np.sqrt((SNR_min)/(2*f_VCO_max_max)**2/fs_max0*(2*BW)**3*2*np.pi**2/9)
80 Nphilist = Nphilist0[Nphilist0 > Nphimin]
81 for Nphi in Nphilist:
82     Nbphi = np.ceil(np.log2(Nphi))
83
84     #defining ftune range
85     fring = fring0*Nphi0/Nphi

```

```

86     #calc ftune for SQNR > target
87     ftunemin = np.sqrt((SNR_min)/(2*Nphi)**2/fs_max0*(2*BW)**3*2*np.pi**2/9)
88     i = len(fring)-1 #iteration variable
89
90     while f_VCO_max_max-fring[i] > ftunemin:
91
92         f_VCO_min = fring[i]
93         j = len(fring[fring > f_VCO_min+ftunemin])-1 #iteration variable
94
95         while fring[j] < f_VCO_max_max:
96
97             #initializing variables
98             f_VCO_max = fring[j]
99
100            #calculate resistors
101            Rconn0 = -900*Vtune0[j]/(Vtune0[j]*Iring0[i]-Vtune0[i]*Iring0[j])
102            Rgnd0 = Vtune0[j]/(Iring0[j]-Vtune0[j]/Rconn0)
103            gring0 = -(Iring0[i]-Iring0[j])/(Vtune0[i]-Vtune0[j])
104
105            # calculate HD3 in steps
106            # calc Vin, indep of wN
107            Vinplus = Vtune0[j:i+1]*(1+Rconn0/Rgnd0)-Iring0[j:i+1]*Rconn0
108            fringVin = fring[j:i+1]
109            deltaVin = (Vinplus-450)
110
111            # Legendre polynomials
112            a_1 = 3/2*(np.trapz(fringVin*deltaVin/450, deltaVin/450))
113            a_3 = 7/2*(np.trapz(fringVin*1/2*(5*deltaVin**3/(450)**3
114                -3*deltaVin/(450)), deltaVin/450))
115
116            #Poly coeffs
117            alpha_3 = 5/2*a_3/450**3
118            alpha_1 = a_1*1/450-a_3/450*3/2
119            HD3 = np.abs(alpha_3*450**2/4/(alpha_1+alpha_3*450**2*3/4))
120
121            #check spec on HD3
122            if HD3 > HD3_max:
123                j -= 1
124            else:
125                #calculate noise
126
127                #white thermal ring noise
128                whiteringnoise = 2*whitenoise0 * gring_nref/(gring0)
129                whiteringnoise *= (Rconn0/Rgnd0 + 1)**2
130
131                # 1/f-noise
132                fnoisedensity_f = 2*whitenoise0*Nphi0/Nphi
133                fnoisedensity_f *= (Rconn0/Rgnd0 + 1)**2*fc0*10**6
134
135                #white resistor noise
136                whiteresistornoise = 4*k_boltzmann*T*Rconn0*(1+Rconn0/Rgnd0)
137

```

```

138     #total noise
139     inputnoise = whiteresistornoise*(BW-BWmin)*10**6
140     inputnoise += whiteringnoise*(BW-BWmin)*10**6
141     inputnoise += fnoisedensity_f*np.log(BW/BWmin)
142     inputnoiseratio0 = inputnoise/(0.45)**2*2
143
144     #determine Wn bounds
145     SQNR_P_max = (2*Nphi*(f_VCO_max-f_VCO_min)/fs_max0)**2
146     SQNR_P_max *= (fs_max0/(2*BW))**3 * 9/(2*np.pi**2)
147     WnL = inputnoiseratio0*Wn0/(1/SNR_min-1/SQNR_P_max)
148     WnL = int(np.ceil(WnL/100)*100)
149     Wnmin = max(Wnmin0, WnL)
150
151
152     fs_inf = fs_max0*SNR_min/SQNR_P_max
153
154     Nbc_max = np.ceil(np.log2(f_VCO_max/fs_inf))
155
156     # Power in func of unknowns
157
158     #P_dig = P_dig_fs*fs
159     P_dig_fs = Pdig0/fsdig0*(Nbc_max+Nbphi)/Nb0
160     #P_ring = P_ring_Wn*Wn
161     P_ring_Wn = (1/Wn0*(Iring0[j]+Iring0[i])/2)*Vdd/1000
162     #P_FF = P_FF_fs*fs
163     P_FF_fs = ((Nphi+2*Nbc_max)*QTFF0)*Vdd/1000
164
165     #calculate upper bound due to minimum
166     WnU2 = (np.sqrt((P_FF_fs+P_dig_fs)*fs_max0
167                 *SQNR_P_max*inputnoiseratio0*Wn0/P_ring_Wn)
168            +SQNR_P_max*inputnoiseratio0*Wn0)/SQNR_P_max*SNR_min
169     WnU2_round = int(np.ceil(WnU2/100)*100)
170
171     if fs_inf < fs_min0:
172         #calculate upper bound due to fs limit
173         WnU1_round = int(np.floor(inputnoiseratio0*Wn0
174                                 /(1/SNR_min-fs_max0/fs_min0*1/SQNR_P_max)/100)*100)
175         Wnmax_1 = min(WnU1_round, WnU2_round)
176     else:
177         Wnmax_1 = WnU2_round
178
179     Wnmax = max(Wnmax_1, Wnmin)
180
181     Wn = np.linspace(Wnmax, Wnmin, (Wnmax-Wnmin)//Wnstepmin+1)
182
183     QNSR_necessary = (1/SNR_min-inputnoiseratio0*Wn0/Wn)
184     fs = (2*BW)**3*2*np.pi**2/(QNSR_necessary
185                               *(2*Nphi*(f_VCO_max-f_VCO_min))**2*9)
186
187     Nbc = np.ceil(np.log2(f_VCO_max/fs))
188
189

```

```

190         #calc power based on data and calculated values, in mW
191         P_dig = Pdig0*fs/fsdig0*(Nbc+Nbphi)/Nb0
192         P_ring = (Wn/Wn0*(Iring0[j]+Iring0[i])/2)*Vdd/1000
193
194         #two cases for Pcc
195         P_CC = (g_L_buff0*Vtune0[i] + (QTbuff0+2*QTCCbuff0)
196               *(f_VCO_max+f_VCO_min)/2) * (Vtune0[i] > 400)*Vdd/1000
197         P_CC += (g_L_CC0*Vtune0[i] + 2*QTCC0*
198               (f_VCO_max+f_VCO_min)/2)*(Vtune0[i] <= 400)*Vdd/1000
199
200         P_FF = ((Nphi+2*Nbc)*QTFF0)*fs*Vdd/1000
201
202         P_tot = P_dig + P_ring + P_CC + P_FF
203
204
205         #test for minimum
206         if np.min(P_tot) < min_P_tot or min_P_tot == 0:
207             min_P_tot = np.min(P_tot)
208             min_P_pos = np.argmin(P_tot)
209             Wnused = Wn[min_P_pos]
210
211         #store relevant values
212         minfactors = [Wnused, Rconn0*Wn0/Wnused,
213                     Rgnd0*Wn0/Wnused, fs[min_P_pos], Nphi]
214         minresults = [f_VCO_max, f_VCO_min, Nbc[min_P_pos],
215                     HD3, Iring0[j]*Wnused/Wn0,
216                     Iring0[i]*Wnused/Wn0, Vtune0[j], Vtune0[i],
217                     inputnoiseratio0*Wn0/Wnused, alpha_1,
218                     alpha_3, whiteringnoise*Wn0/Wnused,
219                     whiteresistornoise*Wn0/Wnused,
220                     gring0/gring_nref*Nphi0/Nphi*fc0]
221         Presults = [P_dig[min_P_pos], P_ring[min_P_pos], P_CC,
222                   P_FF[min_P_pos], P_tot[min_P_pos]]
223         Vinres = Vinplus
224         deltaVinres = deltaVin
225         fringVinres = fringVin
226
227         j -= 1
228     i -= 1

```

## Bibliography

- [1] J. Borgmans and P. Rombouts, "Toward 'digital' analogue-to-digital converters," *Electronics Letters*, vol. 55, no. 10, pp. 568–569, 2019. DOI: <https://doi.org/10.1049/el.2019.1269>. eprint: <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/el.2019.1269>. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/el.2019.1269>.
- [2] J. Borgmans, E. Sacco, P. Rombouts, and G. Gielen, "Methodology for Readout and Ring Oscillator Optimization Toward Energy-Efficient VCO-Based ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 3, pp. 985–998, 2022. DOI: 10.1109/TCSI.2021.3129919.
- [3] G. Taylor and I. Galton, "A Mostly-Digital Variable-Rate Continuous-Time Delta-Sigma Modulator ADC," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 12, pp. 2634–2646, 2010. DOI: 10.1109/JSSC.2010.2073193.
- [4] A. Mukherjee, M. Gandara, B. Xu, *et al.*, "A 1-GS/s 20 MHz-BW Capacitive-Input Continuous-Time  $\Delta\Sigma$  ADC Using a Novel Parasitic Pole-Mitigated Fully Differential VCO," *IEEE Solid-State Circuits Letters*, vol. 2, no. 1, pp. 1–4, 2019. DOI: 10.1109/LSSC.2019.2911874.
- [5] C. Shannon, "Communication in the Presence of Noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949. DOI: 10.1109/JRPROC.1949.232969.
- [6] J. G. Truxal, *Automatic feedback control system synthesis*. New York: McGraw-Hill, 1955.
- [7] G. G. Gielen, L. Hernandez, and P. Rombouts, "Time-Encoding Analog-to-Digital Converters: Bridging the Analog Gap to Advanced Digital CMOS-Part 1: Basic Principles," *IEEE Solid-State Circuits Magazine*, vol. 12, no. 2, pp. 47–55, 2020. DOI: 10.1109/MSSC.2020.2987536.
- [8] J. Borgmans, R. Riem, and P. Rombouts, "The Analog Behavior of Pseudo Digital Ring Oscillators Used in VCO ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2827–2840, 2021. DOI: 10.1109/TCSI.2021.3073817.
- [9] J. Kim, T.-K. Jang, Y.-G. Yoon, and S. Cho, "Analysis and Design of Voltage-Controlled Oscillator Based Analog-to-Digital Converter," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 18–30, 2010. DOI: 10.1109/TCSI.2009.2018928.



- [10] E. Gutierrez, L. Hernandez, F. Cardes, and P. Rombouts, "A Pulse Frequency Modulation Interpretation of VCOs Enabling VCO-ADC Architectures With Extended Noise Shaping," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 2, pp. 444–457, 2018. DOI: 10.1109/TCSI.2017.2737830.
- [11] L. Kleeman and A. Cantoni, "Metastable Behavior in Digital Systems," *IEEE Design & Test of Computers*, vol. 4, no. 6, pp. 4–19, 1987. DOI: 10.1109/MDT.1987.295189.
- [12] C. Perez, R. Garvi, G. Lopez, *et al.*, "A VCO-based ADC with direct connection to a microphone MEMS, 80dB peak SNDR and 438 $\mu$ W power consumption," *IEEE Sensors Journal*, pp. 1–1, 2023. DOI: 10.1109/JSEN.2023.3244663.
- [13] J. Daniels, W. Dehaene, and M. Steyaert, "All-digital differential VCO-based A/D conversion," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2010, pp. 1085–1088. DOI: 10.1109/ISCAS.2010.5537342.
- [14] A. Quintero, C. Buffa, C. Perez, *et al.*, "A Coarse-Fine VCO-ADC for MEMS Microphones With Sampling Synchronization by Data Scrambling," *IEEE Solid-State Circuits Letters*, vol. 3, pp. 29–32, 2020. DOI: 10.1109/LSSC.2020.2964158.
- [15] M. Baert and W. Dehaene, "A 5-GS/s 7.2-ENOB Time-Interleaved VCO-Based ADC Achieving 30.5 fJ/cs," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 6, pp. 1577–1587, 2020. DOI: 10.1109/JSSC.2019.2959484.
- [16] I. Kovacs and M. Neag, "New dual-loop topology for ring VCOs based on latched delay cells," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5. DOI: 10.1109/ISCAS.2018.8351757.
- [17] J. Borgmans and P. Rombouts, "Enhanced circuit for linear ring VCO-ADCs," *Electronics Letters*, vol. 55, no. 10, pp. 583–585, 2019. DOI: <https://doi.org/10.1049/el.2019.0241>.
- [18] Babaie Fishani, Amir and Rombouts, Pieter, "Highly linear VCO for use in VCO-ADCs," eng, *Electronics Letters*, vol. 52, no. 4, 268–269, 2016, ISSN: 0013-5194. [Online]. Available: <http://dx.doi.org/10.1049/el.2015.3807>}.
- [19] J. Borgmans and P. Rombouts, "Noise Optimization of a Resistively-Driven Ring Oscillator for VCO-Based ADCs," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 775–779. DOI: 10.1109/ISCAS48785.2022.9937724.
- [20] H. Xu and A. A. Abidi, "Analysis and Design of Regenerative Comparators for Low Offset and Noise," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 8, pp. 2817–2830, 2019. DOI: 10.1109/TCSI.2019.2909032.
- [21] B. Razavi, "The StrongARM Latch [A Circuit for All Seasons]," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12–17, 2015. DOI: 10.1109/MSSC.2015.2418155.
- [22] J. Borgmans and P. Rombouts, "The Mismatch Performance of Pseudo Digital Ring Oscillators Used in VCO ADCs: PSRR and CMRR," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 2, pp. 579–592, 2023. DOI: 10.1109/TCSI.2022.3222380.

- 
- [23] G. G. Gielen, L. Hernandez, and P. Rombouts, "Time-Encoding Analog-to-Digital Converters: Bridging the Analog Gap to Advanced Digital CMOS-Part 2: Architectures and Circuits," *IEEE Solid-State Circuits Magazine*, vol. 12, no. 3, pp. 18–27, 2020. DOI: 10.1109/MSSC.2020.3002144.
- [24] T.-F. Wu and M. S.-W. Chen, "A 200MHz-BW 0.13mm<sup>2</sup> 62dB-DR VCO-based non-uniform sampling ADC with phase-domain level crossing in 65nm CMOS," in *2018 IEEE Custom Integrated Circuits Conference (CICC)*, 2018, pp. 1–4. DOI: 10.1109/CICC.2018.8357088.
- [25] T.-F. Wu and M. S.-W. Chen, "A 40MHz-BW 76.2dB/78.0dB SNDR/DR Noise-Shaping Nonuniform Sampling ADC with Single Phase-Domain Level Crossing and Embedded Nonuniform Digital Signal Processor in 28nm CMOS," in *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2020, pp. 262–264. DOI: 10.1109/ISSCC19947.2020.9063022.
- [26] J. Daniels, W. Dehaene, M. Steyaert, and A. Wiesbauer, "A 0.02mm<sup>2</sup> 65nm CMOS 30MHz BW all-digital differential VCO-based ADC with 64dB SNDR," in *2010 Symposium on VLSI Circuits*, 2010, pp. 155–156. DOI: 10.1109/VLSIC.2010.5560314.
- [27] A. Babaie-Fishani and P. Rombouts, "A Mostly Digital VCO-Based CT-SDM With Third-Order Noise Shaping," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 8, pp. 2141–2153, 2017. DOI: 10.1109/JSSC.2017.2688364.
- [28] M. Bolatkale, R. Rutten, H. Brekelmans, *et al.*, "A 28-nm 6-GHz 2-bit Continuous-Time  $\Delta\Sigma$  ADC With -101-dBc THD and 120-MHz Bandwidth Using Blind Digital DAC Error Correction," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 12, pp. 3768–3780, 2022. DOI: 10.1109/JSSC.2022.3202977.

