Linköping Studies in Science and Technology
Dissertation No. 1638

Design of Integrated Building Blocks for the Digital/Analog Interface

Niklas U. Andersson

Linköping University
INSTITUTE OF TECHNOLOGY

Linköping University
Department of Electrical Engineering
Electronics Systems
SE-581 85 Linköping, Sweden

Linköping 2015

© Niklas U. Andersson, 2015

ISBN 978-91-7519-163-8

ISSN 0345-7524

URL <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-112215/>

Published articles have been reprinted with permission from the respective copyright holder, see page 9 for details.

Typeset using L^AT_EX

Printed by LiU-Tryck, Linköping 2015

Abstract

The integrated circuit has, since it was invented in the late 1950's, undergone a tremendous development and is today found in virtually all electric equipment. The small feature size and low production cost have made it possible to implement electronics in everyday objects ranging from computers and mobile phones to smart prize tags. Integrated circuits are typically used for data communication, signal processing and data storage. Data is usually stored in digital format but signal processing can be performed both in the digital and in the analog domain. For best performance, the right partition of signal processing between the analog and digital domain must be used. This is made possible by data converters converting data between the domains. A device converting an analog signal into a digital representation is called an analog-to-digital converter (ADC) and a device converting digital data into an analog representation is called a digital-to-analog converter (DAC). In this work we present research results on these data converters and the results are compiled in three different categories. The first contribution is an error correction technique for DACs called dynamic element matching, the second contribution is a power efficient time-to-digital converter architecture and the third is a design methodology for frequency synthesis using digital oscillators.

The accuracy of a data converter, i.e., how accurate data is converted, is often limited by manufacturing errors. One type of error is the so-called matching error and in this work we investigate an error correction technique for DACs called dynamic element matching (DEM). If distortion is limiting the performance of a DAC, the DEM technique increases the accuracy of the DAC by transforming the matching error from being signal dependent, which results in distortion, to become signal independent noise. This noise can then be spectrally shaped or filtered out and hereby increasing the overall resolution of the system. The DEM technique is investigated theoretically and the theory is supported by measurement results from an implemented 14-bit DAC using DEM. From the investigation it is concluded that DEM increases the performance of the DAC when matching errors are dominating but has less effect at conversion speeds when dynamic errors dominate.

The next contribution is a new time-to-digital converter (TDC) architecture. A TDC is effectively an ADC converting a time difference into a digital representation. The proposed architecture allows for smaller and more power efficient data conversion than previously reported and the implemented TDC prototype is smaller and more power efficient as compared to previously published TDCs in the same performance segment.

The third contribution is a design methodology for frequency synthesis using digital oscillators. Digital oscillators generate a sinusoidal output using recursive algorithms. We show that the performance of digital oscillators, in terms of amplitude and frequency stability, to a large extent depends on the start conditions of the oscillators. Further we show that by selecting the proper start condition an oscillator can be forced to repeat the same output sequence over and over again, hence we have a locked oscillator. If the oscillator is locked there is no drift in amplitude or frequency which are common problems for recursive oscillators not using this approach. To find the optimal start conditions a search algorithm has been developed which has been thoroughly tested in simulations. The digital oscillator output is used for test signal generation for a DAC or used to generate tones with high spectral purity using DACs.

Populärvetenskaplig Sammanfattning

Den integrerade kretsen har sedan den uppfanns i slutet av 1950-talet genomgått en enorm utveckling och återfinns idag i princip i all elektronisk utrustning. Den lilla storleken och den låga produktionskostnaden har gjort det möjligt att integrera elektronik i vardagsföremål som datorer och mobiltelefoner och enklare system som till exempel smarta etiketter. Typiska användningsområden för integrerade kretsar är datakommunikation, signalbehandling och datalagring. Data lagras vanligtvis i digitalt format men signalbehandling kan utföras i både den digitala och i den analoga domänen. För att nå bästa prestanda i en krets måste signalbehandlingen delas upp optimalt mellan den digitala och analoga domänen. Denna uppdelning möjliggörs med hjälp av dataomvandlare som översätter data mellan de två domänerna. En krets som omvandlar en analog signal till en digital motsvarighet kallas för en analog-till-digital-omvandlare och en krets som omvandlar digitalt data till en analog signal kallas för en digital-till-analog-omvandlare. Denna doktorsavhandling innehåller resultat från forskning gjord på dessa dataomvandlare och resultaten är sammanfattade i tre huvudkategorier. Det första bidraget är en felkorrigeringsmetod för digital-till-analog-omvandlare, det andra bidraget är en kretsarkitektur för en energieffektiv tid-till-digital-omvandlare och det tredje bidraget är en konstruktionsmetodik för frekvenssyntes med hjälp av digitala svängningskretsar.

Noggrannheten hos en dataomvandlare, med andra ord hur noggrant dataomvandlaren kan omvandla data mellan de två domänerna, begränsas ofta av de fel som uppstår vid tillverkningen av den integrerade kretsen. En typ av fel som uppstår är att dataomvandlarens jämförelsenivåer inte blir lika stora. I frekvensdomänen kommer denna typ av fel resultera i icke önskade harmoniska frekvenser (distorsion) som begränsar dataomvandlarens noggrannhet. Om distorsion, som uppkommer då ett fel beror på dataomvandlarens insignal, begränsar dataomvandlarens prestanda kan den föreslagna felkorrigeringsmetoden omvandla distortionerna till brus genom att göra felet oberoende av insignalen. Det resulterande bruset kan sedan formas spektralt eller filteras bort och därmed öka systemets totala prestanda. Den föreslagna korrigeringsmetoden har undersökts teoretiskt och denna teori har sedan verifierats med mätresultat från en kretsimplementation av en 14-bitars digital-till-analog-omvandlare som använder den föreslagna felkorrigeringsmetoden. Mätresultaten visar att metoden höjer prestandan hos dataomvandlaren för låga insignalfrekvenser då det är felet i jämförelsenivåerna som begränsar prestandan. Vid högre insignalfrekvenser är metoden mindre effektiv då andra dynamiska felkällor hos dataomvandlaren istället begränsar noggrannheten.

Nästa bidrag är en kretsarkitektur till en tid-till-digital-omvandlare. En tid-till-digital-omvandlare är en särskild sorts analog-till-digital-omvandlare som omvandlar tidsskillanden mellan två signaler till en digital representation. Mätresultat från en kretsprototyp visar att den föreslagna kretsarkitekturen är både mindre och mer energieffektiv än tidigare publicerade kretslösningar.

Det tredje bidraget är en konstruktionsmetodik för frekvenssyntes med hjälp av digitala svängningskretsar (oscillatorer). De digitala oscillatorerna genererar en sinusformad utsignal med hjälp av rekursiva algoritmer. Vi visar att prestandan hos digitala oscillatorer, mätt i termer av amplitud- och frekvensstabilitet, till stor utsträckning beror av starttillstånden hos oscillatorerna. Vi visar också att en del starttillstånd tvingar en oscillator att upprepa samma utsignalsekvens om och om igen, vi har då fått vad vi kallar en låst oscillator. Om oscillatorn har låst finns det inte längre någon drift

i amplitud eller frekvens vilka är vanliga problem för rekursiva oscillatorer som inte använder denna metod. För att hitta de optimala startvillkoren för oscillatorerna har en sökalgoritm utvecklats. Denna algoritm har testats noggrannt i datorsimuleringar. En digital oscillator är lämplig att användas för testsignalgenerering för digital-till-analog-omvandlare där kraven på amplitud- och frekvensstabila testsignaler är höga.

Acknowledgments

Firstly, I would like to thank my supervisor Prof. Mark Vesterbacka for the guidance and support he has given me during the work with this dissertation. Also, I would like to thank my co-supervisors Dr. Oscar Gustafsson and Dr. J Jacob Wikner. Your assistance and inputs to my work have been invaluable to me.

I would also like to thank all colleagues, past and present, at Electronics Systems, Linköping University. It has been a pleasure to work with all of you. A special thanks goes to my room mate Joakim Alvbrant for all interesting discussions regarding science and life in general. Also, I would to thank my dear friend Ola Leifler for all discussions and help with typesetting this dissertation.

I would also like thank all my colleagues I have worked with during the years at Ericsson Microelectronics, Infineon Technologies Sweden AB, Acreo Swedish ICT, Sicon Semiconductor AB, Zoran Sweden AB, and Thin Film Electronics AB. In addition to being great colleagues and friends, your professional attitude and experience have meant a lot to me.

My special thanks goes to my parents Ulf Andersson and Viveka Lundmark and also my sister Cecilia Lundmark-Almlöf. Thank you for making me the person I am and thank you for all support you have given me throughout the years.

My very special thanks goes to my family, my wife Karin and my two children Nora and Arvid. Thank you for your very special support and for being who you are.

Contents

Abstract	iii
Acknowledgments	vii
Contents	viii
1 Introduction	1
1.1 Signal Processing in the Analog and Digital Domains	2
1.2 Dynamic Element Matching	4
1.3 Time-to-Digital Converters	5
1.4 Frequency Synthesis using Digital Oscillators	7
1.5 The Work in a Common Context	8
1.6 Papers Included in the Dissertation	9
1.7 Papers Not Included in the Dissertation	10
1.8 Patents	11
2 Data Converters and Performance Measures	13
2.1 Introduction	13
2.2 Digital-to-Analog Conversion	14
2.3 Analog-to-Digital Conversion	17
2.4 Time-to-Digital Conversion	18
2.5 Signal-to-Noise and Quantization Ratio (SNQR)	18
2.6 Static Performance Measures	21
2.7 Frequency Domain Measures	22
3 Dynamic Element Matching	29
3.1 Introduction	29
3.2 Static Mismatch Errors in DACs	30
3.3 Dynamic Element Matching in a 3-level DAC	31
3.4 Extending the DEM Theory to an M -level DAC	33
3.5 Partial Randomization DEM Techniques	36
3.6 DEM with Reduced Glitching	40
3.7 Future Work	43
4 A Vernier TDC With Delay Latch Chain Architecture	45
4.1 Introduction	45

4.2	Exploring the Time-Domain	46
4.3	Digital Phase-Locked Loops, DPLLs	47
4.4	TDC Target Application	48
4.5	Delay-Line Based TDCs	49
4.6	Proposed Vernier TDC Architecture	51
4.7	Digital Support Block	57
4.8	Gray Counter	58
4.9	Simulation Results	59
4.10	Chip Implementation	63
4.11	Measurement Considerations	64
4.12	Measurement Results	67
4.13	Future Improvements	71
5	Digital Recursive Oscillators	73
5.1	Introduction	73
5.2	Recursive Equations and Vector Rotation	74
5.3	Analysis of Recursive Oscillators	75
5.4	Published Oscillators	79
5.5	Steady-State Cycles in Recursive Oscillators	81
5.6	Proposed Search Algorithm	83
5.7	Properties of Locked Oscillators Cycles	83
5.8	Sinusoid Test Signals for Digital-to-Analog Converters	86
5.9	Future Work	92
	Bibliography	95
A	Paper A	103
A.1	Introduction	103
A.2	DEM in DACs	104
A.3	Simulation Results	107
A.4	Conclusions	110
B	Paper B	115
B.1	Introduction	115
B.2	Current-Steering DAC	116
B.3	Oversampling and Interpolating DACs	118
B.4	Dynamic Element Matching in DACs	119
B.5	Simulation Results	121
B.6	Implementation of a PRDEM Structure in a Current-Steering DAC	123
B.7	Conclusions	125
B.8	Acknowledgments	125
C	Paper C	129
C.1	Introduction	129
C.2	Digital-to-Analog Converters	130
C.3	Model of Dynamic Properties in Current-Steering DACs	131

C.4	Dynamic Element Matching Techniques	133
C.5	Implementation of a PRDEM DAC	138
C.6	Comparison of Simulated and Measured Results	139
C.7	Conclusions	141
D	Paper D	145
D.1	Introduction	145
D.2	Proposed TDC Architecture	146
D.3	Measurements	150
D.4	Conclusions	155
E	Paper E	159
E.1	Introduction	159
E.2	Delay Line Based Time-to-Digital Converters	160
E.3	TDC Target Application	161
E.4	Selected TDC Architecture	163
E.5	TDC Implementation	164
E.6	Simulations	167
E.7	Measurements	169
E.8	Conclusions	172
F	Paper F	177
F.1	Introduction	177
F.2	Analysis of Recursive Oscillators	179
F.3	Steady-State Cycles in Recursive Oscillators	185
F.4	Proposed Search Algorithm	186
F.5	Properties of Locked Oscillator Cycles	189
F.6	Comparison of Search Strategies	194
F.7	Conclusion	196

Chapter 1

Introduction

It is often hard to exactly point out the start of a new era, but we know that the electronic revolution started in a physics laboratory at AT&T's Bell Labs in the United States. From November 17, 1947 to December 23, 1947, John Bardeen and Walter Brattain performed experiments leading to the discovery of the transistor, for which they together with William Shockley (also at AT&T) received the Nobel Prize in physics in 1957.

The discovery of the semiconducting transistor paved the way for several important inventions, where the personal computer and the internet often are rated among the top ten most important inventions of all times. The big advantage of the transistor as opposed to earlier technologies, such as the vacuum tube, is that the transistor can be scaled down much more in size allowing for very high system integration. When a transistor is scaled we usually refer to it as process scaling which allows for faster and more power efficient integrated circuits.

A process node is usually named after the smallest transistor length supported by the process and the smallest commercially available technology node (2013) is the 22 nm node which in turn is predicted to be replaced by the 14 nm node in 2014 [1]. It should be noted that only 50 silicon atom layers separate the two terminals (drain and source) in a 22 nm CMOS transistor. The gate oxide thickness in the 22 nm node is even smaller, that is in the order of a few atom layers only. In just above forty years the process scaling has increased the transistor density on a single chip from 2300 transistors in Intel's 4004 processor (1971), to 5 billion transistors in their 62-Core Xeon Phi processor (2012).

A microprocessor (or processor) is a programmable device that process digital data according to given instructions before providing the digital output data. In a personal computer the data is mostly digital but in other systems such as for example a digital radio communication system both analog and digital signals are processed.

To interface between the analog and the digital domain we use data converters. A device converting an analog signal into a digital representation is called an analog-to-digital converter (ADC) and a device converting digital

Figure 1.1: Data converters are the interface between the analog and digital domains.

data into an analog representation is referred to as a digital-to-analog converter (DAC).

In electronics the analog signal usually represents an electric quantity such as a voltage, a current or a charge. Other possible analog representations are for example found in sensor, mechanical or hydraulic systems, where the analog signal represents, e.g., a position, a temperature, or a pressure. How data converters are used to interface between the analog and digital domain are illustrated in Figure 1.1.

1.1 Signal Processing in the Analog and Digital Domains

Signal processing can be performed in either the digital domain or in the analog domain. Which of the domains that is the most beneficial in terms of energy consumption and other performance measures must however be decided for each application. Processing accuracy can be measured using the signal-to-noise ratio (SNR) metric, and a common way to compare performance is to derive the energy consumption for a given SNR. Noise is the limiting factor in both domains and in the analog domain the noise originates from for example thermal fluctuations in the physical devices whereas noise is due to round off errors in the digital domain.

Studies investigating the trade-off between energy consumption and processing accuracy are for example [2, 3]. One conclusion from these investigations is that signal processing in the analog domain can be more energy efficient for low accuracy signal processing. A rule of thumb is that analog signal processing is (theoretically) more energy efficient for SNR values less than 40 dB.

There are however some caveats in these investigations. First, the comparison is theoretical and hence process limitations are for example not taken into account. Secondly, the design time is typically much longer for designing analog systems and thirdly the cost for data conversion between the two domains were not taken into account.

Starting with the process limitations there are some important consequences following from process scaling. While most digital performance measures benefit from process scaling, important analog measures degrades. One such analog measure is the intrinsic gain of the transistors which decreases with each new process node. The intrinsic gain is a good measure on how power efficient analog circuits can be designed and is defined as g_m/g_{ds} , where g_m is the transconductance and g_{ds} is the channel conductance of the transistor. From this perspective, process scaling seems to favor signal processing in the digital domain.

The second caveat, the design time, is always an important factor in product development. If however there are hard requirements on power consumption one might have to consider to implement some functions in the analog domain, despite of the longer design time.

The third caveat, is the energy consumed when converting data between the two domains, which was not taken into account in the derivations in [2, 3]. Energy efficient solutions for data conversion are a key requirement when optimizing the total energy consumption in mixed-mode systems where the signal processing is distributed between the two domains [3]. An example of such a system is described in [4] where the fast Fourier transform (FFT), typically performed in the digital domain, is replaced with an analog counterpart, a so-called analog harmonic transform (AHT).

From the discussion above we conclude that signal processing in the analog domain can be an option for applications with low SNR requirements but also that process scaling seems to favor signal processing in the digital domain. These conclusions however lead to a fourth caveat, not yet mentioned, which is signal processing in the time domain. The theoretical investigations in [2, 3] assumes that information in the analog domain is represented by a voltage or a current. Hence the expressions for SNR and power consumption are typically derived from the voltage amplitude of an analog signal. In the time domain however, the information carrier is a time of phase difference. Hence, even though the time domain is a part of the analog domain, it needs to be treated separately from the conventional analog domain.

Contrary to conventional analog performance measures, the time resolution increases for each new process node. The resolution increases because new process nodes are faster, which is often measured using the so-called cut-off frequency, f_t . In systems using the time domain, phase information is converted to a digital representation using a time-to-digital converter (TDC). In recent years time-domain signal processing has become more and more popular, mainly due to the fact that the performance is expected to increase due to process scaling as discussed above. Circuits using TDCs are for example analog-to-digital converters [5, 6] and digital phase-locked loops (PLLs) as a replacement for the phase comparator [7].

Data converters are and will also in the future be a key component in mixed signal systems. The border between analog and digital will however change, i.e., in which domain the signal processing will be performed. In high performance applications such as for example mobile applications the

Figure 1.2: Illustration of a 3-bit current-steering DAC.

trend is to put as much functionality in the digital domain as possible. In low power applications however, such as the previously mentioned sensor networks [4], the analog domain is an interesting alternative for signal processing.

In this work we suggest and evaluate techniques for efficient data conversion. In Papers A-C we evaluate a technique for increasing the resolution in digital-to-analog converters. This technique is referred to as dynamic element matching (DEM) and will be briefly outlined in Section 1.2.

In Papers D and E we propose a new power efficient TDC architecture. The architecture uses a so-called Vernier delay-line and will be discussed in Section 1.3.

The third contribution in this work is frequency synthesis using digital oscillators. The origin of this research topic was the need to generate fast and accurate test signals for DACs. The same oscillators can however also be used in radio communication systems where accurate sinusoidal signals are required to modulate the signals up or down in frequency [8]. The basic principles of digital oscillators are discussed in Section 1.4.

1.2 Dynamic Element Matching

This section briefly describes the functionality of a digital-to-analog converter and also the proposed dynamic element matching (DEM) technique. Data converters are discussed in more detail in Chapter 2 and the DEM technique is discussed in Chapter 3.

Digital-to-analog converters use a set of internal analog references when converting a digital input code to an analog waveform. These references are for example current sources or resistors. A 3-bit current steering DAC is illustrated in Figure 1.2. The DAC uses three current sources (references) that are scaled in a binary fashion, i.e., 4, 2, and 1 unit currents (I_{unit}) respectively. These currents can be connected to the output via three switches controlled by the three binary bits, x_2 , x_1 , and x_0 as illustrated in the figure. The DAC output can now generate output currents in discrete I_{unit} steps from zero to seven depending on the digital input code.

Figure 1.3: Power spectra for (a) a conventional DAC, and (b) a DAC using DEM.

In an actual circuit implementation however the values of the reference sources will never be exact. These so-called mismatch errors occur during the fabrication of the circuit and puts an upper limit to the performance of high resolution DACs. Matching errors typically result in unwanted distortion terms in the frequency domain as illustrated in Figure 1.3 (a). To reduce the mismatch errors trimming of the reference sources can be used [9, 10]. Trimming are however often associated with extra cost in analog hardware. An alternative to trimming is the so-called dynamic element matching (DEM) technique [11–16].

The main difference between trimming and DEM is that the latter method does not cancel the errors in the references sources. Instead the error is averaged out by manipulating the digital input word. In the frequency domain this corresponds to trading distortion for extra noise. Figure 1.3 illustrates the difference between a conventional DAC and a DAC using DEM. As can be seen in Figure 1.3 (b) the distortion terms seen in Figure 1.3 (a) have been suppressed below the noise floor, but the noise floor level is higher compared to Figure 1.3 (a).

In Paper A different DEM techniques are compared in terms of hardware cost and performance. From this comparison one of the DEM techniques was selected for a circuit implementation. The selected DEM technique and the circuit architecture is described in Paper B. Measurement results and conclusions for the implemented DEM DAC are presented in Paper C.

1.3 Time-to-Digital Converters

Time-to-digital converters (TDCs) are typically used to convert the time difference between the edges of two input signals to a digital output. Many

Figure 1.4: Illustration of a conversion cycle for a delay-line TDC.

types of architectures exist but in this section we focus on the single delay-line TDC.

A single delay-line TDC consists of a number of delay elements connected in series. The outputs from the delay elements are also connected to a sampling register as illustrated in Figure 1.4. The TDC converts the time difference ΔT between the two inputs *start* and *stop*. A complete conversion cycle consists of the following steps, which is also illustrated in Figure 1.4.

The conversion cycle starts with an all-zero state in the delay chain, i.e., all outputs from the delay elements are low. When the *start* input goes high, a pulse (or 1) starts to propagate through the delay chain, gradually setting the inputs to the sampling register high. When the *stop* signal goes high, the input of the sampling register is sampled to the register output. The number of ones, N , at the register output is now linearly dependent on the time difference ΔT between the two edges. The time difference can now be calculated as

$$\Delta T = N\tau, \quad (1.1)$$

where N is the number of ones at the register output and τ is the delay of a single delay element in the delay line.

From the expression in (1.1) we conclude that the resolution or accuracy of which the TDC can measure time is limited by the delay of a single delay element. Hence we are not able to measure time differences which are fractions of τ . One solution to this problem is to use a so-called Vernier delay

line TDC where the *stop* signal propagates through a second delay line [17]. The resolution is now given by the delay difference of the unit delays in the two delay lines.

In Paper D we propose a new Vernier TDC architecture where the D flip-flops commonly used in the sampling register are replaced by a delay latch. The proposed architecture allows for both power and hardware efficiency improvements. An 8-bit TDC using the proposed architecture has also been implemented and measurement results are presented in Paper E. Details on the implementation and measurement results for the chip prototype are also found in Chapter 4.

1.4 Frequency Synthesis using Digital Oscillators

Frequency synthesis is an important part in most electronic systems. Signals with predictive and stable frequencies are for example used as clocks in digital circuits and in radio systems to modulate the baseband signal to a higher (carrier) frequency before transmission.

In this work we focus on frequency synthesis using digital oscillators. Digital oscillators use recursive equations to derive a sinusoidal output, i.e., the next value in a sequence is derived from previous values in the same sequence.

A sinusoid can for example be derived using the following equation

$$y_n = \alpha \cdot y_{n-1} - y_{n-2}, \quad (1.2)$$

where the output y_n is derived by multiplying the previous output y_{n-1} with a multiplier coefficient α and finally subtracting the second previous value y_{n-2} .

However, when expression (1.2) is implemented using digital circuitry the calculations will be performed with a finite accuracy. The accuracy is restricted by the number of binary bits (wordlength) used to represent numbers in the calculations. In order to fit the result into the pre-defined wordlength, all calculated results must be quantized or rounded. This is similar to what we do when we round the decimal number 1.9 up to 2. There are also other rounding schemes where for example truncation discards the decimal part of a number, i.e., 1.9 is truncated down to 1.

Finite wordlength and rounding effects will introduce errors to the calculations, and hence the output y_n in (1.2) will be different from the ideal output as will be illustrated in the following example. First we assign a value to the multiplier coefficient, which in this example is selected to $\alpha = 119/2^6$, and secondly we need to assign values to the first two outputs in the sequence, i.e., y_1 and y_2 . These values are the initial conditions and are in this example set to $y_1 = 0$ and $y_2 = 10/2^6$, respectively. Given these initial conditions the output values can be calculated using the equation in (1.2).

In Figure 1.5 we compare two scenarios, that is with and without rounding effects. As can be seen in the figure the two sequences quickly diverge.

Figure 1.5: Illustration of a recursive oscillator with and without quantization effects.

What can also be seen is that the first and last values are the equal for the sequence derived with rounding, i.e., $y_1 = y_{17}$. If we continue to derive this sequence we will see that the next value is equal to the second value in the sequence, $y_2 = y_{18}$. Hence, in this example, the sequence $y_1 \rightarrow y_{17}$ will repeat over and over again.

In digital oscillators this effect is called locking, or steady-state, and can be used to generate stable sinusoids with predictive frequencies. However, not all initial conditions result in steady-state where the output sinusoid fulfills other performance specifications such as for example spectral purity. Search algorithms to find useful steady-state cycles is the main contribution in Paper F where we also extend the basic theory on digital oscillators.

Another suitable application for digital oscillators are test signal generation for DACs. Suggestions of how to choose good test signal frequencies and how these can be generated are further discussed in Chapter 5.

1.5 The Work in a Common Context

This dissertation targets the interface between the digital and the analog domains. Where this interface should be placed in a mixed signal system for optimal performance must however be decided from application to application as discussed in Section 1.1. An example on how the papers in this dissertation fit in a common mixed signal system, in this case a direct-RF radio architecture, is illustrated in Figure 1.6.

In certain radio systems two digital input streams I and Q are modulated using a digital quadrature oscillator. A design methodology for designing hardware efficient, high performance digital oscillators is proposed in Paper F. After modulation the two data streams are added before conversion into an analog waveform in the DAC. How the resolution can be increased in DACs using the so-called DEM technique is investigated in Papers A-C.

A digital phase-locked loop (PLL) is used to generate a high frequency clock, which in turn is connected to a clock generator block where the high

Figure 1.6: Illustration of how the papers in this dissertation fit a common context.

frequency signal is divided down to generate all frequencies required in the system. A key component in the digital PLL is the time-to-digital converter (TDC). A new hardware efficient TDC architecture suitable for lower power digital PLLs is proposed in Papers D and E.

1.6 Papers Included in the Dissertation

- A. N. U. Andersson and J. J. Wikner, "Comparison of different dynamic element matching techniques for wideband CMOS DACs", in *Proceedings of the 17th Norchip Conference*, 1999

©1999 IEEE. Reprinted, with permission, from N. U. Andersson and J. J. Wikner, Comparison of different dynamic element matching techniques for wideband CMOS DACs, in Proc. of the 17th Norchip Conference, 1999.

- B. N. U. Andersson and J. J. Wikner, "A strategy for implementing dynamic element matching in current-steering DACs", in *Proceedings of Southwest Symposium on Mixed-Signal Design*, 2000, pp. 51–56

©2000 IEEE. Reprinted, with permission, from N. U. Andersson and J. J. Wikner, A strategy for implementing dynamic element matching in current-steering DACs, in Proc. of SSMSD, 2000.

- C. N. U. Andersson et al., “Models and implementation of a dynamic element matching DAC”, *Analog Integrated Circuits and Signal Processing*, vol. 34, no. 1, pp. 7–16, 2003

Springer and the original publisher (Analog Integrated Circuits and Signal Processing, vol. 34, 2003, pp. 7-16, Models and implementation of a dynamic element matching DAC, N.U. Andersson, K.O. Andersson, M. Vesterbacka, and J.J. Wikner), original copyright notice is given to the publication in which the material was originally published, “With kind permission from Springer Science and Business Media.”

- D. N. U. Andersson and M. Vesterbacka, “A Vernier time-to-digital converter with delay latch chain architecture”, *IEEE Trans. Circuits Syst. II*, vol. 61, no. 10, pp. 773–777, Oct. 2014, ISSN: 1549-7747

©2014 IEEE. Reprinted, with permission, from N. U. Andersson and M. Vesterbacka, A Vernier time-to-digital converter with delay latch chain architecture, *IEEE Trans. Circuits Syst. II*, Oct. 2014.

- E. N. U. Andersson and M. Vesterbacka, “Power-efficient time-to-digital converter for all-digital frequency locked loops”, *Analog Integrated Circuits and Signal Processing*, Submitted

- F. N. U. Andersson et al., “Steady-state cycles in digital oscillators”, *IEEE Trans. Circuits Syst. I*, Submitted

1.7 Papers Not Included in the Dissertation

- [1] M. Vesterbacka, M. Rudberg, J. J. Wikner, and N. U. Andersson, “Dynamic element matching in D/A converters with restricted scrambling”, in *Proc. IEEE Int. Conf. Electron. Circuits Syst.*, vol. 1, 2000, pp. 36–39
- [2] M. Rudberg, M. Vesterbacka, N. U. Andersson, and J. J. Wikner, “Glitch minimization and dynamic element matching in D/A converters”, in *Proc. IEEE Int. Conf. Electron. Circuits Syst.*, vol. 2, 2000, pp. 899–902
- [3] K. O. Andersson, N. U. Andersson, J. J. Wikner, “Spectral shaping of DAC nonlinearity errors through modulation of expected errors”, in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 3, 2001, pp. 417–420
- [4] K. O. Andersson, N. U. Andersson, M. Vesterbacka, and J. J. Wikner, “A differential DAC architecture with variable common-mode level”, in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1, 2002
- [5] K. O. Andersson, N. U. Andersson, M. Vesterbacka, and J. J. Wikner, “Combining DACs for improved performance”, in *Proc. 4th IEE Int. Conf. on Advanced A/D and D/A Conversion Techniques and their Applications, ADDA’02*, 2002

- [6] M. Vesterbacka, K. O. Andersson, N. U. Andersson, and J. J. Wikner, "Using different weights in DACs", in *Proc. 4th IEEE Int. Conf. on Advanced A/D and D/A Conversion Techniques and their Applications, ADDA'02*, 2002
- [7] K. O. Andersson, N. U. Andersson, M. Vesterbacka, and J. J. Wikner, "A method of segmenting digital-to-analog converters", in *Southwest Symposium on Mixed-Signal Design*, 2003, pp. 32–37
- [8] K. O. Andersson, N. U. Andersson, M. Vesterbacka, and J. J. Wikner, "A 14-bit dual current-steering DAC", in *Proc. Swedish System-on-Chip Conf., SSoCC'03*, 2003
- [9] A. Jalili, S. M. Sayedi, J. J. Wikner, N. U. Andersson, et al., "Calibration of Sigma-Delta analog-to-digital converters based on histogram test methods", in *Proceedings of the 28th Norchip Conference*, IEEE, 2010, pp. 1–4

1.8 Patents

- [1] M. Rudberg, M. Vesterbacka, N. U. Andersson, and J. J. Wikner, "Scrambler and a method of scrambling data words", pat. US 6462691 B2, 2002

Chapter 2

Data Converters and Performance Measures

2.1 Introduction

Data converters transform information between the analog and digital domains. The analog-to-digital converter (ADC) converts an analog signal to a digital representation and the digital-to-analog (DAC) converter the other way around. The third type of data converter is the time-to-digital converter (TDC). A TDC is essentially an ADC that converts phase information, usually a time difference, to a digital output.

To meet the large range of applications many types of data converters have been developed with different specifications in for example resolution, power consumption, and conversion rate. In the lower performance segment we find for example distributed sensor networks [34] with low requirements on resolution and conversion rate but with high requirements on low power consumption. In the high performance segment we have for example radar and telecommunication applications with high requirements on resolution and conversion rate. The broad range of applications have resulted in the development of a large number of different data converter architectures. Common ADC architectures are for example pipelined, successive approximation and flash ADCs [9, 35–37]. Examples of DAC architectures are current-steering, R2R, and switch capacitor DACs [9, 35–38]. Also TDCs are implemented using different architectures such as the single delay-line, the differential Vernier, or looped architectures [17].

Even though both function and architectures differ between the data converters, they all share the same basic performance measures. The performance measures are used to characterize the converter for different input signals and working conditions. The performance measures are usually divided into static and dynamic performance measures. Static performance measures includes for example the differential-non linearity (DNL) and integral non-linearity (INL) measures, whereas the dynamic measures includes conversion rate, power consumption and signal-to-noise ratio (SNR).

Figure 2.1: Black box representation an N -bit (a) digital-to-analog, (b) analog-to-digital, and (c) time-to-digital converter.

Using a black box representation, the functions of the three different data converters can be illustrated as shown in Figure 2.1. Digital-to-analog conversion is illustrated in Figure 2.1 (a) and will be further discussed in Section 2.2. Figure 2.1 (b) illustrates analog-to-digital conversion, which is discussed in Section 2.3. Time-to-digital conversion is illustrated in Figure 2.1 (c), and is further discussed in Section 2.4. The fundamentals of signal quantization are discussed in Section 2.5, static performance measures in Section 2.6, and frequency domain measures are discussed in Section 2.7.

2.2 Digital-to-Analog Conversion

The ideal digital-to-analog converter as illustrated in Figure 2.1 (a) converts a digital input word D_{in} into an analog output level A_{out} . If the digital input is an N -bit binary coded word, the DAC is referred to as an N -bit DAC. The ideal DC transfer curve for a 3-bit DAC is plotted in Figure 2.2 (a) where each digital input code is mapped to an analog output level. In a linear ideal DAC the amplitude difference between two consecutive codes are equal, i.e., $|A_n - A_{n-1}| = q_s$, where q_s is the quantization step of the converter. For an ideal DAC the quantization step corresponds to an LSB change in the digital input code. In a typical application the digital word is input to the DAC at uniformly spaced time points. Hence, the DAC output is held at a constant value between the samples, as illustrated in Figure 2.2 (b). Hence, the DAC reconstructs the signal using rectangular pulses [39].

If rectangular pulses are used to reconstruct a uniformly sampled analog signal, i.e., the digital input D_{in} , the output spectrum from the DAC is weighted with the sinc function [38].

Figure 2.2: Plot of (a) output amplitude level as a function of input code, and (b) output held constant between samples for an ideal DAC.

Figure 2.3: Illustration of the repeated spectrum and sinc-weighting due to zero-order hold.

Another consequence of the signal reconstruction, using Poissons's formula, is that the output spectrum of the DAC is repeated at multiples of the Nyquist frequency. The transfer function for the sinc-function in the frequency domain is plotted in Figure 2.3 where also the repeated signal spectra are indicated.

As can be seen in the figure, the sinc function attenuates some of the repeated signal spectra. This filtering alone is however not enough in many applications where a so-called image rejection filter is used to filter out the remaining images. We can also see that the sinc attenuates frequencies that are within the Nyquist frequency, i.e., half the sampling frequency, as much as 3.9 dB. In some systems this effect is compensated for using digital pre-distortion of the input signal.

Table 2.1: Binary and thermometer code covering decimal values 0 to 7.

Decimal	Binary	Thermometer
0	000	0000000
1	001	0000001
2	010	0000011
3	011	0000111
4	100	0001111
5	101	0011111
6	110	0111111
7	111	1111111

2.2.1 DAC Codes

For digital-to-analog conversion we need a number of reference levels or weights that are controlled by the digital input bits. These set of weights will in this dissertation also be denoted a DAC code. The input bits select the weights that should be combined to represent a certain digital code at the output. The choice of DAC code is important since it has been shown that it affects both static performance such as DNL [40], as well as dynamic performance measures of the DAC such as for example glitch energy [41, 42].

A generalized digital-to-analog conversion performs in the memory-less (static) case the following operation

$$A(nT) = \sum_{k=1}^K w_k \cdot x_k(nT) \quad (2.1)$$

where w_k is the weight and $x_k(nT)$ is the bit corresponding to bit k . The weights, w_k , can be chosen to be arbitrary as long as we are able to represent all values of A between zero and the sum of all weights w_k . If this requirement is fulfilled, the set of weights $\{w_k\}$ are said to be complete.

To fulfill this requirement, the weights w_k must fulfill Brown's criterion [43] from which we use the corollary in [44] that a sequence $\{w_k\}$ of non decreasing integers is complete if

$$w_1 = 1 \quad \text{and} \quad w_{k+1} \leq 2w_k. \quad (2.2)$$

where w_k corresponds to the k -th DAC weight. From (2.2) we get two extreme codes, that is the binary code where the ratio of two consecutive weights, w_k , is exactly two, and the thermometer code having all weights $w_k = 1$. Table 2.1 illustrates the binary code and the corresponding thermometer code for three binary bits corresponding to the decimal values 0 to 7.

The operation in (2.1) is illustrated in Figure 2.4, where a set of weights, w_k , is multiplied by the input word X where each bit in X can be assigned to the values $x_k \in \{0, 1\}$.

Figure 2.4: Illustration on the general DAC conversion.

The thermometer code is ideal with respect to glitch performance [41, 42] but for larger values of N the encoder complexity might become too large. As a trade off between glitch performance and decoder complexity we can choose to segment the converter [9], i.e., use the binary code for some of the lower significant bits and the thermometer code for the remaining of the bits.

Another important property of a DAC code is code redundancy. In a redundant code there are many combinations of weights w_k that give the same output value. Redundancy allows the use of randomization techniques, such as for example the dynamic element matching technique described in Chapter 3. Typically, it also gives us a higher degree of freedom when designing the circuits, in terms of matching, supply and bias distribution, component sizing, etc. Again the binary and the thermometer codes are the two extremes where the binary code has no redundancy and the thermometer code offers the highest degree of redundancy.

In addition to the binary and thermometer codes other codes have been proposed such as for example the linear code [45–47] and the Fibonacci code [29, 48, 49]. These codes are however not treated further in this dissertation.

2.3 Analog-to-Digital Conversion

The ideal analog-to-digital converter illustrated in Figure 2.1 (b) converts an analog input signal, A_{in} into a digital output word D_{out} . A refined model of the ADC typically consists of a sample and hold circuit followed by a quantizer as shown in Figure 2.5. The sample and hold is not a mandatory function in all types of ADCs, but it is required in for example the successive approximation ADC [9], where the quantizer requires several clock cycles to convert the data.

The operation of the ADC in Figure 2.5 is as follows: The input signal, A_{in} , is sampled and held constant for the time required for the quantizer to convert the intermediate signal A_{sh} to a digital representation D_{out} . Assuming that the digital output word is binary coded, the number of bits in the output word, N , are equal to the resolution of the converter. The transfer function for a 3-bit ADC is a stair case function as illustrated in Figure 2.6.

Figure 2.5: ADC model with sample and hold and quantizer function.

Figure 2.6: Transfer function for an ideal 3-bit ADC.

2.4 Time-to-Digital Conversion

A time-to-digital converter converts phase or time information into a digital output word. The time information can for example be the time difference ΔT between the rising edges of two input signals A_{in} and B_{in} as illustrated in Figure 2.1 (c). If the output word D_{out} is an N -bit binary word the TDC is referred to as an N -bit TDC. Since the TDC essentially is an ADC, they both share the same transfer function illustrated in Figure 2.6.

2.5 Signal-to-Noise and Quantization Ratio (SNQR)

Even though a DAC does not perform any quantization of the input signal as such, the limited resolution in the digital input gives a quantized signal at the DAC output as discussed in [38]. This allows us to derive the quantization error and similar performance measures in a similar way for DACs, ADCs and TDCs. In the following derivation the transfer function of the ADC will be used as the reference.

As previously discussed in Section 2.3, the input signal A_{sh} in Figure 2.5 is sampled and held constant by the sample and hold circuit. The intermedi-

Figure 2.7: Plot of (a) a quantized input ramp,, with (b) corresponding quantization error.

ate signal after the sample and hold, A_{in} , is quantized into 2^N equally large quantization steps where N is the number of bits in the ADC. The quantization of A_{sh} is illustrated in Figure 2.7 (a) where the dotted line is a continuous analog ramp and the solid line is the input ramp quantized into discrete amplitude levels. The smallest distance between two quantization levels is referred to as the quantization step, q_s , and is given by

$$q_s = \frac{A_{FS}}{2^N} \quad (2.3)$$

where A_{FS} represents the full scale analog amplitude level of the ADC and N is the number of bits in the ADC. The full scale amplitude level of the ADC is the maximum input that can be applied to the converter without saturating the converter output. The difference between the analog input and the digital output is called the quantization error, q_e , and is plotted in Figure 2.7 (b). The range of the quantization error should be kept within the following range

$$-\frac{q_s}{2} < q_e < \frac{q_s}{2} \quad (2.4)$$

for full N -bit resolution. This can also be interpreted as that the ADC should keep the absolute quantization error within one least significant bit, LSB, of the digital output code, D_{out} . There are other classes of ADCs that use nonlinear quantization schemes [50] where for example finer steps are used for the small and medium codes while coarse steps are used for large input codes. This technique can be beneficial when statistical knowledge of the input signal is available. If a full scale, or near full scale input is very unlikely to happen, a more coarse quantization for near-fullscale codes can be used without increasing the bit error rate much. We will however in this work

restrict us to ADCs with linear transfer functions, i.e., the quantization steps are equal for all adjacent codes.

By assuming equal quantization steps we can derive a theoretical maximal value of the signal-to-quantization noise ratio (SNQR) for an N -bit ADC. A quantization error, q_ϵ , that is uniformly distributed in the interval given by (2.4), has a mean-squared noise value given by

$$\langle q_\epsilon^2 \rangle = \frac{1}{q_s} \int_{-q_s/2}^{q_s/2} q_\epsilon^2 dq_\epsilon = \frac{q_s^2}{12} \quad (2.5)$$

where q_s is the quantization step as given in (2.3). Since sinusoidal inputs are commonly used to characterize the performance of the ADC, it is interesting to derive the power ratio between a full scale sinusoidal input and the quantization noise, giving us the so-called signal-to-quantization noise, SQNR of an N -bit ADC. A full scale sinusoid has the amplitude $A_{FS}/2$, i.e.,

$$A_{sig} = \frac{A_{FS}}{2} \sin(\omega t + \phi) \quad (2.6)$$

where ω is the angular frequency and ϕ is a constant phase shift of the sinusoid. The input signal given in (2.6) has a mean-square value given by

$$\langle A_{sig}^2 \rangle = \frac{1}{2\pi} \frac{A_{FS}^2}{2^2} \int_0^{2\pi} \sin^2(\omega t + \phi) d(\omega t) = \frac{A_{FS}^2}{8}. \quad (2.7)$$

Using the relation given in (2.3), we can rewrite (2.7) according to

$$\langle A_{sig}^2 \rangle = \frac{A_{FS}^2}{8} = \frac{q_s^2 2^{2N}}{8}. \quad (2.8)$$

The SQNR is now derived as the ratio of the root mean-squared value of the signal and the quantization noise,

$$SQNR = \frac{\langle A_{sig} \rangle}{\langle q_\epsilon \rangle} = \frac{q_s 2^N / 2\sqrt{2}}{q_s / \sqrt{12}} = \frac{\sqrt{3}}{\sqrt{2}} 2^N \quad (2.9)$$

which in decibel scale equals

$$SQNR = 20 \log \left(\frac{\sqrt{3}}{\sqrt{2}} 2^N \right) \approx 6.02N + 1.76. \quad (2.10)$$

Note that we in the SQNR derivation have assumed that the quantization error, q_ϵ , has a rectangular (uniform) distribution. This approximation holds for converters with more than nine bits of resolution, $N \geq 9$, as shown in [51]. For converters with lower resolution the approximation in (2.10) becomes less accurate.

It should also be noted that the expression in (2.10) is valid only for Nyquist range ADCs, if oversampling is used, or oversampling in combination with noise shaping, other expressions will apply [35]. Oversampling in combination with noise shaping is commonly used in so-called sigma delta ADCs [9, 35, 36].

2.6 Static Performance Measures

Static performance measures are used to characterize a data converter for a DC, or slowly varying input signal and are typically used to measure matching errors in the reference levels of the converter. These matching errors occur in the manufacturing of electronic circuits and typically reduce the converter performance at low conversion rates [9, 35, 36]. Commonly used static performance measures are the differential and integral nonlinearity errors (DNL/INL) which will be defined in Section 2.6.1.

2.6.1 Differential and Integral Nonlinearity (DNL/INL)

When investigating the quantization error in Section 2.5 we assumed an ideal quantization of the input signal, i.e., all steps in the transfer function in Figure 2.6 (a) were equally large. Matching errors in the converter will however cause the step sizes to deviate from the uniform staircase in Figure 2.6 (a), resulting in gain and offset errors of the converter. Static nonlinearity of a converter is described by the differential and integral nonlinearities (DNL/INL). Figure 2.8 illustrates how the DNL and INL are defined for a data converter having matching errors in the reference levels.

The DNL describes how much the difference between two adjacent codes deviates from the ideal quantization step q_s , whereas the INL describes how much each code deviates from the ideal staircase. The DNL can be calculated in terms of a quantization step, or LSB, as

$$\text{DNL}_i = \frac{A_{\text{in},i+1} - A_{\text{in},i} - q_s}{q_s}. \quad (2.11)$$

The INL can be expressed in a similar way as

$$\text{INL}_i = \frac{A_{\text{in},i} - \tilde{A}_{\text{in},i}}{q_s}, \quad (2.12)$$

where $\tilde{A}_{\text{in},i}$ is the transition point for the ideal converter and $A_{\text{in},i}$ is the actual transition point for each output code i . The INL can also be calculated from the DNL according to

$$\text{INL}_k = \text{INL}_0 + \sum_{i=0}^k \text{DNL}_i. \quad (2.13)$$

Gain and offset errors are often treated separately from stochastic mismatch since they usually can be accepted or corrected for at a higher system level. Therefore, both DNL and INL are commonly derived by comparing the actual transfer function with a best-fit line derived from the actual transfer function rather than comparing with the ideal transfer function. The best-fit line can for example be derived from the actual transfer function using the least square method [52]. When using DNL and INL to investigate the monotonicity of a converter the best-fit method is required as discussed in Section 2.6.2.

Figure 2.8: Illustration of the DNL and INL errors for a ramp input.

2.6.2 Monotonicity

A data converter is said to be monotonic if the output is steady increasing when applying a ramp at the converter input. Monotonicity is an important property since for example a non-monotonic ADC will have missing codes degrading the performance significantly. Monotonicity is guaranteed if the deviation from a best-fit straight line is less than half an LSB. This gives the following requirements on DNL and INL

$$|DNL_k| \leq \text{LSB}, \quad k = 0, 1, \dots, 2^N - 1 \quad (2.14)$$

and

$$|INL_k| \leq 0.5 \cdot \text{LSB}, \quad k = 0, 1, \dots, 2^N - 1 \quad (2.15)$$

where one LSB corresponds to one quantization step, q_s . If the relations in (2.14) and (2.15) are fulfilled the converter is guaranteed to be monotonic. However, the reversed relation does not apply, data converters with a $|DNL| \geq \text{LSB}$ can still be monotonic. If the best-fit approach is not used, gain and offset errors can cause the DNL and INL to be very large. A constant offset error in the converter of for example +10 LSBs will add about 10 LSBs to the INL and hence violating the requirement in (2.15). The transfer function of the ADC can still be monotonic and have a high linearity despite this offset error, but that is not seen in the DNL/INL measures unless the best-fit approach is used.

2.7 Frequency Domain Measures

The DNL and INL measures specified in Section 2.6.1 are useful when characterizing the converter at DC or very low input frequencies. For higher frequencies however it is more convenient to use frequency domain measures such as for example the signal-to-noise ratio (SNR) or the spurious-free

Figure 2.9: Illustration of common frequency domain measures for a single-tone spectrum.

dynamic range (SFDR). Dynamic measurements are usually carried out by applying a single tone sinusoid at the input of the converter, but also dual and multi-tone test signals are used. In Sections 2.7.1 – 2.7.4 the most commonly used single-tone frequency domain measures are defined. Dual-tone tests such as intermodulation distortion (IMD) are discussed in Section 2.7.5.

Before going into detail on the different frequency domain measures we start by identifying some basic properties of a typical single-tone frequency spectrum. From the spectrum in Figure 2.9 we can identify the fundamental tone, harmonic distortion terms and the noise floor. Harmonics are signal dependent errors and are found at integer multiples of the input signal frequency. The first harmonic is equal to the fundamental tone whereas the remaining terms are so-called overtones where the first overtone equals the second harmonic and so on.

Since ADCs and DACs normally are sampled systems, all signals with a frequency larger than half the Nyquist frequency are folded back at half the Nyquist band [35]. The folding effect can be seen for the 5th harmonic in Figure 2.9, where it has been folded back at half the Nyquist band ending up between the 3rd and 4th harmonic.

The exact frequency positions of the harmonics, taking folding into account, are given by

$$f_h(k) = \frac{f_s}{2} - \left| \frac{f_s}{2} - \text{mod}(kf_0, f_s) \right|, \quad k = 1, 2, 3, \dots \quad (2.16)$$

where $f_h(k)$ is the k -th harmonic, f_s is the sampling frequency, f_0 is the single tone input frequency and $\text{mod}()$ is the modulo (remainder after division) operator. Optionally (2.16) can be written as

$$f_h(k) = \frac{f_s}{2} - \left| \frac{f_s}{2} - kf_0 + f_s \left\lfloor \frac{kf_0}{f_s} \right\rfloor \right|, \quad k = 1, 2, 3, \dots \quad (2.17)$$

where $\lfloor \cdot \rfloor$ is the floor operator.

2.7.1 Harmonic Distortion (HD_k), and Total Harmonic Distortion (THD)

The harmonic distortion, (HD_k), is given by the power ratio of the k -th harmonic and the fundamental tone, i.e., the first harmonic, which in logarithmic scale is given by

$$HD_k = 10 \log \left(\frac{P_k}{P_1} \right), \quad (2.18)$$

where P_1 is the power of the fundamental, and P_k is the power of the k -th harmonic.

The total harmonic distortion (THD) is the ratio between the fundamental and the sum of all harmonics above the fundamental and is given by

$$THD = 10 \log \left(\frac{\sum_{k=2}^{\infty} P_k}{P_1} \right), \quad (2.19)$$

where P_1 again is the power of the fundamental and P_k is the power k -th harmonic. Usually the THD is derived for a limited number of harmonics, typically only for harmonics large enough to be distinguished from the noise floor in the output spectrum.

2.7.2 Signal-to-Noise Ratio (SNR)

The signal-to-noise ratio (SNR) is the power ratio between the fundamental and the total noise power within a specified frequency band,

$$SNR = 10 \log \left(\frac{P_s}{P_n} \right), \quad (2.20)$$

where P_s is the power of the fundamental and P_n is the integrated noise power. Since the signal power from the harmonics (distortion terms) are omitted in the SNR calculation, the SNR can be difficult to define and measure. The reason is that it can be hard to separate distortion terms from the noise floor. A better measure is the SNDR defined in Section 2.7.3.

2.7.3 Signal-to-Noise-and-Distortion Ratio (SNDR)

If the distortion terms are included in the SNR calculation in Section 2.7.2, we get the signal-to-noise-and-distortion ratio (SNDR) which is given by

$$SNDR = 10 \log \left(\frac{P_s}{P_n + \sum_{k=2}^{\infty} P_k} \right), \quad (2.21)$$

where P_s is the power of the fundamental, P_n is the integrated noise power, and P_k is the power of the k -th distortion term. The SNDR can also be derived for a specified frequency band and in that case only harmonics falling into the band of interest should be included in (2.21).

2.7.4 Spurious-Free Dynamic Range (SFDR)

The spurious free dynamic range (SFDR) is the power ratio between the fundamental and the largest (unwanted) harmonic within a specified frequency band and is given by

$$\text{SFDR} = 10 \log \left(\frac{P_s}{P_{d,\max}} \right), \quad (2.22)$$

where P_s the power of the fundamental and $P_{d,\max}$ is the power of the largest harmonic. Since signal powers usually are given in decibels, the SFDR can easily be found from the power spectrum by measuring the distance between the peak of the fundamental and the peak of the largest harmonic as illustrated in Figure 2.9.

2.7.5 Intermodulation Distortion (IMD)

Intermodulation or intermodulation distortion (IMD) occurs when dual or multi-tones are inputs to a non-linear system, such as for example a data converter or an amplifier. The frequency positions for these harmonics are not limited to be integers of the input fundamentals but are also the sums and differences of them. Hence, IMD will occur at frequencies close to the fundamental tones and are hereby more difficult to filter out from the signal band. It should be noted that not only input signals result in IMD but also interfering signals arising from for example crosstalk result in IMD. These crosstalk induced IMD terms also appear close to the fundamental tones and are hence hard to filter out. This motivates why dual and multi-tone tests are very important when characterizing data converters.

The frequency positions for the IMD terms can be derived by modeling the transfer curve of a non-ideal data converter using power series expansion. For the general case, the output Y of the converter is then given by

$$Y = a_0 + a_1 X + a_2 X^2 + a_3 X^3 + \dots + a_k X^k, \quad (2.23)$$

where X is the input to the converter and a_i are the polynomial coefficients. If the input X is a single tone sinusoid with the fundamental frequency f , the non-ideal converter will produce harmonics at $2f, 3f \dots$ etc. If however the input is a two-tone input, i.e., the input X is given by

$$X = \beta_1 \sin(2\pi f_1 t) + \beta_2 \sin(2\pi f_2 t), \quad (2.24)$$

both harmonic distortion and intermodulation will be produced. Inserting the input signal X as defined in (2.24) in (2.23), the output Y is given by

$$\begin{aligned} Y = a_0 &+ a_1(\beta_1 \sin(2\pi f_1 t) + \beta_2 \sin(2\pi f_2 t)) \\ &+ a_2(\beta_1 \sin(2\pi f_1 t) + \beta_2 \sin(2\pi f_2 t))^2 \\ &+ a_3(\beta_1 \sin(2\pi f_1 t) + \beta_2 \sin(2\pi f_2 t))^3 + \dots \end{aligned} \quad (2.25)$$

where a_0 is the DC component from the converter, the a_1 term represents the linear (ideal) transfer of the fundamental frequencies f_1 and f_2 , and the remaining of the terms, $a_2, a_3, \dots$, represent the distortion from the converter.

The second order intermodulation distortion terms are found by expanding the quadratic term in (2.25). By using trigonometric identities, the second term can be expanded to

$$\begin{aligned} X^2 &= \frac{\beta_1^2 + \beta_2^2}{2} \\ &+ \frac{1}{2} \left(\beta_1^2 \cos(2\pi t \cdot 2f_1) + \beta_2^2 \cos(2\pi t \cdot 2f_2) \right) \\ &+ \beta_1 \beta_2 \left(\cos(2\pi t(f_1 - f_2)) + \cos(2\pi t(f_1 + f_2)) \right), \end{aligned} \quad (2.26)$$

where the first term is a DC offset, and the second term are ordinary second order harmonics. The third term contains the second order intermodulation distortion terms whose frequencies are given by $f_1 - f_2$ and $f_1 + f_2$.

A similar expansion of the cubic term in (2.25) gives the third order intermodulation terms as

$$\begin{aligned} X^3 &= \frac{3}{4} \left(\beta_1^3 \cos(2\pi f_1 t) + \beta_2^3 \cos(2\pi f_2 t) \right) \\ &+ 2\beta_1 \beta_2^2 \cos(2\pi f_1 t) + 2\beta_1^2 \beta_2 \cos(2\pi f_2 t) \\ &+ \frac{1}{4} \left(\beta_1^3 \cos(2\pi t \cdot 3f_1) + \beta_2^3 \cos(2\pi t \cdot 3f_2) \right) \\ &+ \frac{3\beta_1^2 \beta_2}{4} \left(\cos(2\pi t(2f_1 + f_2)) + \cos(2\pi t(2f_1 - f_2)) \right) \\ &+ \frac{3\beta_1 \beta_2^2}{4} \left(\cos(2\pi t(2f_2 + f_1)) + \cos(2\pi t(2f_2 - f_1)) \right) \end{aligned} \quad (2.27)$$

As can be seen in (2.26) and (2.27) most IMD terms can be filtered out since they appear at frequencies far from the fundamental frequencies f_1 and f_2 . If however the input frequencies are close in frequency, the third-order IMD ($2f_1 - f_2, 2f_2 - f_1$) will be very close to the fundamental frequencies and cannot easily be filtered out from the signal band. Third-order IMD is of most concern in narrow bandwidth applications since they appear very close to the fundamental frequencies and second-order IMD is of greater concern in broad bandwidth applications.

Figure 2.10 illustrates the IMD frequency positions in a normalized frequency scale for a dual-tone input. We can from the figure identify the fundamental frequencies at f_0 and f_1 with the corresponding single-tone harmonics at multiples of these frequencies. We can also identify the third order IMD terms close to the fundamentals. Note that in this particular example the harmonic at $3f_0$ has folded on the second order IMD $f_0 + f_1$. In a similar way $2f_0$ has folded on $2f_0 + f_1$ and are hereby not visible as separate tones in the figure.

There are also different multi-tone tests used to characterize data converters. One example is the multi-tone power ratio (MTPR) [37], which is

Figure 2.10: Frequency domain measures for two tone input.

of special interest when the converter are used in a communication system. Multi-tone tests will however not be treated further in this dissertation.

2.7.6 Single-Shot Precision

The single-shot precision is mainly used to characterize TDCs [17], but similar tests exist for ADCs. When testing ADCs these tests are referred to as DC input, or constant input tests. The reason for using the single-shot precision test for TDCs is the difficulty to generate a single tone input with high enough linearity. Since the input to the TDC is a phase difference the input must be frequency modulated which are harder to generate than for example a single-tone sinusoid in the voltage domain.

In a single-shot precision test a constant phase or time difference is supplied to the converter inputs. This should ideally generate a constant output, but the presence of noise and other interfering signal sources give an output having a statistical distribution. The standard deviation of this distribution is called the single-shot precision. The single-shot precision is typically code dependent and hence should be measured for all input codes to fully characterize the converter.

Chapter 3

Dynamic Element Matching

3.1 Introduction

The static performance of digital-to-analog converters (DACs) is typically limited by matching errors in the DAC's reference sources. Mismatch errors occur during the circuit fabrication and several techniques have been proposed to trim or calibrate the references in order to reduce the impact of these errors. One technique for on-line calibration of the unit current sources in a current-steering DAC is proposed in [9] and in [10] where the threshold voltages are adjusted to trim the currents.

As an alternative to trimming, the so-called dynamic element matching (DEM) technique have been proposed [11–16]. As opposed to calibration, DEM does not cancel the errors in the references but instead changes nature of the error. The objective of the DEM algorithm is to transform a signal dependent error, which results in harmonic distortion in the frequency domain, into uncorrelated noise. By transforming harmonic distortion into noise the SFDR performance will increase. The SNDR however will not change since the total error power within the Nyquist frequency band remains constant. To improve the SNDR performance of a DEM DAC, oversampling or noise shaping techniques can be applied. These techniques are referred to as noise shaping DEM. The performance of noise shaping DEM is compared to the other DEM techniques in Papers A-B, and are not discussed further in this chapter.

While DEM is able to reduce harmonic distortion for lower update frequencies, other dynamic effects tend to limit the performance for higher frequencies. This leads to the conclusion that the DEM technique does not necessarily increase the performance of a DAC when dynamic errors are dominating the achievable performance. This trade off between the degree of DEM and actual gain in harmonic performance is investigated in Paper C. In this paper we present a model describing the dynamic properties of a DEM DAC and compare the simulated results with measurements of a 14-bit current-steering DEM DAC implemented in a 0.35 μm CMOS process. The measured data agrees well with the results predicted by the model.

Figure 3.1: Illustration of general conversion from digital to analog.

One drawback with the conventional DEM techniques is that it counter effects the good glitch performance of the thermometer code. To overcome this problem a new DEM algorithm was proposed in [24, 25, 33]. The glitch performance of the algorithm equals the performance of the thermometer code, and the DEM performance is similar to the conventional DEM algorithm.

This chapter is organized as follows. In Section 3.2 the transfer function of a DAC with mismatch in the reference sources is derived. Sections 3.3-3.4 explains the theory behind the DEM techniques. Partial randomization DEM is described in Section 3.5 and the glitch minimizing DEM technique is described in Section 3.6.

3.2 Static Mismatch Errors in DACs

In this section we derive the transfer function for a DAC with mismatch in the reference sources. This transfer function will be used later to illustrate how the DEM algorithms transforms the mismatch error from being signal dependent distortion into uncorrelated noise.

As previously shown in Chapter 2 (Sec. 2.2), a generalized digital-to-analog conversion performs in the ideal memory-less case the following operation

$$A(nT) = \sum_{k=1}^K w_k \cdot x_k(nT) \quad (3.1)$$

where w_k is the (analog) reference weight and x_k is the bit corresponding to bit k . This operation was illustrated in Figure 2.4 but is repeated in Figure 3.1 for convenience. In Figure 3.1 a set of weights, w_k , is multiplied with the input word x where each bit in x can be assigned to the values $x_k \in \{0, 1\}$. A fundamental requirement for the DEM algorithm is that a redundant code is used. The code with the highest degree of redundancy is the thermometer code where all weights w_k are equally large. By assuming that the thermometer weights ideally are equal to the quantization step q_s of the DAC we get

$$w_k = q_s, \quad k = 0, 1, \dots, 2^N - 2. \quad (3.2)$$

However, due to imperfections in the processing of microelectronic circuits the weights w_k will deviate from their ideal values. By introducing a statisti-

Figure 3.2: Illustration of 3-level DEM DAC.

cal mismatch variable δ_x the reference weights are now given by

$$w_k = q_s + \delta_k, \quad k = 0, 1, \dots, 2^N - 2. \quad (3.3)$$

where δ_k is the static mismatch error for the k -th reference weight. By replacing w_k as defined in (3.3) the transfer function in (3.1) now expands to

$$\begin{aligned} A(nT) &= \sum_{k=1}^K w_k \cdot x_k(nT) = \sum_{k=1}^K (q_s + \delta_k) \cdot x_k(nT) \\ &= q_s \sum_{k=1}^K x_k(nT) + \sum_{k=1}^K \delta_k \cdot x_k(nT) \end{aligned} \quad (3.4)$$

where the first sum represents the ideal DAC output and the second sum represents the sum of all mismatch errors associated with input code x .

3.3 Dynamic Element Matching in a 3-level DAC

In this section we will show how dynamic element matching works for a 3-level DAC. A similar derivation is made in [53], whereas an alternative way to generalize the theory to also include DEM DACs of any resolution is suggested here.

To keep the notation simple, we will in the remaining of this section view the DAC as a transfer function mapping an input x to an output y , and hence disregard the time dependency. The transfer function in (3.4) can now be written as

$$y(x) = q_s \sum_{k=1}^K x_k + \sum_{k=1}^K \delta_k \cdot x_k. \quad (3.5)$$

A 3-level DEM DAC can be implemented using a thermometer encoder, a scrambler, and two 1-bit DACs as illustrated in Figure 3.2. The DEM DAC has a digital input x that can take the integer values $x \in \{0, 1, 2\}$. The input is connected to a thermometer encoder converting the input to a 2-bit thermometer code t , which in turn is connected to a scrambler controlled by a switch signal s . If $s = 0$ the thermometer bits are directly bypassed to the output, and if $s = 1$ the bits are swapped. The output of the scrambler controls two 1-bit non-ideal DACs with a nominal quantization steps q_s and mismatch errors δ_1 and δ_2 respectively.

Table 3.1: Output values for the 3-level DEM DAC.

x	s	t_2	t_1	x_2	x_1	$y(x)$
0	-	0	0	0	0	0
1	0	0	1	0	1	$q_s + \delta_1$
1	1	0	1	1	0	$q_s + \delta_2$
2	-	1	0	1	1	$2q_s + \delta_1 + \delta_2$

The DEM DAC in Figure 3.2 can now take the states tabulated in Tab. 3.1. We note that the inputs $x = 0$ and $x = 2$ give unique DAC outputs y , while $x = 1$ can give two different outputs depending on the value of the switch signal s . The corresponding DAC transfer function is illustrated in Figure 3.3 where the two possible mid-outputs lies above and below the ideal linear transfer function $\tilde{y}(x)$. The ideal transfer function is the straight line drawn between the start point $y(0)$ and the end point $y(2)$. The equation for the wanted transfer function can be derived as

$$\tilde{y}(x) = \frac{\Delta y}{\Delta x} x = \frac{2q_s + \delta_1 + \delta_2}{2} x = \left(q_s + \frac{\delta_1 + \delta_2}{2} \right) x = \tilde{k}x \quad (3.6)$$

where $\tilde{k}$ is the gain, and x the integer value of the input code. A perfectly linear DAC transfer function would have the mid code $\tilde{y}(1)$ also on the line described by (3.6), that is

$$\tilde{y}(1) = q_s + \frac{\delta_1 + \delta_2}{2}, \quad (3.7)$$

where $\tilde{y}(1)$ denotes the ideal midcode value. By referring to the notation in Figure 3.3 we can derive the deviation from the ideal value $\tilde{y}(1)$ for the two actual outputs $y(1)$ and $y'(1)$ as

$$\begin{aligned} \tilde{y}(1) - y(1) &= \frac{\delta_2 - \delta_1}{2} = \varepsilon \\ \tilde{y}(1) - y'(1) &= \frac{\delta_1 - \delta_2}{2} = -\varepsilon \end{aligned} \quad (3.8)$$

hence the actual outputs lies on the same distance from the ideal transfer function. From this result we can conclude that if the switch signal s is a white noise random variable, the DEM DAC will on average have a perfectly

Figure 3.3: Transfer function for a 3-level DEM DAC.

linear transfer function. This is sometimes referred to mismatch-scrambling DEM [53]. Optionally, the statistical properties of s can be changed to spectrally shape the mismatch error. Instead of spreading the noise evenly over all frequencies the noise can be high-pass filtered. Using oversampling and a low pass filter, the SNDR performance will increase in the signal band. This DEM technique is referred to mismatch-shaping or noise-shaping DEM [54]. For a performance comparison of noise shaping DEM and other DEM algorithms without noise shaping, see Papers A-B.

3.4 Extending the DEM Theory to an M -level DAC

From the derivations in Section 3.3 we concluded that a three level DEM DAC on average has a perfectly linear transfer function assuming that the switch signal s has a white noise distribution. In this section that result is extended to also cover converters with any number of reference levels M , hence we assume that we use an M -bit thermometer code. We also assume that we have an ideal M -bit scrambler, i.e., a scrambler controlled by a switch word s which can permute the input bits t in all possible combinations.

If we again assume that the ideal transfer function is a straight line drawn from the start point $y(0)$ and the end point $y(M)$, as illustrated in Figure 3.4 (b), the gain $\tilde{k}$ of the DAC is (using (3.4)) given by

$$\tilde{k} = \frac{\Delta y}{\Delta x} = \frac{y(M) - y(0)}{M - 0} = q_s + \frac{1}{M} \sum_{k=1}^M \delta_k = q_s + \bar{\delta}, \quad (3.9)$$

where $\bar{\delta}$ denotes the average reference error. This gives the ideal transfer function as

$$\tilde{y} = \tilde{k}x = (q_s + \bar{\delta})x. \quad (3.10)$$

Figure 3.4: Illustrations of (a) an M -level DEM DAC, and (b) the corresponding transfer function.

Assuming that we have an ideal scrambler and also that the switch word s is a white noise random variable, the expectation value for an arbitrary input value p is (using (3.4)) given by

$$\begin{aligned}
 E[y(p)] &= E \left[q_s p + \sum_{k=1}^p \delta_k \right] = q_s p + E \left[\sum_{k=1}^p \delta_k \right] \\
 &= q_s p + \sum_{k=1}^p E[\delta_k] = q_s p + p \bar{\delta} = (q_s + \bar{\delta}) p
 \end{aligned} \tag{3.11}$$

which lies exactly on the ideal transfer function given in (3.10). Concludingly, the expectation values for all input values to an M -bit DEM DAC lies on the same ideal transfer function. When DEM is applied to all bits in the DAC as described in this section, it is referred to as full randomization DEM.

Figure 3.5: Illustrations of (a) unary encoder and (b) thermometer encoder.

Table 3.2: Binary, unary, and thermometer code.

Decimal	Binary	Unary	Thermometer
0	000	0000000	0000000
1	001	0000001	0000001
2	010	0000110	0000011
3	011	0000111	0000111
4	100	1111000	0001111
5	101	1111001	0011111
6	110	1111110	0111111
7	111	1111111	1111111

3.4.1 Thermometer Encoders

Since code redundancy is a requirement for the DEM techniques, the binary code commonly used in the remainder of the system must be converted to a redundant code representation. The most commonly used code in DEM DACs are the thermometer code where all weights in the code are equally large. For a true thermometer code the bits are also ordered as illustrated in Table 3.2. Since all bits in the thermometer code have equal weight the bits can be easily scrambled in a binary switch network as will be discussed in more detail in Section 3.6. The encoder complexity for the thermometer code is also relatively small. An example of a 3-to-7 bit encoder is illustrated in Figure 3.5 (b).

If we however know that the thermometer code always will be scrambled in a succeeding switch network, a simplified version of the thermometer code can be used. By copying the bits in the binary code as many times as the corresponding bit weight, we get a non-ordered thermometer code as illustrated in Figure 3.5 (a). The order of the bits are however not important since the bits will be scrambled by the DEM algorithm. This version of the thermometer code will be referred to as a unary code (Tab. 3.2) in the remaining of this chapter.

Figure 3.6: Illustrations of a 15-stage digital PRBS generator.

3.4.2 Pseudo Random Sequence Generators

The switch signal $s(n)$ in Figure 3.4 (a) should ideally have a white noise distribution in order for the DEM algorithm to work properly, as previously discussed in Section 3.4. Several so called true on-chip random signal generators have been proposed, as for example in [55]. The hardware cost for these generators is however quite large since a control algorithm continuously monitoring the output is required.

As an alternative to true random bit generators a pseudo random bit generator (PRBS) can be used [56]. A PRBS generator is a digital state machine which generates long non-repeating cycles at a low hardware cost. Even though the output is not truly random it will still be able to decorrelate the matching errors in the DAC from the input signal if the sequence is long enough. An example of a 15-stage PRBS generator is illustrated in Figure 3.6. The output from the generator is a non-repeating sequence 2^{15} samples long. Note that all PRBS generators of this type have a forbidden all-zero state, and hence the D-flip-flops must have a reset functionality initializing the generator to one of the allowed states.

3.5 Partial Randomization DEM Techniques

For high resolution DACs the complexity of the DEM decoders might be too large if DEM is applied for all input bits. A possible trade off between DEM performance and decoder complexity is to apply the full randomization DEM scheme, as described in Section 3.4, to a number of the MSBs of the converter. This concept is illustrated in Figure 3.7 where M MSBs are connected to the DEM decoder and the remaining K bits are connected to a conventional K -bit DAC. This approach will in the remaining of this chapter be referred to as segmented DEM.

A second approach is to apply a DEM scheme using a binary switch tree as illustrated in Figure 3.8 (a). This DEM scheme is referred to as partial randomization DEM (PRDEM) and a theoretical performance analysis of this scheme is given in [15].

The PRDEM architecture utilizes a binary switching tree containing switching blocks, $S_{k,r}$, where k denotes the layer and r the position of the switching block in the layer. The switching block in Figure 3.8 (b) has one

Figure 3.7: Illustration of segmented DEM applied to M MSBs.

$(k + 1)$ -bit input and two k -bit outputs, as well as a random control bit $c_k(n)$ equal for all blocks $S_{k,r}$ in the k -th layer. Every $c_k(n)$ is a random or pseudo random bit-sequence (PRBS) uncorrelated with the control bits used in all other layers. $S_{k,r}$ has the following function: when $c_k(n) = 1$, the MSB, x_k , of the input is copied k times and mapped to the top output, while the remaining k bits of the input are mapped directly to the k bits of the bottom output. For $c_k(n) = 0$ the situation is reversed.

In PRDEM we introduce switching in a limited number of layers, i.e., in layers b through R (Figure 3.8(a)), where $2 \leq R \leq b$. Since no randomness is introduced in layers 1 through $R - 1$ we can simply substitute these layers by N^{b-N+1} nominally identical DAC banks – each with an R -bit input (Figure 3.8(c)). The LSB of the input controls a unit DAC element, whereas the remaining $R - 1$ bits control an $(R - 1)$ -bit conventional DAC.

A tree with switching in all layers, i.e., layers 1 through b , are thereby terminated by a set of 1-bit DACs and now resembles of the full randomization DEM system in Figure 3.4 (a).

3.5.1 Simulation Results

This section presents simulation results for the segmented DEM and the PRDEM method. Both architectures were modeled in MATLAB and a Gaussian distributed error current with a standard deviation of $\sigma = 0.05$ were added to the unit current sources. All simulation results show the average of 25 statistical outcomes and the input signal was a full scale single tone with a signal length of 2^{16} samples.

Figure 3.9 shows the output spectra for a 14-bit PRDEM DAC with switching in zero, one, and eight layers. When comparing the two upper plots in Figure 3.9, corresponding to zero and one layer of switching, we see that the SFDR performance is increased by 7 dB to the cost of a somewhat higher noise floor. When using eight layers of switching all distortion terms are hidden in the noise floor.

In Figure 3.10 the SFDR and SNDR performance are plotted against the number of switching layers and the number of randomized MSBs for the PRDEM and the segmented DEM DACs respectively. As can be seen in the

Figure 3.8: Illustrations of a (a) general PRDEM structure, (b) binary switching block, and (c) $(R-1)/1$ bit DAC bank.

figure the SFDR performance increase faster for the PRDEM DAC as compared to the segmented DEM. This is because also the LSBs are randomized in the PRDEM DAC. Hence one layer of switching in a PRDEM solution offers more randomization than applying DEM on two MSB bits in a segmented DEM solution.

This comparison however does not take decoder complexity into account. For a fair comparison, the SFDR performance should be plotted as a function of decoder complexity. In addition the design complexity of the DACs connected to the decoder outputs should also be taken into consideration. For the segmented DEM, the DAC is a conventional segmented DAC whereas several sub-DAC banks must be designed for the PRDEM case.

Figure 3.9: Output spectra for different number of switching layers.

Figure 3.10: SFDR and SNDR versus number of switching layers.

3.5.2 Implementation of a PRDEM DAC

Although the DEM algorithm is able to reduce harmonic distortion for lower update frequencies, other dynamic effects tend to limit the performance for higher frequencies. This trade off between the degree of DEM and actual gain in harmonic performance is investigated in Paper C. In this paper we present a model describing the dynamic properties of a DEM DAC and compare the

simulated results with measurements of a 14-bit current-steering DEM DAC implemented in a $0.35\ \mu\text{m}$ CMOS process. The measured data agrees well with the results predicted by the used model.

3.6 DEM with Reduced Glitching

One drawback with the conventional DEM techniques is that it destroys the glitch performance of the thermometer code. The thermometer code is ideal in this respect since in all code transitions sources are exclusively turned on or off. A proper thermometer code transition from three to five, assuming that $S = \{s_7 s_6 s_5 s_4 s_3 s_2 s_1\}$, would hence be $\{0000111\} \rightarrow \{0011111\}$. In this example bits s_4 and s_5 are turned on. If we apply DEM, the transition might instead look like $\{0000111\} \rightarrow \{1111001\}$, i.e., bits $s_{4 \rightarrow 7}$ are turned on and bits $s_{2,3}$ are turned off. Due to timing imperfections in the digital decoders all bits $s_{1 \rightarrow 7}$ might be turned on for a short time resulting in an unwanted glitch at the DAC output. Note that glitches due to random switching may even occur for a constant input signal.

In order to combine the DEM technique with the good glitch performance of the thermometer code, the restricted DEM algorithms were developed [24, 25, 33] for which there is also a patent. The basic principle of the restricted DEM algorithm is to randomize the switching in such way that no sources are turned on and off at the same time. Assuming a code transition from S_1 to S_2 where S_x is a thermometer coded word, the restricted DEM algorithm can be summarized as follows

- If $S_2 > S_1$, i.e., a positive code transition, then $(S_2 - S_1)$ random sources should be turned on while no sources are turned off.
- If $S_2 < S_1$, i.e., a negative code transition, then $(S_1 - S_2)$ random sources should be turned off while no sources are turned on.

Going back to the previous example with a transition from 3 to 5 where 3 is represented by $\{0000111\}$ we are now allowed to turn on two random bits selected from bits s_{4-7} . A straightforward implementation of this algorithm is described in [24]. A second approach to implement the restricted DEM algorithm is by using a switching network [25] as illustrated in Figure 3.11 (b).

The switch network consists of the switch blocks illustrated in Figure 3.11 (a) with; two inputs $a_{n,k}$ and $b_{n,k}$, two outputs $a_{n+1,k}$ and $b_{n+1,k}$, and a random control signal $s_{n,k}$. The decision table for the switch block is found in Table 3.3 and follows a simple rule. If the inputs $a_{n,k}$ and $b_{n,k}$ are equal, switch the output randomly according to $s_{n,k}$, else keep the previous select settings according to z .

As previously discussed in Section 3.4.1, the input to the switch network does not need to be an ordered thermometer code. Instead we copy the binary input bits as illustrated in Figure 3.11 (b). Further we observe that if

Figure 3.11: Generalized cube network

Table 3.3: Decision table for glitch minimizing DEM

a_n	b_n	w	z_n	Comment
0	0	0	s	Randomize the paths
0	1	1	z_{n-1}	Keep previous switch setting
1	0	1	z_{n-1}	Keep previous switch setting
1	1	0	s	Randomize the paths

both inputs to a switch block are equal, the block can be removed. This is illustrated by the gray shaded areas in Figure 3.11 (b).

An interesting property of the switch network in Figure 3.11 is that if $s_{n,k}$ is set to a constant value, the output of the network is a true thermometer code. This can be explained by the fact that the AND and OR gates in the switch blocks forms the thermometer encoder in Figure 3.5 (b). We also observe that if we hard code the internal signal $w_{n,k}$ in the switch blocks, the glitch reduction algorithm is turned off. The result is a conventional DEM with non-restricted switching and we can hereby easily compare the performance of the glitch reducing DEM with the conventional DEM algorithm.

3.6.1 Simulations

Figure 3.12 shows the power spectra for the reduced glitching DEM with switching in zero, one, and 8 layers respectively. A 10-bit DAC with a Gaussian distributed error of $\sigma = 0.05$ were used in the simulations. As can be seen from the figure the SFDR performance increases with the number of switching layers until all distortion terms are suppressed down below the noise floor.

In Figure 3.13 the performance of the reduced glitching DEM algorithm is compared to a DAC using a non-restricted DEM algorithm. The compar-

Figure 3.12: Output spectra for different number of switching layers using the restricted DEM algorithm.

ison is made by hard coding $w_{n,k}$ to zero when the non-restricted DEM is simulated as previously discussed in Section 3.6.

Figure 3.13: SFDR and SNDR versus number of switching layers for the restricted, and the non-restricted DEM algorithm.

As can be seen from the figure, the gain in SFDR performance for the restricted DEM follows the non-restricted DEM. The same holds for the SNDR performance.

3.7 Future Work

The theory behind the DEM technique is well developed and only small contributions have been made during recent years. An important question however remains to be answered. For which applications is the DEM technique a good alternative? Or in other words, when is it beneficial to trade harmonic distortion for noise? For system using oversampling, or noise-shaping in combination with oversampling, there is an obvious gain in performance since the SNDR is increased within the signal band. For Nyquist rate converters however, the SNDR remains constant. To the best knowledge of the author, this question has never really been answered and it would be interesting to further investigate this issue in the future.

Chapter 4

A Vernier TDC With Delay Latch Chain Architecture

4.1 Introduction

In recent years, time domain signal processing has become a promising alternative to signal processing implemented in the voltage or current domains. Time-to-digital converters, TDCs, are for example used in analog-to-digital converters, ADCs [5, 6], and in digital phase-locked loops, PLLs, as a replacement for the phase comparator [7]. By replacing the phase comparator with a TDC, the charge pump and the analog loop filters can be replaced with digital filters and a digital control loop. An extended motivation on why the time domain signal processing is an interesting alternative to voltage or current domain signal processing is given in Section 4.2.

In this chapter we present a new TDC architecture using a delay line and a chain of delay latches. The delay latches replace the functionality of the second delay chain and sample register commonly found in Vernier converters, hereby enabling for power and hardware efficiency improvements. The chapter is organized hierarchically starting with an introduction to digital PLLs in Section 4.3 and details of the TDC target application are given in Section 4.4. The general principles of delay-line based TDCs are given in Section 4.5 and a new TDC architecture is proposed in detail in Section 4.6. To demonstrate the proposed concept an 8-bit TDC has been implemented in a standard 65 nm CMOS process. Simulation results are presented in Section 4.9 and measurement results are presented in Section 4.12. A comparison with previously published TDCs are made in Section 4.12.4 and suggestions on future improvements are given in Section 4.13.

The proposed TDC architecture and measurement results from a 7-bit version of the TDC are summarized in Paper D. Measurement results from an 8-bit version of the TDC and a presentation of the target architecture are found in Paper E.

Figure 4.1: Extracted process parameters for different process nodes, ©2011 IEEE [57].

4.2 Exploring the Time-Domain

Process scaling towards smaller feature size technology nodes allows for faster and more power efficient digital circuits. The applications driving the process scaling are mainly consumer products such as for example computers and hand held devices. A process node is usually named after the smallest transistor length supported by the process and the current technology node (2013) is the 22 nm node. This node is predicted to be replaced by the 14 nm node in 2014 [1].

While most digital performance measures benefit from process scaling, important analog measures degrades. Analog process measures are for example supply voltage and intrinsic gain of the transistors. The maximal supply voltage in combination with the transistor threshold voltage set an upper limit to the achievable signal-to-noise ratio of a data converter, and the intrinsic gain is a good measure on how power efficient analog circuits can be designed. The intrinsic gain is defined as g_m/g_{ds} , where g_m is the transconductance and g_{ds} is the channel conductance of the transistor.

In Figure 4.1 typical process data are plotted as a function of process nodes [57]. Note that the intrinsic gain of a transistor has a bias current dependency not shown in the table. The other process data are also typical in the sense that the table should not be used as exact values but only to show long term scaling trends.

Two major trends can be seen in Figure 4.1, that is the decrease of intrinsic gain, g_m/g_{ds} , and the increase in cut-off frequency, f_t . The intrinsic gain scales approximately by a factor 0.8 for each new process node which

from a data converter perspective results in a lower resolution in the voltage domain. The resolution in the time domain does however increase due to the higher cut-off frequency, f_t , offered by process scaling. As is shown in Figure 4.1 the cut-off frequency scales with a factor 1.5 for each new process node. A higher f_t allows for shorter gate delays and hence a higher precision in time measurement circuitry used in for example time-to-digital converters, TDCs. Due to the above mentioned reasons, the time domain is a promising candidate for analog tasks currently performed in the voltage domain. If analog functionality can be implemented in the time domain, these circuits can fully utilize the benefits of future process scaling.

Time-to-digital converters, TDCs, are a typical example on how analog functionality can be replaced by digital circuitry. TDCs can for example be used in ADCs [5] or in digital PLLs to replace the phase comparator [7].

4.3 Digital Phase-Locked Loops, DPLLs

A phase-locked loop in which the phase comparator has been replaced by a TDC is sometimes referred to as a digital PLL (DPLL) which also is the target application for the TDC presented in this work. No general description on PLLs will be given in this work. For further reading on general PLLs please see for example [35, 58].

Figure 4.2: Top level block diagrams of a (a) divider-, and (b) counter-assisted digital PLL.

Two main categories of digital PLLs exist, that is divider-assisted DPLLs and counter-assisted DPLLs [17], both illustrated in Figure 4.2. In the divider-assisted DPLL shown in Figure 4.2 (a), the DCO output signal $phiClk$ is divided down to match the frequency of the reference frequency $refClk$

before the TDC converts the phase difference between the two signals to a digital representation. Hence, the conversion range requirement for the TDC is given by the period time of the reference signal, which usually is in the order of MHz.

In the counter-assisted digital PLL illustrated in Figure 4.2 (b), a counter counts the number of full periods of $\phi_{\text{ref}}\text{Clk}$ during the corresponding $\phi_{\text{ref}}\text{Clk}$ period. This gives the integer part of the $\phi_{\text{ref}}\text{Clk}/\phi_{\text{ref}}\text{Clk}$ ratio, whereas the TDC quantizes the fractional part of the frequency ratio. The TDC conversion range is hence given by the period time of the DCO, i.e., in the order of GHz. A counter-assisted digital PLL is sometimes referred to as an all-digital PLL (ADPLL) since all blocks in the PLL, except for the DCO, are now replaced by digital circuitry.

4.4 TDC Target Application

The target application for the proposed Vernier TDC is a low power counter-assisted PLL. A top level diagram of the PLL is illustrated in Figure 4.2 (b). The PLL in Figure 4.2 (b) locks the DCO signal $\phi_{\text{ref}}\text{Clk}$ to the reference clock $\phi_{\text{ref}}\text{Clk}$ using the frequency ratio of the two signals as the target value for the digital control loop. The nominal DCO frequency in the PLL is $f_{\phi} = 2.1$ GHz, and the reference clock has a fixed frequency of $f_{\text{ref}} = 54$ MHz.

Figure 4.3: Timing diagram illustrating the frequency ratio calculation performed in the PLL.

Using the timing diagram in Figure 4.3, the frequency ratio $R = f_{\phi}/f_{\text{ref}}$ is derived as follows: The counter counts the number of rising edges, N , of $\phi_{\text{ref}}\text{Clk}$ between two falling edges of $\phi_{\text{ref}}\text{Clk}$, and the TDC measures the time intervals between the falling edges of $\phi_{\text{ref}}\text{Clk}$ and rising edges of $\phi_{\text{ref}}\text{Clk}$, $T_{\lambda i}$. Using the definitions in Figure 4.3, the frequency ratio can now be expressed as

$$R = \frac{f_{\phi}}{f_{\text{ref}}} = \frac{T_{\text{ref}}}{T_{\phi}} = \frac{NT_{\phi} + T_{\lambda 1} - T_{\lambda 2}}{T_{\phi}}, \quad (4.1)$$

Figure 4.4: Block level diagram of the implemented 8-bit Vernier TDC.

where N is the number of rising edges of ϕ_{Clk} , T_ϕ is the period of ϕ_{Clk} , and T_{λ_1} and T_{λ_2} are two consecutive time intervals measured by the TDC.

By introducing a variable $\lambda_i = T_{\lambda_i} / T_\phi$, hence converting the TDC output to be fractions of a ϕ_{Clk} period, (4.1) can be simplified to

$$R = N + \lambda_1 - \lambda_2, \quad 0 \leq \lambda_i \leq 1, \quad (4.2)$$

where T_{λ_1} and T_{λ_2} are derived in the DSP using information of the TDC time resolution.

The nominal frequency ratio is $R \sim 38.9$, hence a 6-bit integer counter is required. The integer counter was chosen to be a Gray-type counter, this since the sampling error of a Gray counter is limited to one LSB. More details on the Gray counter is found in Section 4.8.

With a nominal DCO frequency of $f_\phi = 2.1$ GHz, the conversion range of the TDC is given by $1/2.1 \text{ GHz} = 0.48 \text{ ns}$. Monte Carlo simulations (Sec. 4.9.2) of the TDC showed that the time resolution could vary between 3.5 and 8.5 ps depending on process corner and transistor mismatch. Hence, the number of delay elements required to cover the 0.48 ns conversion range is between 57 and 137 with an expected value of 84 elements. Although it is possible to design Vernier delay-lines of any length, we decided to use two identical 7-bit TDCs connected in series as illustrated in Figure 4.4. The second 7-bit TDC can optionally be disabled by setting the $8bitMode$ signal in Figure 4.4 low. This reduces the power consumption and almost doubles the conversion rate for the TDC.

4.5 Delay-Line Based TDCs

In this section the basics of delay-line based TDCs are presented. Other architectures such as for example looped delay-line, and noise shaping TDCs are not treated in this work. For further reading on other TDC architectures please refer to the literature, as for example [17].

A TDC can be designed using a single delay-line and a set of D flip-flops connected to the outputs of the delay elements as illustrated in Figure 4.5 (a). Although not included in the figure, the thermometer code, t_i , at the outputs of the D flip-flops are converted to binary code using a thermometer-to-binary encoder.

The operation of the single delay-line TDC in Figure 4.5 (a) is as follows: The first pulse is connected to the *start* input of the TDC and will now propagate through the delay-line generating an increasing thermometer code at the inputs of the D flip-flops. After a time period ΔT , the second pulse is connected to the *stop* input of the TDC hereby sampling the outputs of the delay elements. The time difference between the *start* and *stop* pulse is now given by

$$\Delta T = N\tau, \quad (4.3)$$

where N is the number of delay elements the pulse has propagated through before sampling, and τ denotes the unit delay in the delay-line. From (4.3) we conclude that the time resolution for a single delay-line TDC is limited by the gate delay of the delay elements. Note that τ in a real implementation also includes all parasitic delays in the interconnections of the delay elements.

The maximal sampling or update frequency, $f_{\max}$, for the TDC is limited by the total delay in the delay line, T_{tot} , and can be expressed as

$$f_{\max} = \frac{1}{T_{\text{tot}}} = \frac{1}{\tau M} \quad (4.4)$$

where τ is the unit delay in the delay-line and M is the total number of delay elements. To reduce the total delay in the delay-line, T_{tot} , of the single delay-line TDC in Figure 4.5 (a), τ can be reduced by replacing the delay buffers with inverters.

In order to achieve sub gate-delay resolution, the differential Vernier line architecture is a good candidate [59]. A Vernier TDC uses two delay-lines and a sampling register as illustrated in Figure 4.5 (b). The operation of the Vernier TDC is similar to the operation of the single delay-line TDC with the difference that the *stop* pulse propagates through a second delay line with a shorter unit delay.

The resolution of a Vernier TDC can be derived as follows. If the *start* and *stop* signals are separated ΔT in time, the following relation holds when the *stop* pulse has caught up with the *start* pulse

$$\Delta T + \tau_2 N = \tau_1 N \quad (4.5)$$

where τ_1 and τ_2 are the element delays in the *start* and *stop* delay-lines respectively, ΔT is the time difference between the *start* and *stop* signals, and N is the number of delay elements it takes for the *stop* signal to catch up with the *start* signal. By rewriting (4.5), the time difference between *start* and *stop* can be expressed as:

$$\Delta T = N(\tau_1 - \tau_2) = N\tau_{\text{LSB}} \quad (4.6)$$

Figure 4.5: Gate level schematics of (a) single-, and (b) Vernier delay line based time-to-digital converter.

where τ_{LSB} is the time resolution of the Vernier TDC. From (4.6) we conclude that the resolution of a Vernier TDC is not set by the absolute values of the gate delays but the difference between them. As for the single-delay line TDC, the conversion time of the Vernier TDC can be reduced by replacing the buffers in Figure 4.5 (b) with inverters.

4.6 Proposed Vernier TDC Architecture

In Paper D we propose a new TDC architecture consisting of a chain of delay latches and a delay line. The delay latches have unit delays τ_1 and a delay line have unit delays τ_2 is illustrated in Figure 4.6 (a). The delay latches are transparent if the control input is low and they hold their current output values if the control input is high. The delay latches are modeled using buffers and multiplexers with zero delay connected in feedback. A complete conversion cycle for the proposed architecture in Figure 4.6 (a) follows the timing diagram in Figure 4.6 (b) and consists of the following steps where it is assumed that $\tau_1 > \tau_2$.

1. The TDC is prepared for conversion in the reset phase where the *start* and *stop* inputs are low. All delay latches are now transparent.
2. At the next rising edge of the *start* input, a pulse propagates through the delay latch chain gradually increasing the thermometer code at the t_x outputs.

Figure 4.6: Illustration of (a) the proposed delay latch chain Vernier TDC, and (b) a timing diagram for a conversion cycle.

3. At the next rising edge of the *stop* input, a second pulse propagates through the delay line continuously setting the delay latches in hold state.
4. When the *stop* pulse catches up with the *start* pulse, the N th delay latch is non-transparent hereby stopping the propagation of the *start* pulse.
5. The thermometer code, t_x , at the output of the delay latches is now linearly dependent on the time difference, ΔT , between the two inputs.

The delay latches in the proposed architecture can be implemented in a variety of ways using either standard cells or a full custom solution. A hardware efficient circuit is illustrated in Figure 4.7 where the delay latch chain is implemented using dynamic inverters with alternating NMOS and PMOS enable transistors. It works as follows.

When the gate voltage is set high on an NMOS enable transistor the delay latch works as an inverting delay element and when the gate voltage is low the output of the delay latch becomes a floating node hence holding the current voltage value. The PMOS enable transistors works in the same way as the NMOS transistors but with complementary gate voltages.

To match process, voltage and temperature (PVT) characteristics matching transistors are added to the delay line inverters. The matching transistors are always enabled by connecting the NMOS and PMOS transistor gates to supply and ground potentials respectively. Note that all delay latches and delay elements are inverting in the proposed solution, hence every second thermometer code bit is also inverted. This can however easily be corrected for in the succeeding thermometer-to-binary encoder.

Figure 4.7: Detailed implementation of the proposed TDC.

Since the delay latch outputs, t_x , are floating once the enable transistors are turned off, i.e., no path exists to supply or ground. Pull-up/down circuitry are added as illustrated in Figure 4.7. The pull-up/down circuitry have two additional purposes, which is acting as an extra load to ensure that $\tau_1 > \tau_2$, and also work as buffers driving the inputs of the thermometer-to-binary encoder. More details on the leakage is found in Section 4.6.1.

The regularity of the suggested architecture allows for the design of a single slice that is repeated throughout the Vernier delay line. This slice contains two delay stages and is indicated in Figure 4.7 by the twelve transistors having their corresponding transistor widths written out. Each delay stage now requires nine transistors including the pull-up/down circuitry. This can be compared to the standard Vernier TDC architecture in Figure 4.5 (b) that requires 28 transistors per delay stage in an implementation assuming that one D flip-flop uses 24 transistors. Hence the proposed solution reduces the transistor count by 68%.

4.6.1 Leakage in the Dynamic Nodes

As previously discussed in Section 4.6 the output nodes of the delay latches become floating nodes once the enable transistors are turned off. The major leakage is through the NMOS transistors where a high output leaks towards ground hereby changing state from high to low as is illustrated in Figure 4.8.

Figure 4.8: Illustration of the leakage problem in the dynamic output nodes.

Simulations for the worst process corner show that a floating output node changes state from high to low in ~ 17 ns due to leakage. Hence if the converter is updated at frequencies above 58 MHz the problem with leakage could possibly be ignored. To solve the problem with leakage, pull-up/down circuitry was added to the delay latch outputs as shown in Figure 4.7 and Figure 4.8. The extra transistor controlled by the feedback inverter creates a well defined path to supply.

4.6.2 Reset and Edge Detection Circuit

The TDC requires a reset before each conversion cycle and should also measure the time difference between the falling edge of *refClk* and the next rising edge of *phiClk* as shown in the timing diagram in Figure 4.9 (c).

A high level description of a circuit generating a reset before each conversion and also performing the edge detection is illustrated in Figure 4.9 (a) where the *refClk* and *phiClk* are the inputs to the circuit and the *start* and *stop* signals are inputs to the succeeding Vernier TDC.

An efficient implementation of the circuit in Figure 4.9 (a) is shown in Figure 4.9 (b). The circuit uses less hardware than the D flip-flop implementation in Figure 4.9 (a), which also makes it easier to maintain a constant delay between the *refClk* and *phiClk* inputs.

The circuit in Figure 4.9 (b) works as follows: When *refClk* is high, both delay lines are put into reset by discharging the *start* and *stop* nodes. The *en_start* is now charged allowing a pulse to ripple through the delay latch chain at the next rising edge of the *start* node. At the falling edge of *refClk* the *start* node is charged high and the delay latch chain starts to ripple. At the same time, the *nstop* node is discharged through transistors M2 and M3 thus charging the *stop* input. However, since the *en_stop* node is still low, the *stop* delay line will not start to ripple until the next rising edge of *phiClk*.

Figure 4.9: Illustrations of (a) a high level description, (b) circuit implementation, and (c) timing diagram of the reset and edge detection circuit.

One issue with this circuit is that the reset scheme used limits the maximal conversion rate. This is because the reset is active for half a *refClk* period. As a result of this the maximal conversion rate is reduced almost a factor of two assuming a 50% duty cycle of *refClk*. A new reset scheme using a shorter reset pulse induced directly after each conversion cycle should be developed for the next version of the TDC.

Table 4.1: Comparison of 7-bit thermometer-to-binary encoders.

Decoder Type	Hardware Cost	Critical Path	Relative Costs HW/CP ^a
Wallace tree	360 Γ_{MUX}	22 t_{MUX}	1/1
4-level folded WT	174 Γ_{MUX}	16 t_{MUX}	0.48/0.73
MUX-based	120 Γ_{MUX}	6 t_{MUX}	0.33/0.27

^aHardware and critical path, compared to the Wallace tree

Figure 4.10: A thermometer-to-binary encoder based on multiplexers.

4.6.3 Thermometer-to-Binary Encoder Based on Multiplexers

The thermometer-to-binary encoder converts the thermometer code output, t_x , of the Vernier chain in Figure 4.6 (a) to binary code. In this design, an encoder based on multiplexers was chosen. This since previous investigations show that the multiplexer based encoder [51, 60] requires less hardware and also has a smaller critical path as compared to commonly used one-counter solutions as shown below.

Table 4.1 compares three different 7-bit encoders in terms of hardware cost and critical path. The hardware cost is given in units of multiplexer delays, Γ_{MUX} , where it is assumed that a full adder can be implemented with three 2-to-1 muxes. The critical path is therefore measured in 2-to-1 mux delays, t_{MUX} . The encoder based on multiplexers is compared to a standard Wallace tree adder and an improved 4-level folded Wallace tree adder. As can be seen in Table 4.1 the multiplexer based encoder requires only 33% of the hardware compared to a Wallace tree implementation. The critical path is also reduced 73% as compared to a Wallace tree implementation. The Wallace tree encoders were previously compared in [61], and the multiplexer based encoder in [51, 62].

A high level schematic of a 4-bit encoder based on multiplexers is illustrated in Figure 4.10. Each of the parallel muxes in Figure 4.10 contains of multiple ordinary 2-to-1 muxes where the lower bits, $t_5 \dots t_0$, are connected to the '1' inputs and the upper bits, $t_{14} \dots t_7$, to the '0' inputs of the muxes.

The thermometer-to-binary encoder is a crucial building block since it together with the digital support block accounts for approximately 80% of the total dynamic power consumption of the TDC, as can be seen in Table 4.5.

Figure 4.11: Block level diagram of the serial data read-out circuitry.

One way to reduce the dynamic power consumption in the encoder in the current design is to disable the encoder during the reset phase. Simulations show that this change reduces the total power consumption in the TDC with approximately 30%.

Note that the work on mux-based thermometer-to-binary encoders in [51, 61–63], published in the time period 2004–2007 was performed without the knowledge of the patent [60] granted in 2003 (filed in 2001).

4.7 Digital Support Block

The digital support block shown in the top level diagram in Figure 4.4 has two major tasks. The first is to add the outputs of the two 7-bit TCDs, and the second is to control the optional serial data read-out of the 8-bit TDC. The serial data read-out is mainly used in test mode and the data can be connected either to a separate pad or to the global scan chain of the chip.

A block diagram of the digital support block is illustrated in Figure 4.11 and the timing diagram for a serial data read-out cycle is illustrated in Figure 4.12. Referring to Figure 4.11, a serial data read-out cycle works as follows.

When the *TESTEN* signal is set high, the reset generator resets the counter with a single low pulse at *INITCNT*. After reset, the counter generates a low pulse at *LATCHDATA* every 10th clock period. When *LATCHDATA* is low, the output from the 7-bit adder is loaded into the 9-bit register. For the nine remaining clock periods, the content of the register is serially latched out on the *SCANOUT* output. To mark the start and stop of the serial data the

Figure 4.12: Timing diagram of a serial read-out cycle.

TESTEN is latched out first and a zero '0' is latched out last in the serial data word, that is each 10-bit output word starts with a '1' and ends with a '0'. This zero-one padding of the data is used to identify the serial output data correctly.

4.8 Gray Counter

The Gray counter is used in the PLL to count the number of rising edges of the DCO clock during one reference clock period, as previously discussed in Section 4.4. No measurements were done on the Gray counter, since the counter output was not connected to the digital output bus of the chip. The Gray counter is however a speed optimized design which, despite the lack of measurements results, is of interest anyway. Therefore some details on the counter architecture as well as simulation results will be given in this section.

A Gray counter uses the so called Gray code [64] which is a number system where two successive values differ in only one bit as illustrated in Table 4.2. Hence, the sampling error of a Gray counter is limited to one LSB.

Table 4.2: 3-bit Gray code example

Decimal	Gray code	Binary code
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

The worst case sampling error of a binary counter on the other hand will most likely happen at the MSB transition, $011 \rightarrow 100$, as shown in Table 4.2.

Timing errors in the counter might result in that the output for a short time period equals 111, resulting in a sampling error in the order of one MSB.

The speed requirement for the counter equals the nominal DCO frequency of 2.1 GHz. This is a fairly high frequency and thus a new Gray counter architecture was developed for the project. The proposed counter architecture is shown in Figure 4.13 where the Gray counter core is the part of the schematic connected to the inputs of sampling register. The XOR gates connected to the register outputs is the gray-to-binary code conversion logic. As can be seen in Figure 4.13, the critical path in the counter core is one gate delay thus allowing for high frequency operation.

As no measurement results are available only simulated performance results are presented in Table 4.3. The table shows the maximal counter frequency for six selected worst case working conditions and process corners. The supply voltage is set to 1.1 V, that is 10% below the nominal 1.2 V and the temperature is set to the expected extreme values $T = -40$ and 125°C .

As also can be seen in Table 4.3 both temperature and process corner has a large impact on the maximal counter frequency. This since the maximal counter frequency is basically set by the gate delay. We can also conclude from Table 4.3 that the Gray counter most probably meets the nominal performance requirement of 2.1 GHz.

The standard cells used in the Gray counter are general purpose high-Vt transistors. This transistor type was chosen as a trade off between speed and leakage performance. The Gray counter area is $30\text{ }\mu\text{m} \times 13\text{ }\mu\text{m}$, and the nominal power consumption at 2.1 GHz, and a supply voltage of 1.2 V was simulated to be 0.55 mW.

Table 4.3: Maximal Gray counter frequencies at selected worst case working conditions.

Process Corner	Max Freq. [GHz] $T = +125^\circ\text{C}$	Max Freq. [GHz] $T = -40^\circ\text{C}$
Slow-Slow-Slow	4.8	5.5
Nominal	6.9	7.9
Fast-Fast-Fast	9.2	10.4

4.9 Simulation Results

This section presents simulation results from extracted layout of the implemented TDC. The nominal supply voltage is 1.2 V and the temperature 70°C if nothing else is stated. Monte Carlo simulations have been performed to see how mismatch affects the expected performance. Trade off considerations between sample size and statistical confidence for the Monte Carlo simulations are discussed in Section 4.9.1.

Figure 4.13: Schematic of the 6-bit Gray counter.

4.9.1 Confidence Intervals versus Sample Size

Since Monte Carlo simulations can be very time consuming, it is important to know how the sample size, n , affects the confidence interval of the mean value, $\bar{x}$, and standard deviation, s , of the statistical sample.

Assuming a Gaussian distribution, the two-sided confidence interval for the expectation value μ can, from [65], be calculated by

$$\mu = \bar{x} \pm k \cdot s \quad (4.7)$$

where μ is the expectation value, s the standard deviation, and k a factor factor dependent of the size of the confidence interval and the number of samples, n . The factor k for 95% and 99% confidence intervals will be denoted k_{95} and k_{99} respectively. For large enough values of n , so that the central limit theorem apply, these factors can be approximated by $k_{95} \approx 1.9600/\sqrt{n}$, and $k_{99} \approx 2.5758/\sqrt{n}$ [65]. Using these approximations we can derive how many more samples that are required for a 99% confidence interval as compared to a 95% confidence interval for a given value of k in (4.7) as

$$\frac{n_2}{n_1} = \frac{2.5758^2}{1.9600^2} \approx 1.73, \quad (4.8)$$

that is 73% more samples are required. We also conclude that to reduce the range of the confidence interval in (4.7) a factor of two, four times the number of samples n are required. Some values of k_{95} and k_{99} for different sample sizes n are given in Table 4.4.

The standard deviation, σ , for a Gaussian distribution is a stochastic variable with a χ^2 -distribution, hence the two-sided confidence interval is not symmetric and is given by

$$k_1 \cdot s \leq \sigma \leq k_2 \cdot s \quad (4.9)$$

where s is the sample standard deviation and the factors k_1 and k_2 are dependent of the number of samples n . Some values of k_1 and k_2 for different values of n (confidence interval of 95%) are given in Table 4.4. From the calculations made for k_1 and k_2 , it can be seen that the range of the confidence interval for σ in (4.9) scales as $\sim 1/\sqrt{n}$, that is the same order of magnitude as the confidence interval for the expectation value. Given the values in Table 4.4, a sample size of $n = 300$ was chosen as a trade-off between simulation time and statistical confidence.

4.9.2 Vernier Delay-Line Simulations

Monte Carlo simulations have been performed to predict how the unit delays in of the delay latches and the delay line are affected by process variations and transistor matching errors. The sample size was chosen to $n = 300$, as previously discussed in Section 4.9.1. Since the rise and fall times of the

Table 4.4: Factors for determining confidence intervals for expectation value and standard deviation.

n	μ		σ^a	
	k_{95}^b	k_{99}	k_1	k_2
10	0.715	1.028	0.688	1.826
50	0.284	0.379	0.835	1.246
100	0.198	0.263	0.878	1.162
200	0.139	0.182	0.911	1.109
300	0.113	0.149	0.926	1.087
500	0.088	0.115	0.942	1.066
1000	0.062	0.082	0.958	1.046
10 000	0.020	0.026	0.986	1.014

^aConstants k_1 and k_2 are derived for a 95% confidence interval

^bFor $n \leq 100$ the values are taken from [65]

delays elements in the design are non-symmetric, the average of the rise-to-fall and fall-to-rise delays are calculated. The statistical outcome from the Monte Carlo simulations are shown in Figure 4.14.

From the histograms in Figure 4.14 (a) and (b) it can be observed that the standard deviation, σ , for the individual delay-line delays are approximately 2 ps. The standard deviation for the difference between the delays ($\tau_1 - \tau_2$) in Figure 4.14 (c) is however 1 ps. Hence, the absolute value of the standard deviation is suppressed by a factor of two by the Vernier architecture.

From the expectation value, μ , of the delay difference in Figure 4.14 (c) we conclude that the Vernier TDC has a predicted time resolution of 5.4 ps. The delay difference is also always larger than zero, which is a strict requirement for a monotonic transfer function of the TDC.

4.9.3 Delay Sensitivity to Temperature and Supply Voltage

Transient simulations have been performed to estimate how the unit delays of the delay line and the delay latch chain, are affected by changes in temperature and supply voltage. Figure 4.15 (a) shows the sensitivity to supply voltage variations and (b) the sensitivity to changes in temperature.

As can be seen in Figure 4.15 (a) the unit delay for a delay latch τ_1 drops from 45 ps to 25 ps over the swept supply range, that is a change of 20 ps, and a similar sensitivity is observed for the unit delay of the delay chain τ_2 . The delay difference, $\tau_1 - \tau_2$, however only drops 3.5 ps, hence the absolute supply sensitivity is suppressed more than 5 times. If measured in relative terms however, τ_1 , τ_2 and $\tau_1 - \tau_2$ have the same sensitivity to supply voltage variations. Similarly it can be observed from Figure 4.15 (b) that the absolute temperature sensitivity is suppressed approximately by a factor five, but again have the same relative sensitivity to changes in the temperature.

Figure 4.14: Histogram of the (a) *start* unit delay τ_1 , (b) *stop* unit delay τ_2 , and (c) the delay difference $\tau_1 - \tau_2$.

4.9.4 Power Consumption

The current consumption for the TDC was simulated at a 50 MHz sampling frequency. The simulation results are summarized in Table 4.5 where the total current consumption is split into the building blocks of the TDC as illustrated in Figure 4.4. As can be seen in the table, the digital blocks accounts for 79% of the total current consumption, hence one should first focus on reducing the current consumed in the digital blocks in order to reduce the total current consumption of the TDC.

4.10 Chip Implementation

The 8-bit TDC was implemented in a standard 65 nm CMOS process from STMicroelectronics. The chosen process offers two main transistor flavors, that is general purpose, GP, and low power, LP, transistors. The GP transistors are faster but has a higher current leakage as compared to the LP transistors. Both the GP and LP transistors come in three different threshold voltage, V_t , options, that is high V_t , HVT, standard V_t , SVT, and low V_t , LVT.

Figure 4.15: Simulation results of element delay versus supply voltage (upper) and temperature (lower).

Table 4.5: Simulated current consumption.

Block	Current [μA]	% of total
7-bit Vernier delay-line (1st)	63	10.4
7-bit Vernier delay-line (2nd)	65	10.7
7-bit Therm-to-bin enc. (1st)	188	31.0
7-bit Therm-to-bin enc. (2nd)	187	30.8
Digital support	104	17.1
Total power digital blocks	497	79
Total power Vernier chains	128	21

In the Vernier delay lines, GPHVT transistors were chosen as the best trade off between speed and leakage current and to minimize the power consumption in the thermometer-to-binary encoder, LPLVT was selected.

The layout of the 8-bit TDC is shown in Figure 4.16 where also the layout of the 6-bit Gray counter is included. The sizes of the individual building blocks are indicated in the figure and the total core size of the TDC is $75 \mu\text{m} \times 120 \mu\text{m}$. The 8-bit implementation and measurement results were presented in Paper E.

4.11 Measurement Considerations

The predicted TDC resolution of ~ 5 ps sets high performance requirements on the measurement system. Not only the signal generators must have a phase and jitter performance better than 5 ps, but also noise and distortion sources from the surrounding environment must be minimized to not affect the measurement results.

Figure 4.16: Chip photo and floorplan of the 8-bit TDC.

A theoretical analysis on how voltage glitches affect the phase of input test signals as well as suggestions on how the glitch sensitivity can be improved are given in Section 4.11.1. The test equipment and the measurement setup are presented in Section 4.11.2.

4.11.1 Minimizing Glitch Sensitivity for Input Signals

The TDC presented in this work measures the time or phase difference between two edges of two input signals. Hence, the information carrier is a phase difference rather than a voltage difference as is the case for a conventional ADC. We will in this section analyze how a voltage glitch affects the phase difference between two input signals. Since phase and time are exchangeable in this context, we will analyze the phase shift due to voltage glitches as a time error in the remaining of this analysis.

A TDC can in principle be measured using any test signal with a well defined phase. Periodic signals are typically preferred, such as for example sinusoids, square, or trianglewave signals. The sensitivity to disturbances however differ between these signals, i.e., how much the phase information is affected by a voltage glitch. Figure 4.17 illustrates how the phase difference between two sinusoids change when one sinusoid experience a voltage change of ΔV_1 , and the other a voltage change of ΔV_2 . Note that the

polarity of the sinusoid determines the direction of the time changes, Δt_1 and Δt_2 . Hence we have the worst scenario if signals with opposite polarity experience correlated voltage glitches with the same direction. This is the completely opposite scenario as compared to differential signaling when voltage is the information carrier for which correlated voltage glitches are suppressed.

Figure 4.17: Illustration of timing error in a sinusoid due to DC voltage glitch.

To derive how a voltage change, ΔV , change the phase, Δt , of a sinusoid we rearrange the input signal $u(t) = \beta \sin(\omega t)$ so that t becomes a function of u as

$$t(u) = \frac{1}{\omega} \arcsin\left(\frac{u}{\beta}\right), \quad (4.10)$$

where β is the amplitude, and ω the phase rotation of the sinusoid. The sensitivity is now derived by differentiation of (4.10) at $u = u_0$, that is

$$dt(u) = \left. \frac{\partial t(u)}{\partial u} \right|_{u=u_0} du = \left. \frac{1}{\beta \omega \sqrt{1 - (u/\beta)^2}} \right|_{u=u_0} du. \quad (4.11)$$

Since the phase information usually is extracted at the zero crossings of a sinusoid we derive (4.11) in $u_0 = 0$ as

$$dt(u) = \frac{du}{\beta \omega} = \frac{du}{\beta 2\pi f}. \quad (4.12)$$

By changing notation and replacing $dt(u)$ and du in (4.12) with Δt and ΔV respectively, the gain factor G between voltage and time in the zero crossings of a sinusoid is given by

$$G = \frac{\Delta t}{\Delta V} = \frac{1}{\beta 2\pi f}. \quad (4.13)$$

Assuming the scenario in Figure 4.17 where two signals of opposite polarity are used, the total phase error due to glitches is now given by $\Delta t_{\text{tot}} = \Delta t_1 + \Delta t_2$ where Δt_1 and Δt_2 can be derived from (4.13).

Hence, if two 1 MHz sinusoids with amplitudes $\beta = 1$ V experience the same voltage glitch of 1 mV, the resulting phase error, or time skew, between them are given by

$$\Delta t_{\text{tot}} = \Delta t_1 + \Delta t_2 = 2 \cdot \frac{1 \cdot 10^{-3}}{2\pi \cdot 10^6} \approx 318 \text{ ps}, \quad (4.14)$$

which corresponds to 64 LSBs for a TDC with 5 ps resolution. The gain factor for the sinusoids are $G \approx 159 \text{ ns/V}$.

For a square wave input, the gain factor $G = \Delta t / \Delta V$ is directly given by the rise/fall times and the amplitude of the signal. The signal generator used in the measurements have a gain factor of $G = 6.9 \text{ ns/V}$ for a 1 V_{pp} square wave signal, which is about 20 times smaller than for a corresponding sinusoidal signal.

Concludingly, TDC test signals should have as low G value as possible, that is short rise and fall times, in order to suppress the influence of voltage glitches. If two input signals are used represent a phase difference, they should be of the same polarity to in order to suppress correlated glitches and noise.

For the reasons mentioned above, square wave signals were used as inputs to the TDC. However, the input signals were of opposite polarity in order to match the internal signal polarities in the TDC chip. This should be changed in the next version of the TDC.

4.11.2 Measurement Setup

The linearity of the prototype chip was measured using the instrument setup shown in Figure 4.18. The SMBV100A vector signal generator from Rohde & Schwarz was used to generate two square wave inputs, *refClk* and *phiClk*, and the MSO9404a oscilloscope from Agilent sampled the digital output bus, *tdcOut*<7:0>.

The phase difference of the two inputs was controlled by introducing a time skew between the I and Q outputs of the vector signal generator. The time skew could be swept in 1 ps steps which was enough for our application. The DC levels of *phiClk* and *refClk* was set to half the IO voltage of the chip pads using RF bias tees before connecting them to the TDC inputs. Both inputs were terminated over 50 Ω .

4.12 Measurement Results

4.12.1 Time Resolution

The time resolution of the TDC was measured using a Rohde & Schwarz SMBV100A vector signal generator where the I and Q outputs from the RF baseband generator were used as inputs to the TDC. The phase difference between the I and Q outputs could be controlled with 1 ps accuracy, hence sufficient to measure the expected 5 ps resolution. RF-bias tees were used to

Figure 4.18: TDC measurement setup.

set a DC level of 800 mV on the input signals and the standard digital inputs pads were supplied with a 1.6 V I/O voltage. This a lower voltage than the nominal 2.5 V I/O voltage. Linearity measurements where the I/O voltage was swept showed that the lower I/O voltage introduced less disturbances to the measurements.

The phase difference between the input signals to the TDC was swept in 5 ps steps and 10-K samples was collected for each of the phase settings. The average of these 10-K samples was derived and the resulting differential and integrated non-linearity, INL/DNL, curves for the TDC are shown in Figure 4.19 and Figure 4.20 respectively.

Figure 4.19 shows the measured DNL for the TDC in 8-, and 7-bit mode. As can be seen in the figure, the DNL is always larger than -1 LSB for both settings guaranteeing that the TDC has a monotonic transfer function.

In Figure 4.20 we find the INL to be $-5/-9$ LSBs respectively for lower-end codes. This comparatively high non-linearity is caused by an insufficiently sized inverter, INV1 in Figure 4.9 (b). The relatively long rise time of the inverter unfortunately sets the latch (i.e., the path through transistors M1, M2 and M3) in a metastable state for low input codes, that is when the falling edge of *refClk* and the rising edge of *phiClk* are close in time. The metastability in turn increases the delay through the latch resulting in the non-linear INL curves. The hypothesis was verified through INL simulations as shown in Figure 4.21. Figure 4.21 (a) shows the INL with the weak driver and Figure 4.21 (b) shows the INL with a correctly sized driver.

Figure 4.19: Measured DNL of the TDC in (a) 8-bit, and (b) 7-bit mode.

Figure 4.20: Measured INL of the TDC in (a) 8-bit, and (b) 7-bit mode.

4.12.2 Single Shot Precision

The single shot precision measures the output of the TDC for a constant input. This catches noise and other non-ideal behavior from on-chip as well as off-chip sources. The TDC input was swept in 1 ps steps and 10-K samples was sampled for each input. The standard deviation, σ , was derived for each input code and plotted in the upper plot in Figure 4.22. The histogram for the standard deviation over all TDC codes is plotted in the lower plot in Figure 4.22. From this histogram it can be seen that the standard deviation over all TDC codes is 1.1 LSB on average with a standard deviation of 0.33 LSB.

Figure 4.21: Simulated INL error for the (a) implemented reset and phase detect circuit, and (b) same circuit with re-tuned inverter size.

Figure 4.22: Single shot standard deviation as a function of TDC code (upper), and corresponding histogram (lower).

4.12.3 Power Consumption and Maximal Sampling Rate

The power consumption of the TDC was derived by measuring the voltage drop over a $10\ \Omega$ resistor connected to the output of a voltage regulator on the test PCB. Power simulations and measurements show that the TDC has a signal dependent power consumption, but all power figures given here are for the worst case input. The power consumption was measured for the TDC in 7- and 8-bit modes, and the resulting curves are plotted in Figure 4.23. As can be seen in the figure, the power consumption increases approximately

Figure 4.23: Measured power consumption versus signal update frequency.

linear with the sampling frequency and the TDC consumes 1.75 mW in 7-bit mode at 100 MHz and 1.85 mW in 8-bit mode at 50 MHz sampling frequency.

The maximal sampling rate for the TDC was measured up to 100 MHz in 7-bit mode and 50 MHz in 8-bit mode. The limiting factor for the sampling rate is the reset scheme used in the TDC, as was previously discussed in Section 4.6

4.12.4 Comparisons with Previously Published TDCs

In Table 4.6 the implemented TDC is compared with recently published TDCs with a resolution in the range 4-6 ps. The TDCs in Table 4.6 are selected with respect to small area and low power consumption. Note that there are converters with sub-picosecond resolution [66, 67]. The finer time resolution does however come with a significantly larger chip area and power consumption.

From Table 4.6 it can be concluded that the proposed TDC offers competitive performance in terms of area and power consumption. The delay line TDC has shorter conversion range than a looped architecture [68]. Intended application areas for the proposed TDC are counter-assisted digital PLLs [17] and all-digital ADCs [5, 69].

The limited measured non-linearity will be addressed in future designs by mainly resizing the inverter in the edge detect circuit as described in Section 4.12.1. Note that the prototype chip still shows a high potential of the proposed architecture.

4.13 Future Improvements

During the TDC project a couple of improvement areas have been identified. These improvements have been discussed in this chapter but will also be summarized here.

As has been discussed in Section 4.6.2, the reset scheme reduces the maximal conversion speed of the TDC. This since the reset pulse is active for

Table 4.6: Published time-to-digital converters.

Type	[70] Passive interp.	[71] Cyclic Vernier	[72] 2-D delay-line	[73] Vernier + GRO	This Work 7-bit/8-bit
Samp. Rate [MS/s]	180	10	50	25/100	100/50
Resolution [ps]		5.5	4.8	5.8	5.7
OSR	4			16	
Res. w. interp. [ps]	4.7			3.2	
Power Supply [V]	1.2	1.0	1.2	1.2	1.2
Power [mW]	3.6	2.0	1.7 ^a	3.6 ^b	1.75 ^c /1.85 ^d
Range [ns]	0.6	100	0.6	40	0.73/1.46
Number of bits	7	15	7		7/8
Area [mm ²]	0.02	0.006	0.02	0.027	0.004/0.008
Technology [nm]	90		65	90	65
Year	2008	2011	2010	2012	2013

^ameasured at 50 MS/s

^bmeasured at 25 MS/s

^cmeasured in 7-bit mode at 100 MS/s

^dmeasured in 8-bit mode at 50 MS/s

one half conversion period. A short reset pulse induced directly after each conversion is sufficient and should be implemented in the next version.

To reduce the over all power consumption the thermometer-to-binary encoder should be disabled in the reset phase as was discussed Section 4.6. The expected power reduction for the TDC due to this improvement is around 30%.

The linearity of the TDC can be improved by mainly resizing the inverter in the edge detect circuit as described in Section 4.12.1.

In order to suppress measurement noise the polarity of the test signals should be changed to have the same polarity as discussed in Section 4.11.1.

Chapter 5

Digital Recursive Oscillators

5.1 Introduction

Digital frequency synthesis is an essential part in a variety of applications such as software defined radio [74] and radio using frequency shift keying (FSK) or quadrature amplitude modulation (QAM) [8, 75, 76]. Other application areas using digitally synthesized frequencies are audio effect design [77] and built-in self-test (BIST) for mixed-signal systems [78, 79].

One candidate for real-time sinusoid generation is the recursive digital oscillator structure previously investigated in [74–77, 80–84]. A digital oscillator can offer a simple approach to generate sinusoids only using recursion of arithmetic expressions and avoids the need for lookup tables imminent in the DDS approach [85]. However, there are a number of issues related to finite word length effects that must be handled. In theory these effects lead to infinite infinite round-off noise accumulation [74], eventually leading to overflows in the oscillator.

However, the digital oscillator forms a deterministic state machine with a finite number of states. Once one of the states is visited again, the sequence will continue to repeat as there are no input signal to the oscillator. Hence, it will eventually lock in a periodic sequence of states [81, 82]. In addition, if the oscillator is initialized to one state of a sequence it will follow that sequence.

In Paper F we propose a new search algorithm for finding all such sequences (initial states) for a given oscillator configuration. These sequences can then be evaluated with respect to spectral purity. The improvement in spurious-free dynamic range is between 7 and 40 dB compared to previous reported results. A key part of the search algorithm is the reduction of the search space. This reduction is made possible by an extension of existing theory on digital oscillators.

This chapter is organized as follows: A theoretical background to digital oscillators are given in Section 5.2 and Section 5.3. A summary of previously published recursive oscillators are given in Section 5.4. Steady state-cycles and a new algorithm for finding these cycles are discussed in Section 5.5 and Section 5.6 respectively. A practical example on how the proposed algorithm

can be used to find test signals for digital-to-analog converters is presented in Section 5.8.

5.2 Recursive Equations and Vector Rotation

The oscillators in this chapter use recursive equations to compute the sinusoidal outputs. These equations are often derived from trigonometric relations such as for example

$$\cos \varphi \cos \theta = \frac{1}{2} (\cos(\varphi - \theta) + \cos(\varphi + \theta)). \quad (5.1)$$

By reordering and interpreting θ as the phase increment (or step angle) derived in each iteration, the following formula can be identified from (5.1)

$$y(n) = 2 \cos \theta \cdot y(n-1) - y(n-2), \quad (5.2)$$

hence the new output value $y(n)$ is derived from the two previous values $y(n-1)$ and $y(n-2)$. An oscillator using the expression in (5.2) is sometimes referred to as a biquad oscillator. If the step angle in each iteration is given by θ , the biquad oscillator has the output

$$y(n) = \cos(n\theta + \varphi) \quad (5.3)$$

where φ is a phase offset set by the initial state of the oscillator. The initial state is the value we assign to the oscillator at time-point $n = 0$ and has a large impact on the oscillator output as will be shown throughout this chapter.

Another example of trigonometric identities useful for oscillators are the following two expressions

$$\begin{cases} \cos(\varphi + \theta) = \cos \varphi \cos \theta - \sin \varphi \sin \theta \\ \sin(\varphi + \theta) = \cos \varphi \sin \theta + \sin \varphi \cos \theta \end{cases} \quad (5.4)$$

By again interpreting θ as the step angle we get

$$\begin{cases} y_1(n) = \cos \theta \cdot y_1(n-1) - \sin \theta \cdot y_2(n-1) \\ y_2(n) = \sin \theta \cdot y_1(n-1) + \cos \theta \cdot y_2(n-1) \end{cases} \quad (5.5)$$

which in turn can be written as a matrix multiplication as

$$\begin{bmatrix} y_1(n) \\ y_2(n) \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} y_1(n-1) \\ y_2(n-1) \end{bmatrix}. \quad (5.6)$$

The two element vectors now corresponds to an oscillator with two outputs and we also identify the matrix in (5.6) as the commonly used rotation matrix. An oscillator using the rotation matrix is often referred to as a coupled form complex oscillator [76]. One interpretation of the matrix multiplication

Figure 5.1: A vector rotation an angle θ in the y_1, y_2 -plane using the rotation matrix in (5.6).

in (5.6) is a rotation of a vector in the two dimensional y_1, y_2 -plane as illustrated in Figure 5.1. As can be seen in the figure, the vector $y(n)$ is rotated an angle θ counterclockwise in the y_1, y_2 -plane.

Other properties of the rotation matrix is that the vector length is preserved and also that the outputs $y_1(n), y_2(n)$ are in quadrature, that is phase shifted $\pi/2$ radians relative to each other. The vector length is preserved since the determinant of the matrix is equal to one, and the quadrature relationship can be seen by using the relation, $\sin(\theta + \pi/2) = \cos(\theta)$, in (5.4). Length preservation is a strict requirement for recursive oscillators as will be discussed further in Section 5.3.

Interestingly enough, not only the coupled form complex oscillator can be described using matrix multiplication. The corresponding matrix multiplication for the biquad oscillator in (5.4) is for example given by

$$\begin{bmatrix} y_1(n) \\ y_2(n) \end{bmatrix} = \begin{bmatrix} 2 \cos \theta & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_1(n-1) \\ y_2(n-1) \end{bmatrix}. \quad (5.7)$$

Fact is, all commonly used recursive oscillators can be written in this form, which makes it possible to analyze them using a common theory [76].

The existing theory on recursive oscillators has been extended to also include sinusoids with arbitrary amplitude and phase. The most important steps in the derivations are summarized in the next section (Sec. 5.3), while we refer to Paper F for details.

5.3 Analysis of Recursive Oscillators

In this section, a general recursive oscillator structure is analyzed using state-space equations. The use of a state-space representation makes it possible to analyze recursive oscillator structures in terms of stability, relative output

Figure 5.2: A general discrete-time state-space structure [86].

amplitudes, and relative output phases by considering their corresponding state-space matrices only.

The general state-space structure in Figure 5.2 is a discrete-time and time-invariant system. We will in this analysis restrict ourselves to a system with two inputs, two outputs and two state-space variables. The corresponding state-space representation for the structure in Figure 5.2 is

$$\begin{aligned} v(n+1) &= Av(n) + Bx(n) \\ y(n) &= Cv(n) + Dx(n) \end{aligned} \quad (5.8)$$

where A , B , C , and D all have the dimensions 2×2 . In this analysis the state-space matrix A will be written in its general form (5.9). The input matrix B and output matrix C are unit matrices and the feedforward matrix D is the zero matrix, hence we have that

$$\begin{aligned} A &= \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad a, b, c, d \in \mathbb{R}, \\ B &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned} \quad (5.9)$$

The output matrix C is in this analysis set to unity since it does not affect the stability of the oscillator. It can however be used for scaling and/or constant phase shift of the output signal as will be further discussed in Section 5.3.3. Combining (5.8) with the matrix definitions in (5.9), the state-space representation in (5.8) simplifies to

$$y(n+1) = Ay(n) + x(n), \quad (5.10)$$

where $x(n)$ and $y(n)$ are input and output vectors, and A is the state-space matrix.

The recursive oscillators investigated in this work are self-sustained oscillators and hence the output is directly set by the initial condition. By setting the input vector $x(n)$ to zero for all values of n and also defining an initial condition x_i , we can rewrite (5.10) as

$$y(n) = A^n x_i, \quad (5.11)$$

where $y(n)$ is the oscillator output, x_i is the initial condition, and A is a general state-space matrix. A signal flow graph implementing the recursive general matrix multiplication in (5.11) is shown in Figure 5.3, [87].

Figure 5.3: Signal flow graph of a general recursive oscillator [87].

Figure 5.4: Mapping of poles between the S-plane and the z-plane.

5.3.1 Requirements for Oscillation

The system stability for the discrete-time state-space structure in Figure 5.2 is determined by the placement of the system poles in the z-domain. The rules for stability with respect to pole placement can be summarized as follows [88]:

- If the outermost system poles are *inside* the the unit circle, then the system is stable.
- If the outermost system poles are *outside* the the unit circle, then the system is unstable.
- If the outermost system poles are *on* the the unit circle, then the system is marginally stable.

The stability rules are illustrated in Figure 5.4 where also the corresponding rules for a time continuous system are shown for reference. The figure also illustrates the pole mapping between the S-domain and the z-domain.

From the stability rules above, we conclude that the requirement for oscillation is that the system poles of the structure in Figure 5.2 are placed on the unit circle. The system poles are identical to the eigenvalues of A [86], which always come in pairs for the matrix A , and hence the requirement can be re-phrased according to the following. The poles (eigenvalues) of A should be

a complex pole pair placed on the unit circle. The eigenvalues of A are given by

$$\lambda_{1,2} = \frac{(a+d)}{2} \pm j \frac{\sqrt{4(ad-bc) - (a+d)^2}}{2}, \quad (5.12)$$

from which we find that in order to create a complex pole pair placed on the unit circle, the following requirements must be met

$$\begin{aligned} |\lambda_{1,2}| &= 1 \\ 4(ad-bc) - (a+d)^2 &> 0. \end{aligned} \quad (5.13)$$

From the requirements in (5.13) we finally derive the following requirements on the state-space matrix A

$$\begin{aligned} \det(A) &= (ad-bc) = 1 \\ |a+d| &< 2. \end{aligned} \quad (5.14)$$

As seen in (5.14), the matrix determinant of A must be one. This corresponds to the requirement of vector length preservation as was briefly discussed in Section 5.2.

We have now derived the fundamental requirement for oscillation, hence as long the requirements in (5.14) are fulfilled the system in Figure 5.4 will oscillate with a constant amplitude. By continuing the analysis, other interesting properties of the oscillator can be derived directly from the state-space matrix A . From this analysis it can be concluded that if $b = -c$ (same value but opposite signs in the non-main diagonal ↙) we get equi-amplitude outputs and if $a = d$ (identical values in the main diagonal ↘) the phase difference γ equals $\pm\pi/2$, i.e., the output is in quadrature.

5.3.2 Initial Condition

An important result from the theoretical analysis is a closed form expression for the relation between the initial condition x_i and the resulting oscillator output $y(n)$. It is shown that an oscillator will generate the following output

$$y(n) = \begin{bmatrix} \beta \Omega \sin(n\theta + \gamma + \varphi_0) \\ \beta \sin(n\theta + \varphi_0) \end{bmatrix} \quad (5.15)$$

if initialized with the following initial condition

$$x_i = \begin{bmatrix} \beta \Omega \sin(\gamma + \varphi_0) \\ \beta \sin(\varphi_0) \end{bmatrix}, \quad (5.16)$$

where β is the base amplitude, Ω the relative amplitude, θ the step angle, φ_0 the common phase offset, and γ is the phase difference. In earlier derivations [76], the output signals were restricted to full-scale signals, $\beta = 1$, and a common phase offset $\varphi_0 = 0$.

The closed form expression for the initial condition in (5.16) is a key component in the proposed search algorithm in Section 5.6. The algorithm uses these expressions to significantly reduce the search space for the algorithm and hence it also reduces the total search time.

5.3.3 Constant scaling and rotation

The output matrix C (5.9) was in the analysis set to unity, but it can also be used for scaling and/or rotation of the output signal $y(n)$. This can be useful in applications where the amplitude or phase of a signal must be swept. A fixed amplitude scaling and rotation can also be applied if the search algorithm in Section 5.6 finds high linearity cycles with amplitude or phase values that are out of specification.

The oscillator outputs can be scaled by factors β_{c1} and β_{c2} using the output matrix according to

$$\begin{aligned} y'(n) = Cy(n) &= \begin{bmatrix} \beta_{c1} & 0 \\ 0 & \beta_{c2} \end{bmatrix} \begin{bmatrix} \beta \Omega \sin(n\theta + \gamma + \varphi_0) \\ \beta \sin(n\theta + \varphi_0) \end{bmatrix} \\ &= \begin{bmatrix} \beta_{c1}\beta \Omega \sin(n\theta + \gamma + \varphi_0) \\ \beta_{c2}\beta \sin(n\theta + \varphi_0) \end{bmatrix}. \end{aligned} \quad (5.17)$$

If the outputs of the oscillator are in quadrature, a constant gain as well as a constant rotation an angle φ_c can be applied using a general rotation matrix as

$$\begin{aligned} y''(n) = Cy(n) &= \begin{bmatrix} \beta_{c1} \cos \varphi_c & -\sin \varphi_c \\ \sin \varphi_c & \beta_{c2} \cos \varphi_c \end{bmatrix} \begin{bmatrix} \beta \Omega \cos(n\theta + \varphi_0) \\ \beta \sin(n\theta + \varphi_0) \end{bmatrix} \\ &= \begin{bmatrix} \beta_{c1}\beta \Omega \cos(n\theta + \varphi_0 + \varphi_c) \\ \beta_{c2}\beta \sin(n\theta + \varphi_0 + \varphi_c) \end{bmatrix}. \end{aligned} \quad (5.18)$$

5.4 Published Oscillators

From the theory developed in [76], it was shown that all recursive oscillators can be described by their corresponding state-space matrices. In Section 5.4.1 we will revisit the biquad oscillator and in Section 5.4.2 the coupled form complex oscillator, both earlier introduced in Section 5.2. A summary of other previously published recursive oscillator structures are presented in Section 5.4.3.

5.4.1 The Biquad Oscillator

The biquad oscillator is probably the simplest of the recursive oscillators in terms of arithmetic complexity and will also be used in most simulation examples in this work. From Section 5.2 we recapture the corresponding state-space matrix for the biquad oscillator as

$$A = \begin{bmatrix} \alpha & -1 \\ 1 & 0 \end{bmatrix} \quad \alpha = 2 \cos \theta, \quad (5.19)$$

where α is the multiplier coefficient, and θ is the step angle of the oscillation. Using the conclusions drawn in Section 5.3.1, we can directly from the state-space matrix A in (5.19) conclude that the biquad oscillator outputs are

Figure 5.5: Flow graph of a recursive biquad oscillator.

equi-amplitude (same value but opposite signs in the non-main diagonal $\swarrow$) and non-quadrature (non-equal values in the main diagonal $\searrow$). Another important property is that since $\det(A) = 1$, the poles are always placed on the unit circle.

The outputs from the biquad oscillator is given by

$$y(n) = \begin{bmatrix} \beta \sin((n+1)\theta + \varphi_0) \\ \beta \sin(n\theta + \varphi_0) \end{bmatrix} \quad (5.20)$$

where β is the amplitude, θ is the step angle and φ_0 is the common phase offset. A flowgraph for the biquad oscillator is illustrated in Figure 5.5. As can be seen in Figure 5.5, and also from (5.20), the biquad oscillator has essentially only one output which is delayed one clock cycle yielding the second output.

5.4.2 The Coupled-Form Complex Oscillator

The coupled-form complex oscillator is the only recursive oscillator simultaneously satisfying the requirements of quadrature and equi-amplitude outputs Section 5.3. From Section 5.2 we recapture the corresponding state-space matrix (rotation matrix) as

$$A = \begin{bmatrix} C & -S \\ S & C \end{bmatrix}, \text{ where } \begin{cases} C = \cos \theta \\ S = \sin \theta \end{cases} \quad (5.21)$$

Ideally, this matrix satisfies the requirements for oscillation as given in (5.14), but a hardware implementation using finite word length radix-2 arithmetics does not. The pole radius, given by $\det(A) = C^2 + S^2$, can never be unity, resulting in either increasing or decreasing output amplitudes. A formal proof for this statement is given in Paper F. This issue has been discussed previously, by for example Vankka [74], however without proof.

Even though the radius is never unity for the complex oscillator other non-linear effects, such as rounding, will enable it to sometimes lock in steady-state cycles offering outputs with high spectral purity. Steady-state cycles will be discussed further in Section 5.5.

5.4.3 Collection of Oscillators

To overcome the problem with a non-unity pole radius, a number of approximations of the rotation matrix in (5.21), guaranteed to have $\det(A) = 1$,

have been proposed with either equi-amplitude *or* quadrature outputs. Approximations with quadrature outputs but with non-equi-amplitude are for example the digital waveguide [84], and the quadrature staggered update oscillator [76].

A collection of recursive oscillators and their corresponding amplitude- and phase properties is shown in Table 5.1. As can be seen in the table only the coupled-form complex oscillator has both equi-amplitude and quadrature outputs. For the remaining of the oscillators in Table 5.1 the poles are guaranteed to be placed on the unit circle.

5.5 Steady-State Cycles in Recursive Oscillators

The analysis and conclusions in Section 5.3 assumes that all coefficients have infinite precision. However, when implementing the recursive oscillator in Figure 5.3 as a digital state-machine using finite word length arithmetics, the oscillator outputs might deviate from their ideal values. Round-off noise in the arithmetic operations may result in either increasing or decreasing amplitudes at the outputs of the oscillator. If, however, the state-machine after a number of iterations returns to a previously visited state, the state-machine is forced into a locked loop, or cycle, as illustrated in Figure 5.6.

The state machine in Figure 5.6 is initiated to state S_0 and updates its internal state after each iteration. After fourteen iterations state S_{14} equals

Table 5.1: Recursive oscillators

Oscillator	Equi-amp.	Quadrature	$\kappa =$	State-space matrix
Biquad	Yes	No	$2 \cos(\theta)$	$\begin{bmatrix} \kappa & -1 \\ 1 & 0 \end{bmatrix}$
Coupled-form complex oscillator [83]	Yes	Yes	$\sin(\theta)$	$\begin{bmatrix} \sqrt{1-\kappa^2} & \kappa \\ -\kappa & \sqrt{1-\kappa^2} \end{bmatrix}$
Digital waveguide [84]	No	Yes	$\cos(\theta)$	$\begin{bmatrix} \kappa & \kappa-1 \\ \kappa+1 & \kappa \end{bmatrix}$
Equi-amplitude-staggered update [76]	Yes	No	$2 \sin(\theta/2)$	$\begin{bmatrix} 1-\kappa^2 & \kappa \\ -\kappa & 1 \end{bmatrix}$
Quadrature-staggered update [76]	No	Yes	$\cos(\theta)$	$\begin{bmatrix} \kappa & 1-\kappa^2 \\ -1 & \kappa \end{bmatrix}$

Figure 5.6: Illustration of a locked state-machine.

state S_2 and the cycle $S_2 \rightarrow S_{13}$ repeats. This locking effect was previously investigated in [81, 82].

In a locked oscillator, noise is no longer accumulated and since all errors now occur periodically all non-linearities will transform into distortion, hence a locked oscillator is noiseless. Note that we only consider digital noise in the output signal, other physical noise sources such as for example thermal noise is not considered. Due to the fixed cycle length, a locked oscillator by definition also has absolute periodicity [82].

The output frequency f_{sine} , of a recursive oscillator locked in a steady-state cycle is given by

$$f_{\text{sine}} = \frac{\theta'}{2\pi T} = \frac{M}{NT} \quad (5.22)$$

where θ' is the actual (simulated) step angle, and T is the sampling period of the system. N is the total number of samples of M consecutive periods of the sinusoid where $2M \leq N$. Referring to Figure 5.6, N is the number of samples in the cycle and M is the number of turns around the origin. From (5.22) we conclude that there are many combinations of M and N resulting in the same output frequency of the oscillator.

Other methods have been proposed to control the accumulated noise problem. In [80] the oscillator is periodically reset and the accumulated noise is also reduced by switching between additional multiplier coefficients resulting in increasing and decreasing amplitudes respectively. A second method suggested in [76] is amplitude regulation using a feedback loop. The hardware cost for the general control loop in [76] is large but can be reduced with restrictions on settling time and output amplitude of the oscillator [76].

Locked oscillator cycles are typically found using search algorithms where multiplier coefficients, arithmetic word lengths, rounding schemes and most importantly initial conditions are swept. The cycles are then evaluated in terms of spectral purity. In Paper F we propose a new search algo-

rithm and two new search strategies for finding steady-state cycles. The improvement in spurious-free dynamic range is between 7 and 40 dB compared to previous reported results. The proposed search algorithm is described in Section 5.6.

5.6 Proposed Search Algorithm

In order to find all steady-state cycles for a given oscillator implementation, an algorithm could test all initial conditions in the $y_{1,2}$ -plane in Figure 5.6. This approach works for shorter word lengths but for larger word lengths the search time must be reduced.

With the specification as input, the proposed search algorithm uses the closed form expression in (5.16) to derive all possible initial conditions. Strict requirements on amplitude and phase gives a small number of possible start points whereas tolerances in the specification might result in a significantly larger search space. This search space is further reduced using the cuts described in Paper F. The remaining start values are tested one by one until a cycle fulfilling the specification is found. If no cycle fulfills the specification, another rounding scheme, new multiplier coefficients and/or a different word length can be tested.

An important part of the search algorithm is a bit-true model of the oscillator where all implementation specific effects such as word length, rounding scheme, signal over flow etc. are correctly modeled.

The search algorithm follows the decision diagram illustrated in Figure 5.7. As can be seen from the diagram, the cycle length N is not used as an end condition as it was in [82]. This strategy allows the algorithm to find cycles with the same step angle θ' 5.22, i.e., the same M/N ratio, for all combinations of N and M . This increases the probability to find a cycle with a high linearity. Another difference as compared to [82] is that the algorithm keeps track of all previously visited points, and hence no sequence is ever tested twice. This gives a second end condition greatly reducing the search time.

5.7 Properties of Locked Oscillators Cycles

The locking effect can be utilized in the design of all recursive oscillators and they all share the same specific properties. The most important properties are listed below.

- An oscillator locked in steady state generates noiseless outputs with absolute periodicity [82].
- The output amplitude can be changed by simply changing initial condition, hence no extra multipliers are required for scaling.

Figure 5.7: Decision diagram for the proposed algorithm.

- Two or more oscillators can be connected in a time-interleaved configuration to give a combined output signal with an output rate higher than possible for a single oscillator.
- Oscillators locked in steady-state have a much finer frequency resolution than initially suggested by the equations.

In the remainder of this chapter a number of simulation examples are given where two different rounding schemes are used. Rounding is defined such that half an LSB is added to the number before truncating all bits below the LSB bit. In truncation all bits below the LSB are discarded.

The binary number representation used when simulating the biquad oscillator is two's complement with two integer bits and $(W - 2)$ fractional bits where W denotes the total word length. Two integer bits gives the required number range to represent the multiplier coefficient α as defined in (5.19), i.e., $\alpha \in [-2, 2 - 2^{-(W-2)}]$. For increased readability, the values of coefficients and initial conditions are sometimes given as the corresponding integer representation divided by $2^{(W-2)}$.

The corresponding binary number representation for the coupled-form complex oscillator is two's complement with one integer bit and $(W - 1)$ fractional bits. The coefficients are given as a coefficient pair, $[C \ S]$, and the

Figure 5.8: Impact of initial condition.

values of coefficients and initial conditions given by the corresponding integer representation divided by $2^{(W-1)}$.

5.7.1 Sensitivity to initial conditions

Recursive oscillators are very sensitive to initial conditions where a small change in initial condition may result in very different outputs as illustrated in Figure 5.8. The figure shows a simulation of an 8-bit, coupled-form complex oscillator, coefficients $[125/2^6 \ 28/2^6]$, where the difference in initial condition between the two signals is one LSB only. In this example we have assumed an infinite signal range to illustrate the accumulation of round-off errors. In a real implementation the output would either saturate or overflow depending on the strategy used.

Even though the outputs follow each other for the first hundred samples they soon quickly diverge. As can also be seen in the figure, while one initial condition results in a signal diverging towards infinity the other results in a steady-state cycle as discussed in Section 5.5.

The proposed search algorithm uses different strategies to search the state-space for initial conditions giving high output sequences. The difficulty of predicting which initial condition that gives a high linearity cycle can be illustrated by simulating all initial conditions for a fixed coefficient. If all initial conditions are plotted in the state-space domain where the SFDR for each of the initial condition are represented by a color, we get the plot shown in Figure 5.9. An 8-bit biquad oscillator with multiplier coefficient $\alpha = 70/2^6$ is used in the simulation. We can conclude from the plot in Figure 5.9 that the red dots representing a high SFDR value are surrounded by colored dots representing a much lower SFDR performance. Hence the oscillator has a large sensitivity to the initial conditions. What can also be observed in Figure 5.9

Figure 5.9: SFDR simulated for all initial conditions for an 8-bit biquad oscillator, $\alpha = 70/2^6$.

is that almost all initial conditions in the elliptic area result in a locked cycle, this since the poles of a biquad oscillator are always placed on the unit circle, as previously discussed in Section 5.4.1.

As a comparison Figure 5.10 shows a similar plot for the coupled-form complex oscillator (Table 5.1, Section 5.4). The coupled-form complex oscillator also locks in cycles, but much fewer as compared to the biquad oscillator in Figure 5.9. This is because the poles for the coupled-form complex oscillator can never be placed on the unit circle using a radix-2 implementation, as proven in Paper F. The reason it still locks in cycles is due to other non-linear effects, such as rounding, in the finite word length implementation.

5.8 Sinusoid Test Signals for Digital-to-Analog Converters

In this section we will illustrate how the proposed search algorithm can be used to find test signals for digital-to-analog converters. This is a suitable test case for the algorithm since these test signals have high requirements on spectral purity. The biquad oscillator will be used in all examples.

The test signals in this example are single-tone and two-tone sinusoids testing dynamic measures such as SFDR, SNDR and intermodulation IMD. In order to test static linearity measures such as INL and DNL a binary counter can be used to generating a digital ramp input to the DAC. Static test signals will however not be treated in this work.

In Section 5.8.1 the advantages and disadvantages of internal and external test signal generation are discussed. The requirements on test signals

Figure 5.10: SFDR simulated for all initial conditions for an 8-bit coupled-form complex oscillator, $\alpha = 33/2^7$.

suitable for DAC testing are discussed in Section 5.8.2. Simulation results for single tone sinusoids using shift-only coefficients are presented in Section 5.8.4, and suggestions on how hardware efficient two tone test patterns can be implemented are given in Section 5.8.5.

5.8.1 Internal versus External Test Pattern Generation

A major obstacle when testing high-speed digital-to-analog converters, DACs, is to feed the DAC with test patterns at high enough sampling rates. If an external pattern generator is used, the sampling rate of the test signal is most probably limited by the speed of the chip edge interface. The chip edge interface can be either single ended, using for example standard digital CMOS pads, or differential low voltage swing LVDS pads. LVDS pads offer higher signal rates as compared to digital CMOS pads but the number of input pads are doubled.

One drawback with external pattern generation is that cross chip-edge communication typically disturbs the measurement due to the simultaneous switching noise generated by the large number of pads switching simultaneously in the pad frame. There is a potential risk that this switching noise will be transferred to the DAC via the chip substrate and hereby limit the signal-to-noise ratio, SNR, of the DAC.

These issues with external pattern generation calls for the use of on-chip generated test signals. Examples of such methods are on-chip memories

which are simply programmed with the test signal or sinusoid generators using a phase accumulator and a look-up table [74, 85].

5.8.2 Single Tone Test Signal Requirements

A single tone test signal suitable for testing DAC performance should fulfill the requirements as outlined in the list below. Even though we focus on single tone test signals these requirements also holds for dual- and multi-tone test inputs.

- Test signals should have sufficient linearity not to hide performance limitations in the DAC, and also meet the coherently sampling criterion.
- Distortion terms due to non-linearities in the DAC should not fold on other distortion terms, i.e., they should be uniquely defined on the frequency axis.
- Test signals must be generated with high enough sampling frequency so that dynamic limitations of the DAC can be accurately measured.
- Test signals should ideally cover all input codes in the DAC.

The last bullet in not a necessary requirement but it adds important information on matching errors and other errors originating from the manufacturing of the DAC. In Tables 5.2-5.3 the relative code coverage is measured as the ratio of all unique codes in the cycle and the number of possible codes in the converter.

5.8.3 Suitable Single Tone Signal Frequencies

If sampling frequency is a critical performance measure for our test signal generator a first attempt would be to investigate multiplier coefficients, α , using shifts only. Hence multiplier coefficients that can be written on the form $\alpha = \pm 2^i$ where i is an integer in the range $i \in [-W/2 \ W/2]$ and W is the coefficient wordlength.

The distribution of frequencies for shift-only coefficients is illustrated in Figure 5.11. As can be seen in Figure 5.11, the density of output frequencies is higher around the step angle $\theta = \pi/2$. Although not obvious from the figure, a longer wordlength W only adds frequencies closer to $\theta = \pi/2$, that is, a longer word length in the multiplier coefficient, α , does not add more unique test frequencies.

Since the harmonics generated by non-linearities in the DAC should not fold onto each other, we have derived the frequency positions of the ten first harmonics for all shift-only coefficients. Out of these coefficients ten coefficients have been found satisfying the above mentioned folding conditions. These coefficients are, $\alpha = \pm 2^{-i}, i \in [-5 \ -1]$. The harmonics positions for the negative coefficient of each pair are plotted in Figure 5.12.

 Figure 5.11: Frequency distribution of shift-only coefficients $\alpha = \pm 2^i$.

Figure 5.12: Illustration of harmonic frequency positions for selected shift only coefficients.

Note that the plots in Figure 5.12 only indicates the frequency position of the harmonics. No information on harmonic power or similar can be read out from the figure.

5.8.4 Simulation Results for Shift Only Coefficients

The initial conditions for a biquad oscillator have been optimized for the multiplier coefficients, $\alpha = \pm 2^{-i}$, $i \in [-5 - 1]$, as described in Section 5.8.3. The initial conditions have been optimized for highest SFDR using the proposed search algorithm. If a high code coverage is required, one long cycle with reasonable high SFDR has also been found. The results for 12-, 14-, and 16-bit word length implementations are summarized in Tables 5.2-5.3.

From Tables 5.2-5.3 we conclude that we are able to find cycles with high SFDR values and also with reasonable high amplitudes. If full-scale sinu-

Figure 5.13: Power spectrum of two sines (top) and (middle) and the sum of these sines (lower)

soids are a requirement, a new search restricted to full-scale sinusoids can be performed. A second option is constant scaling of the outputs as described in Section 5.3.3.

5.8.5 Two-Tone Signal Generation

The two-tone test is a second option for measuring dynamic non-linearities in the DAC. There are two main advantages of the two tone test as compared to the single tone test. First there are no problems with harmonics folding onto other harmonics, with the exception of frequencies very close to the Nyquist frequency. Second the intermodulation distortion is unaffected by the sinc-weighting.

One way of creating a two tone test signal is to add two single tones as illustrated in Figure 5.13. The upper and middle plots in Figure 5.13 show the spectrum of two 14-bit single tones from Tab. 5.2 with multiplier coefficients $\alpha_1 = -2048/2^{12}$ and $\alpha_2 = -1024/2^{12}$ respectively. The lower plot in Figure 5.13 show the spectrum of the sum these sines. To prevent overflow, the sum of the sines has been divided by two and then truncated to 14-bits.

Another interesting property of the two single tones is that the their individual cycle lengths are relatively prime (603 and 752 respectively). This means that the sum of these cycles is a non-repeating cycle $603 \cdot 752 = 453456$ samples long. The relative output code coverage for this sequence, that is how many of the DAC output values that are triggered by the sequence is 93.6%.

A second option to create a two tone test is by mixing (multiply) two sinusoids with frequencies ω_1 and ω_2 . The resulting output Y after mixing is

Figure 5.14: Power spectrum of a sine (top) and the same sine mixed with $f_{\text{samp}}/4$.

Figure 5.15: Example of a built in self test using sinusoidal generators.

two sinusoids with the frequencies $\omega_1 - \omega_2$ and $\omega_1 + \omega_2$ according to

$$Y = \cos(\omega_1 t) \cos(\omega_2 t) = \frac{1}{2} (\cos(\omega_1 t - \omega_2 t) + \cos(\omega_1 t + \omega_2 t)). \quad (5.23)$$

A hardware efficient mixer can be implemented if the mixing frequency is chosen to $\omega_1 = \pi/2$ (or in frequency $f_{\text{clk}}/4$) which only requires multiplication with the sequence $\{1, 0, -1, 0, \dots\}$.

The spectrum of a 14-bit single tone signal ($\alpha = 8177/2^{12}$, initial condition = $[332 \ 109]/2^{12}$), and the same signal after mixing with $f_{\text{clk}}/4$ is shown in Figure 5.14. Note that all DC content in the single tone must be removed since all DC power will be mixed up to $f_{\text{clk}}/4$. Figure 5.15 illustrates the three different test signal options described in this section. A single tone test

can be applied to the DAC via the lower input of the mux. Two tone tests can be applied by either adding two signals, middle input of the mux, or by mixing, upper input of the mux.

Note that the system in Figure 5.15 only shows a conceptual example of a DAC test system. In a real test system, more test frequencies are required.

5.9 Future Work

In this work we have investigated digital recursive oscillators and extended existing theory to cover also discrete-time sinusoids with any phase and amplitude. A search algorithm for finding steady-state cycles has been proposed and evaluated in numerous of simulations. The next step in this work is to evaluate the performance of the oscillators in a hardware implementation. A possible test vehicle for the hardware implementation is a built-in-test (BIST) system for DACs, following the IEEE test standard [89]. Oscillators utilizing steady-state are suitable candidates in such a system since relatively few fixed frequencies with a high spectral purity are required. The update frequency can also be high while maintaining a low hardware cost for the overall BIST system.

Table 5.2: Search results from selected shift only multiplier coefficients , α , highest linearity, truncation.

Word length 12-bit						
α	Freq. ^a	SFDR [dB]	Length ^b	Amp. ^c	Init State ^d	Coverage ^e
-2^{-1}	0.290	78.8	1175	-2.48	[2 1489]	0.13
-2^{-2}	0.270	76.4	752	-0.52	[0 1916]	0.16
-2^{-3}	0.260	78.7	427	-1.17	[18 1787]	0.05
-2^{-4}	0.255	77.6	1608	-0.18	[7 2002]	0.18
-2^{-5}	0.252	77.8	1307	-1.07	[2 1811]	0.15
2^{-5}	0.248	76.7	1608	-1.24	[4 1774]	0.18
2^{-4}	0.245	76.6	1608	-2.66	[8 1507]	0.31
2^{-3}	0.240	76.3	1708	-2.45	[2 1542]	0.31
2^{-2}	0.230	79.2	539	-2.15	[4 1585]	0.06
2^{-1}	0.210	79.0	1206	-1.64	[8 1641]	0.25

Word length 14-bit						
α	Freq.	SFDR [dB]	Length	Amp.	Init State	Coverage
-2^{-1}	0.290	90.3	603	-0.94	[56 7104]	0.02
-2^{-2}	0.270	86.4	752	-0.20	[35 7943]	0.02
-2^{-3}	0.260	88.4	427	-1.04	[80 7250]	0.02
-2^{-4}	0.255	87.3	1357	-0.05	[28 8139]	0.04
-2^{-5}	0.252	87.8	1307	-2.11	[10 6419]	0.04
2^{-5}	0.248	89.5	1709	-0.05	[23 8147]	0.10
2^{-4}	0.255	87.9	1759	-1.27	[6 7072]	0.09
2^{-3}	0.240	86.5	1733	-1.02	[4 7271]	0.05
2^{-2}	0.230	87.5	752	-0.35	[47 7808]	0.05
2^{-1}	0.210	90.3	1206	-0.56	[38 7447]	0.04

Word length 16-bit						
α	Freq.	SFDR [dB]	Length	Amp.	Init State	Coverage
-2^{-1}	0.290	97.1	5713	-0.01	[36 31672]	0.04
-2^{-2}	0.270	95.3	4675	-0.19	[9 31812]	0.03
-2^{-3}	0.260	95.9	4320	-0.34	[22 31456]	0.03
-2^{-4}	0.255	98.7	3518	-0.55	[1 30745]	0.05
-2^{-5}	0.252	96.3	2111	-1.83	[32 26542]	0.03
2^{-5}	0.248	101.3	1709	-0.42	[20 31223]	0.03
2^{-4}	0.245	102.2	1759	-0.15	[22 32182]	0.03
2^{-3}	0.240	97.3	2587	-1.38	[5 27898]	0.04
2^{-2}	0.230	95.7	5803	-1.71	[21 26691]	0.04
2^{-1}	0.210	98.1	2555	-2.93	[29 22653]	0.04

^aNormalized to Nyquist frequency.^bNon-repeating cycle length.^cAmplitude in dB relative to full-scale.^dDivide by 2^{W-2} for actual value, where W is the word length.^eRelative code coverage of the cycle.

Table 5.3: Search results from selected shift only multiplier coefficients , α , longest cycle, truncation

Word length 12-bit						
α	Freq. ^a	SFDR [dB]	Length ^b	Amp. ^c	Init State ^d	Coverage ^e
-2^{-1}	0.290	68.2	8411	-0.76	[1 1819]	0.57
-2^{-2}	0.270	68.2	8435	-0.58	[2 1892]	0.56
-2^{-3}	0.260	71.5	9494	-0.39	[0 1952]	0.62
-2^{-4}	0.255	69.4	11558	-0.60	[1 1910]	0.69
-2^{-5}	0.252	69.8	23526	-0.32	[1 1972]	0.86
2^{-5}	0.248	68.5	21910	-0.36	[0 1967]	0.88
2^{-4}	0.245	69.1	16635	-0.60	[0 1915]	0.82
2^{-3}	0.240	68.6	10323	-1.75	[0 1674]	0.61
2^{-2}	0.230	68.5	16983	-0.13	[0 1998]	0.83
2^{-1}	0.210	68.2	9934	-3.90	[0 1262]	0.53

Word length 14-bit						
α	Freq.	SFDR [dB]	Length	Amp.	Init State	Coverage
-2^{-1}	0.290	80.1	15044	-0.03	[1 7908]	0.34
-2^{-2}	0.270	80.1	9350	-0.87	[8 7355]	0.24
-2^{-3}	0.260	81.2	6028	-1.64	[4 6767]	0.17
-2^{-4}	0.255	81.5	10001	-1.65	[1 6773]	0.25
-2^{-5}	0.252	81.0	11058	-0.29	[7 7927]	0.27
2^{-5}	0.248	80.0	21514	-0.39	[3 7835]	0.45
2^{-4}	0.245	80.9	12464	-0.10	[0 8093]	0.30
2^{-3}	0.240	80.4	9494	-0.66	[1 7583]	0.24
2^{-2}	0.230	80.1	11606	-0.93	[0 7308]	0.27
2^{-1}	0.210	82.1	8871	-0.72	[1 7307]	0.23

Word length 16-bit						
α	Freq.	SFDR [dB]	Length	Amp.	Init State	Coverage
-2^{-1}	0.290	92.5	17139	-0.50	[7 29965]	0.12
-2^{-2}	0.270	92.5	14777	-0.44	[3 30918]	0.20
-2^{-3}	0.260	92.8	9494	-0.33	[10 31500]	0.07
-2^{-4}	0.255	92.9	15429	-1.40	[3 27888]	0.11
-2^{-5}	0.252	92.9	8947	-0.07	[3 32508]	0.07
2^{-5}	0.248	92.9	11058	-0.11	[10 32331]	0.08
2^{-4}	0.245	92.6	14072	-0.53	[76 30810]	0.09
2^{-3}	0.240	94.1	11227	-0.08	[3 32387]	0.08
2^{-2}	0.230	93.0	10102	-1.95	[6 25974]	0.07
2^{-1}	0.210	92.2	17742	-0.54	[1 29810]	0.23

^aNormalized to Nyquist frequency.^bNon-repeating cycle length.^cAmplitude in dB relative to full-scale.^dDivide by 2^{W-2} for actual value, where W is the word length.^eRelative code coverage of the cycle.

Bibliography

- [1] ITRS, *International technology roadmap for semiconductors*, Jan. 2013. [Online]. Available: <http://www.itrs.net>.
- [2] E. A. Vittoz, "Low-power design: ways to approach the limits", in *Proc. IEEE Int. Solid-State Circuit Conf.*, IEEE, 1994, pp. 14–18.
- [3] R. Sarpeshkar, "Analog versus digital: extrapolating from electronics to neurobiology", *Neural computation*, vol. 10, no. 7, pp. 1601–1638, 1998.
- [4] D. J. White, P. E. William, M. W. Hoffman, and S. Balkir, "Low-power analog processing for sensing applications: low-frequency harmonic signal classification", *Sensors*, vol. 13, no. 8, pp. 9604–9623, 2013.
- [5] J. Daniels, W. Dehaene, M. Steyaert, and A. Wiesbauer, "A/D conversion using asynchronous Delta-Sigma modulation and time-to-digital conversion", *IEEE Trans. Circuits Syst. I*, vol. 57, no. 9, pp. 2404–2412, 2010.
- [6] M. M. Elsayed, V. Dhanasekaran, M. Gambhir, J. Silva-Martinez, and E. Sanchez-Sinencio, "A 0.8 ps DNL time-to-digital converter with 250 MHz event rate in 65 nm CMOS for time-mode-based modulator", *IEEE J. Solid-State Circuits*, vol. 46, no. 9, pp. 2084–2098, 2011.
- [7] T. Rahkonen and J. Kostamovaara, "Low-power time-to-digital and digital-to-time converters for novel implementations of telecommunication building blocks", in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 3, 1994, pp. 141–144.
- [8] J. B. Anderson, *Digital transmission engineering*. Piscataway, NJ: Wiley-IEEE Press, 2005, ISBN: 0471694649.
- [9] R. van de Plassche, *Integrated analog-to-digital and digital-to-analog converters*. Kluwer Academic Publishers, 1994, ISBN: 0-7923-9436-4.
- [10] A. Biman and D. Nairn, "Trimming of current mode DACs by adjusting V_t ", in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1, May 1996, 33–36 vol.1.
- [11] R. J. van de Plassche, "Dynamic element matching for high-accuracy monolithic D/A converters", *IEEE J. Solid-State Circuits*, vol. 11, no. 6, pp. 795–800, 1976.

- [12] B. H. Leung, "Architectures for multi-bit oversampled A/D converter employing dynamic element matching techniques", in *Circuits and Systems, 1991., IEEE International Symposium on*, 1991, pp. 1657–1660.
- [13] P. Carbone and I. Galton, "Conversion error in D/A converters employing dynamic element matching", in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2, 1994, pp. 13–16.
- [14] I. Galton and P. Carbone, "A rigorous error analysis of D/A conversion with dynamic element matching", *IEEE Trans. Circuits Syst. II*, vol. 42, no. 12, pp. 763–772, 1995.
- [15] H. T. Jensen and I. Galton, "A low-complexity dynamic element matching DAC for direct digital synthesis", *IEEE Trans. Circuits Syst. II*, vol. 45, no. 1, pp. 13–27, 1998.
- [16] —, "An analysis of the partial randomization dynamic element matching technique", *IEEE Trans. Circuits Syst. II*, vol. 45, no. 12, pp. 1538–1549, 1998.
- [17] S. Henzler, *Time-to-digital converters*, ser. Springer Series in Advanced Microelectronics. Springer, 2010, ISBN: 9789048186273.
- [34] D. Marsh, R. Tynan, D. O’Kane, and G. M. P. O’Hare, "Autonomic wireless sensor networks", *Engineering Applications of Artificial Intelligence*, vol. 17, no. 7, pp. 741–748, 2004.
- [35] D. Jones and K. Martin, *Analog integrated circuit design*. John Wiley & Sons, 1997, p. 706, ISBN: 0-471-14448-7.
- [36] B. Razavi, *Design of analog CMOS integrated circuits*. McGraw-Hill, 2000, ISBN: 0-07-238032-2.
- [37] M. Gustavsson, J. Wikner, and N. Tan, *CMOS data converters for communications*. Kluwer Academic Publishers, 2000, ISBN: 0-7923-7780-X.
- [38] J. Wikner, "Studies on CMOS digital to analog converters", Linköping Studies in Science and Technology, Dissertation No. 667, PhD thesis, Linköping University, 2001.
- [39] R. E. Ziemer, W. H. Tranter, and D. R. Fannin, *Signals and systems: continuous and discrete*. Prentice Hall, 1998, vol. 4.
- [40] Y. Cong and R. L. Geiger, "Formulation of INL and DNL yield estimation in current-steering D/A converters", in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, vol. 3, 2002.
- [41] M.-H. Shen, J.-H. Tsai, and P.-C. Huang, "Random swapping dynamic element matching technique for glitch energy minimization in current-steering DAC", *IEEE Trans. Circuits Syst. II*, vol. 57, no. 5, pp. 369–373, 2010.
- [42] K. O. Andersson and M. Vesterbacka, "Modeling of glitches due to rise/fall asymmetry in current-steering digital-to-analog converters", *IEEE Trans. Circuits Syst. I*, vol. 52, no. 11, pp. 2265–2275, 2005.

- [43] J. Brown J. L., "Note on complete sequences of integers", English, *The American Mathematical Monthly*, vol. 68, no. 6, pp. 557–560, 1961, ISSN: 00029890.
- [44] R. Honsberger and M. A. of America, "Mathematical gems iii", in: ser. Dolciani mathematical expositions. Published and distributed by the Mathematical Association of America, 1985, pp. 123–130, ISBN: 9780883853139.
- [45] J. Wikner and M. Vesterbacka, "D/A conversion with linear-coded weights", in *Mixed-Signal Design, 2000. SSMSD. 2000 Southwest Symposium on*, 2000, pp. 61–66.
- [46] —, "Characteristics of linear-coded D/A converters", in *Mixed-Signal Design, 2000. SSMSD. 2000 Southwest Symposium on*, 2000, pp. 67–72.
- [47] M. Vesterbacka and J. Wikner, "Design of encoders for linear-coded D/A converters", in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1, May 2001, 524–527 vol. 1.
- [48] R. Kubokawa, T. Ohshima, A. Tomar, P. Ramesh, H. Kanaya, and K. Yoshida, "Development of low power DAC with pseudo fibonacci sequence", in *Proc. IEEE Asia-Pacific Conf. Circuits Syst.*, 2010, pp. 370–373.
- [49] K. Hokazono, D. Kanemoto, R. Pokharel, A. Tomar, H. Kanaya, and K. Yoshida, "A low-glitch and small-logic-area fibonacci series DAC", in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, 2011, pp. 1–4.
- [50] A. Pacut and K. Hejn, "Analog-to-digital converters: towards a generalization of widrow's theorem", in *Instrumentation and Measurement Technology Conference, 1998. IMTC/98. Conference Proceedings. IEEE*, vol. 2, May 1998, pp. 1190–1197.
- [51] E. Säll, "Implementation of flash analog-to-digital converters in silicon-on-insulator CMOS technology", PhD thesis, Linköping University, Electronics Systems, 2007, p. 173.
- [52] Å. Björck, *Numerical methods for least squares problems*. Philadelphia, Pa.: SIAM, Society for Industrial and Applied Mathematics, cop. 1996, 1996, ISBN: 0898713609.
- [53] I. Galton, "Why dynamic-element-matching DACs work", *IEEE Trans. Circuits Syst. II*, vol. 57, no. 2, pp. 69–74, 2010.
- [54] —, "Spectral shaping of circuit errors in digital-to-analog converters", *IEEE Trans. Circuits Syst. II*, vol. 44, no. 10, pp. 808–817, 1997.
- [55] C. Tokunaga, D. Blaauw, and T. Mudge, "True random number generator with a metastability-based quality control", *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 78–85, Jan. 2008, ISSN: 0018-9200.
- [56] R. N. Mutagi, "Pseudo noise sequences for engineers", *Electronics Communication Engineering Journal*, vol. 8, no. 2, pp. 79–87, Apr. 1996, ISSN: 0954-0695.

- [57] F. Ellinger, M. Claus, M. Schroter, and C. Carta, "Review of advanced and beyond CMOS FET technologies for radio frequency circuit design", in *IEEE MTT-S International Microwave Optoelectronics Conference (IMOC)*, 2011, pp. 347–351.
- [58] B. Razavi, *RF microelectronics*. Prentice Hall New Jersey, 1998, vol. 1.
- [59] T. E. Rahkonen and J. T. Kostamovaara, "The use of stabilized CMOS delay lines for the digitization of short time intervals", *IEEE J. Solid-State Circuits*, vol. 28, no. 8, pp. 887–894, 1993.
- [60] P. Capofreddi, "Method and apparatus for low power thermometer to binary coder", pat. 6,542,104 B1, Apr. 2003.
- [61] E. Säll, M. Vesterbacka, and K. Andersson, "A study of digital decoders in flash analog-to-digital converters", in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1, 2004, pp. 129–132.
- [62] E. Säll and M. Vesterbacka, "Thermometer-to-binary decoders for flash analog-to-digital converters", in *Proc. Europ. Conf. Circuit Theory Design*, 2007, pp. 240–243.
- [63] —, "A multiplexer based decoder for flash analog-to-digital converters", in *TENCON 2004. 2004 IEEE Region 10 Conference*, 2004, pp. 250–253.
- [64] G. Frank, "Pulse code communication", pat. 2 632 058, Mar. 1953.
- [65] L. Råde and B. Westergren, *Mathematics handbook for science and engineering*. Lund : Studentlitteratur, 1998, 1998, ISBN: 9144008392.
- [66] M. Lee and A. A. Abidi, "A 9 b, 1.25 ps resolution coarse-fine time-to-digital converter in 90 nm CMOS that amplifies a time residue", *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 769–777, 2008.
- [67] Y.-H. Seo, J.-S. Kim, H.-J. Park, and J.-Y. Sim, "A 0.63ps resolution, 11b pipeline TDC in 0.13 μ m CMOS", in *IEEE Symposium on VLSI Circuits (VLSIC)*, 2011, pp. 152–153.
- [68] M. Z. Straayer and M. H. Perrott, "A multi-path gated ring oscillator TDC with first-order noise shaping", *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1089–1098, 2009.
- [69] Q. Chen, Z. H. Shen, N. Yan, X. Tan, and H. Min, "Monolithic digitally controlled buck converter with TDC-based ADC sharing delay cells with DPWM", *Electronics Letters*, vol. 48, no. 20, pp. 1303–1304, 2012.
- [70] S. Henzler, S. Koeppe, W. Kamp, H. Mulatz, and D. Schmitt-Landsiedel, "90nm 4.7ps-resolution 0.7-LSB single-shot precision and 19pJ-per-shot local passive interpolation time-to-digital converter with on-chip characterization", in *Proc. IEEE Int. Solid-State Circuit Conf.*, 2008, pp. 548–635.
- [71] Y. Park and D. D. Wentzloff, "A cyclic Vernier TDC for ADPLLs synthesized from a standard cell library", *IEEE Trans. Circuits Syst. I*, vol. 58, no. 7, pp. 1511–1517, 2011, ISSN: 1549-8328.

- [72] L. Vercesi, A. Liscidini, and R. Castello, "Two-dimensions Vernier time-to-digital converter", *IEEE J. Solid-State Circuits*, vol. 45, no. 8, pp. 1504–1512, 2010.
- [73] P. Lu, A. Liscidini, and P. Andreani, "A 3.6 mW, 90 nm CMOS gated-Vernier time-to-digital converter with an equivalent resolution of 3.2 ps", *IEEE J. Solid-State Circuits*, vol. 47, no. 7, pp. 1626–1635, 2012.
- [74] J. Vankka, *Digital synthesizers and transmitters for software radio*. Kluwer Academic Pub., 2005.
- [75] N. J. Fliege and J. Wintermantel, "Complex digital oscillators and FSK modulators", *IEEE Trans. Signal Process.*, vol. 40, no. 2, pp. 333–342, 1992.
- [76] C. S. Turner, "Recursive discrete-time sinusoidal oscillators", *IEEE Signal Process. Mag.*, vol. 20, no. 3, pp. 103–111, 2003.
- [77] J. Dattorro, "Effect design: part 3 oscillators: sinusoidal and pseudonoise", *J. Audio Eng. Soc.*, vol. 50, no. 3, pp. 115–146, 2002.
- [78] B. Kim and J. A. Abraham, "Efficient loopback test for aperture jitter in embedded mixed-signal circuits", *IEEE Trans. Circuits Syst. I*, vol. 58, no. 8, pp. 1773–1784, Aug. 2011, ISSN: 1549-8328.
- [79] G. Starr, J. Qin, B. Dutton, C. Stroud, F. Dai, and V. Nelson, "Automated generation of built-in self-test and measurement circuitry for mixed-signal circuits and systems", in *Proc. IEEE Int. Defect Fault Tolerance VLSI Syst.*, 2009, pp. 11–19.
- [80] F. Curticapean, K. Palomäkia, and J. Niittylahti, "Hardware implementation of a quadrature digital oscillator", in *Proc. IEEE Nordic Signal Process. Symp.*, Jun. 2000, pp. 291–294.
- [81] I. Hartimo, "Self-sustained stable oscillations of second order recursive algorithms", in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 8, 1983, pp. 635–638.
- [82] K. Furuno, S. K. Mitra, K. Hirano, and Y. Ito, "Design of digital sinusoidal oscillators with absolute periodicity", *IEEE Trans. Aerosp. Electron. Syst.*, no. 6, pp. 1286–1299, 1975.
- [83] B. Gold and C. Rader, *Digital processing of signals*. McGraw-Hill, 1969.
- [84] J. Smith and P. Cook, "The second-order digital waveguide oscillator", in *Proc. Int. Computer Music Conf.*, San Jose, CA, Oct. 1992, pp. 150–153.
- [85] J. M. P. Langlois and D. Al-Khalili, "Phase to sinusoid amplitude conversion techniques for direct digital frequency synthesis", *IEE Proc. Circuits Devices Syst.*, vol. 151, no. 6, pp. 519–528, Dec. 2004, ISSN: 1350-2409.
- [86] L. B. Jackson, *Digital filters and signal processing: with MATLAB exercises*. Boston, MA: Kluwer Academic Pub., 1996, ISBN: 079239559X.

- [87] A. Wenzler and E. Luder, "New structures for complex multipliers and their noise analysis", *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2, pp. 1432–1435, 1995.
- [88] L. Tan and J. Jiang, *Digital signal processing : fundamentals and applications*. Amsterdam ; Boston : Elsevier/Academic Press, 2013, ISBN: 9780124158931.
- [89] *IEEE standard for terminology and test methods of digital-to-analog converter devices*, Feb. 2012.
- [90] N. U. Andersson and J. J. Wikner, "A comparison of dynamic element matching in DACs", English, *Proc. Norchip*, pp. 385–390, Nov. 1999.
- [91] —, "A strategy for implementing dynamic element matching in current-steering DACs", in *Proc. SSMSD Mixed-Signal Design 2000 Southwest Symp*, IEEE, 2000, pp. 51–56.

Papers

The articles associated with this thesis have been removed for copyright reasons. For more details about these see:

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-112215>