Simon Louwsma
Ed van Tuijl
Bram Nauta

ACSP
Analog Circuits And Signal Processing

Time-interleaved Analog-to-Digital Converters

Springer

Time-interleaved Analog-to-Digital Converters

ANALOG CIRCUITS AND SIGNAL PROCESSING SERIES

Consulting Editor*: Mohammed Ismail.** ***Ohio State University

For other titles published in this series, go to
www.springer.com/series/7381

Simon Louwsma • Ed van Tuijl • Bram Nauta

Time-interleaved Analog-to-Digital Converters

Springer

Simon Louwsma
Axiom IC
Colosseum 28
Enschede, 7521 PT
Netherlands
simon.louwsma@gmail.com

Ed van Tuijl
Axiom IC/University of Twente
Colosseum 28
Enschede, 7521 PT,
Netherlands
ed.van.tuijl@axiom-ic.com

Bram Nauta
MESA+ Institute
University of Twente
P.O. Box 217
Enschede, 7500 AE
Netherlands
b.nauta@utwente.nl

ISBN 978-90-481-9715-6 e-ISBN 978-90-481-9716-3
DOI 10.1007/978-90-481-9716-3
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2010935602

Cover design: eStudio Calamar S.L.

Cover background was designed by Erik Bosgra.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

“Everything should be made as simple as possible, but not simpler.”
— Albert Einstein

Preface

This book describes the research carried out by our PhD student Simon Louwsma at the University of Twente, The Netherlands in the field of high-speed Analog-to-Digital (AD) converters. AD converters are crucial circuits for modern systems where information is stored or processed in digital form. Due to increasing data rates and further digitization of systems, the demands on the AD converters are increasing in both sample-rate and number of bits. A fast and accurate AD converter combined with digital signal processing offers an attractive alternative for the analog signal chain still present in many actual receivers. This book offers an exploration of fundamental and practical limits of high speed AD conversion, aiming at a step forward in number of bits and sample-rate, while keeping the power consumption low. To achieve high performance, a technique called time interleaving is used. Time interleaving is the analog equivalent of parallel processing in the digital domain. To implement this, instead of a single Track-and- Hold (T&H), we use a whole series of them, each sampling a bit later than the previous one. In the design example in this book we use 16 T&H circuits, followed by 16 sub-AD converters. The timing alignment of these T&H circuits needs to be extremely accurate, and conventionally, complex timing calibration is used to achieve this. Here however, it is shown that even better performance can be achieved by a compact and good design of the timing circuit without requiring any timing calibration. The circuits use a minimum of transistors that cause timing inaccuracies and special layout techniques are the finishing touch. Thanks to the absence of a control range for the timing, the amount of jitter is also reduced. To save power and to keep the input capacitance low, small sized transistors are used in the time-interleaving T&H circuitry. Only simple DC calibrations are needed to make the 16 paths behave equally over the whole input frequency range. An extensive analysis of accuracy and timing requirements is given and circuit solutions are described in detail. After the input signal is sampled by a T&H section, a sub-ADC finalizes the conversion. Pipeline AD converters are popular for conversion rates around 100 MS/s, but they suffer from the fact that even in the first stage of the pipeline the full accuracy for settling is required. This makes the design of high speed in combination with a high accuracy quite a challenge. Instead of that, we use sub-ADCs based on Successive Approximation (SA). As explained in this book, this has quite some advantages: A SAR ADC contains

less critical analog blocks, and its power consumption can be ten times less than a comparable pipeline ADC. A potential disadvantage of Successive Approximation converters is the relatively low maximum sample-rate. This problem is tackled with a new overrange technique that greatly reduces the demands on settling time per conversion step and that postpones the critical decision to the last conversion step. This offers great advantage over a Pipeline ADC, where the first residue amplifier must settle to full accuracy to avoid unrecoverable analog errors in the conversion process. The work described in this book shows state-of-the art performance and describes techniques, which gain popularity among today's AD converter designers. We enjoyed carrying out the research with Simon and we hope you will enjoy reading the results.

University of Twente, Enschede, The Netherlands

Professor Ed van Tuijl
Bram van Nauta

Contents

About the Author

Simon M. Louwsma was born on 1 January 1976, in Wommels, The Netherlands. He received the M.Sc. degree in electrical engineering from the University of Twente, Enschede, The Netherlands, in 2001. Following that, he has been working towards his Ph.D. degree on time-interleaved ADCs, within the IC-Design group at the same university. The results of his research are contained in this book. He holds several patents and is co-founder of Axiom-IC, which specializes in data converters and other mixed-signal circuits and systems.

Nomenclature

List of Symbols

β	Gain factor
σ	Standard deviation
τ	Time-constant
C	Capacitance
f	Frequency
f_S	Sample-rate
g_m	Transconductance
I	Current
k	Boltzmann constant
N	Number of channels
n	Resolution
P	Power consumption
Q	Charge
R	Resistance
T	Temperature
t	Time
V	Voltage

List of Abbreviations

ADC	Analog-to-Digital Converter
BW	Bandwidth
CMOS	Complementary Metal Oxide Semiconductor
DAC	Digital-to-Analog Converter
DNL	Differential Non-Linearity
ENOB	Effective Number of Bits
ERBW	Effective Resolution Bandwidth
FoM	Figure of Merit
FRS	Frontend Sampler
INL	Integral Non-Linearity

LSB	Least Significant Bit
MOST	Metal Oxide Semiconductor Transistor
MSB	Most Significant Bit
RMS	Root Mean Square
SA-ADC	Successive Approximation ADC
SNDR	Signal-to-Noise-and-Distortion Ratio
SNR	Signal-to-Noise Ratio
T&H	Track and Hold
THD	Total Harmonic Distortion

Chapter 1
Introduction

1.1 Analog-to-Digital Conversion

Analog-to-digital conversion is all around us. Whether you are making a phone call, taking pictures, browsing the internet or even when doing the laundry, it all involves signal conversion between the analog and the digital domain. The proliferation of analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) and the amount of R&D spent on it can be explained by five reasons.

The first is that the physical world around us is analog[1] and will remain analog: Music traveling through the air as sound waves, candlelight reflected from someone's face, electromagnetic waves from GSM devices or GPS satellites, the temperature of the laundry, the orientation of your fancy smart-phone, and so on.

The second reason is that the processing of signals in the digital domain has many advantages. Digital signals are quantized in both time and amplitude, and can be stored and processed with (almost) unlimited accuracy. Moreover, up to a certain noise margin, signal integrity is not affected by distortion and noise. Also in terms of power consumption, digital signal processing can be advantageous: For an increase in the signal-to-noise ratio (SNR) of 3 dB, the power consumption typically doubles in the analog domain, while in the digital domain it only increases with a fraction of $1/n$ to $2/n$, with n the resolution.[2] Certain types of signal processing are hardly feasible in the analog domain, while they are not a problem in the digital domain. For example, the implementation of wideband filters with a constant group delay, or the application of OFDM schemes, which have many advantages [62]. Testing of digital systems and porting digital circuitry to newer technologies can be automated relatively easy, in contrast to analog systems. Thanks to the use of software, digital processing can also be made flexible.

The third reason is that new applications require higher data rates and better power efficiencies. For example, software defined and cognitive radios require

[1] Let's neglect quantum effects here.

[2] The actual fraction depends on the function and implementation.

S.M. Louwsma et al., *Time-interleaved Analog-to-Digital Converters,*
Analog Circuits and Signal Processing,
DOI 10.1007/978-90-481-9716-3_1, © Springer Science+Business Media B.V. 2011

ADCs with high sample-rates, to create a flexible radio suitable for multiple standards; the same holds for wideband conversion, where a complete radio band is digitized (e.g. satellite or cable-TV), such that the channel selection and filtering can be performed in the digital domain. This saves energy needed for analog filtering and it is more flexible: for example the amount of filtering and the sample-rate of the ADC can be adapted to the actual strength of interferers. Moreover, there is a continuous demand for more features and more performance: more mega-pixels, higher bandwidths of radio links (e.g. WiFi, 3G, wireless USB, and Bluetooth 3), the use of video instead of pictures, and so on. Mobile applications run on batteries and need ADCs with a good power efficiency. There is a market demand for devices with the same processing power as a workstation of a few years ago, but now they must fit in a pocket and work on batteries for days.

The fourth reason is that digital signal processing continuously becomes less expensive with respect to power consumption and die area. In 1965, Gordon Moore stated that the number of components on a chip would double every year and that as a result the cost would decrease exponentially [30]. Ten years later he corrected this to a double amount of components every 2 years[3] and to date "Moore's law" still holds, it has become a self-fulfilling prophecy. Assuming constant field scaling [11], the area scales with $1/s^2$, while energy scales with $1/s^3$, with s the scaling factor. So, the power efficiency of digital logic decreases even faster than its area.

To explain the fifth reason, the increasing power efficiency of digital signal processing is compared to the development of the power efficiency of ADCs over the years. The well known Figure of Merit (FoM) for ADCs[4] [60] is a measure of the power efficiency:

$$\text{FoM} = \frac{P}{2^{\text{ENOB}} \cdot f_{\text{S}}} \tag{1.1}$$

with P the power consumption, ENOB the effective number of bits and f_{S} the sample-rate. In Fig. 1.1 the FoM is plotted as a function of the year of publication for Nyquist ADCs presented at the ISSCC and VLSI conferences from 1998 to 2009 [32].

From this figure, it can be concluded that the increase in power efficiency of ADCs is about a factor of 2 every 2 years. In [31] the same conclusion is drawn with a slightly different FoM. The increase in ADC power efficiency is thus slower than that of digital signal processing. So, in the course of time, the power consumption of ADCs will become more dominant in a mixed-signal system, increasing the demand for power efficient ADCs.

[3] Despite popular misconception, Moore is adamant that he did not predict a doubling "every 18 months". However, an Intel colleague had factored in the increasing performance of transistors to conclude that integrated circuits would double in performance every 18 months [59].

[4] Although this figure of merit is well known, not everyone agrees with it [31], since for thermal noise limited designs, the power scales with $2^{2 \cdot \text{ENOB}}$. This book focuses on converters with resolutions up to 10 bits, which are usually not noise limited. Therefore, it does make sense to use this FoM here.

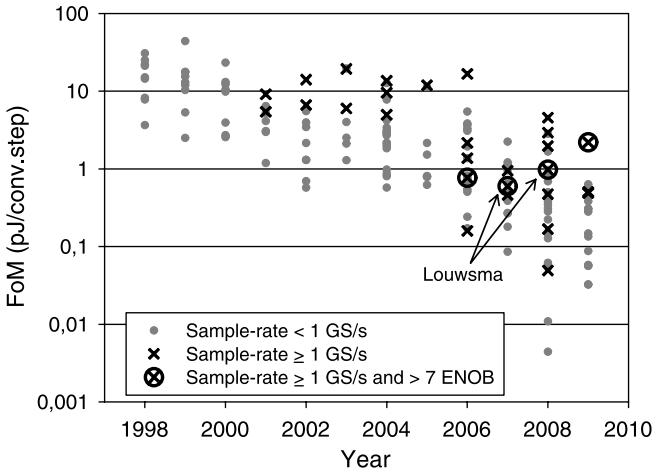

Fig. 1.1 ADC Figure of Merit as a function of the year of publication

In conclusion, the world is analog, while digital signal processing has many advantages, the requirements of systems increase and the power consumption of digital processing decreases rapidly. This creates a large demand for ADCs with high sample-rates, high resolutions and low power consumption.

In this book the feasibility is described of an analog-to-digital converter with a sample-rate of 1–2 GS/s, a resolution of 8–10 bits, and a power efficiency of less than 1 pJ/conversion-step. Example applications include wideband conversion of cable-TV, software-defined radios and future, high-speed communication standards.

At the start of this project in 2002, a FoM of 1 pJ/conversion-step was state-of-the-art for converters with a much lower sample-rate. For converters with a sample-rate of 1 GS/s and above, indicated by crosses in Fig. 1.1, the best power efficiencies were 10 times lower. Moreover, converters with a sample-rate of at least 1 GS/s, and an accuracy of 7 ENOB or more could not even be found in open literature, so the target specifications were challenging indeed.

Transferring the digital data stream of the ADC of up to 20 Gb/s to another chip is costly in terms of power consumption and number of pins. To realize a low-cost solution, the digital signal processing should be integrated together with the ADC on a single chip. For digital signal processing, CMOS technology is preferred and therefore, the ADC should be realized in this technology as well.

1.2 Architecture

Sample-rates beyond 1 GHz can be reached by the full flash architecture [7, 13], the time-interleaved architecture [8], and to a lesser extent by the folding architecture [51]. Due to the large number of parallel comparators in a flash converter, its input capacitance becomes impractically large for resolutions above 6 bits, limiting

the input bandwidth. Moreover, since the input capacitance is non-linear, it is hard to drive without introducing much distortion. The power consumption of a full flash converter increases with a factor of 8 per bit, resulting in a low power efficiency for resolutions above 6 bits. Techniques like folding help, but at the cost of the maximum bandwidth.

In the time-interleaved architecture, multiple converters (sub-ADCs) are used in parallel to increase the sample-rate. The sub-ADCs are clocked such that the combination appears as a single, fast ADC. The relatively low sample-rate of the sub-ADCs enables a high power efficiency [31], which is desired for the target specifications.

However, time-interleaving involves more than just placing a number of ADCs in parallel: The Track and Holds[5] (T&Hs) need to handle much higher frequencies than the T&H of a single non-interleaved sub-ADC, while maintaining good linearity. Moreover, matching between different channels is required to avoid spurious tones.

Another parallel architecture is frequency interleaving. As, the different sub-ADCs need input filtering to avoid aliases, and analog filtering is expensive in terms of power consumption, this technique is rarely used and will not be further investigated in this book.

For the desired specifications, the time-interleaved architecture is therefore the most suitable architecture.

1.3 Outline

Chapter 2 describes the time-interleaved Track and Hold (T&H) and design choices for the implementation are derived. The necessity of channel matching is explained, different architectures are described, and the buffer driving the sub-ADC is discussed. Moreover, the optimum number of channels is discussed, and calibration and jitter issues are explained.

Chapter 3 treats the sub-ADC for a time-interleaved ADC. An important aspect is the power efficiency, as the sub-ADCs usually dominate the total power consumption. As ADCs with the Successive Approximation (SA) architecture can achieve a very good efficiency, this architecture is described in detail and it is compared with the popular opamp based pipeline ADC. Techniques to increase the sample-rate and to decrease the power consumption are presented.

In Chap. 4 the implementation of a 16 channel, time-interleaved ADC is presented. All circuit blocks are described in detail including the interaction between different channels, calibration setting and layout issues. Measurement results are presented and compared to other state-of-the-art converters. Chapter 5 summarizes and concludes this book.

[5]The terms Track and Hold (T&H) and Sample and Hold (S&H) usually refer to the same process. Since a sampling action in an actual silicon implementation is not performed instantaneously, the term T&H is more appropriate, and is used in this book.

Chapter 2
Time-interleaved Track and Holds

2.1 Introduction

This chapter describes the time-interleaved Track and Hold (T&H) for the use in a time-interleaved ADC. In Fig. 2.1 an implementation of a time-interleaved ADC is shown consisting of several channels, each with a T&H section and a sub-ADC. The sample-rate of an interleaved ADC is N times the sample-rate of a sub-ADC, with N the number of channels. The main benefit of the time-interleaved architecture is that the overall sample-rate can be very high, while the sub-ADCs only need a moderate sample-rate, enabling a high power efficiency.

For $N = 16$, an example of a corresponding timing diagram is shown in Fig. 2.2. At each falling edge of the master-clock (MCLK), one of the T&Hs goes from track-mode to hold-mode and takes a sample of the input-signal. For a master-clock with sample-rate f_S and a number of channels N, each T&H and sub-ADC has a sample-rate of f_S/N.

Making a time-interleaved ADC involves more than just placing a few non-interleaved ADCs in parallel, since the requirements for a non-interleaved T&H and ADC differ from that of a time-interleaved T&H and sub-ADC: Aspects like offset, gain error and absolute timing, which are usually not an issue for a general-purpose non-interleaved T&H and ADC, are important for a time-interleaved architecture, as will be explained in this chapter. Moreover, the Nyquist frequency of a time-interleaved architecture is N times higher than that of a non-interleaved ADC, so the T&Hs should have a much higher bandwidth and should be able to sample signals with an N times higher frequency.

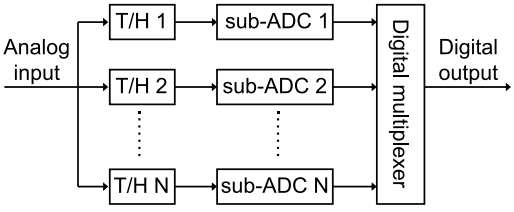

Fig. 2.1 The time-interleaved ADC architecture

S.M. Louwsma et al., *Time-interleaved Analog-to-Digital Converters,*
Analog Circuits and Signal Processing,
DOI 10.1007/978-90-481-9716-3_2, © Springer Science+Business Media B.V. 2011

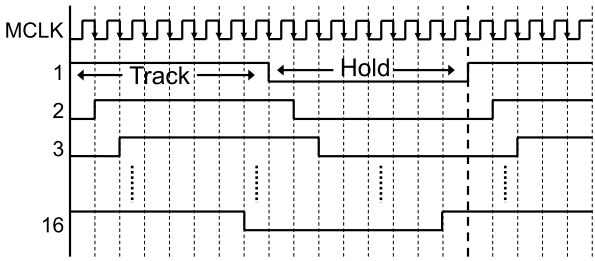

Fig. 2.2 Timing diagram of a time-interleaved ADC

In this chapter, aspects associated with high-speed time-interleaved T&Hs will be discussed, starting with matching aspects between channels in Sect. 2.2, followed by the description of time-interleaved T&H architectures in Sect. 2.3. In Sect. 2.4 T&H buffers will be treated, and Sect. 2.5 discusses the use of bottom-plate sampling in a time-interleaved T&H. This is followed by a discussion on the optimum number of channels in Sect. 2.6. The chapter ends with aspects associated with calibration in Sect. 2.7 and jitter requirements in Sect. 2.8.

2.2 Mismatch Between Channels

As stated above, the requirements for a non-interleaved general-purpose ADC for offset, gain and timing (e.g. the delay from the sample-clock to the actual sample moment) are usually not strict. As long as they are constant, they do not affect the Signal-to-Noise-and-Distortion Ratio (SNDR), Integral Non-Linearity (INL) and Differential Non-Linearity (DNL). For a time-interleaved ADC consisting of multiple channels, the situation is different. Differences in e.g. offset, gain or timing/phase between channels cause spurious tones [8, 21]. Offset mismatch causes distortion tones at multiples of f_S/N, while mismatch in gain or timing results in tones at multiples of $f_S/N \pm f_{IN}$. In Fig. 2.3 the spectrum of a reconstructed sinusoid is shown for $N = 8$ and band-limited to $f_S/2 + f_{IN}$, with f_{IN} the frequency of the input signal. The upper part shows the case where only offset mismatch is present and the lower part shows the effect of gain or phase mismatch. The amplitude of the spurious tones depends on the offset/gain distribution of the channels and on the number of channels: for a larger number of channels, the error energy is divided between more tones, so the amplitude per tone decreases.

2.2.1 Origin of Spurious Tones

To give insight in the origin of spurious tones, three graphical examples are given for a time-interleaved T&H with two channels. An analytical analysis is given in [21]. Channel offset is analyzed first. In the upper part of Fig. 2.4, the sampled output of the two channels is shown. From this, a common signal and a difference signal can

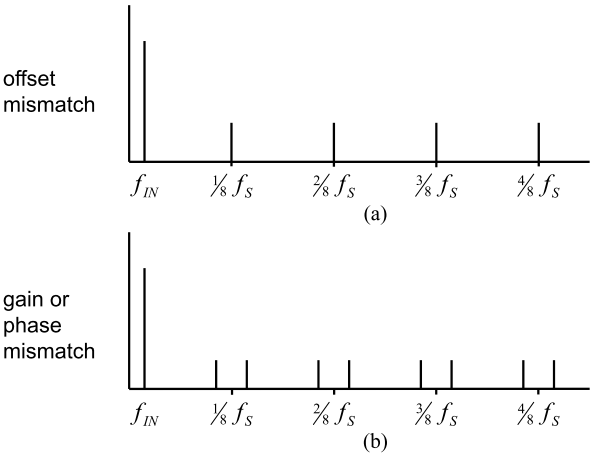

Fig. 2.3 Spectrum of a reconstructed sinusoid for a time-interleaved ADC with 8 channels and mismatch in (**a**) offset and (**b**) gain or phase

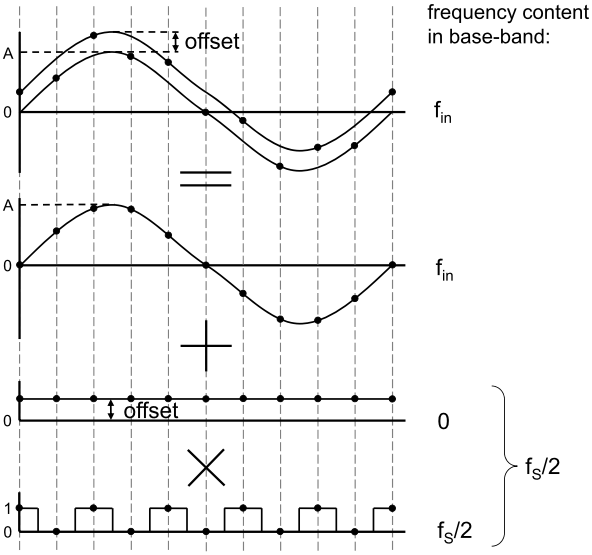

Fig. 2.4 Visualization of frequency content in the case of offset between 2 channels

be derived, see the rest of the figure. The common signal is one of the input signals and the difference signal is the offset multiplied by a square-wave with frequency $f_S/2$. In a spectrum limited to the Nyquist frequency, this results in a tone at $f_S/2$. For a larger number of channels, the situation is similar, however the number of square-waves (tones) is higher [21]. The tones are static and do not depend on the input signal.

In the case of gain mismatch between two channels, the situation is illustrated in Fig. 2.5. The common signal is again one of the input signals and the difference signal is the difference between the input signals multiplied by a square-wave with

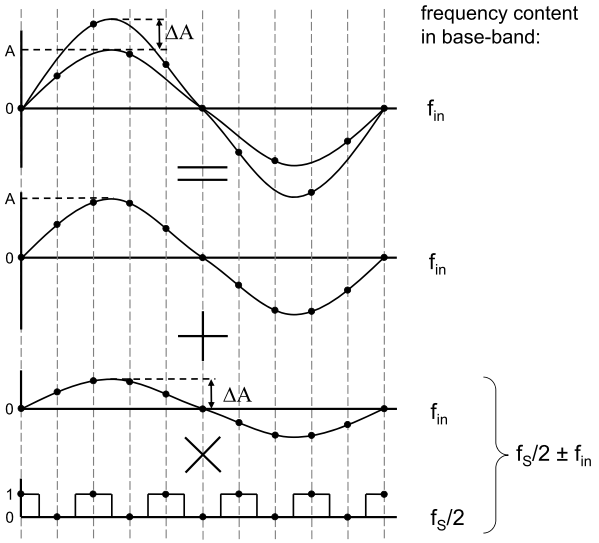

Fig. 2.5 Visualization of frequency content in the case of gain mismatch between 2 channels

frequency $f_S/2$. Due to the multiplication, sum and difference frequencies arise:

$$\sin\left(2\pi\frac{f_S}{2}t\right)\cdot\sin(2\pi f_{IN}t)=\frac{1}{2}\sin\left(2\pi\left(\frac{f_S}{2}+f_{IN}\right)t\right)+\frac{1}{2}\sin\left(2\pi\left(\frac{f_S}{2}-f_{IN}\right)t\right) \tag{2.1}$$

Note that due to aliasing higher-order products of the square-wave[1] result in the same frequencies as the fundamental of the square-wave ($f_S/2$). In summary, gain differences between channels result in scaled copies of the input spectrum around $f_S/2$. And again for more channels, more scaled copies of the input spectrum appear.

Finally, timing-misalignment between channels is considered. The signals are shown in Fig. 2.6 and are similar to the case of gain mismatch, except that the phase of the difference signal is shifted by 90°. The resulting spectrum has the same frequency content. It is easy to understand that phase-differences between channels have the same effects as timing misalignment, since for a sinusoid holds: $\Delta\varphi = 2\pi f_{IN}\Delta t$.

Apart from those channel mismatches, other differences between channels e.g. differences in linearity or bandwidth will also degrade the performance. Bandwidth mismatch is discussed in the next section. The difference signal caused by linearity mismatch depends on the input signal, like in the case of gain mismatch, and it has therefore the same effect on the output spectrum.

[1] In this example, only 2 samples per period are of importance, so the square-wave can also be replaced by a sine-wave.

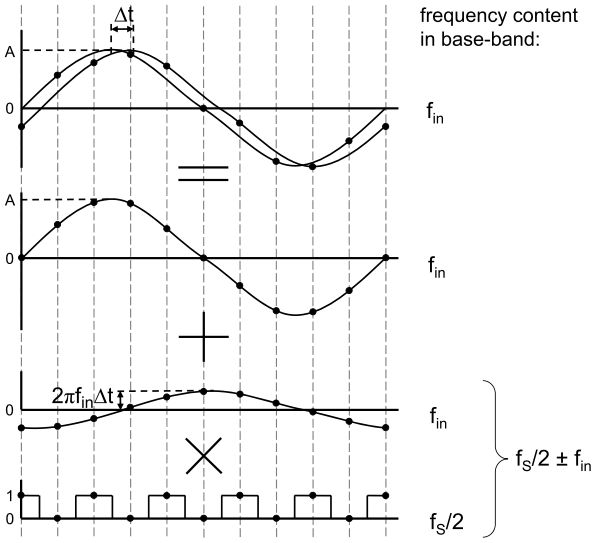

Fig. 2.6 Visualization of frequency content in the case of timing mismatch between 2 channels

2.2.2 Bandwidth Mismatch

As discussed above, the performance of a time-interleaved ADC is affected by mismatch in offset and gain, and this is not dependent on the signal frequency. For bandwidth mismatch however, the performance degradation is dependent on the signal frequency. In this section, phenomena related to bandwidth mismatch are described and quantified.

Two different channel bandwidth limitations can be distinguished: (1) limitations common to all channels caused by the input resistance and interconnect capacitance and (2) per channel limitations caused by the sample capacitor and the resistance of the sample-switch and that of interconnect.

To get some feeling for the quantitative effects of bandwidth mismatch, a numerical example is given. This requires a few assumptions, which are taken from an implementation presented in Chap. 4. For an overview see Fig. 2.1. In this implementation, the SNR due to kT/C noise [50] needs to be about 10 bits, so for a differential system, a sample-capacitor of 150 fF is sufficient with a signal swing of 0.4 V. With a bandwidth requirement of 1 GHz, the switch resistance should be lower than 1 kΩ. In a 0.13 µm process, a switch with a geometry of 1 µm/0.13 µm has a $\sigma(V_{GS})$ of 13.5 mV in which both V_T spread and spread in β (gain factor) contribute for about the same amount. Under typical bias conditions this leads to a $\sigma(R_{ON})/R_{ON}$ of 3.5%, so $\sigma(R_{ON})$ is 35 Ω.

Mismatch parameters about the interconnect resistance could not be found in literature. Devices like MOSTs ($I_{D,sat}$), active resistors [48] and back-end capacitors have more or less a standard deviation of around 1 %µm and this can also be assumed for interconnect resistance as long as its dimensions are above the minimum feature-size [58]. The interconnect in the example implementation is 170 µm long

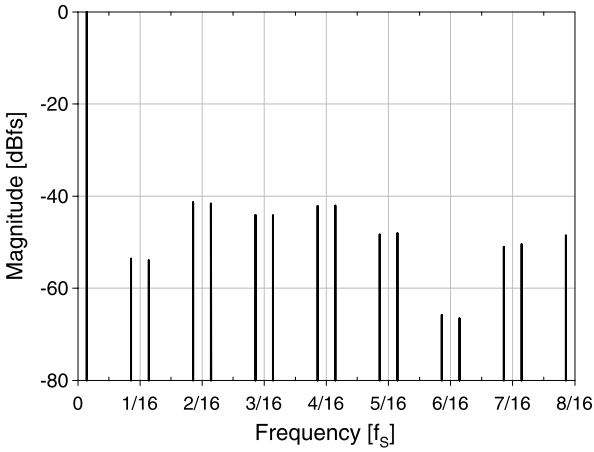

Fig. 2.7 Typical simulated spectrum of a reconstructed sinusoid in the presence of bandwidth mismatch, with $N = 16$ and $\sigma(BW)/BW = 3.5\%$

and 0.4 μm wide (two times minimum width), and has a nominal resistance of 26 Ω. The standard deviation of the resistance is approximately:

$$\sigma\left(R_{\text{intc}}\right) = \frac{R_{\text{intc}} \cdot 1\%}{\sqrt{170 \cdot 0.4}} = 0.12\% \cdot 26\ \Omega = 0.03\ \Omega \tag{2.2}$$

This is much smaller that the standard deviation of the switch resistance, and therefore the variation in interconnect resistance can be neglected.

Capacitor matching in modern CMOS processes is relatively good. The process technology used for the implementation, has capacitors with $\sigma(C_{\text{sample}})/C_{\text{sample}} = 0.03\%$. Taking the three relative standard deviations together leads to a $\sigma(BW)/BW$ of 3.5%, dominated by mismatch in the sample-switch resistance.

To demonstrate the effects of bandwidth mismatch on the spectrum, a 16-channel time-interleaved T&H is simulated in which bandwidth mismatch is the only error and the rest is assumed ideal. In this example $\sigma(BW)/BW = 3.5\%$ and the sample-rate is 1 GHz. The nominal channel bandwidth is also 1 GHz and the signal frequency is close to the sample-rate, such that its alias appears close to zero and the spurious tones appear near multiples of $f_S/16$, resulting in an orderly spectrum. The reconstructed spectrum is shown in Fig. 2.7 and has an SNDR of only 33 dB (or[2] 5.1 bits), which is much less than required for a 10 bits converter. When the input frequency is reduced to the Nyquist frequency, the SNDR slightly improves to 37 dB (5.8 bits).

In Fig. 2.8 the maximum achievable SNDR is shown as a function of $\sigma(BW)/BW$ for a 16-channel time-interleaved T&H. The input frequency is equal to half the nominal channel bandwidth f_0, while the SNDR is independent of the sample-rate. For an SNDR of 10 bits and input frequencies up to half the nominal channel bandwidth, $\sigma(BW)/BW$ should be smaller than 0.2%. To improve the matching of R_{ON} from 3.5% to 0.2% by device scaling [38] (increasing both width and length), the

[2]The relation between the scales in dB and bits is: $\text{dB} \approx 6.02n + 1.76$.

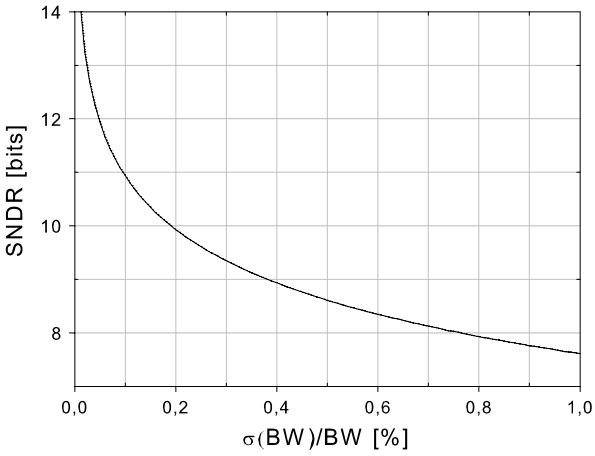

Fig. 2.8 Achievable SNDR as a function of $\sigma(BW)/BW$ with $N = 16$ and $f_{IN} = {}^1/_2 f_0$ (average SNDR of Monte-Carlo simulation)

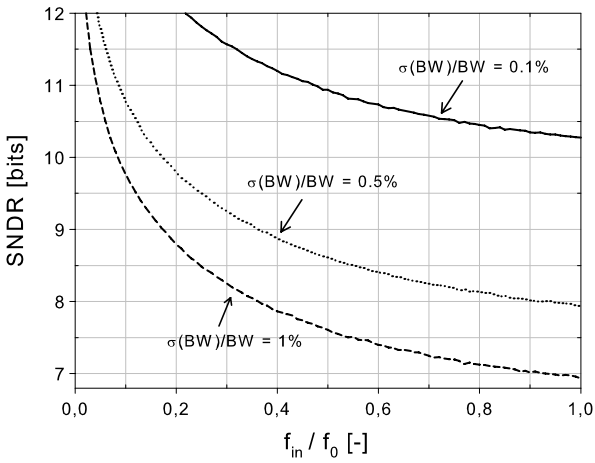

Fig. 2.9 SNDR as a function of the normalized signal frequency for different values of $\sigma(\Delta BW/BW)$

area should be increased $17.5^2 = 306$ times, which would lead to an unacceptably large switch. Scaling of the width only is discussed in the next section.

Performance Improvement by Increasing the Nominal Channel Bandwidth

The amplitude of the spurious tones depends on the ratio of the signal frequency and the nominal channel bandwidth. For a relatively large nominal channel bandwidth, bandwidth mismatch between channels has little impact. To demonstrate this effect, the achievable SNDR as a function of the signal frequency normalized to the nominal channel bandwidth f_0 is shown in Fig. 2.9 for different values of $\sigma(BW)/BW$. The system is equal to the one described above and the normalized frequency is the signal frequency divided by the nominal channel bandwidth.

From the above it becomes clear that instead of improving bandwidth matching, it is also possible to increase the channel bandwidth, such that for the frequencies of interest the gain matching and phase alignment is better. When increasing the width of the sample-switch, both the bandwidth and the matching of the bandwidth improve.[3] For example, if a switch of 10/0.13 is used, $\sigma(V_T) = 4$ mV, $\sigma(\beta) = 0.9\%$, $\sigma(R_{ON}) = 1.1\%$ and the bandwidth is about 10 GHz. The achievable SNDR for input signals at 10 GHz is then only about 44 dB or 7 ENOB. However, for signal frequencies up to 1 GHz, an SNDR of 10 bits is achievable, without the need for bandwidth calibration.

Bandwidth Mismatch Split into Resulting Gain and Phase Mismatch

Bandwidth mismatch between channels causes frequency dependent differences in both gain and phase [10]. It is useful to distinguish between these two effects, and therefore a simulation result of a 16-channel time-interleaved T&H is shown in Fig. 2.10, when taking into account: (a) only gain errors due to bandwidth mismatch, (b) only phase errors due to bandwidth mismatch and (c) both errors. On the horizontal axis the normalized signal frequency f_{IN}/f_0 is shown, where f_0 is the nominal channel bandwidth. $\sigma(BW)/BW$ is 1%. For signal frequencies close to the nominal channel bandwidth, the errors caused by (bandwidth mismatch induced) gain mismatch and (bandwidth mismatch induced) phase mismatch degrade SNDR by the same amount. Towards lower frequencies the effect of gain mismatch decreases rapidly (increasing SNDR), while the effect of phase errors only decreases slowly. So, in conclusion, phase errors are dominant for relatively low input frequencies.

2.3 Time-interleaved Track and Hold Architectures

In this section two time-interleaved T&H architectures are discussed: the normal time-interleaved architecture without a frontend sampler and the time-interleaved architecture with a frontend sampler. Optional improvements on both architectures are discussed, and interleaving limits are discussed for both architectures in relation to bandwidth and accuracy. The section ends with a comparison of the architectures.

[3] The amount of channel charge dump mainly depends on the area of the transistor channel. Since sample switches usually have a large aspect ratio (large W, small L) e.g. 10/0.13, a small absolute variation in W has little impact, while the same absolute variation in L has a much larger impact. The relative mismatch in charge dump is therefore quite independent on W, resulting in an absolute mismatch proportional to W. So, increasing the switch width leads to an increase in the mismatch of the charge dump. For a bootstrapped sample-switch this results in increased offset mismatch. As offset calibration is often required for time-interleaved ADCs, this is not considered to be a problem.

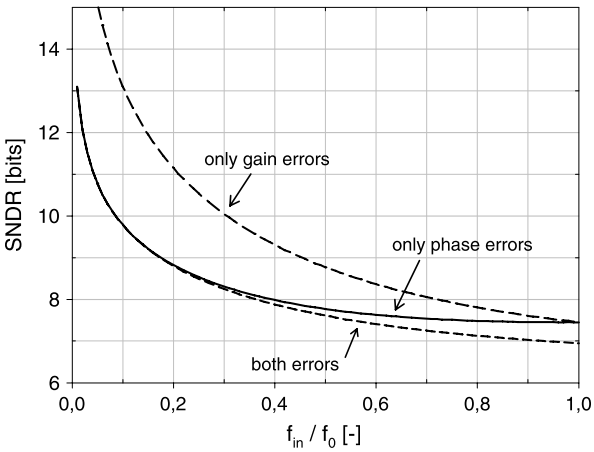

Fig. 2.10 SNDR as a function of the normalized signal frequency with $\sigma(BW)/BW = 1\%$, when taking into account: only gain errors due to bandwidth mismatch, only phase errors due to bandwidth mismatch, both errors

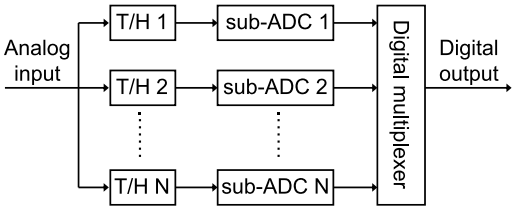

Fig. 2.11 The time-interleaved ADC architecture

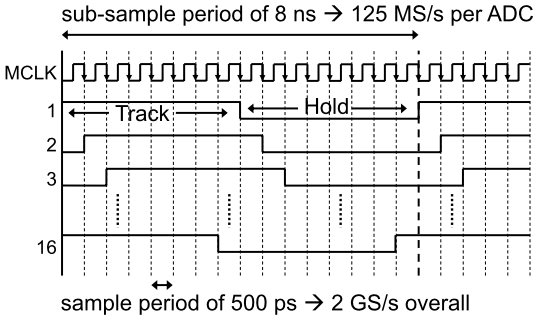

Fig. 2.12 Timing diagram of time-interleaved ADC with $N = 16$

2.3.1 Architecture Without a Frontend Sampler

The most straightforward configuration of a time-interleaved T&H is shown in Fig. 2.11, where each sub-ADC has its own T&H circuit [8, 39]. The corresponding timing diagram is shown in Fig. 2.12. At each falling edge of the master-clock, one of the T&Hs goes from track-mode to hold-mode and takes a sample of the input-signal. This signal is then converted to the digital domain by the ADC in the same channel.

An important consideration is the input capacitance of the time-interleaved T&H. For high speed input signals often transmission lines with on-chip 50 Ω termination

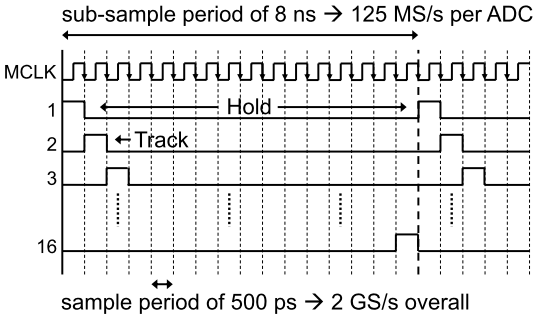

Fig. 2.13 Timing diagram with track-time of 1 period and $N = 16$

are used to mitigate reflections. The resistance at the input node is therefore 25 Ω: 50 Ω of the on-chip termination parallel to 50 Ω of the external source. With this fixed input resistance, the input capacitance determines the bandwidth.

If the resulting bandwidth is not large enough, an input buffer can be used to increase the bandwidth. The input capacitance of the T&H determines the power consumption of this buffer and for a very large capacitive load it can be unfeasible to drive it with sufficient bandwidth. Also, due to the high demands on this buffer (the combination of a high speed and a large capacitive load), it requires a lot of power. In [40] a front-stage buffer in SiGe technology is used that consumes 1 W of power to drive a 4 pF T&H load. The aim of the research described in this book, is to investigate low power solutions. Therefore, no input buffer is used, instead it is assumed that the T&H is driven by an external 50 Ω source.

When the timing diagram of Fig. 2.12 is used, at each moment in time $N/2$ sample-capacitors are connected to the input. The input capacitance can be decreased by reducing the track-time. In Fig. 2.13 the timing diagram is shown for a track-time of one period of the master clock. In this case only one sample-capacitor is connected to the input at a time, lowering the input capacitance and enabling higher number of channels for a given bandwidth.

Now, it is calculated whether one period is sufficiently long to let the voltage on the sample capacitor settle sufficiently close to the input value. This clearly depends on the bandwidth of the sampler, determined by the combination of sample switch and capacitor. For a limited attenuation of the input signal at the Nyquist frequency ($f_S/2$), a sampler bandwidth of two times the Nyquist frequency is assumed, so $BW_{\text{sampler}} = f_S$. The time-constant of the sampler is thus:

$$\tau_{\text{sampler}} = \frac{1}{2\pi f_S} \tag{2.3}$$

There is one period used for settling, so:

$$T_S = \frac{1}{f_S} = (n+1) \cdot \tau_{\text{sampler}} \cdot \ln(2) \tag{2.4}$$

with n the resolution in bits.[4] Combining (2.3) and (2.4) yields:

$$n = \frac{2\pi}{\ln(2)} - 1 \approx 8 \tag{2.5}$$

So, in the case the sampler bandwidth is equal to f_S, settling is accurate up to a resolution of 8 bits.

In the previous section it was argued that a large channel bandwidth is advantageous to improve matching. For a bandwidth larger than f_S, one period of tracking is also enough for resolutions of more than 8 bits.

A short track-time implies a long hold-time, which is advantageous in most ADC architectures, as the ADC has more time to do the conversion.

Resetting of the Sample Capacitor

In the above calculation, it was implicitly assumed that when the T&H entered track-mode, an unknown value of the previous sample action was still present on the sample capacitor. To prevent inter-symbol interference, the T&H needs to settle to the full accuracy, as calculated above.

It is possible to use a reset switch, to remove the previous sampled signal from the sample capacitor, before going into track-mode. Assuming the sample process is linear, incomplete settling only leads to attenuation, and inter-symbol interference does not occur.

To exploit the advantage of a reset switch however, the settling time should be constant. So, not only should the track-to-hold moment be defined well, also the hold-to-track moment should be defined well. This increases the complexity of the T&H circuit significantly. As a result, a reset switch should only by used when the T&H settling-time requirement can not be fulfilled easily.

Input Capacitance

Besides one or more sample capacitors, the wiring and the sample-switches also contribute to the input capacitance. For a (half) circular layout, see Fig. 2.14, the capacitance of the sample-switches is proportional to N, while the wiring capacitance is proportional to N^2: For a fixed channel height, both the number of channels and the wiring length are proportional to N. An approximation for the total input capacitance when using one clock period for tracking is:

$$C_{IN} = C_{sample} + N \cdot C_{switch} + N^2 \cdot C_{wire} \tag{2.6}$$

[4] A derivation is given in Sect. 3.2.2 starting at p. 42. However, a step-size of half the range is assumed there, while here the step-size equals the entire range for an input signal at the Nyquist frequency. Therefore, n needs to be replaced by $n + 1$.

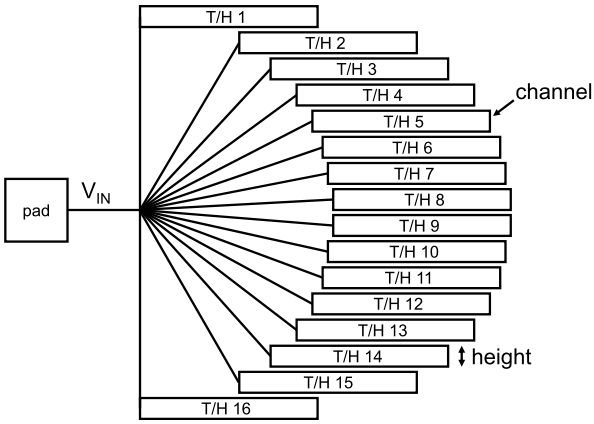

Fig. 2.14 The semi-circular layout for minimizing bandwidth differences

where C_{switch} is the switch capacitance and C_{wire} is the capacitance of a wire of a certain length. The length of the wires depends on the configuration. To give some actual numbers, an example is introduced taken from an actual implementation [24], presented in Chap. 4. In this example a semi-circular layout is used as shown in Fig. 2.14. This layout is chosen in order to make the bandwidth for all channels equal. This could also be accomplished by a full-circle layout, but it has three disadvantages: (1) the distance from the bond-pads to the middle of the circle is longer, resulting in a lower input bandwidth, (2) routing the clock and input signals from the bond-pad to the middle of the circle and shielding these, requires a few metal layers, so less layers are available, and (3) the placement of the ADCs is less practical.

The wire-length depends on the channel height, which has a lower limit for practical reasons. The implementation resulted in T&H blocks of 20 µm by 300 µm and this height is assumed here for determining the wire capacitance.

To get some feeling for the total input capacitance, the values for the capacitances will be approximated and are again taken from the same implementation. C_{sample} has a value of 150 fF, limiting the achievable ENOB to 10.4 bits for a peak-to-peak signal swing of 0.4 V due to kT/C noise. R_{ON} of the switch is 150 Ω, and together with the interconnect resistance of 26 Ω, this results in a bandwidth of 6 GHz. This bandwidth is chosen to be large for reasons explained in Sect. 2.2.2 on p. 9. This value of R_{ON} is reached with an NMOST switch in 0.13 µm CMOS technology for a switch width of 10 µm. The total switch capacitance at the source node is 15 fF, assuming the switch is off, since all but one switches are off. The capacitance of a wire depends on its width and length. When using a minimum width of 0.2 µm, a 3D EM-field simulation shows a capacitance of 0.12 fF per µm length. Assuming the structure and block-height discussed above results in a C_{wire} of 1.3 fF, which can be used in (2.6).

The input capacitances and the input bandwidth can now be derived. In Fig. 2.15 the various capacitances are plotted as function of the number of channels N for: the sample-capacitors, the sample-switches, the wires and the sum of these three. On the right y-axis, the input bandwidth is shown. Above 20 channels, the wiring

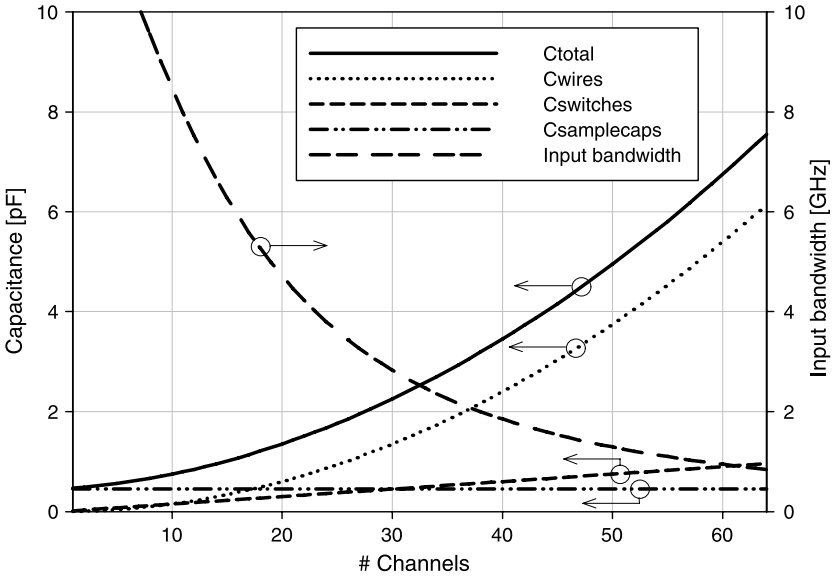

Fig. 2.15 Capacitances and input bandwidth as a function of the number of channels N

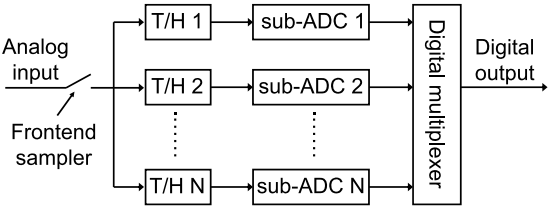

Fig. 2.16 Time-interleaved ADC architecture with frontend sampler

dominates the total capacitance. For an input bandwidth of 2 GHz, the number of channels is limited to about 40. For 6 GHz of bandwidth, N should not exceed 16.

2.3.2 Architecture with a Frontend Sampler

In the architecture discussed in the previous section, each channel has its own T&H. Mismatch between these T&Hs results in timing-misalignment of the sample-moments. Depending on the signal frequency, this degrades the SNDR. In this section an architecture is described that does not have this disadvantage.

To avoid timing-misalignment between channels, a frontend sampler (FRS) [16] can be added to the conventional architecture as shown in Fig. 2.16. The essence of this architecture is that the frontend sampler determines all sampling moments, avoiding timing-misalignment.

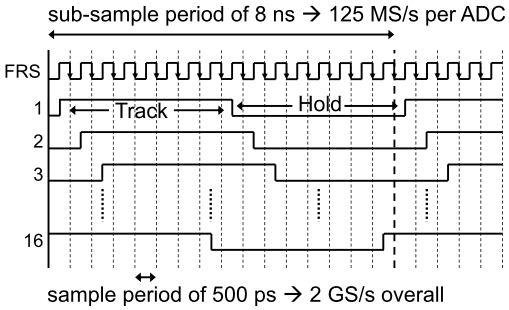

Fig. 2.17 Timing diagram of a time-interleaved ADC with frontend sampler and $N = 16$

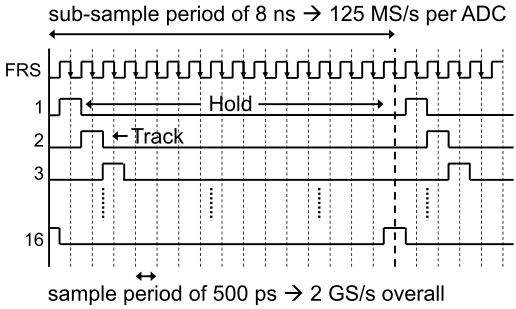

Fig. 2.18 Timing diagram with track-time of 1 period and frontend sampler

In the timing diagram of Fig. 2.17 the timing of the frontend sampler (FRS) is shown and it equals the master clock of the architecture without a frontend sampler. The timing of the T&Hs is also equal to this architecture with the exception the T&Hs are now delayed by half a clock-period of the master clock. This way the frontend sample switch opens first and determines the sample moment. The timing of the T&H sample-switches is therefore not critical and distortion tones at multiples of $f_S/N \pm f_{IN}$ due to timing-misalignment between channels [8] are avoided.

To reduce the input capacitance and to increase the conversion-time available for the ADC, the track-time can be reduced to one clock period, in the same way as in the conventional architecture as described in the previous section. The resulting timing diagram is shown in Fig. 2.18.

Without a frontend sampler the track-time can be made one or more periods. A disadvantage of the architecture with a frontend sampler is that the track-time is limited to about half a clock-cycle of the master clock, because the frontend sampler has to operate at the full sample-rate. The track-time can be slightly increased by using a clock with a duty-cycle larger than 50%, but it can never reach a full clock period, as the sample-switch in the channel has to be opened while the frontend switch is still open. Ensuring that the clocks are non-overlapping at high sample-rates, takes a significant part of the sample-period.

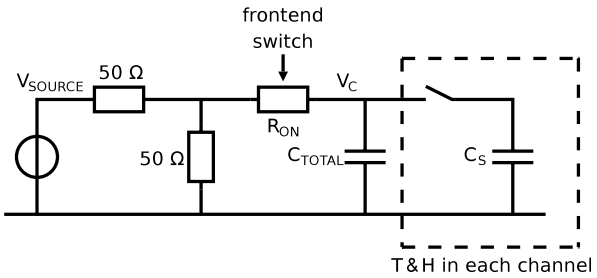

Fig. 2.19 Schematic for calculation of the bandwidth and settling-time requirements

Input Bandwidth and Settling-time Requirements

The input capacitance as calculated in the previous section is also present in this architecture, however not at the input, but instead after the frontend sample-switch.

The requirements for the frontend switch will now be calculated based on bandwidth and settling-time. The bandwidth can be calculated using the schematic of Fig. 2.19. It is assumed that both the impedance of the signal source and the on-chip termination are 50 Ω and the resistance of the frontend switch is called R_{ON}. These resistors and capacitance C_{total} cause a first pole.

Each channel contains a T&H switch and a sample capacitor, indicated by the dashed box. These cause a second pole, which for practical implementation would be far away from the first pole, and therefore it is neglected.

So, assuming a first-order system, the bandwidth at node V_{C} is:

$$BW_{V_{\mathrm{C}}} = \frac{1}{2\pi \cdot R_{\mathrm{eff}} \cdot C_{\mathrm{total}}} \tag{2.7}$$

with $R_{\mathrm{eff}} = 50/2 + R_{\mathrm{ON}}$ and C_{total} the total input capacitance of wires, sample switches and a sample capacitor as calculated in the previous section. Figure 2.20 shows the required switch resistance R_{ON} as a function of the number of channels N for various bandwidths. From this graph it becomes clear that for the example of an input bandwidth of 2 GHz and 16 channels, the switch resistance needs to be 50 Ω or less.

The settling-time requirement is calculated as follows: The signal should settle to the required accuracy within half the sample-period, again assuming a first-order system. For an example sample-rate of 2 GS/s and settling up to an accuracy of $^1/_2$ LSB at 8 bits level, 6.2τ of settling is required in 250 ps, so τ should be smaller than 40 ps. Using $\tau = R_{\mathrm{eff}} \cdot C$, for each N the required R_{ON} can now be calculated. In Fig. 2.21 the required resistance of the frontend switch is plotted as a function of the number of channels for different accuracies.

For e.g. 16 channels and 10 bits of accuracy, the required switch resistance is 8 Ω. For an NMOST in 0.13 µm technology with $V_{\mathrm{GS}} = 0.6$ V and $R_{\mathrm{ON}} = 8$ Ω, the required switch width is 160 µm. Such a large sample switch has a large parasitic capacitance of a few hundred fF, which has two consequences: (1) It is hard to drive

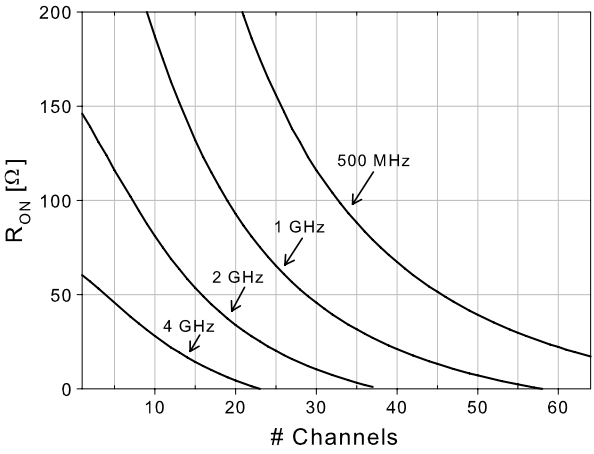

Fig. 2.20 Required frontend sample-switch resistance as a function of the number of channels N for various bandwidths

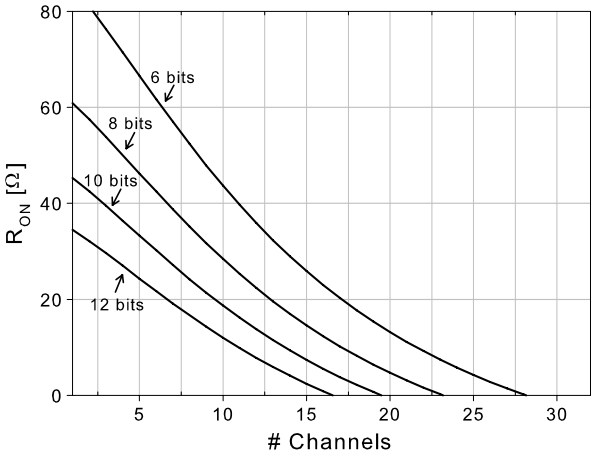

Fig. 2.21 Required frontend sample-switch resistance as a function of the number of channels N for various accuracies

the gate-node with a steep edge to make the sample process close to ideal, and (2) the sample-to-hold step becomes [50] unacceptably large. Under the above assumptions and an accuracy of 10 bits, the number of channels should be limited to about 5 for the frontend sampler architecture.

Note that when the frontend sampler is omitted, the settling time requirements are significantly relaxed: The series resistance of the switch is not there and the settling time can be twice as long, as explained in the previous section.

Increasing the Input Bandwidth

The main disadvantage of a frontend sampler is the decrease in bandwidth, due to the large capacitance of the wires and switches after the sampler. This capacitance can be decreased by using additional switches. An example is shown in Fig. 2.22,

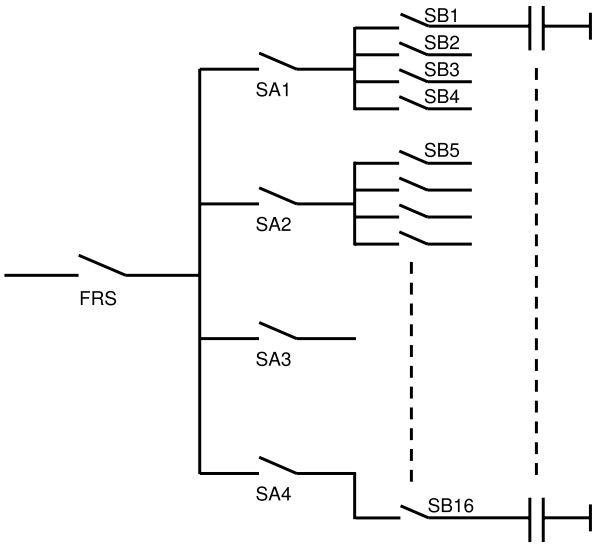

Fig. 2.22 Architecture with a frontend sampler and additional switches to increase the bandwidth

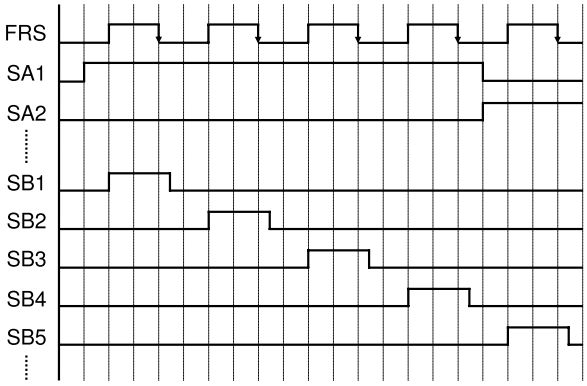

Fig. 2.23 Timing diagram of architecture with frontend sampler and additional switches

where 4 additional switches (SA1–SA4) are placed between the frontend sampler (FRS) and the T&H switches in the channels (SB1–SB16). The additional switches will also increase the resistance, however the net effect on the bandwidth can still be positive.

The corresponding timing diagram is shown in Fig. 2.23 and the operation is as follows: Suppose switches FRS, SA1 and SB1 are conducting, such that the first channel is in track mode. Then, FRS opens first and determines the sample moment. Next, SB1 opens and fixes the charge on the sample capacitor. After this, FRS and SB2 close and the second channel is in track-mode. After a sample period, FRS opens again followed by SB2 and so on.

When SB4 is opened (after FRS is opened), also SA1 is opened and SA2 is closed, such that the next four channels can take samples of the input signal.

The A-switches can be opened, after the B-switches are opened. Since the charge on the sample-capacitor is then already fixed, charge injection of the A-switches does not degrade the performance.

The advantage of this architecture is that timing misalignment is avoided, and the bandwidth is larger than without using additional switches. A disadvantage of this architecture is that the (in this example) four quarters of the circuit will have bandwidth mismatch due to spread in the A switches (w.r.t. R_{ON} and $C_{parasitic}$) and the capacitance of the wire and the B switches. This can limit the performance or require bandwidth calibration.

2.3.3 Conclusions on Architectures

In conclusion it can be said that the use of a frontend sampler has the advantage of good alignment of sampling moments between different channels, but that the product of bandwidth and accuracy is limited. To achieve moderate accuracy (8–10 bits) together with a high bandwidth (>1 GHz), the use of a frontend sampler is not viable under the assumptions made. The use of additional switches between the FRS and the T&H switches increases the bandwidth and accuracy, but can cause bandwidth differences between the groups of T&Hs sharing an intermediate switch.

2.4 Track and Hold Buffers

This section handles buffers for the use in a time-interleaved T&H. In a time-interleaved ADC multiple sub-ADCs operate in parallel, resulting in an N times higher sample-rate, with N the number of channels. If the ratio of the maximum input frequency and the sample-rate (e.g. $^1/_2$ for Nyquist operation) is kept constant, the T&Hs need to operate with N times higher input frequencies than if used non-interleaved.

To achieve good linearity, closed loop configurations using feedback are commonly used in T&Hs for medium signal frequencies [2, 37, 64]. These configurations are however not suitable for high-frequency input signals: the gain-bandwidth product is limited, so for high frequencies the gain is limited and the feedback mechanism for correction of imperfections is less effective, resulting in reduced linearity at higher input frequencies. For signal frequencies in the gigahertz range, closed loop configurations are not considered feasible in the target process technology.

Open loop configurations offer a higher bandwidth at the cost of accuracy and linearity. A bandwidth of e.g. 1 GHz is easily achievable with a configuration with a source-follower buffer,[5] see Fig. 2.24. This configuration suffers from two problems: (1) distortion introduced by the sample process, for which solutions are presented in Sect. 4.3.1 and (2) distortion caused by the buffer, which is discussed in this section.

[5]This buffer does use feedback, but it is only local.

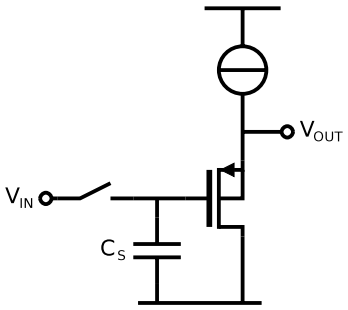

Fig. 2.24 High-speed, open-loop T&H configuration

The problem of buffer distortion is tackled in three steps. First, even-order distortion is discussed, second the dominant distortion mechanism of a conventional buffer is solved, and third the effect of a capacitive load is treated.

2.4.1 Even-order Distortion

Depending on the blocks in front of the ADC, a (quasi) differential implementation of the T&H can reduce even-order harmonics by a large amount. The actual reduction depends on the matching of the halves of the circuit. In general, matching of up to about 1% is realistic, resulting in a decrease of even-order distortion products with 40 dB.

The non-suppressed odd-order distortion products will now dominate the total distortion.

2.4.2 Buffer Distortion

Higher-order harmonics and most inter-modulation products can be canceled in the same way as second-order distortion, although additional phase-shifted versions of the input signal are required [29]. For example, by using three input signals, with phase-shifts of 120 degrees in between, all but the 4^{th}, 7^{th}, 10^{th}, etc. order harmonics are canceled.[6] It is however hard to generate signals with a constant phase-shift over a wide bandwidth, and therefore it is not considered to be a feasible option for a broadband ADC.

Therefore, other solutions are considered. Consider the source follower buffer of Fig. 2.24. The bulk of the PMOST is tied to its source, to mitigate the non-linear body effect. Assuming an ideal current source, the small signal transfer function is

[6]Increasing the number of phase shifted input signals does not only remove harmonics. For example, when going from 2 to 3 signals, some even-order harmonics appear again.

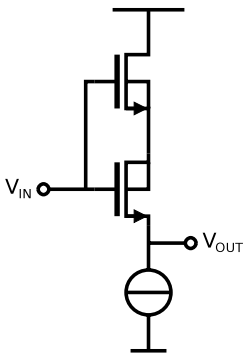

Fig. 2.25 The cascode source follower

given by:

$$V_{\mathrm{OUT}} = V_{\mathrm{IN}} \frac{1}{1 + \frac{1}{g_{\mathrm{m}}(V_{\mathrm{IN}}) \cdot r_{\mathrm{out}}(V_{\mathrm{IN}})}} \tag{2.8}$$

with g_{m} the transconductance of the transistor and r_{out} the output resistance of the transistor. Both g_{m} and r_{out} are functions of the drain-source voltage V_{DS} due to channel length modulation, and as V_{DS} depends on the input voltage, they are functions of the input voltage. So, when the input voltage varies, the transfer function varies, and the output signal becomes distorted.

In modern sub-micron CMOS processes, the non-linearity of the output resistance is the dominant source of distortion in the configuration of Fig. 2.24. To get a high bandwidth, the length of the transistor must be small, so the absolute value of the output resistance is small. If the intrinsic gain ($g_{\mathrm{m}} \cdot r_{\mathrm{out}}$) is small and nonlinear, the output signal is significantly distorted.

The best way to increase the linearity is to decrease the variation of the drain-source voltage. An example of a circuit where this is implemented is the cascode switch source follower [17, 18], shown in Fig. 2.25. A disadvantage of this implementation is the increased input capacitance. Moreover, the upper transistor needs to have a much smaller threshold voltage than the lower transistor to keep the lower transistor in saturation. This can be accomplished by scaling the transistors, which can be disadvantageous for other circuit aspects like speed or it can be accomplished by using a process option such as the low-V_{T} option [18], which requires additional process steps.

The schematic of a new unity-gain buffer is shown in Fig. 2.26 [26]. It is in fact a P-type source-follower (SF), with an additional N-type SF aiming to keep the drain-source voltage of the PMOS transistor constant.

The second SF decreases the variation in V_{DS} of the PMOST, such that the effective output resistance of the PMOST is increased and that the gain and linearity of the buffer are increased. This is explained in the next paragraphs.

The second SF transistor needs to have a short channel length to achieve a large SF bandwidth, and its bulk is connected to ground, since this is required by most standard CMOS processes. Due to the small output resistance and the body-effect,

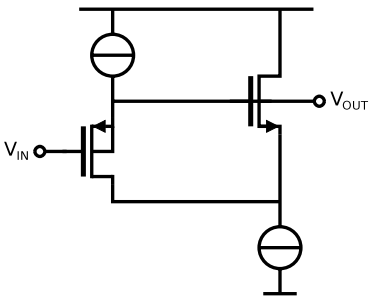

Fig. 2.26 The schematic of a new unity gain buffer

the voltage gain of the 2nd SF buffer is only around 0.9. The signal swing over the source-drain of the first SF transistor is therefore only $0.1V_{OUT}$ instead of V_{OUT}. Consequently, there will flow 10 times less current in the output resistance, and its effective resistance is increased by the same factor. The g_m of the first SF transistor is unchanged, so the intrinsic gain ($g_m \cdot r_{out}$) is increased by a factor of 10 as well, and the voltage gain of the buffer will be closer to 1.

For the linearity the following holds: Suppose the output resistance is described by the following equation:

$$r_{out} = a + bV_{DS} + cV_{DS}^2 + dV_{DS}^3$$

Compared to a conventional SF, V_{DS} is 10 times less (−20 dB), the second-order distortion component cV_{DS}^2 is reduced by 40 dB (100 times) and the third-order distortion component dV_{DS}^3 is reduced by 60 dB (1000 times).

Note that this only holds for the linearity of the output resistance and does not imply that the distortion of the complete buffer is reduced by these amounts. Other distortion components (such as limited output resistance of the current sources) will now dominate the distortion.

Input Capacitance

It is important that the input capacitance of the T&H buffer is low and linear, to avoid distortion at the input of the buffer for high-frequency input signals. The new buffer has less non-linear input capacitance than a conventional or the cascoded source-follower, as described in the following: In the conventional source-follower, the gate-source capacitance is effectively lowered thanks to the Miller effect:

$$C_{eff} = (1 - A_V) \cdot C_{real}$$

with C_{eff} the effective capacitance when looking into the gate, A_V the voltage gain between the gate and the source and C_{real} the real gate-source capacitance. For a source-follower, the gain A_V is close to 1 and the effective capacitance is only a small fraction of the real capacitance. This is true for both the gate-source and the gate-bulk capacitance, assuming the bulk is connected to the source. What remains

is the gate-drain capacitance and this is the dominant input capacitance for both the conventional and the cascoded source-follower.

In the new unity-gain buffer, the drain terminal of the input transistor also tracks the input signal. The gate-drain capacitance is therefore mitigated as well, resulting in a very small input capacitance.

2.4.3 Distortion at High Frequencies with a Capacitive Load

If a buffer, implemented as a switch source follower (or similar) is loaded with a capacitance (e.g. an ADC), the current through the input transistor of the buffer varies when the capacitance is charged or discharged, see Fig. 2.28 with switch S2 closed. If the bias current is not constant, the gate-source voltage V_{GS} of the input transistor is not constant and the output will be distorted.

Before going into detail, the difference between non-interleaved and time-interleaved T&Hs is described first, as this has impact on the requirements for settling-time and bandwidth. For a non-interleaved (NI) T&H and a buffer with first-order settling behavior, the bandwidth requirement for the buffer with respect to settling is[7]:

$$BW_{\mathrm{NI,settle}} > \frac{(n+1)\cdot \ln(2)\cdot 2\cdot f_S}{2\pi} \tag{2.9}$$

with n the resolution in bits, f_S the sample-rate and assuming half the sample-period for settling. This equation is derived in the next chapter. For the example of $n = 10$, the resulting bandwidth requirement yields: $BW_{\mathrm{NI,settle}} > 4.9\, f_{\mathrm{Nyquist}}$. An input buffer with this bandwidth even tracks input signals at the Nyquist frequency closely.

For a time-interleaved T&H the bandwidth requirement for settling is relaxed by the interleaving factor (number of channels). The bandwidth requirement for a time-interleaved T&H is:

$$BW_{\mathrm{INT,settle}} > \frac{(n+1)\cdot \ln(2)\cdot 2\cdot f_S}{2\pi\cdot N} \tag{2.10}$$

with N the interleaving factor and again assuming half the sample-period for settling. For the example of $n = 10$ and an interleaving factor of 16, the bandwidth requirement is: $BW_{\mathrm{INT,settle}} > 0.3\, f_{\mathrm{Nyquist}}$.

If a buffer with minimal bandwidth for settling is used to save power, the buffer output no longer tracks input signals at the Nyquist frequency, but a large attenuation and phase-shift is present, and the problem as shown in Fig. 2.27 arises: During tracking, the buffer output V_{BUF} cannot follow the input signal $V_{\mathrm{T\&H}}$ and at the sample moment (t_{SAMPLE}), the output signal V_{BUF} is not yet fully settled. After the sample moment, the buffer output V_{BUF} will slowly settle to its final value. During

[7] Compare (2.3) and see footnote 4 on p. 15.

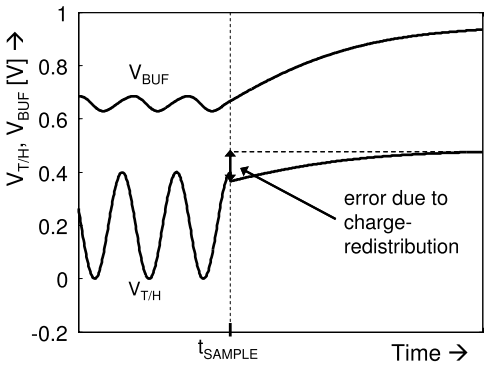

Fig. 2.27 Sampling a high-speed input signal with limited buffer bandwidth

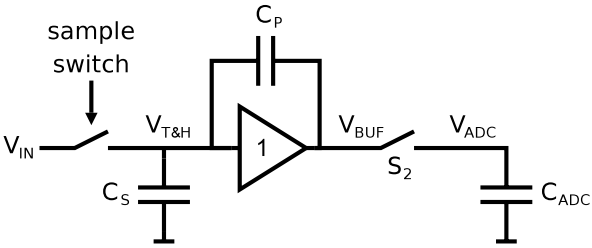

Fig. 2.28 T&H configuration with additional switch S2, which is open during tracking to increase the buffer bandwidth and avoid distortion

this settling, charge-redistribution between (1) the non-linear parasitic capacitance C_P between the input and output of the buffer and (2) the sample capacitor C_S, causes distortion of the voltage on the sample capacitor $V_{T\&H}$ and the buffer output V_{BUF}, as indicated in the figure.

To avoid distortion, the buffer bandwidth could be increased, but this increases its power consumption significantly. Moreover, up-scaling of the buffer is limited, as this also increases the nonlinear input capacitance of the buffer, which requires more drive-power and introduces distortion at the input of the buffer. Up-scaling is therefore always a compromise between the required bandwidth on one side, and linearity, power and available drive on the other side.

To overcome this compromise, switch S2 is introduced between the buffer output and the input capacitance of the ADC as shown in Fig. 2.28 [26]. In track-mode this switch is open and the load capacitance of the buffer is small. Hence the buffer bandwidth is high and output V_{BUF} can now follow the input $V_{T\&H}$ closely, as shown in Fig. 2.29. In this case, the distortion due to charge redistribution is mitigated, without decreasing the linearity or increasing the power consumption.

When the ADC is connected at $t = t_{SWITCH}$, the buffer output will first make a step to the value of the previous sample, still present on the ADC input capacitance. Then the buffer will charge the ADC load to the new sample value. charge redistribution after $t = t_{SWITCH}$ causes a signal dependent step in $V_{T\&H}$, marked by S. This seems to cause distortion, however as V_{BUF} settles to its final value, the process of charge redistribution is reversed and $V_{T\&H}$ returns to its initial, undistorted value. This is thanks to charge conservation at the capacitor plates connected to the input node of the amplifier.

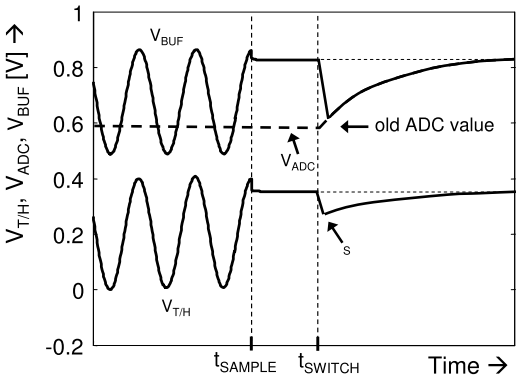

Fig. 2.29 Sampling a high-speed input signal with enhanced buffer bandwidth in track-mode, thanks to disconnected capacitive load

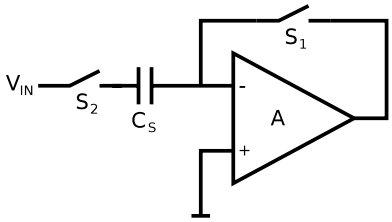

Fig. 2.30 Schematic of a T&H using bottom-plate sampling

In conclusion: in an interleaving architecture the settling time can be relatively long. If the buffer has a large capacitive load, its bandwidth can be reduced to save power. However, this causes distortion. Now, by disconnecting the load during tracking, the distortion is avoided and the buffer bandwidth can remain reduced and power is saved.

2.5 Bottom-plate Sampling in a Time-interleaved ADC

Pipeline ADCs are broadly used and especially the voltage-mode type using an opamp for residue amplification is popular [2, 37, 64]. To make the sample process linear, these converters use bottom-plate sampling. This section considers the use of bottom-plate sampling in a time-interleaved ADC.

Bottom-plate sampling works as follows: The opamp creates a virtual ground node and the input signal is tracked on the sample capacitor as shown in Fig. 2.30 with switches S_1 and S_2 closed. When switching to hold-mode, S_1 is opened first and fixes the charge on capacitor C_S. Ideally this operation does not cause distortion and the operation is independent on the input signal, as the drain and source potentials of switch S_1 are at virtual ground potential. After this, switch S_2 is opened, which does not affect the amount of charge on C_S, since the other side of the capacitor is floating. This is called bottom-plate sampling, because the actual sampling takes place at the bottom-side, the (virtual) ground-side, of the capacitor.

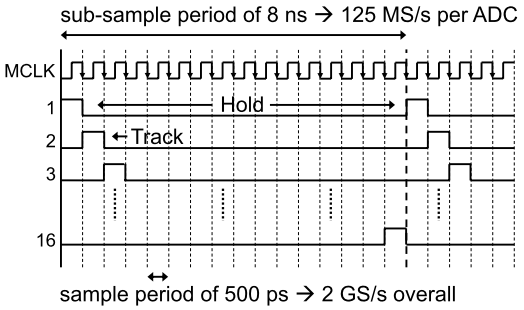

Fig. 2.31 Timing diagram with a track-time of 1 period

The virtual ground is created by the opamp, which has a limited closed-loop bandwidth. When combining a number of such sampling structures in a time-interleaved ADC without a frontend sampler, two problems arise: The first is that it is difficult to reduce the track-time significantly. The opamp requires a certain amount of time to restore the virtual ground potential in case the input signal differs from the previous sample, and most time is required for signals close to the Nyquist frequency. The timing scheme with a track-time of 1 period shown in Fig. 2.31 and described in Sect. 2.3.1 is then not viable. For a longer track-time, more sample-capacitors are connected to the input at a time, limiting the input bandwidth.

The second problem is that the virtual-ground cannot be maintained for high-frequency input signals. Except for over-sampling it is therefore not useful to deploy the described bottom-plate sampling technique in a time-interleaved T&H without a frontend sampler.

The use of bottom-plate sampling in a time-interleaved T&H with a frontend sampler is considered next. When using a frontend sampler, it turns off before the switch in the T&H, see Figs. 2.16 and 2.17. So, the signal conducting switch turns off first, which is contrary to the clocking scheme for bottom-plate sampling as explained above. Bottom-plate sampling can therefore not be used in combination with a frontend sampler.

To use opamp-based pipeline converters in a time-interleaved ADC, alternative sampling techniques have to be used such as in [16], where a separate T&H is used in front of the pipeline converters to perform the actual sampling. Consequently, the advantage of linearity of bottom-plate sampling is lost.

2.6 Number of Channels

In this section, aspects determining the number of channels of a time-interleaved ADC are discussed. The relation between the number of channels and the input bandwidth was already treated: In Sect. 2.3.1 it was argued that for the architecture without a frontend sampler, the input bandwidth depends on the number of channels, and in Sect. 2.3.2 it became clear that the number of channels determines the bandwidth together with R_{ON} of the frontend sample-switch.

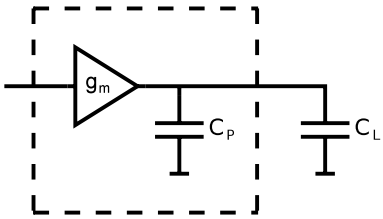

Fig. 2.32 Transconductor with parasitic capacitance C_P and load capacitor C_L

If a frontend sampler is used, the achievable resolution decreases with the number of channels. Without a frontend sampler, this is much less an issue, as the settling time is significantly relaxed.

The input bandwidth can be increased by using a buffer with a low output impedance in front of the T&H instead of driving it with a 50 Ω source. Due to the high demands on this buffer (low impedance, high speed and high linearity) it requires a lot of power[8] [40].

2.6.1 Sub-ADCs

Another important factor for determining the number of channels is the specification of the sub-ADCs. The sample-rate of a sub-ADC is: $f_{S,\text{subADC}} = f_S/N$ with f_S the sample-rate of the time-interleaved ADC and N the number of channels. For a lower number of channels the sample-rate of the sub-ADCs needs to be higher. The sample-rate of a non-interleaved medium resolution ADC (8–12 bits) is practically limited to a few hundred MS/s, as becomes clear from overviews of published ADCs [32, 60]. Besides a maximum sample-rate there is also a trend visible suggesting that beyond a certain sample-rate the power consumption increases more than proportional with the sample-rate. This can be explained as follows [53]: Suppose a transconductor with transconductance g_m and parasitic capacitance C_P, is charging a capacitive load C_L, see Fig. 2.32. The accompanying time-constant is:

$$\tau = \frac{C_P + C_L}{g_m} \tag{2.11}$$

This buffer is used in an ADC and needs to charge the load capacitor within a certain accuracy in a clock period. Suppose $C_P \ll C_L$. In this case the time-constant is relatively large and the sample-rate limited. For a higher sample-rate, the time-constant needs to decrease, which can be accomplished by putting multiple buffers in parallel. The new time-constant is given by:

$$\tau(p) = \frac{p \cdot C_P + C_L}{p \cdot g_m} \tag{2.12}$$

[8] This is in contrast with the buffers after the T&H switch, as these have relaxed speed requirements due to the interleaving [26].

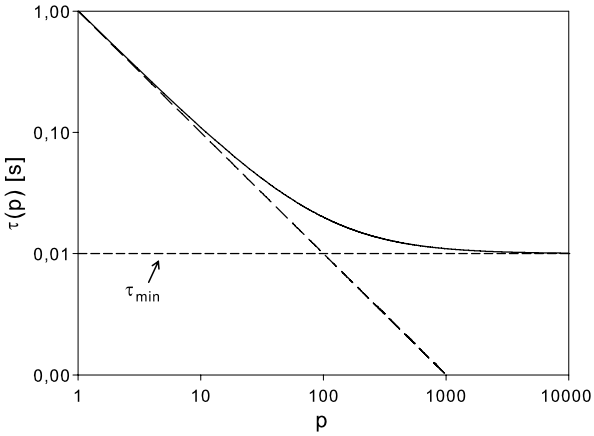

Fig. 2.33 The time-constant τ as a function of the number of parallel buffers p, with $C_P = C_L/100$. Both axes contain only relative numbers and are not related to a physical quantity

with p the number of parallel buffers. For small values of p, the time-constant decreases linearly with p, but when the value of $p \cdot C_P$ approaches C_L, the decrease of the time-constant becomes less than linear and for very large values of p, the time-constant even becomes independent of p. In this case:

$$p \cdot C_P \gg C_L \tag{2.13}$$

and so:

$$\tau = \tau_{min} = \frac{C_P}{g_m} \tag{2.14}$$

This is graphically shown in Fig. 2.33, where C_P is chosen $C_L/100$. The values on both axes are just relative numbers and are not related to a physical number of buffers or transistor sizes.

From this, it can be concluded that starting from a low sample-rate, the power increases proportional with the sample-rate, while for higher sample-rates, the power increases more than linear with the sample-rate. The power efficiency therefore decreases for high sample-rates.

Dependency on Resolution

Another trend is that the sample-rate decreases with increasing resolution. For most architectures this is easy to explain. For example in a pipeline converter: for a higher resolution, the opamps require a longer settling time, so the achievable sample-rate is lower. In a successive approximation (SA) ADC, a higher resolution means more steps per conversion and more settling time per step, both lowering the achievable sample-rate. Finally, in flash architecture, more resolution implicates more com-

parators in parallel and a higher accuracy per comparator. This results in more capacitance[9] and longer settling times, both degrading the achievable sample-rate.

2.6.2 Guidelines

The optimum number of channels is thus a trade-off between T&H architecture, bandwidth, resolution, power consumption and sub-ADC architecture. Moreover, it also depends on the process technology. Most of these trade-offs are hard to quantify and depend on a lot of variables and implementation details. It is therefore not possible to derive the exact optimum number of channels, however some guidelines can be given:

- In literature, ADCs with a medium resolution and good power efficiency can be found with sample-rates up to about 50–150 MS/s. f_S/100 MS/s could therefore serve as a starting point for the number of channels. So, e.g. 20 channels for a 2 GS/s time-interleaved ADC.
- A higher ADC resolution requires longer settling times and more conversion steps, limiting the maximum sub-ADC sample-rate. If the required resolution is relatively low (6–7 bits), the number of channels can be a bit lower, and when the resolution is relatively high (10–12 bits) the number of channels should be increased.
- Newer technologies (e.g. 65/45 nm) offer more speed than older technologies (e.g. 0.18 µm) [57] and allow a higher sample-rate of the sub-ADCs, and therefore the optimum number of channels is slightly lower for newer technologies.
- If a high input bandwidth of the T&H is required the number of channels should be reduced. This will probably increase the power consumption of the sub-ADCs.
- The use of a frontend sampler reduces the input bandwidth considerably. If a high bandwidth is required and timing-alignment can be made sufficiently accurate, do not use it.
- The optimum number of channels is relatively flat, adding or removing a few channels has no major impact.

2.7 Calibration

Mismatch causes errors like offset, and gain and timing differences. Calibration can be used to compensate for these errors. In this section, different kinds of calibrations are discussed together with their implementations.

Calibration of a circuit can be split-up into several parts, see Fig. 2.34. From left to right, the following blocks can be found: a test-signal generation block, the circuit which is calibrated, an error detection block and a correction block.

[9] Depending on the architecture, the capacitance increases between 4 and 8 times more per bit.

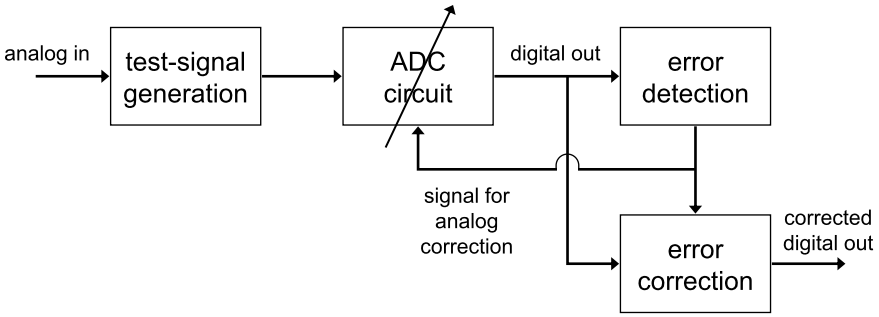

Fig. 2.34 Overview of blocks in a calibration system

The blocks are now discussed, starting with the test-signal generation block. Depending on the calibration, this may have a relatively easy function of shorting the inputs, swapping the input signals or generating a noncritical DC test-signal, or it may have a more sophisticated function, like generating accurate high-speed test-signals.

In order to perform calibration of a circuit, the error should be measured first and usually some signal processing and memory is needed for the calibration, therefore, the result of the measurement should preferably be in a digital format. As a T&H is normally followed by an ADC, this ADC can also serve as measurement device for the T&H. Moreover, the combination of T&H and ADC can be calibrated as one system. Therefore, the circuit to be calibrated is assumed to be a combination of T&H and ADC.

The detection block detects the errors and controls the signal generation block and correction blocks. Several techniques exist for performing a calibration. In foreground calibration, the ADC is not usable during calibration and specific input-signals are applied. This kind of calibration can be performed at start-up, after warm-up or periodically if the application allows for this.

Also, several background calibration techniques exist and the ADC is usable during this kind of (usually continuously running) calibration. Depending on the application, the normal input signal can be used to determine the errors or pseudo random data is added to the input (or the differential input signals can be interchanged) to distinguish between ADC errors and the input signal.

The correction of errors can be done both in the analog and in the digital domain. An advantage of digital correction is that the correction is fully transparent: e.g. an addition of 1 LSB is always exactly an addition of 1 LSB. A disadvantage is that the digital correction can consume a significant amount of power, especially in the case of more complex operations such as needed for bandwidth or timing correction.

Analog correction has the advantage that it can be implemented with little or no additional power consumption, that it is possible to do sub-LSB corrections (e.g. subtract 0.3 LSB from the input signal, or decrease the gain with 1%) without increasing the number of output bits and that the input range is not decreased. Moreover, it can undo errors made in the analog domain rather than correct for the

consequence of errors afterwards, as in digital correction. In the case of e.g. timing misalignment, it is usually not possible to reconstruct the original input from a distorted signal.

2.7.1 Offset Calibration

Mismatch of the sample-switches causes channel-to-channel variation of clock feedthrough and channel-charge injection, which leads to offset between channels. If a buffer is used in a channel it can also cause offset. Reducing buffer offsets by circuit scaling to a small enough level (for example $^1/_2$ LSB) can, even for moderate resolutions, be unfeasible [24] as the input capacitance becomes very large and limits the input bandwidth unacceptably.

Both the detection and the correction of offset errors is relatively easy. In the case of foreground calibration, the differential inputs can simply be shorted to detect the channel offsets and for background calibration the average signal level over a long period can be determined, assuming the input signal is DC free. Correction of the error in both the analog [25] and digital domain is feasible.

2.7.2 Gain Calibration

Gain differences between channels can have the same origin as offset errors. Foreground detection of gain errors is not complicated: after offset correction, a non-critical DC input signal can be applied to determine the channel gain. Analog correction of e.g. the gain of a buffer is feasible and does not have to cost power. An example is [25], where a digital adjustable impedance is placed between the two halves of a pseudo differential buffer. Compared to this, digital correction will always cost some power needed for the digital processing.

For some applications it can be beneficial to perform calibration based on the measured temperature or supply voltage. This is possible if e.g. the gain is dependent on the temperature and this dependency can be determined accurately. Also in this case, the adjustment can be performed in the analog domain.

2.7.3 Timing Calibration

Calibration of timing [40] is complicated. In the case of foreground calibration, high-frequency test signals are required and the detection of timing differences requires sophisticated algorithms. The correction of timing errors is also non-trivial and flexibility in timing tends to increase the power consumption and/or jitter of the clock signal. Background calibration often relies on spectral characteristics of

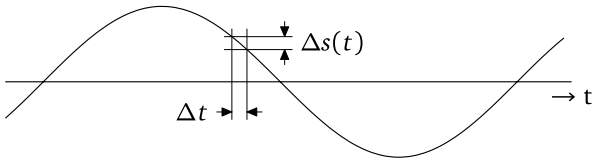

Fig. 2.35 An input signal as a function of time and the effect of sampling jitter

the input signal and usually involves additional hardware [19]. So, although timing calibration is possible, it is better to make the timing alignment accurate by itself, such that calibration can be avoided.

2.7.4 Bandwidth Calibration

Calibration of per-channel bandwidth is not trivial, as the detection of bandwidth differences is not straightforward: it requires high-frequency test-signals and complicated detection algorithms. Adjustment of the bandwidth could be another problem. It is therefore advantageous if bandwidth matching is accurate by design. This can be accomplished by making the channel bandwidth larger than strictly required, so that phase and gain match well within the band of interest. More information can be found in Sect. 2.2.2 on p. 9.

2.8 Jitter Requirement on the Sample-clock

Jitter in the sample-clock means uncertainty in the exact sample moment. In Fig. 2.35, it is graphically shown that a deviation from the ideal sample-moment leads to an error in the sampled signal. For a full-scale sinusoid with frequency f_{IN}, the signal-to-noise ratio is given by:

$$SNR_{\Delta \mathrm{t}} = \frac{1}{\sigma(\Delta t) \cdot 2\pi \cdot f_{\mathrm{IN}}} \tag{2.15}$$

with $\sigma(\Delta t)$ the RMS value of the timing jitter. Figure 2.36 shows this equation graphically, with $\sigma(\Delta t)$ as a parameter. For e.g. an input signal of 1 GHz, and a state-of-the-art timing jitter of 0.5 ps RMS, the resulting SNR is 50 dB or 8 ENOB.

Depending on the application, this definition may be too strict and may lead to over-design. In for example wireline communication systems, channels have increased insertion loss at high signal frequencies [16]; a maximum signal amplitude at the maximum (Nyquist) frequency, does not occur.

In [14] an example is given which compares the variance on the phase-skew between channels when assuming either (1) a sinusoid at the maximum frequency or (2) white noise filtered with an ideal filter with a cutoff frequency equal to the maximum frequency. The requirement on the phase-skew variance is a factor three lower for the filtered white noise, independent of the number of channels. Random

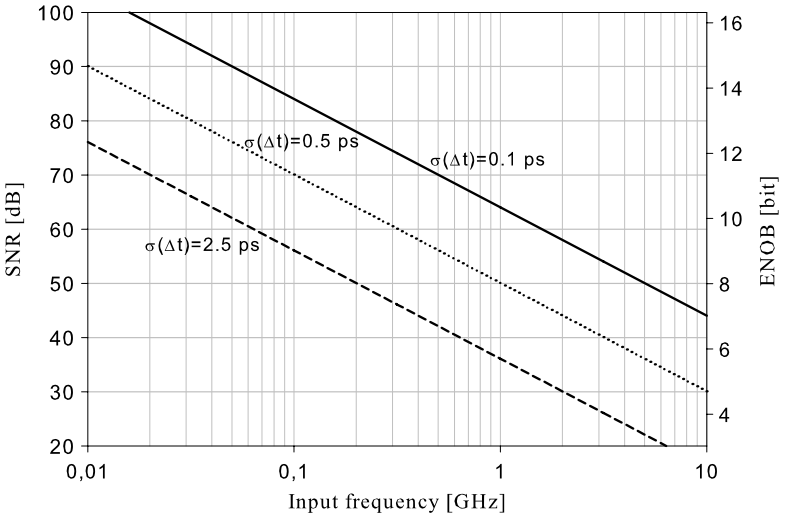

Fig. 2.36 Signal-to-noise ratio and ENOB as a function of the input frequency with $\sigma(\Delta t)$ as parameter

clock jitter can be considered as phase skew for an infinite number of channels, and this conclusion therefore also holds for random clock jitter. So, assuming a sinusoid at the maximum input frequency leads to an overestimate of the required jitter variance.

Also for a software (-defined) radio receiver the above definition can be far too strict. It can be shown that [5]:

$$\mathcal{F}(\Delta s_\tau(k)) \approx \mathcal{F}\big(\Delta t(k\tau)\big) * \mathcal{F}\left(\frac{\partial}{\partial t} s(t)\bigg|_{k\tau}\right) \tag{2.16}$$

where $\mathcal{F}$ denotes the DTFT, $*$ denotes convolution, $\Delta s_\tau(k)$ is the error in the sampled signal and $\Delta t(k\tau)$ is the sampling time error. When the sampling clock is derived from a synthesizer containing a VCO, $\Delta t(t)$ can be assumed to have a f^{-2} power spectrum outside the synthesizer loop bandwidth [41]. Due to its f^{-2} nature, most energy in the error of the sampling clock is at low frequencies. Assuming such a spectrum, (2.16) is illustrated in Fig. 2.37.

Knowing that in the frequency domain this is convoluted with the derivative of the input signal leads to the following [4]:

- The jitter spectrum is convoluted with the input spectrum, and therefore the jitter-induced error is concentrated around the input frequencies.
- Input signals with higher power are surrounded by more jitter-induced error in the output than input signals with lower power.
- Input signals of higher frequencies are surrounded by more jitter-induced error in the output than signals at lower frequencies.

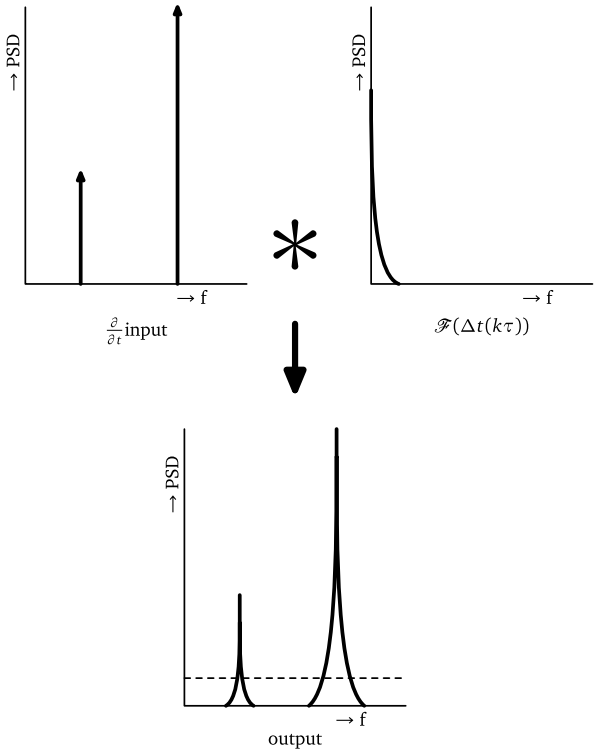

Fig. 2.37 Illustration of (2.16). The convolution of the input spectrum (*upper left*) with the spectrum of $\Delta t(k\tau)$ (*upper right*) gives the output spectrum of the ADC (*bottom*)

So, if a small input signal needs to be received in the presence of a strong out-of-band interferer, the jitter requirement should be based on the small input signal (with possibly also a lower frequency) rather than on the strong interferer. This can relax the jitter specification by two orders of magnitude [4].

2.9 Summary and Conclusions

In this chapter the time-interleaved Track and Hold architecture was discussed. Mismatch between channels, like differences in offset, gain and timing, degrade the performance. In the case of bandwidth mismatch it is shown that it is advantageous to have a large bandwidth, such that for the frequencies of interest the effect of the mismatch is mitigated.

Two T&H architectures were discussed, one with a frontend sampler and one without. The use of a frontend sampler has the advantages of good timing alignment between channels, the resistance of the switch is however a problem: it limits both the input bandwidth and the achievable resolution. The input bandwidth and accuracy can be improved by placing additional switches, which decrease the capacitance after the frontend sampler.

A buffer is introduced which has a large bandwidth and improved linearity compared to a switch source follower. Driving a large capacitive load with a large bandwidth consumes a lot of power. The time-interleaved architecture offers the possibility to decrease the buffer bandwidth, as there is a relatively long time available for settling of the buffer. However, the reduced bandwidth causes distortion. To avoid this, the following solution is introduced: a switch is inserted between the buffer and the capacitive load, such that the buffer bandwidth is increased and the distortion is mitigated. This solution can save a significant amount of power.

Some guidelines are given for determining the number of channels of a time-interleaved ADC. This depends on resolution, bandwidth, technology, and whether a frontend sampler is used or not. For the target specifications and process technology, and considering the current state-of-the-art, a sub-ADC sample-rate of about 100 MS/s can serve as a starting point, as literature shows that power efficient sub-ADCs with this sample-rate are currently feasible.

The topic of calibration is discussed and it turns out that offset and gain calibration is relatively easy to implement, while the implementation of timing and bandwidth calibration is much harder. One should therefore strive for good timing-alignment and bandwidth matching in the frequency band of interest.

Finally, the jitter requirements on the sample-clock are discussed. The traditional requirement assumes a full-scale sinusoid at the maximum frequency. For many applications this requirement is too strict and leads to over-design.

Regarding the desired implementation, the following design choices are derived:

- Use the architecture without a frontend sampler
- The channel bandwidth should be large compared to f_{IN}
- Use the presented T&H buffer
- In track-mode, disconnect the capacitive load from the buffer
- The number of channels should be around 20
- Timing alignment should be made accurate by design
- Use analog calibration to correct for offset and gain errors

Chapter 3
Sub-ADC Architectures for Time-interleaved ADCs

3.1 Introduction

This chapter discusses sub-ADC architectures for time-interleaved ADCs. A sub-ADC is in fact a general purpose ADC, but there are a few differences: a time-interleaved T&H frontend is assumed, so a T&H is not needed as part of the sub-ADC, and the sub-ADC should be able to handle signal frequencies close to the Nyquist frequency well, since for the time-interleaved ADC this corresponds to frequencies at Nyquist/N, with N the number of channels.

The different channels of a time-interleaved ADC need to match each other well. In non-interleaved ADCs for e.g. video applications, the INL is not of great importance, since the human eye is not very sensitive to INL errors. If multiple ADCs with INL errors of a few LSBs would be combined, and they would all have a different INL curve, the resulting spectral tones do become visible in the video image. To avoid this, the INL curves of the different sub-ADCs need to match, which is difficult to realize.

The presence of multiple sub-ADCs on a chip also has advantages. For example, it enables sharing of blocks between sub-ADCs, like reference voltages, DACs and clocking.

In literature many ADC architectures can be found: flash, folding, pipeline, two-step, successive approximation, algorithmic, slope converters, etc. This book is about time-interleaved ADCs with an overall sample-rate of 1–2 GS/s and a medium resolution (8–10 bits). In this discussion about sub-ADC architectures, this is taken as a boundary condition.

For the target specifications, the flash architecture is not a suitable option: It is a parallel architecture and therefore its input capacitance increases exponentially with the number of bits. Depending on the application of techniques like interpolating and averaging, each additional bit increases the input capacitance with a factor between 4 and 8. Above 6 bits this leads to poor power efficiencies and large capacitances which are impractical to drive.

The folding architecture has a reduced number of comparators and can achieve higher resolutions than the flash architecture. The high number of parallel operating

S.M. Louwsma et al., *Time-interleaved Analog-to-Digital Converters,*
Analog Circuits and Signal Processing,
DOI 10.1007/978-90-481-9716-3_3,

folding amplifiers still require quite some power and even in a recent state-of-the-art paper [47] the power consumption is relatively high (2.4 pJ/conversion step). Therefore, this architecture is not investigated further.

Counting and slope converters are relatively slow and would require a huge number of channels to achieve the desired sample-rate. It is therefore not a suitable architecture for this purpose.

The Successive Approximation ADC (SA-ADC) architecture is not fast, it can however have a very good power efficiency. In [54] an SA-ADC is demonstrated which achieves a FoM of 4.4 fJ/conversion-step. Section 3.2 describes the SA-ADC in detail.

Pipeline, two-step and algorithmic converters are more or less based on the same principle: subtraction and residue amplification. As these converters are quite similar [53], only the pipeline converter is considered, since it is popular in both industry and topic of many research papers. In Sect. 3.3 a comparison between pipeline and SA-ADC converters is made regarding power efficiency.

3.2 The Successive Approximation ADC

This section describes the Successive Approximation Analog-to-Digital converter (SA-ADC). This type of converter can have a high power efficiency, because for an n bits converter, only n comparator decisions are needed, and high-gain amplifiers—used in traditional pipeline and two-step architectures—are not required. The absence of the need for high-gain amplifiers makes the architecture suitable for nanometer scale technologies. The sample-rate is however limited: for an n bits converter, n iterations are required.

In Sect. 3.2.1 the operation of a standard SA-ADC is described. The section following that shows that the DAC settling time is the major limitation for the maximum sample-rate, and solutions to reduce the total DAC settling time are presented. In Sect. 3.2.3 the optimum number of steps is derived for a specific architecture, and the section following that describes the look-ahead logic, which can be used to reduce the delay of the digital logic and thus increase the maximum sample-rate. This section ends with a discussion of the comparator circuit.

3.2.1 Standard SA-ADC

A standard SA-ADC [6, 28, 44] consists of three components: a comparator, a Digital-to-Analog Converter (DAC) and a digital control block. These components are configured as shown in Fig. 3.1. A frontend T&H is assumed, so the input signal V_{IN} has a fixed value within one sample-period. The converter finds the value of the input signal by doing a so-called binary search. Firstly, the digital control block switches on the MSB of the DAC, so it settles to the middle of the range. Secondly, the comparator compares the input signal with the DAC signal. If the input signal

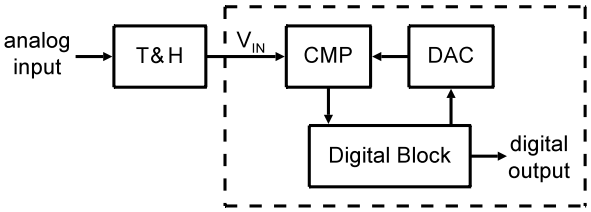

Fig. 3.1 Block diagram of an SA-ADC

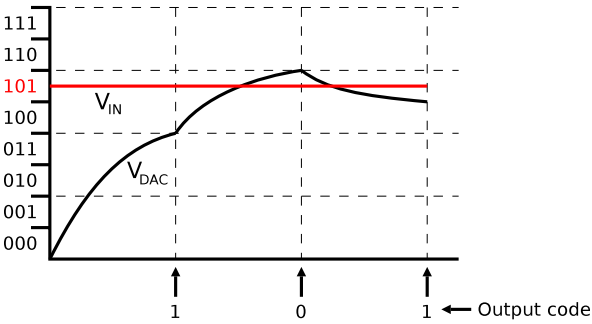

Fig. 3.2 The conversion process of an SA-ADC, which is in fact a binary search. The numbers at the *bottom* represent the comparator decisions and equals the binary output code

is larger than the DAC value, the MSB stays on, otherwise the MSB is turned off. Next, the MSB-1 bit of the DAC is switched on, the comparator makes a decision and the logic keeps the MSB-1 bit on or turns it off. After repeating this n times, the DAC has approximated the input signal within 1 LSB and the binary code is known. Figure 3.2 shows this conversion process graphically for $n = 3$.

The digital control can be implemented using a register, which holds the output code when the conversion is finished. Therefore, this kind of converter is also referred to as a Successive Approximation Register (SAR). In more advanced implementations discussed later, the logic contains more than just a register and the term SAR is not appropriate. Therefore, the term SA-ADC is preferred, and refers to the conversion mechanism rather then to the implementation of the digital control.

As stated above, an SA-ADC needs n steps to reach an accuracy of n bits. In each step, there are three operations: the settling of the DAC, the comparator making a decision and the control logic determining the next DAC level. All three operations require some time for completion and this limits the maximum sample-rate of the converter. A technique to reduce the delay caused by the DAC settling is discussed next, and a technique to reduce the delay of the logic is described in Sect. 3.2.4.

3.2.2 Architectures to Reduce the DAC Settling Time

In this section, different architectures will be described regarding the settling time of the DAC.

Conventional SA-ADC Architecture

For a conventional converter with an accuracy of e.g. ½ LSB, in each step the DAC should settle to within ½ LSB of its final value as well. For an RC limited DAC, the worst-case settling is in the first step[1] when the DAC settles from the minimum or maximum value to the middle of the range. Without loss of generality, a rising DAC signal is assumed here. This signal should be larger than half-range minus the required accuracy of e.g. ½ LSB:

$$V_{\mathrm{DAC}}(t = t_{\mathrm{settle}}) = \frac{1}{2} V_{\mathrm{FS}} \left(1 - e^{t_{\mathrm{settle}}/\tau}\right) \geq V_{\mathrm{FS}} \left(\frac{1}{2} - \frac{1}{2^{n+1}}\right) \tag{3.1}$$

with V_{FS} the full-scale voltage, τ the time-constant of the DAC and n the resolution of the converter. Working out this equation yields the minimum required settling time:

$$t_{\mathrm{settle}} > n \cdot \ln(2) \cdot \tau \tag{3.2}$$

For the example of a 6 bits converter, 4.2τ of DAC settling is required before the comparator can make a decision in the first step. Conventionally, the settling time in each step is set to this worst case step. For the example of a 6 bits converter the sum of all settling times in a conversion is $6 \cdot 6 \cdot \ln(2) \cdot \tau \approx 25\tau$. One way to decrease the total settling time is described next.

Variable Settling Times

In the second step of the conversion process, the step-size is half that of the first (see Fig. 3.2), which has the same effect on the settling time as a reducing the resolution by 1 bit. The required settling time is thus 3.5τ and in the third step it is 2.8τ, and so on. So, for each consecutive step the settling time could be made shorter. For a 6 bits converter, the total settling time could be made as short as:

$$(6 + 5 + 4 + 3 + 2 + 1) \cdot \ln(2) \cdot \tau \approx 14.6\tau \tag{3.3}$$

Compared to the above described system with a constant settling time determined by the worst-case first step, this saves 42% of the total settling time. The implementation of the required clocking scheme is however not trivial.

A clocking scheme that is easier to implement, uses only two different delays for DAC settling. Delay A is used in the first steps and the delay B is used in the remaining steps. For the 6 bits example, an overview of the required settling times for the different configurations is shown in Table 3.1. The 2nd and 3rd columns show the configurations already discussed and the 6th column (3–3) shows the shortest total settling time with two different delays (2 dd). In this case, half of the steps are

[1] The track-phase can be used to extend the DAC settling for the first step. In this case the worst-case step is the second step.

Table 3.1 Minimum DAC settling time in each step for a 6 bits converter and different configurations. Basic = standard configuration with a fixed delay, Short = shortest total settling time with 6 different delays (dd), and the remaining configurations ($f - p$) have two different delays with f steps using the first delay and p steps using the second delays

Settling time $[\ln(2) \cdot \tau]$	Basic 1 dd	Short 6 dd	1–5 2 dd	2–4 2 dd	3–3 2 dd	4–2 2 dd	5–1 2 dd
1st step	6	6	6	6	6	6	6
2nd step	6	5	5	6	6	6	6
3rd step	6	4	5	4	6	6	6
4th step	6	3	5	4	3	6	6
5th step	6	2	5	4	3	2	6
6th step	6	1	5	4	3	2	1
Total	36	21	31	28	27	28	31

performed with delay A and the other half with delay B. This results in 25% less settling time compared to the standard case.

The general case is now derived for a converter with arbitrary resolution. The total settling time is:

$$t_{\text{settle}} = \big((n-p)\,n + p \cdot p\big) \cdot \ln(2) \cdot \tau = (p^2 - np + n^2) \cdot \ln(2) \cdot \tau \tag{3.4}$$

with n the number of bits of the converter and p the number of steps with a reduced settling time. The derivative of the settling time is:

$$\frac{d(t_{\text{settle}})}{dp} = (2p - n) \cdot \ln(2) \cdot \tau \tag{3.5}$$

Setting this derivative equal to zero yields:

$$\frac{d(t_{\text{settle}})}{dp} = 0 \rightarrow p = \frac{n}{2} \tag{3.6}$$

So, the total settling time is minimal when half the steps are done with delay A and half the steps are done with delay B. This does not depend on the resolution of the converter.

If the same delay is used in all steps, the total settling time is:

$$t_{\text{settle,1delay}} = n^2 \cdot \ln(2) \cdot \tau \tag{3.7}$$

For the case of two different delays, the total settling time is:

$$t_{\text{settle,2delays}} = \left(\frac{n}{2} \cdot n + \frac{n}{2} \cdot \frac{n}{2}\right) \ln(2) \cdot \tau = \frac{3}{4} n^2 \ln(2) \cdot \tau \tag{3.8}$$

The ratio between the total DAC settling times is 3/4. So, by using two different delays, the total settling time can be reduced by 25%.[2]

[2]This assumes an even number of bits, for an odd number of bits the reduction is slightly lower, since the number of steps cannot be divided into two equally sized groups.

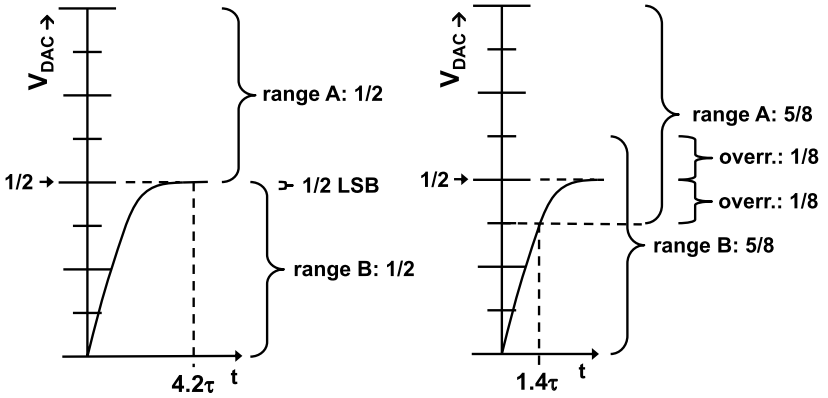

Fig. 3.3 *Left-hand* side: basic configuration; *right-hand* side: configuration with overrange

Although it can be attractive to use the scheme with variable settling times, for the remainder of this chapter the DAC settling time is assumed to be fixed, since the solutions presented in the next sections reduce the settling time even more and it is not useful to combine these with a variable settling time scheme.

Now, different SA-ADC architectures will be described, and they will be compared based on comparison time and power efficiency.

SA-ADC with Overranging

The required settling time can also be reduced by employing the principle of overranging [22]. This is schematically shown in Fig. 3.3. On the left-hand side the operation of a basic SA-ADC is shown for the example of a 6 bits converter, in which 4.2τ of settling is required to make the DAC error smaller than $^1/_2$ LSB. On the right-hand side, the operation of an SA-ADC with overranges of $^1/_8$ of the range is shown. If the input signal is below $^3/_8$ of the range, the comparator output should be 0, to indicate that the signal is in the lower range. If the input signal is between $^3/_8$ and $^5/_8$ both 0 and 1 are correct outputs, since the two ranges overlap in this region. If the input signal is above $^5/_8$ the comparator output should be 1, to indicate the upper range.

When using overranging, the output code does not follow directly from the comparator decisions as with standard SA-ADC, but some decoding is needed to resolve the binary code.

The advantage of overranging is that the comparator and the DAC together are allowed to make an error of $^1/_8$ of the range ($+^1/_2$ LSB) instead of only $^1/_2$ LSB for the conventional case. For the two architectures just described, comparator offset results in ADC offset and for most applications this is not critical. Other error sources like noise and hysteresis, do however cause errors: In the last step of the conversion, errors of the comparator directly lead to ADC errors. So, despite the overrange,

comparator errors (except offset) should be smaller[3] than $^1/_2$ LSB. Since the comparator needs to be accurate, the complete overrange can be spent to compensate for incomplete settling of the DAC.

The required DAC settling for an overrange of $^1/_8$ of the range is calculated next. When again assuming a rising DAC signal (without losing generality), the worst-case settling is from the minimum of the range to the middle of the range. The DAC signal is then described by:

$$V_{\mathrm{DAC}}(t) = \frac{1}{2} R\left(1 - e^{t/\tau}\right) \tag{3.9}$$

with R the size of the current range. The DAC needs to settle to at least $^3/_8$ of the range, so:

$$V_{\mathrm{DAC}}\left(t = t_{\mathrm{settle}}\right) \geq \frac{3}{8} R \tag{3.10}$$

Solving this using (3.9) yields:

$$t_{\mathrm{settle}} \geq \ln(4) \cdot \tau \approx 1.4\tau \tag{3.11}$$

A general expression for the minimum settling time for a first-order system on a step response is:

$$t_{\mathrm{settle}} \geq \ln\left(\frac{SZ}{\mathrm{error}}\right) \cdot \tau \tag{3.12}$$

with SZ the step-size, error the allowed error and τ the time-constant.

Equation (3.11) shows that overranging can reduce the settling time requirement significantly: without overranging 4.2τ of settling would be required for a 6 bits SA-ADC and 6.9τ for 10 bits resolution, see (3.2). Note that with overranging the required settling time is independent of the resolution of the converter and is the same for all conversion steps.

In a conventional SA-ADC the search-range (the range in which the input signal is) is halved each step, in the architecture with an overrange of $^1/_8$, the new range is $^5/_8$ of the previous range. If the search-range is halved each step, the so-called radix[4] is 2, which means that each step $^2\log(2) = 1$ bit is resolved. If the next range is $^5/_8$ of the previous range, the radix is $^8/_5$ and only $^2\log(^8/_5) = 0.68$ bit is resolved per step. To reach an accuracy of e.g. 6 bits, there are $6/0.68 \approx 9$ steps required. In general, the required number of conversion steps for the conventional overrange architecture is:

$$\mathrm{conversion\text{-}steps}_{\mathrm{overrange}} = \left\lceil \frac{n}{^2\log(\mathrm{radix})} \right\rceil = \left\lceil \frac{n}{^2\log\left(\frac{1}{\frac{1}{2}+\mathrm{OR}}\right)} \right\rceil \tag{3.13}$$

[3] For a pipeline converter things are different. Since the residue signal is amplified in each stage, an error of $^1/_8$ of the input range can be tolerated.

[4] The radix is the ratio of the old range and the new range.

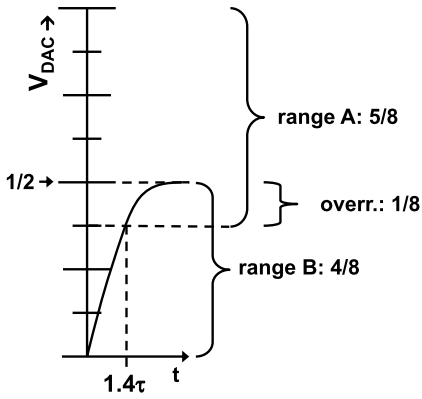

Fig. 3.4 Operation with an overrange on only one side of the comparator

with n the resolution and OR the overrange. The increased number of steps is a major disadvantage of using overranging. A technique to resolve more bits per step is described in the next section.

Single-sided Overrange Technique

The overrange technique described in the previous section has an overrange on both sides of the comparison level. An RC limited DAC however, does not show overshoot during transitions and the DAC error is only due to incomplete settling. An overrange region is therefore only needed at the side from which the DAC is settling. Based on this criterion, a new overrange technique is introduced, as shown in Fig. 3.4 for a rising DAC signal.[5] The DAC settling requirement is the same as in the case of normal (double-sided) overranging: 1.4τ. The size of the lower range is $^1/_2 R$, so 1 bit is resolved if this range is selected. However, if the upper range is selected still only 0.68 bit is resolved, and worst case there are still 9 steps required for an accuracy of 6 bits.

To lower the worst-case number of steps, the two new ranges need to have the same size. To achieve this, the DAC level is moved from mid-range to 9/16 of the range for a rising DAC signal[6] [25], as shown in Fig. 3.5. The size of the next range is now always 9/16 of the previous range, resulting in 0.83 resolved bits per step. For the 6 bits example, now only 7 instead of 9 steps are required,[7] which is a large improvement.

The settling requirement is slightly increased from 1.4τ to 1.5τ due to the larger maximum step-size (9/16 instead of 1/2) of the DAC signal.

[5] For a falling DAC signal the diagram should be flipped vertically.

[6] For a falling DAC signal, the DAC level is moved to 7/16 of the range.

[7] Actually the required number of steps is slightly above 7. However, for the last steps in the conversion the overrange is smaller than LSB/2. If an error of LSB/2 is allowed, the radix of these last steps can be slightly larger than 16/9, and 7 steps is sufficient to reach 6 bits accuracy.

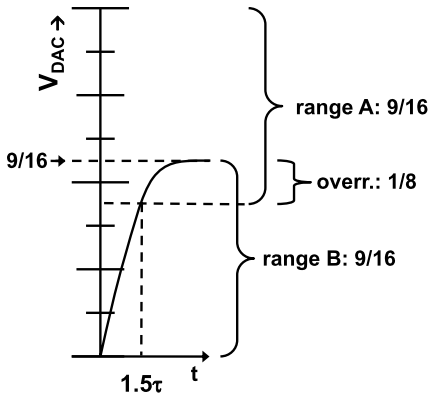

Fig. 3.5 Single-sided overrange technique

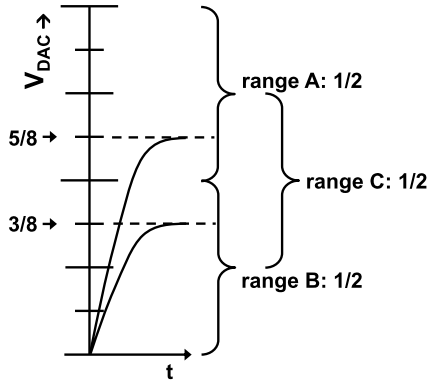

Fig. 3.6 SA-ADC operation with two comparators and two DACs in parallel

A general expression for the required number of conversion steps for the single-sided overrange architecture can be derived:

$$\text{conversion-steps}_{\text{singlesided}} = \left\lceil \frac{n}{^2\log(\frac{2}{\text{OR}+1})} \right\rceil \tag{3.14}$$

with again n the resolution and OR the overrange.

SA-ADC with Two Comparators in Parallel

To increase the maximum sample-rate, an SA-ADC can also operate with multiple comparators and DACs in parallel. An example is shown in Fig. 3.6, where two comparators and two DACs are used to select one out of three ranges. The lower range is selected when the signal is below the comparator at $^3/_8$, the middle range is selected when the signal is above the comparator at $^3/_8$, and below the comparator at $^5/_8$ and the upper range is selected when the signal is above the comparator at $^5/_8$.

For this configuration, the only sensible overrange is $^1/_8$. The new ranges are half the size of the range, so the radix is 2 and consequently in each step 1 bit is resolved.

Table 3.2 Overview of successive approximation ADC architectures

Architecture	Number of steps	Settling time [τ]	Conversion time [τ]	FoM static	FoM dynamic
Conventional	6	4.2	25.2	25.2	6
1/8 overrange	9	1.4	12.5	12.5	9
Single-sided OR	7	1.5	10.5	10.5	7
2 comparators	6	1.6	9.7	19.3	12

The required number of steps is thus simply:

$$\text{conversion–steps}_{\text{2CMPS}} = n \tag{3.15}$$

When using 2 DACs, there are two settling requirements. Since the upper DAC has the largest step-size, it has the most stringent settling requirement of 1.6τ. This architecture is similar to the one commonly used in 1.5 bit/stage pipeline converters [2, 10].

SA-ADC Architecture Comparison

The discussed architectures will now be compared. In Table 3.2 an overview is given for a resolution of 6 bits. For each architecture the following information is shown: (1) the number of steps, (2) the required DAC settling time per step, (3) the conversion time considering DAC settling time only (4) a Figure of Merit (FoM) representing energy units per conversion assuming all power consumption static and (5) a FoM representing energy units per conversion assuming all power consumption dynamic.

Since the power consumption of the various circuit blocks is not considered here, it is not possible to derive absolute Figures of Merits. To be able to make a power efficiency comparison nevertheless, relative FoMs are used instead.

The static FoM is the conversion time divided by τ, and the dynamic FoM is the number of comparator actions. For the two-comparator architecture, the static FoM is doubled, since it uses two comparators and two DACs in parallel.

It depends on the architecture of the comparator and the DAC, which of the two energy per conversion figures is more appropriate. For the design presented in this book, both the DAC and the comparator pre-amplifier consume static power and the first definition is more applicable.

For designs that use dynamic comparators [42] and e.g. charge based DACs, dynamic power consumption will be dominant and the second FoM is more appropriate.

Looking at the table, different conclusions can be drawn: If conversion time is not an issue and the power consumption is only dynamic, the conventional architecture shows the best power efficiency, since it uses the least number of comparisons.

The architecture with two comparators has the shortest total settling time and the single-sided overrange architecture is second best on this criterion, however, the

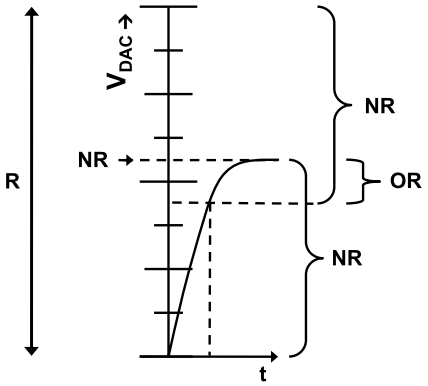

Fig. 3.7 Schematic for calculation of the optimum number of steps

latter uses much less energy per conversion. This is true if either static or dynamic power consumption is dominant.

Compared to the normal (1/8) overrange architecture commonly used in SA-ADCs [22], the single-sided overrange architecture uses 16% less static energy per conversion and 22% less dynamic energy per conversion.

The conversion time for the conventional architecture is proportional to n^2, while for the converters with overrange, the conversion time is proportional to n. So, for resolutions higher than 6 bits, the conventional architecture becomes worse on conversion time and static FoM, while the ratio between the architectures with overrange does not change on the aspects conversion time, dynamic and static energy consumption.

3.2.3 Optimum Number of Conversion Steps

In the previous section it was shown that overranging reduces the required DAC settling time and increases the power efficiency. An overrange with a size of 1/8 of the range was used as an example. In this section the optimum overrange size and hence the optimum number of conversion steps is derived. For this derivation, the single-sided overrange technique is used, as it has the best power efficiency. First the total DAC settling time is calculated, and after this the delay of the comparator and the logic is included.

The DAC settling time is calculated using the diagram shown in Fig. 3.7. R is the size of the current range, NR is the size of the new ranges and OR is the size of the overrange. For the moment, a DAC with an infinite number of levels is assumed, in practice the levels will have to be rounded to the closest DAC level.

To reach an accuracy of n bits in s steps, n/s bits have to be resolved each step. The ratio between the new range and the old range is therefore:

$$\frac{\mathrm{NR}}{\mathrm{R}} = 2^{-n/s} \tag{3.16}$$

and the overrange OR is:

$$\mathrm{OR} = 2 \cdot \mathrm{NR} - \mathrm{R} = \mathrm{R}\left(2 \cdot 2^{-n/s} - 1\right) \tag{3.17}$$

Worst case, the DAC signal will come from 0 and is described by[8]:

$$V_{\mathrm{DAC}}(t) = \mathrm{NR}\left(1 - e^{-t/\tau}\right) \tag{3.18}$$

The DAC signal has to settle to at least the level NR-OR:

$$V_{\mathrm{DAC}}\left(t = t_{\mathrm{settle}}\right) \geq \mathrm{NR} - \mathrm{OR} \tag{3.19}$$

The minimum settling time is therefore:

$$t_{\mathrm{settle}} \geq -\tau \ln\left(2 - 2^{n/s}\right) \tag{3.20}$$

This equation is valid for $s > n$. Note that for a given number of steps s, the required settling time is equal for all steps, until the overrange OR becomes smaller than the required accuracy of the converter of e.g. $^1/_2$ LSB. This is because both the step-size and the overrange (the tolerable error) decrease with the same factor. This is in contrast with a converter without overrange, where the required settling time decreases with each step as described in Sect. 3.2.2.

A 6 bits converter is again used as an example. In Table 3.3 the required settling time for one step and the total settling time are shown for different numbers of steps. In the case of 6 steps there is no overrange, for comparison however the settling times are included in the table. For a larger number of steps, the DAC is adjusted towards the input signal more often in a conversion and consequently the total settling time decreases. In the limit towards an infinite number of steps, the total DAC settling approaches $\ln(2^n)\tau = 4.2\tau$, which is the time required to let the DAC settle to 63/64 of the range. The DAC is then just in the upper LSB range, and the comparator can decide whether the ADC output code should be 111110 or 111111.

Considering the table, it seems attractive to increase the number of steps as far as possible, to decrease the total DAC settling time. However, as stated in the beginning of this section, there are more factors to take into account. The comparator and the logic also need time and both actions also consume power. The total conversion time is equal to the total DAC settling time, plus s (the number of steps) times the delay caused by the comparator and the logic. To determine the minimum conversion time, the time constant of the DAC τ and the delay of the comparator and logic are needed.

Time-constant of a DAC

There are many types of DACs, e.g. current-mode, voltage-mode and charge-mode DACs and each of them has a large number of implementations with different characteristics.

[8] Assuming first-order settling again.

Table 3.3 DAC settling time per step en total DAC settling time for different number of steps, for a resolution of 6 bits. The equations are valid for $n > 6$, for $n = 6$ there is no overrange

Number of steps s	Settling time per step $[\tau]$ $-\ln(2-2^{n/s})$	Total settling time $[\tau]$ $-s \cdot \ln(2-2^{n/s})$
6	4.2	25
7	1.67	11.7
8	1.15	9.2
9	0.89	8.0
20	0.26	5.3
100	0.043	4.3
∞	0	4.2

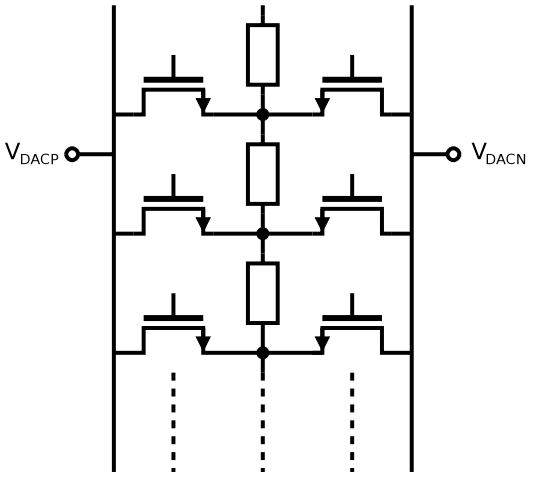

Fig. 3.8 Schematic of the differential resistor-ladder DAC in 0.13 µm CMOS

To derive a time-constant nevertheless, a DAC from the implementation discussed in Chap. 4 is used. It is a 6 bits DAC and has only 32 taps as explained in the section "DAC of the SA-ADC" on p. 103. The schematic of this differential resistor ladder DAC is shown in Fig. 3.8. The signal range is from 0.2 V to 0.6 V, and this allows the use of NMOST-only switches, which have more intrinsic bandwidth than PMOST switches or transmission gates (NMOST and PMOST in parallel). It is assumed that the top and bottom of the ladder are connected to low impedance reference nodes.

The time constant of the DAC is determined by the resistance of the ladder and a switch, and the capacitance, which is determined by the non-conductive ladder switches and the capacitive load of the comparator. The parasitic capacitance of the ladder itself is neglected.

The switch resistance and drain capacitance of an NMOST switch for different values of the source and drain voltage (assuming equal potentials) are shown in Table 3.4. The first column lists the voltage of the source and drain, the second

Table 3.4 R_{ON} in on-state, capacitances of an NMOST in off-state and the DAC time-constant for 32 switches. $W = 1$ µm, $L = 0.13$ µm, process is 0.13 µm CMOS

$V_S = V_D$ [V]	R_{ON} ($V_G = 1.2$) [kΩ]	C_{GDov} ($V_G = 0$) [fF]	C_{jDB} ($V_G = 0$) [fF]	C_{total} ($V_G = 0$) [fF]	τ_{DAC} ($R_{ON} \cdot 32 \cdot C_{total}$) [ps]
0	0.4	0.40	1.18	1.58	22
0.2	0.5	0.39	1.06	1.45	24
0.4	0.7	0.36	0.98	1.34	30
0.6	1.5	0.34	0.92	1.26	59
0.8	21	0.32	0.87	1.19	785

column lists the on-resistance of the switch when the gate-voltage is equal to V_{DD} (1.2 V). Bootstrapping is not used and the bulk terminal is connected to ground, as the process used does not support a triple well option. The next columns in Table 3.4 list the various capacitances when the switch is off. C_{GDov} is the gate-drain overlap capacitance, C_{jDB} is the drain-bulk junction capacitance and C_{total} is the sum of these capacitances. Note that the transistor is off and that the intrinsic transistor capacitance at the drain C_{DD}, which equals dQ_D/dV_D [45] is negligible. The last column shows the time-constant of the DAC, which is equal to $R_{ON} \cdot 32 \cdot C_{total}$.

Since the maximum signal level is 0.6 V, the worst case time constant is 59 ps for the switches alone. Note that this is independent of the width of the switches, since the resistance is inversely proportional to the width, and the capacitance is proportional to the width.

To get a practical DAC time-constant, the time constant is simply doubled to account for the ladder resistance and comparator capacitance. The factor of two is only a rough estimate and follows from simulation of the referred implementation.

Total Conversion Time as a Function of the Number of Steps

For the comparator time and logic delay together, three values are taken: 100 ps, 200 ps and 300 ps. For the design presented in this book, the latter value is most realistic. Again, a 6 bits converter is assumed. Table 3.5 shows the total settling time for different numbers of steps for the three values of the comparator and logic delay. From this table it can be concluded that going from 6 to 7 steps reduces the total conversion time significantly, while going from 7 to 8 steps gives only a slight increase for very short comparator and logic delays. Taking into account the fact that the dynamic comparator power increases with about 14% and the amount of logic doubles, it can be concluded that using more than 7 steps to resolve 6 bits is not useful in this technology.

For the described DAC, both the time-constant of the DAC and the delays of the comparator and logic depend roughly in the same way on the f_T of a process. It is therefore expected that the conclusions are also valid for other CMOS processes.

Table 3.5 DAC settling time for $\tau = 120$ ps and total conversion time for different values of the comparator plus logic delay, for a 6 bits converter

Number of steps s	Total DAC settling [ns] $s \cdot \tau \cdot \ln(2 - 2^{n/s})$	Total time per conversion [ns] for three different values of the delay for comparator and logic $s \cdot \tau \cdot \ln(2 - 2^{n/s}) + s \cdot t_{\text{logic,comp}}$		
		100 ps	200 ps	300 ps
6	3	3.6	4.2	4.8
7	1.4	2.1	2.8	3.5
8	1.1	1.9	2.7	3.5
9	0.96	1.9	2.8	3.7
20	0.64	2.6	4.6	6.6
100	0.52	11	21	31
∞	0.50	∞	∞	∞

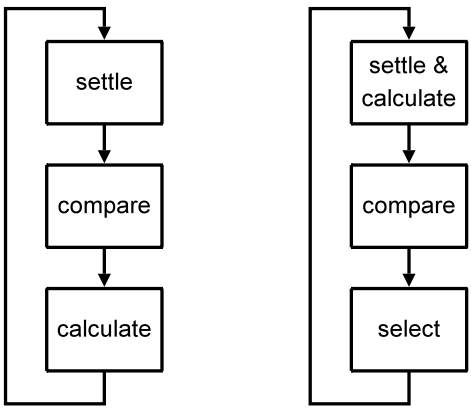

Fig. 3.9 SA-ADC flowcharts without look-ahead logic (*left*) and with look-ahead logic (*right*)

3.2.4 Look-ahead Logic

After each comparator decision in an SA-ADC, the logic has to calculate the next DAC level. This is indicated in the SA-ADC flowchart in Fig. 3.9 on the left side. For a conventional SA-ADC this task is simple and does not consume a significant amount of time. For architectures using overranging, this process is more complicated and requires more time, so it limits the maximum sample-rate. It is therefore advantageous to move the determination of the next DAC level out of the critical path.

Each comparator decision results in a choice between two DAC levels as indicated in Fig. 3.10. These two new DAC levels can already be calculated before the comparator makes its decision. Once the decision is made, the correct level only has to be selected as indicated on the right side of Fig. 3.9. This avoids most of the logic delay and increases the maximum sample-rate. The implementation is described in Sect. 4.4.2 on p. 99.

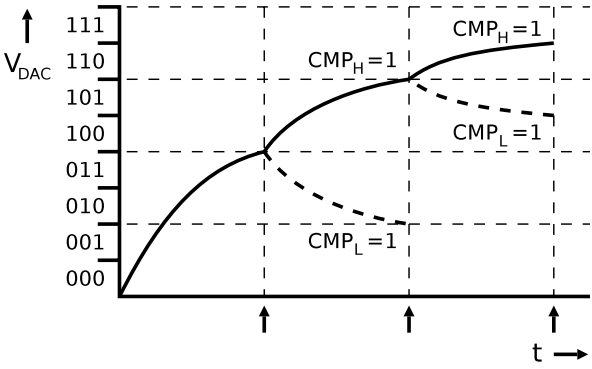

Fig. 3.10 After each comparator decision (indicated by the *arrows*) there are two possible new values for the DAC

3.2.5 Comparator

A comparator can be considered a differential amplifier with a very high gain. If the positive input is higher than the negative input, the output clips to the positive supply and if the negative input is higher than the positive input, the output clips to the negative supply. In the field of ADCs, a comparator is usually clocked and has two clock phases. The first phase is the reset phase and resets the comparator such that the memory of the previous decision is erased. The second phase is the comparison phase. In this phase the (possibly) small differential input voltage can first be integrated, after which a regenerative action amplifies the (small) differential input to a full swing output.

A possible implementation is shown in Fig. 3.11 [42, 43]. In the reset phase CLK is low (0 V) and MP_3 and MP_4 reset the outputs to V_{DD}. In the comparison phase CLK is high (V_{DD}), the tail current source is activated and the differential input voltage is integrated on nodes DI. After a while, MN_1 and MN_2 become active and slightly later MP_1 and MP_2 are also activated. These four transistors regenerate the input signal to a full-swing signal [43].

Comparator Accuracy

The accuracy of an SA-ADC depends on the accuracy of the comparator: comparator errors in the last comparator decision directly result in ADC errors. When an ADC needs an accuracy of e.g. ½ LSB, and half of this error is allowed to come from the comparator, the comparator needs an accuracy of ¼ LSB. In the previous sections it was assumed that there was one comparator making all decisions, and therefore this comparator should have an accuracy of ¼ LSB.

An accurate comparator requires more energy than a comparator with a lower accuracy. In an SA-ADC, not all comparisons require the same accuracy: In an architecture using overranging, the requirements on comparator and DAC errors are relaxed, in all but the last steps, and some imperfection is allowed, without degrading the final ADC accuracy. In the conventional overrange architecture and the

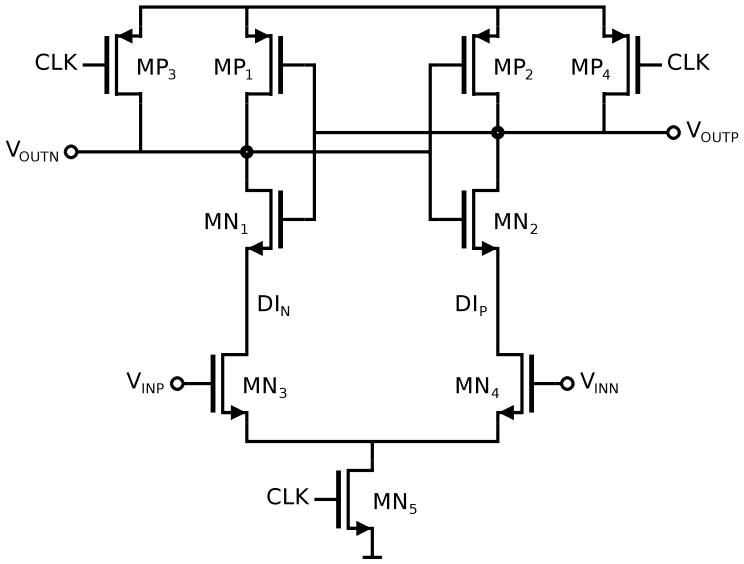

Fig. 3.11 Implementation of a comparator

architecture with 2 comparators, an overrange of $^1/_8$ of the range was used. This overrange was completely used to allow for limited DAC settling, as only one accurate comparator was used. However, part of the overrange can be reserved for comparator imperfections like noise and hysteresis.

To take advantage of the relaxed comparator requirements, another comparator is needed, since the last comparator decision still needs the full ADC accuracy.

Now, an analysis is made to demonstrate the amount of power that can be saved if two different comparators are used. A 10 bits SA-ADC is assumed that uses two comparators and two DACs in parallel (1.5 bit/stage) and has overranges of $^1/_8$ of the range. The accuracy is $^1/_4$ LSB. In the standard solution two comparators of the same type are used to do all conversions, so they need an accuracy of $^1/_4$ LSB and will consume a certain amount of energy E per conversion. A conversion of an input sample therefore uses $2 \cdot 10 \cdot E = 20 \cdot E$ for the comparators alone.

Next, two different comparators are used: a low power type for the first comparisons and an accurate, high power type for the last comparisons. Half of the overrange is used for DAC errors and the other half is used for comparator errors.[9] In the first step, the comparator is allowed to make an error of $^1/_8 \cdot {}^1/_2 \cdot 2^{10} = 64$ LSB thanks to the overrange, and an additional $^1/_4$ LSB thanks to allowed error of

[9]This increases the required DAC settling time, compared to the case where the error is completely spent on incomplete DAC settling. At the end of this section, it will become clear that the error the comparator can make is relatively the largest in the 8^{th} step. In the steps before this, the error is relatively smaller, requiring a smaller increase of the settling time. Moreover, in the 8^{th} step, the DAC step is very small and is no longer only limited by linear settling. In practice, the increase in settling time is therefore minimal.

Table 3.6 comparator accuracy and required power per step for a 10 bits ADC with 1/8 overrange

Step	Accuracy	Power [E]
1	$64\frac{1}{4}$	1/66k
2	$32\frac{1}{4}$	1/17k
3	$16\frac{1}{4}$	1/4225
4	$8\frac{1}{4}$	1/1089
5	$4\frac{1}{4}$	1/289
6	$2\frac{1}{4} = \frac{9}{4}$	1/81
7	$1\frac{1}{4} = \frac{5}{4}$	1/25
8	$\frac{1}{2} + \frac{1}{4} = \frac{3}{4}$	1/9
9	$\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$	1/4
10	$\frac{1}{4}$	1

$\frac{1}{4}$ LSB. So a comparator accuracy of $64\frac{1}{4}$ LSB is needed in the first step, $32\frac{1}{4}$ in the second step and so on, see Table 3.6.

In the 3rd column of the table the required power per step is shown. The power scaling is based on noise and offset requirements. For an optimally scaled design, the best way to decrease the noise and offset is using impedance level scaling [20, 49], also known as w-scaling [35]. All component widths are scaled with factor α, such that the noise and offset scale with $\sqrt{\alpha}$. For the first steps, this scaling is not entirely accurate, since other factors like drive capability determine the minimum power consumption.

For the last step(s) the same comparator type is used as in the standard solution, with an energy per comparison E. For the first steps a comparator type with less accuracy is used. When the low power comparator is used for the first 8 steps, then this comparator needs an accuracy (in the 8th step) which is 3 times less than the accurate type, as indicated in Table 3.6. If the comparator is noise limited, the required energy is 9 times less.

In Table 3.7 an overview is shown of different configurations with 2 comparator types. The first column lists the configuration with two numbers. The first number indicates the number of steps with the first comparator type and the second number is the number of steps with the second comparator type. The second column shows the accuracy of the first comparator, and the third column shows the corresponding relative power. The fourth and fifth column show respectively the accuracy and relative power of the second comparator and the last column shows the total power.

From this table it can be concluded that the 8–2 configuration is most efficient and uses 3 times less than the 10–0 configuration with one type of comparator.

It is possible to save even more energy by using more different comparators (or comparator settings) but this makes the design also more complex. For example, more selection switches and logic is required, and more offsets should be compensated, as described next.

Table 3.7 Overview of different configurations with 2 types of comparators. The accuracy and energy for both comparator types are shown and the total energy is listed in the last column

Config	Accuracy 1st CMP	Power 1st CMP	Accuracy 2nd CMP	Power 2nd CMP	Total power [E]
10–0	1/4	E	–	–	$2(10 \cdot 1) = 20$
9–1	1/2	$E/4$	1/4	E	$2(9 \cdot 1/4 + 1) = 6.50$
8–2	3/4	$E/9$	1/4	E	$2(8 \cdot 1/9 + 2) = 5.78$
7–3	5/4	$E/25$	1/4	E	$2(7 \cdot 1/25 + 3) = 6.56$
6–4	9/4	$E/81$	1/4	E	$2(6 \cdot 1/81 + 4) = 8.15$

Comparator Offset Requirements

In SA-ADC architectures with one comparator, comparator offset only results in ADC offset, which is acceptable in a lot of applications. If multiple comparators are used as in the architecture with two comparators and two DACs in parallel, or when using different comparators for different steps, offsets between comparators will degrade ADC performance. In these cases, attention should be paid to comparator offset, for example by using offset cancelation or calibration.

3.3 Efficiency of SA-ADC Versus Pipeline ADC

In this section a comparison of the power efficiency of an SA-ADC and that of a pipeline ADC is made.[10] An ADC contains many blocks, each with a lot of different implementations. It is therefore not feasible to find an analytical expression for the power consumption for a complete ADC. To make a comparison nevertheless, some simplifications are made.

Both converters require a DAC to generate reference voltages. The requirements and power consumption of this DAC are comparable for both ADCs, and even in a design with a very good power efficiency [54], the power consumption of the DAC can still be relatively small. The power consumption of the DAC is therefore not taken into account in this comparison.

Clock generation and control logic are also required in both architectures and regarding the two architectures, their complexity is comparable. Their power consumption will not be part of the comparison; for very power efficient designs how-

[10] In [53] a power efficiency comparison between different ADC architectures is made based on intrinsic capacitance. The required capacitance is determined based on noise and matching, and it is assumed that the power consumption is proportional to the amount of intrinsic capacitance. Here, mismatch is neglected, since calibration can compensate for mismatch with little power, and a fixed relation between the amount of capacitance and the power consumption is not assumed, as there is a significant difference in required power between e.g. a high-gain opamp charging a capacitor and a charge redistributing DAC.

ever their contribution in the total power consumption is not negligible. At the end of this section, this will be discussed.

In order to make a fair and fundamental comparison, the absolute minimum power consumption of the remaining blocks is determined. In order to do so, powerless calibrations are assumed, as it can be implemented with negligible power consumption. Moreover, optimal scaling is assumed, for example between different pipeline stages, and for the comparator in the SA-ADC.

For the SA-ADC, the comparator is the only remaining block. The comparator used in this efficiency comparison is a dynamic type comparator as described in Sect. 3.2.5. For the power, only the integrating frontend is considered. The backend latch is neglected for two reasons: (1) the contribution of the latch to the input referred noise of the comparator is mitigated, due to the voltage gain of the frontend that is typically about 10. (2) both type of converters need latches, and only the difference here is considered here.

For the pipeline ADC, opamps and comparators remain. The power consumption of the latter is neglected, since the resolution of the comparators is low compared to the resolution of the converter,[11] and therefore the power consumption of the comparator is low compared to that of the opamp. The commonly used topology with 1.5 bit per stage has an overrange of 1/8 of the full range, so comparator error of up to 1/8 of the range can be tolerated. Due to the gain in each stage, this holds for all stages.

In this comparison, the popular pipeline architecture using residue amplification with an opamp [2, 37, 64] is assumed. Traditionally, it uses an opamp with a large gain to cancel the input offset and to mitigate the non-linearity of the opamp, such that the stage-gain is completely determined by the feedback network. Typically more than 70 dB of gain is used for a 10 bits converter, and to maximize the signal swing (for increased SNR) 2-stage opamps are used, which require frequency compensation.

Moreover, traditionally large capacitors are used. The capacitors in a pipeline ADC need to satisfy two requirements: (1) the SNR requirement due to sampled kT/C noise, and (2) the matching requirement. The first criterion depends on the signal swing, while the second does not. Above a certain signal swing, the mismatch requirement is therefore dominant and converters usually operate in this region.

The use of high-gain two-stage opamps with frequency compensation and large capacitors based on matching, results in a high power consumption. In order to improve the power efficiency, opamps with a much lower gain are used in recent publications, and the capacitor sizes are only based on the required signal-to-noise ratio [12]. To reach the desired function of a pipeline stage nevertheless, many calibrations are required. An example of such a design is presented in [56], which describes a 10 bits converter using an opamp with a gain of only 28 dB. In that design, calibration is used to correct for opamp gain error, opamp nonlinearity and capacitor mismatch. Since minimal power consumption is the aim, the calculations are based on a similar design.

[11] This is under the assumption of a limited stage-gain.

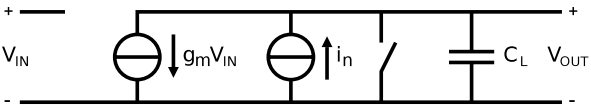

Fig. 3.12 Model of the input stage of a comparator

Next, the required power for an SA-ADC is derived, followed by the calculation of the required power of a pipeline converter. This section ends with comparison and discussion of the power consumptions.

3.3.1 SA-ADC

The required energy for a comparator with a certain accuracy will now be derived. An SA-ADC with overrange requires the most accurate comparator in the last conversion step, while in earlier steps the required accuracy is lower. Therefore, the required power for the last step is calculated, and this is scaled to include the power of the other steps as well.

The input stage of the comparator is modeled as an ideal transconductance amplifier[12] with noise current i_n, a capacitor and a reset switch, as shown in Fig. 3.12.

At $t = 0$ the switch is opened, and noise caused by the on-resistance of the switch is sampled on the capacitor with a noise-bandwidth of $\pi/2 \cdot f_{-3\text{dB}}$, with $f_{-3\text{dB}} = 1/(2\pi R_{\text{ON}} C)$ and R_{ON} the on-resistance of the switch. This results in the well known kT/C noise variance:

$$\overline{V^2_{\text{n,sample}}} = \frac{kT}{C_{\text{L}}} \tag{3.21}$$

After $t = 0$, the amplifier starts to integrate the fixed input voltage V_{IN} on the capacitor. The output voltage of the amplifier caused by the fixed input voltage is:

$$V_{\text{OUT}}(t) = V_{\text{IN}} \frac{g_{\text{m}}}{C_{\text{L}}} t \tag{3.22}$$

The impulse response from the noise current source to the output voltage is:

$$h_{\text{n}}(t) = \frac{1}{C_{\text{L}}} u(t) \tag{3.23}$$

with $u(t)$ the unity step function. Based on [15] and [46] the noise variance at the output node caused by the integrator is calculated, using $i^2_{\text{n}} = 4kT\gamma g_{\text{m}} \Delta f$ with γ the noise excess factor and Δf the noise bandwidth:

$$\overline{V^2_{\text{n,OUT},g_{\text{m}}}} = \frac{2kT\gamma g_{\text{m}}}{{C_{\text{L}}}^2} \cdot t \tag{3.24}$$

[12] An ideal transconductance amplifier has an infinite output resistance.

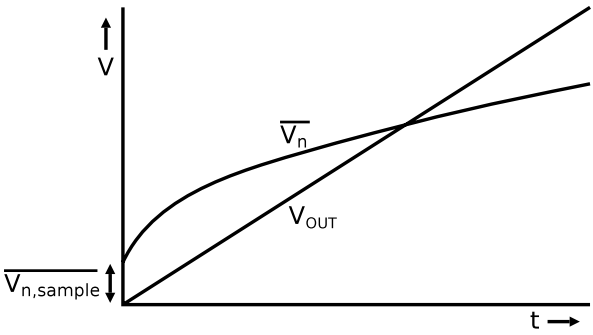

Fig. 3.13 The output signal V_{OUT} due to integration of the input signal, the standard deviation of the total output noise $\overline{V_{\mathrm{n}}}$ and the sampled kT/C noise indicated by $\overline{V_{\mathrm{n,sample}}}$

The variance of the total output noise at the output of the amplifier is thus:

$$\overline{V_{\mathrm{n,OUT,total}}^2}(t) = \frac{kT}{C_{\mathrm{L}}} + \frac{2kT\gamma g_{\mathrm{m}}}{{C_{\mathrm{L}}}^2} \cdot t \tag{3.25}$$

The output signal V_{OUT} due to integration of the input signal, and the standard deviation of the total output noise $\overline{V_{\mathrm{n}}}$ are shown in Fig. 3.13. The sampled kT/C noise, indicated by $\overline{V_{\mathrm{n,sample}}}$, is present from[13] $t = 0$, while the output signal and the output noise start rising after $t = 0$.

In order to relate the total noise variance to the input signal, it is divided by the square of the gain of the integrator A_{int}. This gain equals:

$$A_{\mathrm{int}} = \frac{V_{\mathrm{OUT}}}{V_{\mathrm{IN}}} = \frac{g_{\mathrm{m}}}{C_{\mathrm{L}}} t \tag{3.26}$$

So, the input referred noise variance is equal to:

$$\begin{aligned} \overline{V_{\mathrm{n,IN,total}}^2}(t) &= kT\left(\frac{C_{\mathrm{L}}}{g_{\mathrm{m}}^2 \cdot t^2} + \frac{2\gamma}{g_{\mathrm{m}} \cdot t}\right) \\ &= kT\left(\frac{C_{\mathrm{L}}}{(\frac{g_{\mathrm{m}}}{I})^2 \cdot I^2 \cdot t^2} + \frac{2\gamma}{(\frac{g_{\mathrm{m}}}{I}) \cdot I \cdot t}\right) \end{aligned} \tag{3.27}$$

From this equation it can be concluded that for best efficiency (minimal noise for a given current), the transistor should operate in weak inversion, for maximum g_{m}/I. It can also be noted that time and current can be traded. Regarding this, there is however a practical limitation: To increase the comparator speed (smaller t), the current should increase. To keep the transistor in weak inversion, an increasing width of the transistor is required. At a certain point, the parasitics of this transistor become impractically large and put a limit on the maximum sample-rate.

[13] For $t < 0$, kT/C noise is also present, however by opening the switch, a specific value is sampled.

Minimizing the Load Capacitance to Increase the SNR

From the above equation it also follows that the load capacitance should be minimized. Thermodynamics dictates that generating more signal power to improve the signal-to-noise ratio costs more power. It may therefore sound counterintuitive that the load capacitance should be decreased for a better SNR. This paradox is now discussed. The minimum amount of energy consumed by the transconductor to charge the capacitor is:

$$E_{\text{charge}} = \frac{1}{2} C_{\text{L}} V_{\text{C}}^2 \tag{3.28}$$

with V_{C} the voltage across the capacitor. If the capacitor is charged with a fixed current I_{charge}, voltage V_{C} equals:

$$V_{\text{C}} = \frac{I_{\text{charge}} \cdot t}{C_{\text{L}}} \tag{3.29}$$

The amount of energy consumed is then:

$$E_{\text{charge}} = \frac{I_{\text{charge}}^2 t^2}{2C_{\text{L}}} \tag{3.30}$$

So, for a given current I_{charge} and integrating time t, a smaller capacitor results in a higher energy consumption and a higher SNR, which is in agreement with thermodynamics.

Neglecting kT/C Noise

Combining (3.26) and (3.27) yields:

$$\overline{V_{\text{n,in,total}}^2} = \frac{kT}{g_{\text{m}} t} \left(\frac{1}{A_{\text{int}}} + 2\gamma \right) \tag{3.31}$$

For a single MOST, γ is approximately 2/3 and the gain is assumed to be at least 10, the sampled kT/C noise is therefore negligible compared to the noise of the integrator. Therefore, the equation of the input referred noise variance is simplified to:

$$\overline{V_{\text{n,in,total}}^2} \approx \frac{2kT\gamma}{g_{\text{m}} t} \tag{3.32}$$

Signal-to-Noise Ratio

The signal-to-noise ratio at the input of the comparator is now derived. For a sine-wave, the signal power[14] S is equal to:

$$S = \left(\frac{V_{\mathrm{PP}}}{2\sqrt{2}}\right)^2 = \frac{V_{\mathrm{PP}}^2}{8} \tag{3.33}$$

The SNR at the input is:

$$\mathrm{SNR} = \frac{V_{\mathrm{PP}}^2 \cdot g_{\mathrm{m}} t}{8 \cdot 2kT\gamma} \tag{3.34}$$

Assuming an integration time of half the sample-period ($t = 1/2f_{\mathrm{S}}$), (3.34) can be rewritten to get an expression for the required current for a given SNR, sample-rate and signal-swing[15]:

$$I_{\mathrm{CMP}} = \frac{32kT\gamma \cdot \mathrm{SNR} \cdot f_{\mathrm{S}}}{V_{\mathrm{PP}}^2 (\frac{g_{\mathrm{m}}}{I_{\mathrm{CMP}}})} \tag{3.35}$$

In this comparison, it is chosen to make the thermal noise equal to half the quantization noise. Therefore, SNR = $2 \cdot \frac{3}{2}\, 4^n$ is substituted:

$$I_{\mathrm{CMP}} = \frac{96kT\gamma \cdot 4^n \cdot f_{\mathrm{S}}}{V_{\mathrm{PP}}^2 (\frac{g_{\mathrm{m}}}{I_{\mathrm{CMP}}})} \tag{3.36}$$

I_{CMP} is the current required for the last comparator decision in the SA-ADC. In this analysis it is assumed that overranging is used and that the noise of the comparator of the second last step is within the overrange, so that it does not degrade the SNR.

For simplicity, it is assumed that the comparator in the second last step uses half the power of the last comparator, the third last uses a quarter, and so on.[16] Assuming a large number of stages, the total current is then twice that of the last stage comparator:

$$I_{\text{SA-ADC,CMPs}} = \frac{192kT\gamma \cdot 4^n \cdot f_{\mathrm{S}}}{V_{\mathrm{PP}}^2 (\frac{g_{\mathrm{m}}}{I})} \tag{3.37}$$

Substituting $V_{\mathrm{PP}} = \alpha V_{\mathrm{DD}}$ with α the fraction of the supply voltage used for signal swing, yields the total SA-ADC power consumption:

$$P_{\text{SA-ADC,CMPs}} = \frac{192kT\gamma \cdot 4^n \cdot f_{\mathrm{S}}}{\alpha^2 V_{\mathrm{DD}} (\frac{g_{\mathrm{m}}}{I})} \tag{3.38}$$

[14] Although it is common practice to use the term signal (noise) power instead of signal (noise) variance, it is of course not correct. However, if the SNR is determined at a certain node, the result is correct, since the impedance for both powers is the same and it drops out of the equation.

[15] In fact both sides are multiplied by I_{CMP} to get the current into the equation.

[16] Looking at Table 3.6 on p. 56, this is a slightly pessimistic approximation.

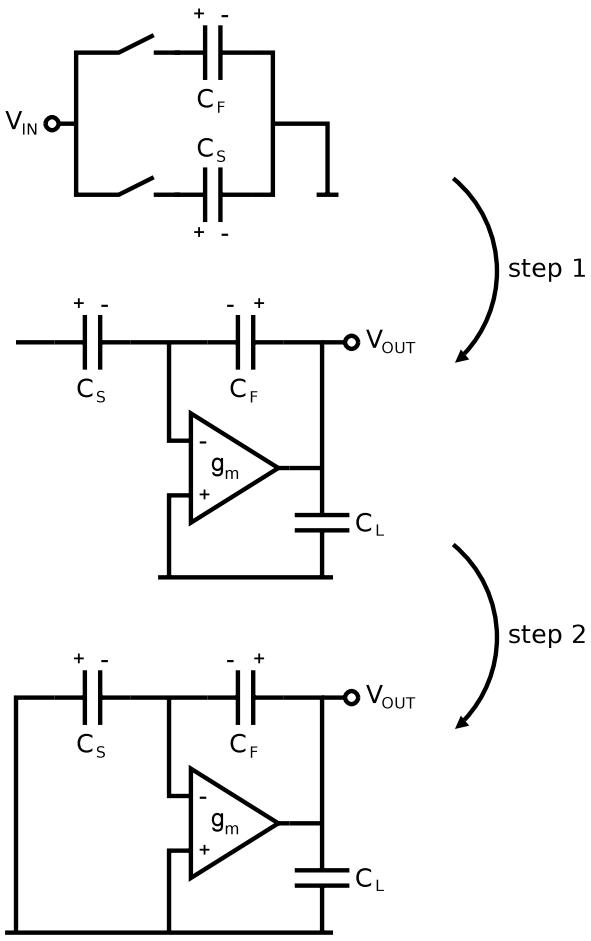

Fig. 3.14 Two modes of the pipeline stage with a virtual mode in between for ease of understanding

3.3.2 Pipeline Converter

Now, the power consumption of the opamp in a pipeline converter is derived. Compared to an SA-ADC, the stage scaling of a pipeline converter is the other way around: the first stage has to be most accurate. This is a result of the gain in each stage, which attenuates the noise of the subsequent stages. The required power for the first stage opamp is calculated first, and next, the amount of power is scaled to include all stages.

A pipeline stage has two modes: sample-mode and amplify-mode. In sample-mode, the opamp can be used in unity-gain feedback mode to automatically cancel the input offset of the opamp. Since minimum power consumption is the aim, the opamp is not used in sample-mode. Instead, the sample-mode is configured as shown in the upper part of Fig. 3.14 and it is assumed that offset calibration is used to mitigate opamp offset.

The sampling action results in kT/C noise on capacitors C_S (sample capacitor) and C_F (feedback capacitor). After sampling, the noise variance on both capacitors is:

$$\overline{V_{n,CS}^2} = \frac{kT}{C_S} \quad \text{and} \quad \overline{V_{n,CF}^2} = \frac{kT}{C_F} \tag{3.39}$$

For ease of understanding, going from sample-mode to amplify-mode is performed in two steps: First, the configuration as shown in the middle part of Fig. 3.14 is used. The amplifier enforces a virtual ground at the negative input of the amplifier and the output voltage of the amplifier is charged to V_{IN}. In this phase a noiseless amplifier is assumed, and the output noise variance due to this first step is:

$$\overline{V_{n,OUT,1}^2} = \overline{V_{n,CF}^2} \tag{3.40}$$

Secondly, the left-side of C_S is switched to ground, so the charge from C_S is transferred to C_F, and causes the output voltage to rise by $V_{IN} \cdot C_S/C_F$. For a noiseless amplifier, the output noise variance due to this second step is:

$$\overline{V_{n,OUT,2}^2} = \left(\frac{C_S}{C_F}\right)^2 \cdot \overline{V_{n,CS}^2} \tag{3.41}$$

The output noise variance due to both actions is:

$$\overline{V_{n,OUT,1,2}^2} = \frac{kT}{C_F}\left(1 + \frac{C_S}{C_F}\right) \tag{3.42}$$

Amplifier Noise

For calculating the noise of the amplifier, the configuration at the bottom of Fig. 3.14 is used. The stage-gain H is:

$$H = \frac{C_S + C_F}{C_F} = \frac{C_S}{C_F} + 1 \tag{3.43}$$

In order to calculate the required transconductance g_m, the time-constant of this configuration is derived:

$$\tau_{pipe} = \frac{C_{OUT}}{g_m \cdot \beta} = \frac{C_{OUT} \cdot H}{g_m} \tag{3.44}$$

with β the feedback factor equal to $1/H$, and C_{OUT} the effective capacitance at the output of the amplifier, which equals:

$$C_{OUT} = \frac{C_S \cdot C_F}{C_S + C_F} + C_L \tag{3.45}$$

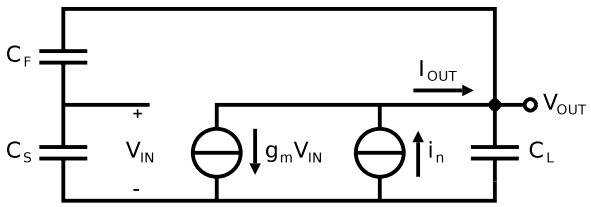

Fig. 3.15 Small signal circuit equivalent of the amplify-mode of a pipeline stage

with C_L the load capacitance of the next stage. Stage-scaling is assumed in such a way that the next stage contributes half the noise power of the current stage. Consequently, the total noise variance of all stages approaches twice that of the first stage, assuming a large number of stages.

The noise power of a stage is inversely proportional to the capacitor values. If the capacitors would be scaled with $1/H^2$, the input referred noise contribution of all stages would be equal. To halve the noise contribution of each next stage, the capacitors should be scaled with $2/H^2$. In this case, the value of the load capacitance is:

$$C_\mathrm{L} = \frac{2}{H^2}(C_\mathrm{S} + C_\mathrm{F}) \tag{3.46}$$

and the total output capacitance is equal to:

$$C_\mathrm{OUT} = \frac{C_\mathrm{S} \cdot C_\mathrm{F}}{C_\mathrm{S} + C_\mathrm{F}} + \frac{2}{H^2}(C_\mathrm{S} + C_\mathrm{F}) = C_\mathrm{F}\frac{H+1}{H} \tag{3.47}$$

The time-constant of the amplifier now becomes:

$$\tau_\mathrm{pipe} = \frac{C_\mathrm{F}}{g_\mathrm{m}}(H+1) \tag{3.48}$$

Using the small signal equivalent circuit of Fig. 3.15, the output noise variance caused by the amplifier is now calculated. The node voltage V_IN equals:

$$V_\mathrm{IN} = \frac{C_\mathrm{F}}{C_\mathrm{S} + C_\mathrm{F}} \cdot V_\mathrm{OUT} = \frac{V_\mathrm{OUT}}{H} \tag{3.49}$$

The noise voltage at the output of the amplifier is equal to:

$$V_\mathrm{n,OUT,amp} = I_\mathrm{OUT} \cdot Z_\mathrm{OUT} = \frac{i_\mathrm{n} - g_\mathrm{m} \cdot \frac{V_\mathrm{OUT}}{H}}{j\omega C_\mathrm{OUT}} = \frac{i_\mathrm{n}\frac{H}{g_\mathrm{m}}}{j\omega C_\mathrm{OUT}\frac{H}{g_\mathrm{m}} + 1} \tag{3.50}$$

So, the standard deviation of the output noise for low frequencies is equal to $i_\mathrm{n} \cdot H/g_\mathrm{m}$, and the noise bandwidth is:

$$BW_\mathrm{noise} = f_{-3\mathrm{dB}} \cdot \frac{\pi}{2} = \frac{g_\mathrm{m}}{4C_\mathrm{OUT} \cdot H} \tag{3.51}$$

Again using $i_n^2 = 4kT\gamma g_m \Delta f$, the noise variance at the output caused by the amplifier is:

$$\overline{V_{n,OUT,amp}^2} = 4kT\gamma g_m \cdot \frac{g_m}{4C_{OUT} \cdot H} \cdot \frac{H^2}{g_m^2} = \frac{kT}{C_{OUT}} \cdot \gamma H = \frac{kT}{C_F} \cdot \frac{\gamma H^2}{H+1} \quad (3.52)$$

The total output noise variance due to sampling, amplification and the amplifier is thus:

$$\overline{V_{n,OUT,total}^2} = \frac{kT}{C_F}\left(H + \frac{\gamma H^2}{H+1}\right) \quad (3.53)$$

Dividing this by the square of the stage gain H yields the total input referred noise variance:

$$\overline{V_{n,IN,total}^2} = \frac{kT}{C_F}\left(\frac{1}{H} + \frac{\gamma}{H+1}\right) = \frac{kT}{C_F} \cdot Q \quad (3.54)$$

The term between brackets (referred to as Q) represents in the first term both the noise caused by the sampled noise on C_F and the noise sampled on C_S and transferred to C_F, and the second term represents the noise caused by the amplifier.

Signal-to-Noise Ratio

For a sinewave, the signal power at the input of the stage is $V_{PP}^2/8$, so the signal-to-noise ratio at the input of the stage can be determined and this leads to a minimum requirement for capacitance C_F.

$$\text{SNR} = \frac{V_{PP}^2 C_F}{8kTQ} \quad \Rightarrow \quad C_F = \frac{\text{SNR} \cdot 8kTQ}{V_{PP}^2} \quad (3.55)$$

Combining this with (3.48) yields:

$$\tau_{pipe} = \frac{\text{SNR} \cdot 8kTQ}{g_m \cdot V_{PP}^2}(H+1) \quad (3.56)$$

For a settling error less[17] than LSB/4 in half a sample period, the following relation holds:

$$\frac{T_S}{2} = \frac{1}{2f_S} = (n+1) \cdot \ln(2) \cdot \tau \quad (3.57)$$

Again, it is chosen to make the thermal noise equal to half the quantization noise, so SNR = $2 \cdot \frac{3}{2}\, 4^n$ is substituted in (3.56) and combine this with (3.57) to get an

[17] In a pipeline converter, there is no correction for incomplete settling, so it can result in errors in the ADC output code. To avoid that settling errors dominate over thermal noise, the settling error should be smaller than LSB/4.

expression for the current required in the opamp in the first pipeline stage:

$$I_{\text{opamp}} = \frac{48kT \cdot 4^n \cdot (n+1) \cdot \ln(2) \cdot f_S}{\frac{g_m}{I_{\text{opamp}}} \cdot V_{PP}^2} \cdot \left(\frac{H+1}{H} + \gamma \right) \tag{3.58}$$

For simplicity, it is assumed that the current required for all opamps is twice that of the first-stage opamp, as is done for the SA-ADC. This is realistic for a stage-gain of two and a bit pessimistic for higher stage-gains.

The noise of all stages adds up and due to stage scaling with $2/H^2$, the total SNR is half that of the first stage only. To compensate for this, one bit is added to the SNR requirement. Now, the current consumption of all opamps in the pipeline ADC can be derived:

$$I_{\text{pipeline,opamps}} = \frac{192kT \cdot 4^n \cdot (n+1) \cdot \ln(2) \cdot f_S}{\frac{g_m}{I} \cdot V_{PP}^2} \cdot \left(\frac{H+1}{H} + \gamma \right) \tag{3.59}$$

Substituting $V_{PP} = \alpha V_{DD}$ with α the fraction of the supply voltage used for signal swing, yields the total pipeline ADC power consumption:

$$P_{\text{pipeline,opamps}} = \frac{192kT \cdot 4^n \cdot (n+1) \cdot \ln(2) \cdot f_S}{\frac{g_m}{I} \cdot \alpha^2 V_{DD}} \cdot \left(\frac{H+1}{H} + \gamma \right) \tag{3.60}$$

3.3.3 Comparison and Conclusions on Power Efficiency

In the previous sections, the minimum power consumption for an SA-ADC and a pipeline ADC were derived. This derivation is based on the comparator power for the SA-ADC and the opamp power for the pipeline. The power consumption of the T&H, reference DAC, digital logic and backend latches is not included and for the pipeline converter the comparator power is neglected as well. These assumptions are made since the power consumption of these blocks is either common to both architectures or it is negligible compared to other blocks.

The ratio of the power consumption of a pipeline converter and that of an SA-ADC, regarding the analyzed blocks is:

$$\frac{P_{\text{pipeline,opamps}}}{P_{\text{SA-ADC,CMPs}}} = \ln(2) \cdot (n+1) \left(\frac{H+1}{H\gamma} + 1 \right) \tag{3.61}$$

with H the stage-gain of the pipeline converter. For $\gamma = 2/3$, $n = 10$ and $H = 2$, this ratio is 25 and for H going to infinity, it is 19. So, the SA-ADC shows a significantly lower power consumption, thanks to the better power efficiency of an open-loop comparator compared to a closed-loop opamp.

If the power consumption of the additional blocks (DAC, control logic and clock generation) are included, the ratio of the powers becomes smaller. In an SA-ADC

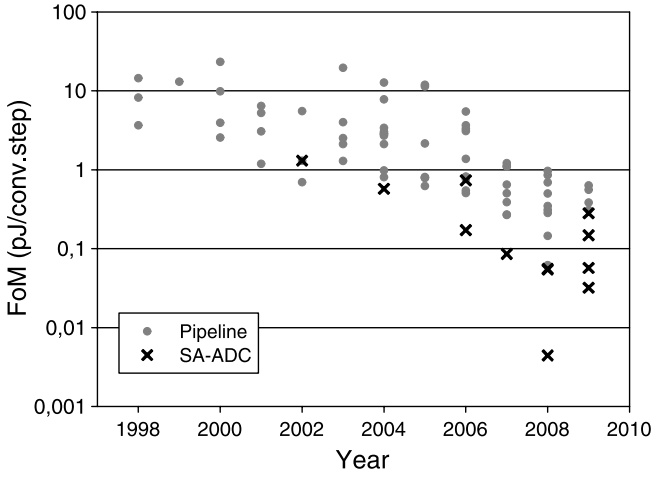

Fig. 3.16 Figure of Merit as a function of the year of publication for pipeline converters and SA-ADCs

design with a very good power efficiency [54], the power consumption of the control logic and clock generation is 44% of the total, however this includes the power of a delay-line to generate the clock phases. Even when including this power, the ratio of the power consumptions of the pipeline ADC and the SA-ADC is above 10.

To compare this power ratio with actual implementations, all non-interleaved pipeline and SA-ADCs published at the ISSCC and VLSI conferences from 1998 to 2009 are taken from [32]. The converters are compared using the well known Figure of Merit (FoM) defined as:

$$\text{FoM} = \frac{P}{2^{\text{ENOB}} \cdot f_S} \tag{3.62}$$

with P the power consumption, ENOB the effective number of bits, and f_S the sample-rate.

From Fig. 3.16 it is clear that the SA-ADC architecture has gained (renewed) interest from 2002, and that overall, the SA-ADC shows a significant better power efficiency than a pipeline converter. The most power efficient pipeline converter has a FoM of 62 fJ/conversion-step [9], while the most efficient SA-ADC has a FoM of 4.4 fJ/conversion-step [54]. Both implementations were presented in 2008 and are converters with a resolution of 10 bits. The ratio of the FoM of these converters is 14. These results show that also for actual implementations, the SA-ADC has clearly the best power efficiency.

3.4 Summary and Conclusions

The SA-ADC architecture can achieve a very good power efficiency, so it is an attractive option for the desired implementation and therefore it is discussed in detail.

Its sample-rate is however limited, since it needs multiple steps per conversion. The settling of the DAC and the delay of the logic are the main limiting factors for this, and solutions to increase the sample-rate are presented.

Using different DAC settling times for different conversion steps can reduce the DAC settling time with 42% for a 6 bits converter, the implementation of the required clocking scheme is however not trivial. Moreover, it is not useful to combined this technique with overranging, since the settling time is already short in that case.

Overranging techniques can reduce the required DAC settling time even more. Regarding power efficiency the newly presented single-sided overrange technique shows the best performance. Compared to the normal overrange architecture commonly used in SA-ADCs [22], the single-sided overrange architecture uses 16% less static energy per conversion and 22% less dynamic energy per conversion.

The optimum number of steps for a 6 bits SA-ADC using the single-sided overrange technique is 7. A larger number of steps can slightly decrease the total DAC settling time, the additional time required by the comparator and logic counteract this.

By calculating the next two possible DAC values in advance, the digital delay can be minimized. This is referred to as look-ahead logic.

By using comparators with different accuracies for different steps in the SA-ADC, power can be saved. However, for correct operation, the comparator offset should be small.

An analytical comparison between the power efficiencies of a pipeline ADC and an SA-ADC shows that the latter has a much better efficiency. This is confirmed by publications about actual implementations.

Regarding the desired implementation, the following design choices are derived:

- Use the SA-ADC architecture for the sub-ADCs, providing that the number of channels can be made sufficiently large to allow for the limited sample-rate of the sub-ADCs, and the specified sample-rate of the time-interleaved ADC is met.
- Implement the single-sided overrange technique
- Use 7 conversion steps for a 6 bits SA-ADC
- Exploit the use of look-ahead logic
- Use comparators with different accuracies for different steps

Chapter 4
Implementation of a High-speed Time-interleaved ADC

4.1 Introduction

In this chapter, the implementation of a high-speed time-interleaved ADC is presented, using the design choices described in the second and third chapter of this book. This design aims for a sample-rate of about 2 GS/s, an effective resolution[1] of 8–9 bits and an effective resolution bandwidth (ERBW) of 1 GHz. An even larger bandwidth is beneficial, as this would enable sub-sampling of signals in the GHz-range.

The design consists of 16 channels, consisting of a Track and Hold (T&H) and a sub-ADC. An overview of one channel is shown in Fig. 4.1. The core of the T&H part consists of a sample-switch, a capacitor and two buffers. Two additional switches are used to improve the performance and will be explained in this chapter. The sample-switch is driven by two blocks, which are combined in the figure: (1) a bootstrap block to enable close tracking of the input signal, which is required for good linearity, and (2) a low-skew switch-driver to define the sample moment accurately, and to minimize timing misalignment between channels.

The sub-ADC consists of a first Successive Approximation (SA) ADC, a DAC, a 16× amplifier, a re-sampler and a second SA-ADC.[2] The resolution of the SA-ADCs is 6 bits, and the total sub-ADC resolution is 10 bits.

Apart from the 16 channels, there are several additional blocks: The clock input is connected to a clock-buffer and the output signal of this buffer directly controls the sample moment through the switch-driver. The distributed clock-generation block is also connected to the clock-buffer and controls all switches and synchronous circuits. The overlapping or non-overlapping of various clock signals is guaranteed by this block.

For the target sample-rate of 2 GS/s, it is hardly feasible to bring all data off-chip with one LVDS (Low Voltage Differential Signaling) driver per bit, and moreover

[1] The number of output bits is 10.

[2] The reason that the sub-ADC consists of more than a single SA-ADC is motivated in Sect. 4.4.

S.M. Louwsma et al., *Time-interleaved Analog-to-Digital Converters,*
Analog Circuits and Signal Processing,
DOI 10.1007/978-90-481-9716-3_4, © Springer Science+Business Media B.V. 2011

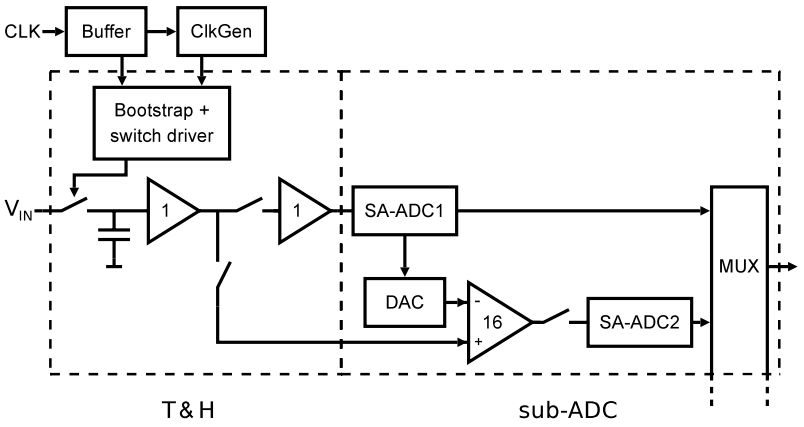

Fig. 4.1 Overview of a channel of the time-interleaved ADC

the data acquisition device does not support such a high data-rate. Therefore, a sub-sampling output multiplexer is integrated for evaluation purpose. It combines the data from all channels, and brings 1-out-of-9 samples to the outside world. This way, some data[3] from every sub-ADC is available and the full performance of the interleaved ADC can still be determined.

Finally, the design includes a digital interface to control calibration settings, and various IO-buffers and ESD protections.

This chapter is organized as follows: First, the implementation of the circuit blocks is described, starting with the clock generation in Sect. 4.2, followed by the Track and Hold in Sect. 4.3 and the sub-ADCs in Sect. 4.4. Second, the calibrations and adjustments of the ADC are treated in Sect. 4.5, followed by a description of the layout in Sect. 4.6. Section 4.7 presents the measurement results of the design and evaluates the design. The chapter ends with a description of an improved design [27], its measurement results and a comparison with other state of the art designs in Sect. 4.8.

This chapter contains quite some implementation details. Readers not interested in these can skip Sects. 4.2.2, 4.2.3, 4.3.4, and from Sect. 4.4.2 the subsections: Clock generation, Comparator, Digital control logic implementing..., and Decoder, and from Sect. 4.4.3 subsection: Binary to 1-out-of-32 decoder.

4.2 Clock Generation

An ADC channel requires quite a number of clock signals, with each a specific duty-cycle and phase. Some examples are: a T&H clock, two non-overlapping clocks for

[3] In fact it is also 1-out-of-9 samples from each sub-ADC. Note that not every factor can be used. If, for example, a factor of 2 is used instead of 9, only the data of half of the sub-ADCs can be analyzed.

the amplifier, two clocks for each SA-ADC and a DAC clock. For the complete ADC with 16 channels, the number of clock signals is huge. It is therefore not practical to generate all clock signals in one central block, since the distribution would take a large amount of space and power, and clock skew could become a significant problem due to the high clock frequency of up to 2 GHz and the chip area of approximately 2 by 2 mm.

Instead, the clock generation block only generates clock signals for the first channel, and in each channel D-flipflops are used to generate clock signals for the next channel. These flipflops are clocked with the high-frequency clock of around 2 GHz, so that the clock signals for the next channel are delayed by one period.

This requires the distribution of only one high-frequency clock over the complete chip, while the clock-signals going from channel to channel only need to be transferred over a relative short distance.

A Current Mode Logic (CML) clock is used for the T&H circuit and a CMOS logic clock is used for the sub-ADCs and the MUX. The input clock buffer is described next, followed by a control circuit for CML logic, a CML clock divider and a CML to CMOS conversion circuit. The generation of the non-critical clock signals is not described any further, as this is common design practice.

4.2.1 Clock Buffer

The chip has a differential clock input to suppress common-mode interference and to minimize cross-talk from the clock. This chip does not contain a Phase Locked Loop (PLL) or another clock cleaning mechanism; instead the clock signal is directly used as the sample clock, and the frequency equals the sample-rate.

The maximum amount of jitter for a 1 GHz input signal and 50 dB SNR is only 0.5 ps RMS. It is therefore important that the incoming clock signal is accurate and that it is kept clean. The Power Supply Rejection Ratio (PSRR) of CMOS logic is insufficient for this purpose, and instead Current Mode Logic (CML) is used in the T&H.

The use of CML in the T&H also minimizes crosstalk between the clock and the input signal. The rest of the chip works with sampled signals and therefore the timing is less critical. Apart from the T&H, regular CMOS logic is used to save power and area. Since the timing of the sampler is most critical and the input clock is already differential, the clock buffer generates the CML clock first. The full-swing clock for the CMOS logic is generated from this CML clock.

The schematic of the clock-buffer is shown in Fig. 4.2 and consists of two CML buffers. The second buffer is scaled such that it can drive its capacitive load of around 1 pF, with steep edges.[4] The resistors in the second stage are 57 Ω, the signal

[4]It is important to have steep clock edges, since this increases the maximum clock frequency at which the CML can operate, and it improves the accuracy of the sample moment regarding both random jitter and channel alignment. This is discussed in detail in Sect. 4.3.3.

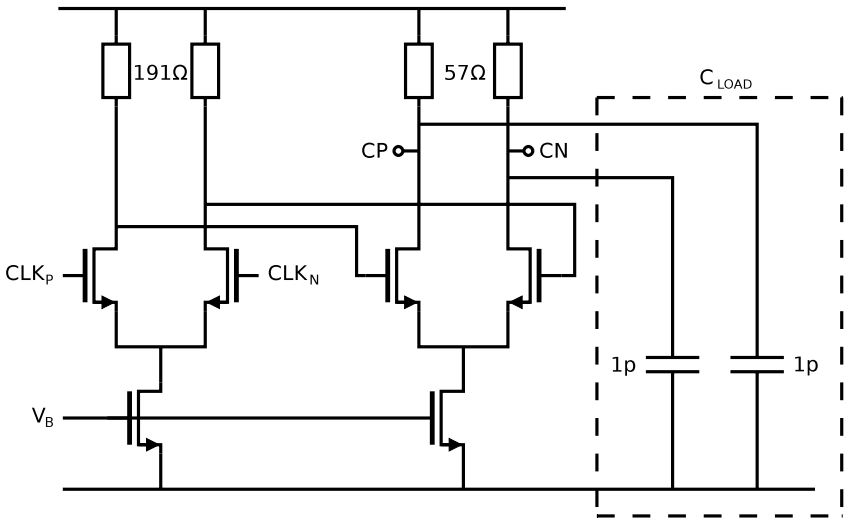

Fig. 4.2 Schematic of the CML clock buffer

swing is $V_{DD}/2$, with V_{DD} equal to 1.2 V, so the current is: $I = 0.6/57 = 10.5$ mA. At the zero-crossing of the clock, half of this current flows in or out the capacitor, and the slope of the edge is then:

$$\frac{dV}{dt} = \frac{I}{C} = \frac{5.25\text{ mA}}{1\text{ pF}} = 5.25\text{ GV/s} \tag{4.1}$$

Extrapolating this to the whole transition leads to a rise/fall-time of 110 ps.

The first buffer is used to increase the steepness of the incoming clock edges, such that the chip can be driven with a sinusoid, while the output signal of the buffer is steeper.

The capacitive load of the buffer consists largely of wiring. The rest is gate capacitance of clock generation circuits and the low-skew switch-driver circuit, which will be described in Sect. 4.3.3.

Control Circuit for the CML Signal-swing

The poly-silicon resistors used in the circuit have a large process spread of around $\pm 20\%$ and also the transistors show process spread. This leads to wafer-to-wafer variation of the CML signal swing. To make this signal swing independent of process spread, the circuit of Fig. 4.3 is used, with $\text{MP}_1 = \text{MP}_2$ and $R_1 = R_2 = R_3 = R$. The circuit generates a current of $V_{DD}/2R$. If this current flows through a (matched) resistance R, the voltage over the resistance is $V_{DD}/2$, independent of the exact value of R [55].

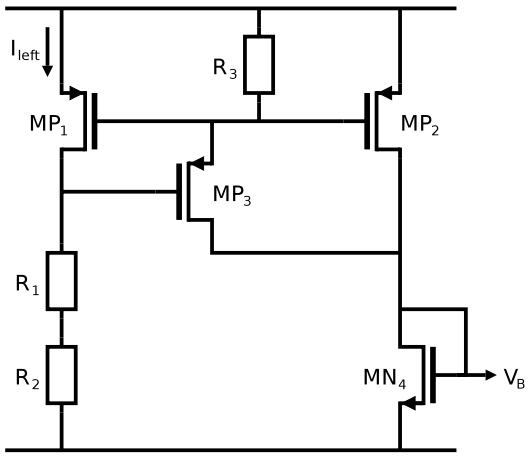

Fig. 4.3 Circuit to control the CML signal-swing. $R_1 = R_2 = R_3 = R$ and $\mathrm{MP}_1 = \mathrm{MP}_2$

The value of $V_{\mathrm{DD}}/2$ is the signal swing in the CML blocks and is a compromise between two criteria: A larger signal-swing increases the maximum speed [49], but it limits the number of devices that can be stacked to implement logic functions.

The circuit works as follows: the current in the left branch is:

$$I_{\mathrm{left}} = (V_{\mathrm{DD}} - V_{\mathrm{GS1}} - V_{\mathrm{GS3}})/2R \tag{4.2}$$

Via MP_2, this current is mirrored into the output diode MN_4, which generates the bias voltage V_{B} for the current sources in the CML circuits. The terms $-V_{\mathrm{GS1}}/2R$ and $-V_{\mathrm{GS3}}/2R$ are unwanted. The current through R_3 is:

$$I_{\mathrm{R3}} = V_{\mathrm{GS1}}/R \tag{4.3}$$

Via MP_3, this current also flows into the diode MN_4 and compensates for the two unwanted terms. If V_{GS1} equals V_{GS3}, then the current into the output diode is $V_{\mathrm{DD}}/2R$. The transistors are scaled such that the gate-source voltages are equal under nominal conditions, and they only change slightly with changes in e.g. the supply voltage.

To save power the circuit is scaled down compared to the buffers. This circuit is used a few times on the chip to generate local copies of the bias voltage. Mismatch will cause variations between the bias voltages and therefore cause differences in the signal swing over the chip. However, since the voltage gain of the CML cells is large enough, these differences do not cause any problems.

4.2.2 CML Clock-phase Generator

The T&H in each channel requires a 1-out-of-16 (1 period high, 15 periods low) CML clock signal, to implement the clocking scheme as shown in Fig. 4.4, which

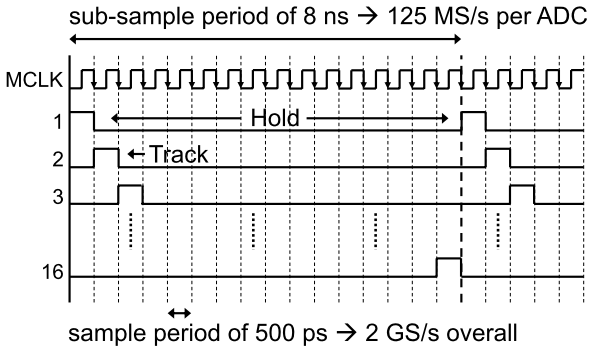

Fig. 4.4 Timing diagram with track-time of 1 period

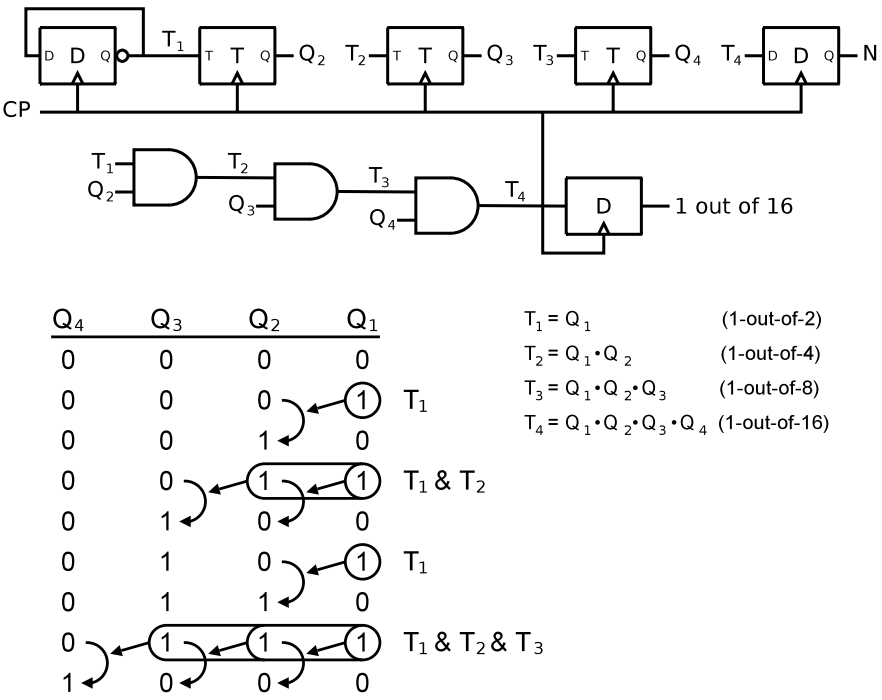

Fig. 4.5 Schematic of the 1-out-of-16 clock-phase generator (*top*) and part of the corresponding state diagram (*bottom*)

was discussed in Sect. 2.3.1. The schematic of the clock-phase generator is shown on the top of Fig. 4.5 and it is in fact a binary clock-divider using T-flipflops.[5] The first part of the state diagram is shown at the bottom of the same figure.

[5] At the rising edge of the clock, a T-flipflop toggles (hence the T) the output if the input is logical 1, otherwise the output does not change.

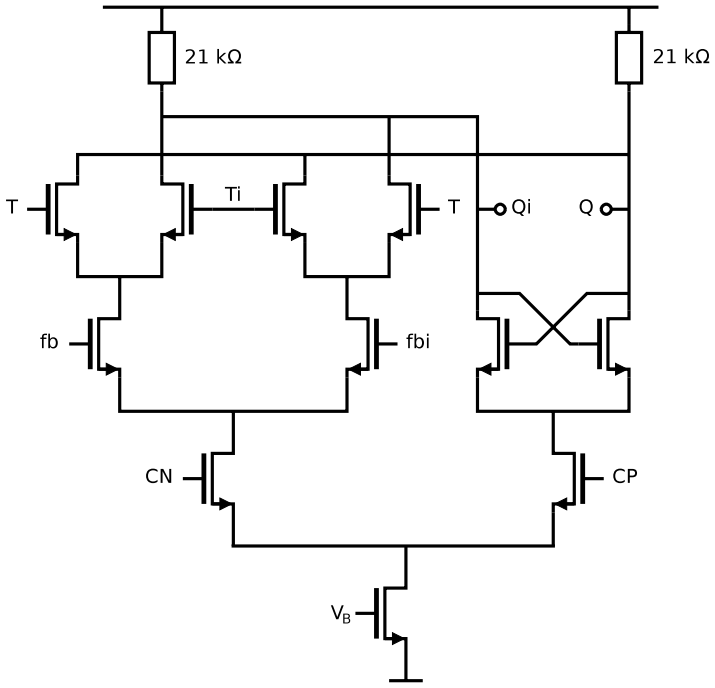

Fig. 4.6 Implementation of the first stage of a CML T-flipflop

The first stage generates the LSB and can be made with a regular D-flipflop with the inverted output connected to the input, as it needs to toggle on each rising edge of the clock. The second stage toggles, when the first stage output is logical high, the third stage toggles when both the first and the second stage are high, and so on. The toggle-4 signal (T_4) is the desired 1-out-of-16 signal. An additional D-flipflop is used to keep the delay small with respect to the main clock.

The T-flipflops are implemented with two stages, the first stages include an XOR function of the input signal and the toggle signal to implement the toggle function. The second stage is a latch, to hold the output signal when the clock-signal is high (CP > CN). The schematic of the first stage is shown in Fig. 4.6. For the second stage, the transistor widths are doubled and the resistor values are halved, to double the drive capability. Resistors are used instead of PMOST loads to minimize the capacitive load and maximize the speed.

4.2.3 CML to CMOS Conversion Circuit

In this chip, part of the circuitry uses CML signals and part uses CMOS logic signals, therefore a conversion circuit is needed to convert CML signals into full-swing CMOS signals. For timing uncritical signals, the conversion circuit as shown in

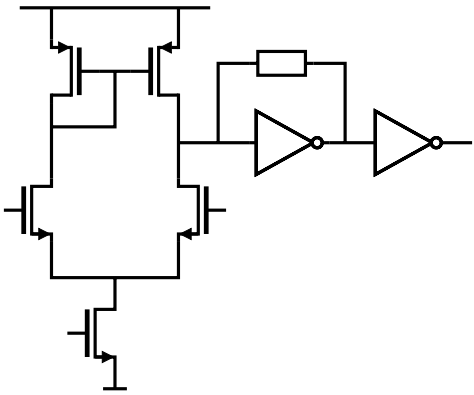

Fig. 4.7 CML to full-swing CMOS conversion circuit

Fig. 4.7 is used.[6] It consists of three stages, the first a differential input stage with a current-mirror load to perform the differential to single-ended conversion. The second trans-impedance stage is a low impedance load for the first stage to limit the signal swing at the output of the first stage. It is used to keep all transistors in saturation, such that the slow process of recovering from a discharged state is avoided. Moreover, the output stays symmetric with around 50% duty-cycle.[7] The last stage arranges the conversion to a full-swing output signal, suitable for CMOS logic.

The circuit is not optimized for low jitter, as the circuitry using this clock only handles sampled signals, and quite some jitter can be tolerated.

4.3 Track and Hold

This section treats the architecture and implementation of the T&H. The bootstrap mechanism applied to the sample-switch and its implementation is discussed first. Next, the low-skew switch-driver that accurately determines the sample moment is described. Finally the dual buffer which drives two different loads of the sub-ADC is described.

4.3.1 Bootstrapping of the Sample-switch

The main advantage of a bootstrapped clock-signal for a T&H is that it decreases and linearizes the resistance of the sample switch. A second advantage is that the sample-switch can be implemented with a single NMOST instead of a transmission gate (NMOST and PMOST in parallel), because even for signals close to the (positive) supply, the V_{GS} of the NMOST is sufficient to conduct well. The use of only an

[6] A conversion circuit for timing critical signals will be discussed in Sect. 4.3.3.

[7] Assuming that the input duty-cycle is 50% as well.

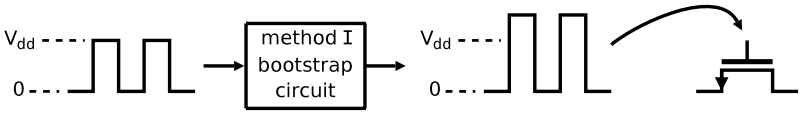

Fig. 4.8 Bootstrapping according to method I: the clock-signal is bootstrapped above the supply voltage

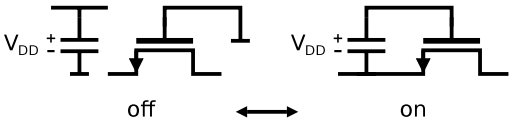

Fig. 4.9 Bootstrapping according to method II: switch in the off-state and in the on-state

NMOST is advantageous, as the parasitic capacitance from the PMOST is avoided and only one clock-phase is required.

In the area of T&Hs, bootstrapping is known in two incarnations. In bootstrapping according to method I, the clock-signal for the sample-switch is bootstrapped above the supply voltage as shown in Fig. 4.8. In [10] an implementation is described that generates a square-wave clock-signal of 0/5 V from a 3.3 V supply, with a technique published in [34].

In the implementation of [10], an additional bootstrap circuit is needed to charge the well of a PMOST, to prevent latch-up. Device reliability is not a concern there, since 5-volt-capable devices are used.

So for method I, when the switch is conducting, the gate voltage is fixed. This is not the case for method II:

Bootstrapping according to method II is described in [2], where the gate of the switch is lifted with respect to its source. This operation is shown in Fig. 4.9. In the off-state, the gate of the switch is connected to ground and a capacitor is charged to V_{DD}. In the ON-state the charged capacitor is connected between the source and the gate. Bootstrapping according to method II has many advantages:

- The on-resistance of the sample-switch is minimized.

The switch resistance is minimized as V_{GS} is always equal to the maximum allowed voltage V_{DD}. A small switch resistance results in a large sampler bandwidth, such that variations in the resistance have little impact for the signal frequencies of interest, as described in section

- The on-resistance of the sample-switch is constant over varying input signals.

In general, the relationship between the on-resistance and the input voltage is nonlinear, and causes distortion. If bootstrapping according to method II is used, the gate-source voltage is constant and it does not depend on the input signal, so the on-resistance is constant as well. This increases the linearity of the T&H.

Note that the bulk terminal of the sample-switch will be usually connected to ground. Due to the back-gate effect, the on-resistance still depends on the input signal, however much less than without bootstrapping.

- Channel-charge injection is independent of the momentary input signal.

Depending on the ratio of the impedances on either side of the switch, part of the channel-charge is injected into the impedance on one side of the switch and part is injected into the other side of the switch. For a non-bootstrapped switch or bootstrapping according to method I, the gate-voltage is fixed, and the source voltage is equal to the momentary input signal. So, V_{GS} of the sample-switch depends on the input voltage and as the relation between channel-charge and V_{GS} is non-linear, charge injection causes distortion.

If bootstrapping according to method II is used, V_{GS} is fixed and so is the amount of channel-charge. Therefore, when going from track-mode to hold-mode, only the signal independent pedestal step remains. This only causes channel offset which is relatively easy to compensate for.

- The turn-off delay of the sample switch is independent of the momentary input signal.
- Except for transitions, the voltage difference between two terminals of a device is less than V_{DD}, which minimizes device reliability issues.

These last two items have a longer motivation, which will be given in the next two subsections. In conclusion, bootstrapping according to method II is clearly advantageous over method I, and therefore it is implemented in this design.

Signal Independent Turn-off Delay

Neglecting sub-threshold conduction, the sample-switch turns off when the gate-source voltage V_{GS} becomes smaller than the threshold voltage V_T. If a sample-switch is not bootstrapped, or bootstrapped according to the method I, the sampling process with a finite slope is shown in Fig. 4.10. If the input signal is high, the switch turns off earlier than if the input signal is low. The switch turns off when

$$V_{GS} = V_T \quad \Rightarrow \quad V_G - V_T = V_S \tag{4.4}$$

Instead of the actual gate voltage V_G, $V_G - V_T$ is drawn, so that the crossing of this line with the input signal V_S marks the actual sample-moment. The time between the ideal sample-moment and the actual sample-moment is:

$$t_{delay} = \frac{V_{DD} - V_T - V_S}{\frac{dV_G}{dt}} \tag{4.5}$$

The voltage error made compared to ideal sampling (with an infinitely steep gate slope), can be approximated by the product of t_{delay} and the derivative of the input signal V_S:

$$\text{error}_{\text{finite slope}} = \frac{dV_S}{dt} \cdot t_{delay} = \frac{dV_S}{dt} \cdot \frac{V_{DD} - V_T - V_S}{\frac{dV_G}{dt}} \tag{4.6}$$

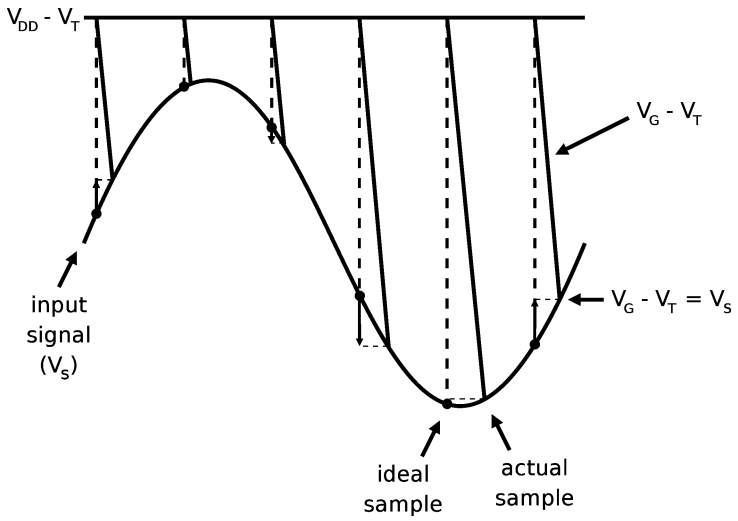

Fig. 4.10 Sampling with a finite gate-slope without using bootstrapping

For a sinusoidal input signal $V_S = A \cdot \sin(\omega t)$, the error is:

$$\begin{aligned} \text{error}_{\text{finite slope}} &= A\omega \cdot \cos(\omega t) \cdot \frac{V_{DD} - V_T - A \cdot \sin(\omega t)}{\frac{dV_G}{dt}} \\ &= \frac{A\omega}{\frac{dV_G}{dt}} \left(\cos(\omega t) \cdot (V_{DD} - V_T) - \cos(\omega t) \cdot A \cdot \sin(\omega t)\right) \\ &= \frac{A\omega}{\frac{dV_G}{dt}} \left(C \cdot \cos(\omega t) - \frac{A}{2} \sin(2\omega t)\right) \end{aligned} \tag{4.7}$$

with C the constant $V_{DD} - V_T$. From the last line of this equation the following conclusions can be drawn: (1) the error increases with signal frequency, (2) the error is inversely proportional with the slope of the gate voltage, (3) the first term between the large brackets, implies an amplitude and phase error, but no distortion and (4) the second term implies second-order distortion.

In reality, the gate slope will not be completely constant as assumed above, and also higher-order distortion products will arise. This is a major cause of sampler non-linearity.

If bootstrapping according to method II is applied, the sampling process is shown in Fig. 4.11. For the same reason as above, $V_G - V_T$ instead of V_G and $V_S + V_{DD} - V_T$ instead of $V_S + V_{DD}$ are drawn. Ideally, the infinite gate-slope only causes a fixed delay and for a certain frequency the difference between the ideal and the actual sampled signal is only a phase-shift. In practice, the gate-slope depends slightly on the input-signal and some non-linearity remains.

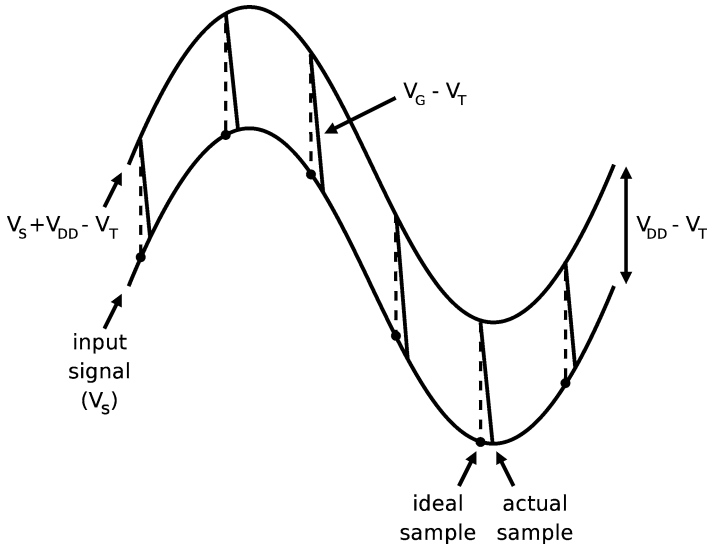

Fig. 4.11 Sampling with a finite gate-slope using bootstrapping according to method II. The finite gate-slope only causes delay

Reliability

In [2] an implementation of bootstrapping according to method II is described. The schematic is shown in Fig. 4.12 and operates as follows: in hold-mode $\varphi = 0$, and MN_3 and MN_4 charge C_3 to V_{DD}. MN_1, MN_2, C_1, C_2 and the inverter generate a voltage of about $2 \cdot V_{DD}$ to keep MN_3 on while its source approaches V_{DD}.

In track-mode $\varphi = V_{DD}$, and MN_7 and MP_2 connect C_3 between the source and gate of the sample switch MN_8. Now, bootstrapping according to method II is a fact. At the end of track-mode, MN_{10} is turned on and together with cascode MN_9, they turn the sample switch off.

This circuit works well for its applications, however for target specifications of this book it has three disadvantages. The first disadvantage is the additional capacitance at the source node of the sample-switch. In the referred paper it is stated that this is one of the factors limiting the reduction in overall power consumption.

The second disadvantage is that when the gate is charged rapidly, the gate-source voltage may exceed the supply voltage, possibly leading to device reliability issues [2]. This can be understood as follows, see Fig. 4.9. Suppose the switch is off, the input voltage connected to the source is around V_{DD} and the previous sample, still present on the drain node, is close to ground. When the charged capacitor is now connected between the gate and the source terminals, the gate node will rapidly rise to about $2 \cdot V_{DD}$, while the drain node is still close to ground. As it takes some time to form a channel in the transistor, the gate-drain voltage is momentarily larger than V_{DD}. This reliability issue can be avoided by charging the gate more slowly, but this limits the sample-rate and is not an option for the target sample-rate.

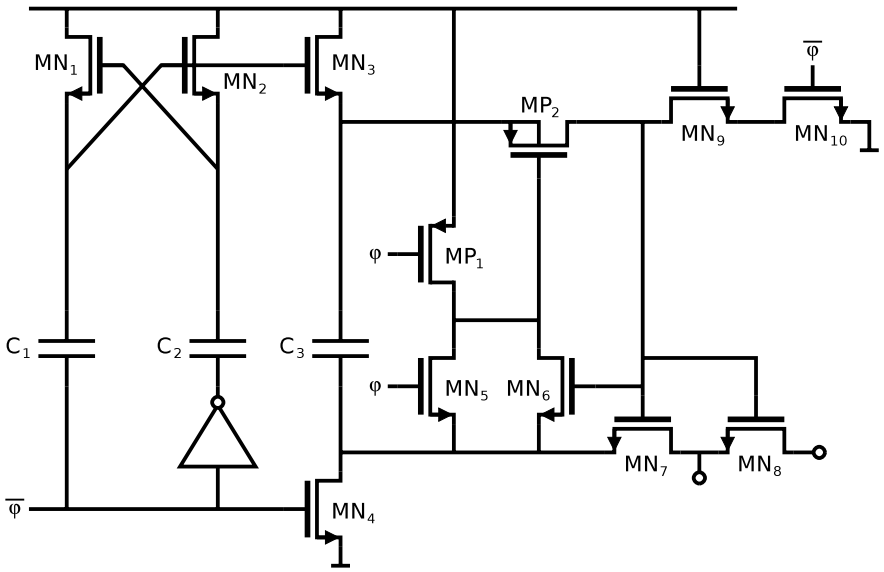

Fig. 4.12 Bootstrap circuit according to method II [2]

In track-mode, the gate of the sample switch (MN_8) is charged above the supply voltage. When the gate is discharged, reliability could be a problem. To avoid reliability issues, the transistor turning off the sample switch (MN_{10}) is cascoded. This is a third disadvantage, since the cascode transistor adds series resistance and parasitic capacitance, which decreases the gate slope on the sample switch and makes the sampling process less ideal. Moreover, for a time-interleaved architecture, mismatch of cascode transistors also increases timing skew.

Simplified Bootstrap Implementation

In this design, bootstrapping according to method II is implemented, based on the same idea as [2], however with a different implementation that is suitable for higher sample-rates. The schematic is shown in Fig. 4.13 and will be described in the next section. The value of the bootstrap capacitor is chosen such that, after charge redistribution between the capacitor and the gate of the sample-switch, the gate voltage never exceeds the supply voltage. By doing so, all voltages stay between the supply rails and reliability issues are avoided completely.

This implementation is not suitable for rail-to-rail input signals. This is not a problem, since such large signals are not used, as the rest of the circuitry is also not suitable for these.

In [2] a bootstrap capacitor of 0.5–1.8 pF is used, while here only 60 fF is used. This results in a much smaller parasitic capacitance of only 8 fF, and mitigates the disadvantage of the increased capacitance at the source node of the sample switch.

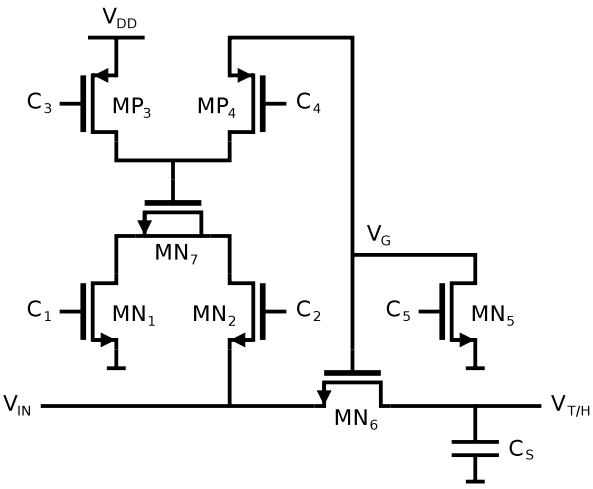

Fig. 4.13 Implementation of the bootstrapped T&H core

The most important goal of bootstrapping is that the value of the on-resistance is kept constant and this is also achieved with this simple implementation.

In some designs, a dummy transistor is added to lower the effect of charge injection and clock feed-through. Bootstrapping makes the injected charge independent of the input signal. A fixed pedestal step is not a problem and the use of a dummy transistor to decrease the step would only cause more spread in this step. Therefore, a dummy transistor is not used in this design.

4.3.2 Implementation

An overview of the T&H core is shown in Fig. 4.13, for clarity only half of the differential circuit is shown. Transistor MN_6 is the sample-switch, MN_7 serves as bootstrap capacitor, and the other transistors operate as switches. The circuit operation is as follows: First, MN_2 and MP_4 are off, while MN_1, MP_3 and MN_5 are on, such that capacitor MN_7 is charged to V_{DD} and the sample-switch is off. Next, MN_1, MP_3 and MN_5 turn off, after which MN_2 and MP_4 turn on, and the gate of MN_6 is charged with respect to its source. The sample-switch is now conducting and the T&H is in track-mode.

At the end of the track-mode MN_2 and MP_4 turn off, however, due to the parasitic gate-source capacitance, MN_6 stays conducting. All these timing signals are not critical, as they do not influence the sample moment.[8]

Finally, MN_5 is turned on and discharges the gate of MN_6 rapidly. This defines the sample moment and changes the state of the T&H from track-mode to hold-mode. The turn-off signal for MN_5 (called C_5) is the only time critical clock signal, and the generation of this clock-signal is described in the next section.

[8]This is the moment the T&H changes from track-mode to hold-mode.

The target resolution is about 9 effective bits and since kT/C noise is just one of the many noise and distortion components, the SNR due to kT/C noise should be about 10 bits. So, the required size of the sample capacitor is:

$$C_{\text{required},kT/C} = \frac{\text{SNR}_{kT/C} 8kT}{V_{\text{PP}}^2} \tag{4.8}$$

For an SNR of 62 dB or 10 bits and a signal swing of 0.4V_{PP} (SE), the required capacitance is 325 fF. Since the circuit is (quasi) differential, half of this capacitance is required on each side. In this implementation 150 fF is used on each side.

In Sect. 2.2.2 it was motivated that a very large channel bandwidth is advantageous, since for the frequencies of interest bandwidth mismatch is mitigated.

The sample switch (MN_6) has a geometry of 10/0.13 µm, which has a maximum resistance[9] over the signal swing of 150 Ω. The resulting channel bandwidth is 7 GHz.

The time-constant of capacitor MN_7 with switches MN_1 and MP_3 should be small enough to charge the capacitor in a few clock-cycles of the hold mode. Switches with a geometry of 1/0.13 µm for the NMOST and 2/0.13 µm for the PMOST are sufficient. Switches MN_2 and MP_4 should charge the gate of the switch-transistor (16 fF) in a fraction of a period. These switches should be minimal in size to minimize the charge-dump and clock feed-through into the source and the gate of the sample-switch. These switches have the same geometry as MN_1/MP_3.

4.3.3 Low-skew Switch-driver

Calibration of timing mismatch requires high-frequency test-signals and complicated calibration algorithms, as discussed in Sect. 2.7.3. Instead, the aim is to achieve a timing alignment within the required accuracy by careful design. This also helps to avoid adjustable timing blocks required for calibration, which generate a significant amount of jitter compared to their non-adjustable counterparts.

For a large number of channels, the timing offset per channel can be approximated by a Gaussian distribution. In this case, the SNR caused by timing offsets is:

$$\text{SNR}_{\Delta t} = \frac{1}{\sigma(\Delta t) \cdot 2\pi \cdot f_{\text{IN}}} \tag{4.9}$$

with $\sigma(\Delta t)$ the RMS value of the timing offset. For an SNR of 50 dB and an input frequency f_{IN} of 1 GHz, the required timing offset between channels should be smaller than 0.5 ps RMS. In [16] a technique to prevent timing errors in a time-interleaved T&H is presented. It uses a frontend sampling switch, which is closed

[9] Thanks to bootstrapping, the resistance is almost constant. Only the body-effect causes some dependence on the input signal.

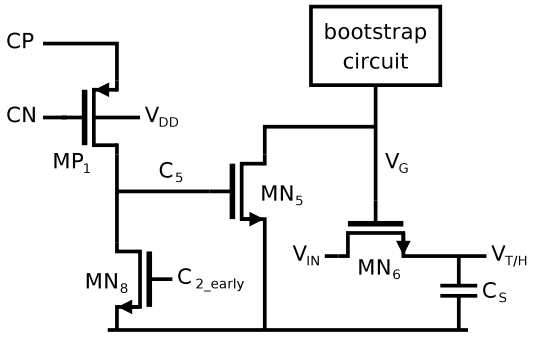

Fig. 4.14 Schematic of the low-skew sample-switch driver

only half of the period of the master clock. As described in Sect. 2.3.2, a disadvantage of this method is the decrease in bandwidth, which makes it unsuitable for high signal frequencies and high interleaving factors.

In the design presented in this chapter, a good timing alignment is achieved by using a master clock [33] to synchronize the different sampling instants and matched lines in the layout are used (same width, length and spacings) to distribute clock and input signals to the channels (see Fig. 4.50 on p. 116).

In applications where supply noise can degrade performance, CML (Current-Mode Logic) is commonly used, because it generates little supply noise, due to its nearly constant current consumption. It is also quite insensitive to supply noise. CML uses differential signaling, with a typical signal swing of half of the supply voltage. To convert the CML master clock into a full-swing signal suitable for the sample switch, a conversion circuit is needed.

In an earlier experiment, the implementation as shown in Fig. 4.7 on p. 78 was used for this purpose. A timing misalignment of 6 ps RMS was measured, which is much too high for the target specification. Therefore, a new circuit topology is proposed that minimizes the number of components in the path from the common master-clock to the sample switch.

The circuit diagram and the waveforms are shown in Figs. 4.14 and 4.15 respectively. Signals CP and CN are the clean CML clock signals directly from the clock buffer. The circuit operates as follows: the T&H is put in track-mode by the bootstrap circuit, and at the end of the track mode ($t = t_2$), node V_G is left floating by the bootstrap circuit, and for further bootstrapping is relied on parasitic capacitance.

Now, to switch into hold mode, node V_G has to be discharged to ground rapidly: transistors MP_1 and MN_5 take care of this. Assume switch transistor MN_8 is conducting at $t = 0$. From $t = 0$ until $t = t_2$ $V_{CP} < V_{CN}$, so node C_5 is at ground potential and MP_1 is not conducting. At $t = t_2$, transistor MN_8 is made nonconducting, without influencing the potential on C_5.

When the differential voltage of the master clock ($V_{CP} - V_{CN}$) becomes larger than the threshold voltage V_T of MP_1, MP_1 starts conducting and node C_5 will be charged to V_{CP}. This in turn will make MN_5 conducting, discharging node V_G rapidly and putting the circuit into hold mode. This is the only time-critical event in the T&H. Advantages of this solution are:

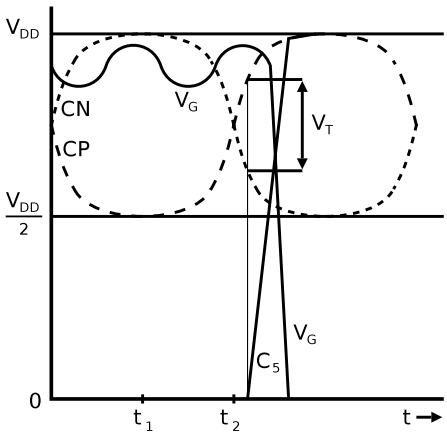

Fig. 4.15 Waveforms of the low-skew sample-switch driver

- Only MP_1 and MN_5 cause skew, so the complete "spread budget" can be spent in these transistors.
- Both differential clocks are used, so the effective slope is doubled, which halves the influence of the threshold voltage variations of MP_1.
- By minimizing the number of transistors between the clock input of the chip and the sample switch, jitter is minimized as well.
- By cascading MP_1 and MN_5, the gain from the CML clock to the gate of the sample-switch is maximized, making the sample action closer to ideal.

By multiplying the (simulated) switching slopes by the $\sigma(\Delta V_T)$ of the respective transistors, the expected timing misalignment is calculated to be 0.45 ps RMS.

In a time-interleaved T&H, the channels sample one after the other, with a delay of one clock-period (see Fig. 2.13 on p. 14). Within each period, only one of the channels should switch into hold mode. This is accomplished by applying so called clock-gating to the circuit. The additional transmission gate (NMOST and PMOST in parallel) and a PMOST are added to the low-skew switch-driver circuit as shown in Fig. 4.16.

At the rising edges of the master clock ($V_{CP} - V_{CN}$), 15 out of 16 times the circuit should just stay in hold-mode. In these cases the TM signal is low, such that the gate of MP_1 is at V_{DD} potential and MP_1 does not conduct, so the circuit stays inactive.

In the case the T&H should sample the input signal, TM becomes active high when V_{CN} is close to V_{DD}. MP_2 turns off and clock-signal V_{CN} is connected to the gate of MP_1. For the rest, the operation is as described above: when ($V_{CP} - V_{CN}$) becomes larger than V_T of the MP_1, it starts conducting, and causes the T&H to enter hold-mode.

The addition of this clock gating does not influence the performance. Note that the load for clocks CP and CN is not symmetrical and causes some imbalance. However, this does not impact timing alignment.

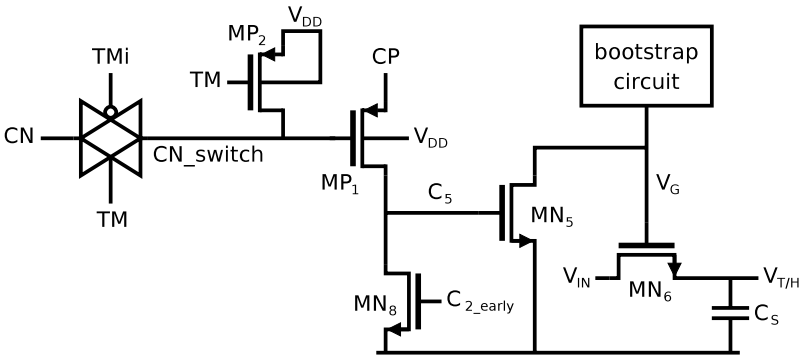

Fig. 4.16 Schematic of the low-skew sample-switch driver including clock-gating to implement the 1-out-of-16 sample scheme

4.3.4 Clock Generation for the T&H

The T&H requires quite some clock signals: the bootstrap circuit requires clocks C_1–C_4 and the low-skew switch-driver requires clock-signals TM, TMi and $C_{2,\text{early}}$ to make C_5. All these signals must be full-swing CMOS signals.

The circuit of Fig. 4.17 generates those clock-signals. All logic before the CM2SE[10] blocks is Current Mode Logic (CML) to limit noise generation in the sensitive T&H circuit. All flipflops and latches (noted by L) are connected to the CML clock. The waveforms are shown in Fig. 4.18. The incoming signal D, is a 1-out-of-16 signal and all other signals are derived from this signal and the CML clock.

Signal A is a delayed version of signal D, and serves as D signal for the next channel. It is also converted to CMOS swings to make $C_{2,\text{early}}$, which is used to control MN_8 in the T&H core (see Fig. 4.16).

C_2 is a delayed version of $C_{2,\text{early}}$ and C_4 is the inversion of C_2. When C_2 is high, the bootstrap capacitor is connected between the gate and source of the sample-switch. C_2 needs to become inactive before the sample-moment.

C_3 is made by the logical AND of F and $\overline{A}$, delaying it by one clock period, and converting it to CMOS levels. C_1 is the inverse of C_3. C_1 and C_3 determine when the bootstrap capacitor is connected between the supply rails. The AND-gate is added to allow the circuit to operate with a x-out-of-16 signal, with x the number of tracking periods. In this design, $x = 1$ is used, but the circuit also allows for higher values of x.

Signal I is made by inserting signal A into a cascade of three latches, so it changes on the falling edge of the clock. TMi and TM are generated from this signal.

Chronologically, the following happens, see Fig. 4.19: First $C_{2,\text{early}}$ becomes active high and activates switch MN_8 (Fig. 4.16), such that node C_5 is discharged.

[10]This block implements the conversion from Current-Mode Logic to full-swing Single-Ended CMOS signalling.

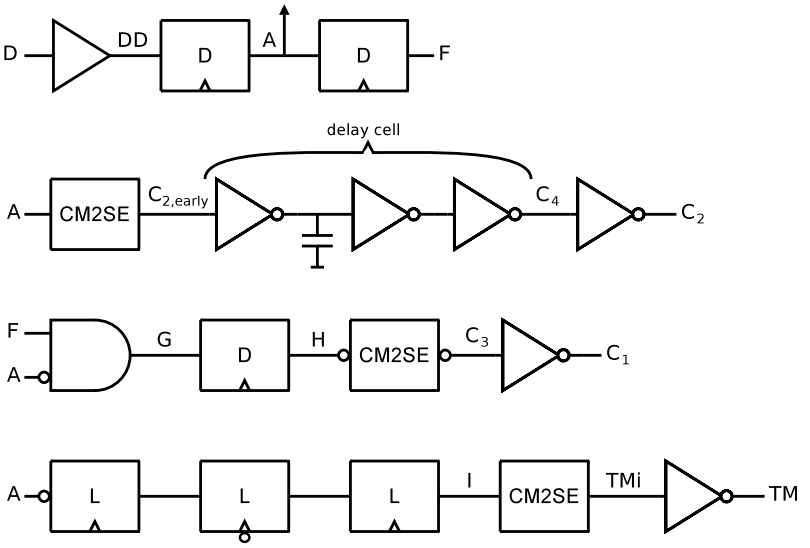

Fig. 4.17 Diagram of the clock generator

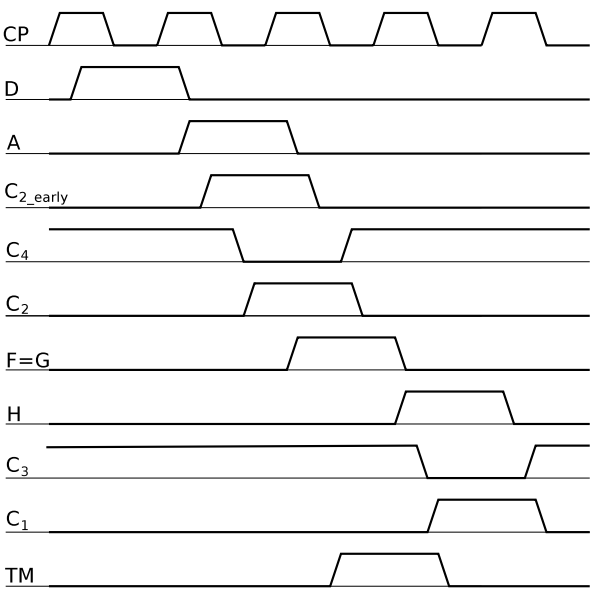

Fig. 4.18 Waveforms of the clock generator for the T&H

Next, C_2 and C_4 become active, putting the T&H in track-mode (Fig. 4.13). $C_{2,\text{early}}$, C_2 and C_4 become inactive before the sample moment. TM becomes active while CN is around V_{DD} and the T&H changes to hold-mode when $V_{\text{CP}} - V_{\text{CN}} > V_{\text{T,MP}_1}$. Finally C_1 becomes active and inactive again, such that the bootstrap capacitor is charged for the next cycle.

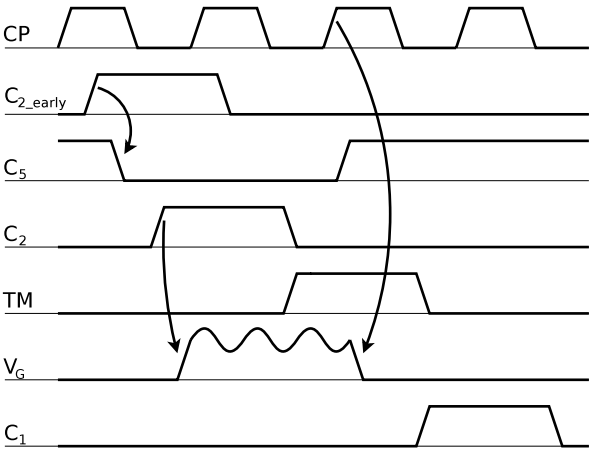

Fig. 4.19 Important signals controlling the T&H core

When a channel is put in track-mode, a sample capacitor charged with a previous sample is connected to the central input node, causing a small distortion at this node. Therefore, the track-time is just shorter than two periods, such that a sample is taken just before a next channel enters track-mode. This is a break-before-make switching scheme.

4.3.5 Buffer

The T&H includes two buffers, see Fig. 4.1 on p. 72. The output of the first buffer is connected to the switched-capacitor multiplying DAC (MDAC), that multiplies the residue signal by 16, and has an input capacitance of 600 fF. Errors in the buffer output directly appear in the ADC output, and therefore it should at least be as accurate as the converter (∼10 bits). The common-mode level of this output is not critical, as the MDAC suppresses common-mode signals.

The output of the second buffer is connected to SA-ADC1, which has a resolution of 6 bits. Therefore, the second buffer needs a relatively low accuracy of 6 bits as well. However, the common-mode level is important here, since the input window of the SA-ADC is limited.

Non-linear input capacitance at the input node causes distortion. To minimize this capacitance, the buffers are cascaded, see Fig. 4.20. The first buffer, described in detail in Sect. 2.4 on p. 22, is a PMOST source-follower (MP_1–MP_3) with an additional NMOST switch source follower (MN_1 and MN_2) aiming to keep the V_{DS} of MP_3 constant to improve linearity.

The relatively large capacitive load of the MDAC (∼600 fF) is disconnected from the buffer in track-mode. This increases the buffer bandwidth and avoids distortion without increasing the power consumption. A detailed explanation is given in Sect. 2.4.3 on p. 26.

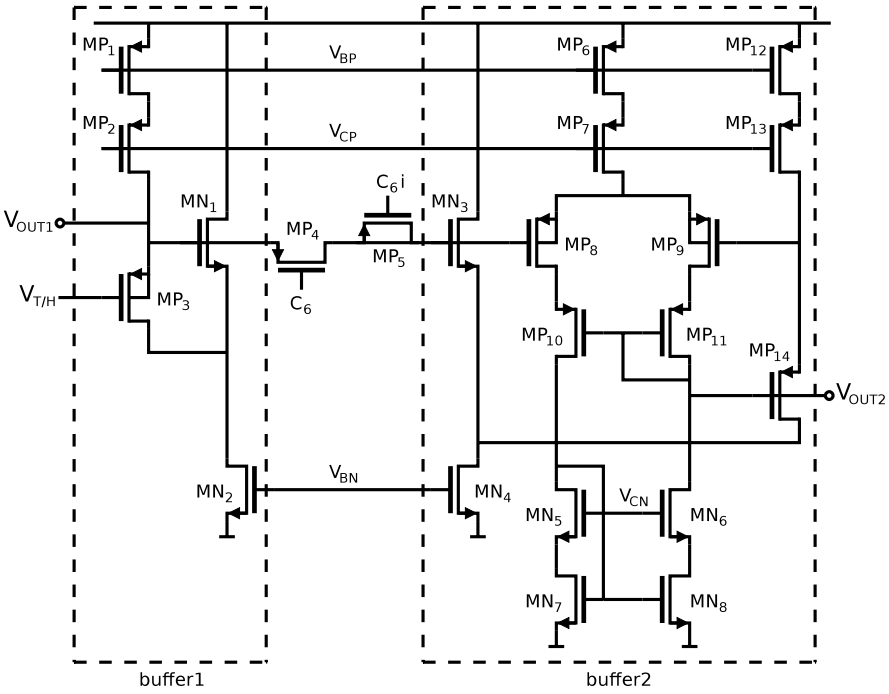

Fig. 4.20 Schematic of the T&H buffer consisting of a cascade of two buffers

The second buffer consists of MP_6–MP_{14} and MN_3–MN_8, and its goal is to replicate the input voltage of the first buffer $V_{T\&H}$. MP_{12}–MP_{14} is a down-scaled copy of the input source-follower (MP_1–MP_3). The drain of MP_{14} has the same potential as the drain of MP_3 as switch source follower MN_3–MN_4 is a replica of MN_1–MN_2.

The opamp consisting of MP_6–MP_{11} and MN_5–MN_8 aims to make the source potential of MP_{14} equal to that of MP_3. Now, the gate voltage of MP_{14} (V_{OUT2}) will be close to that of MP_3.

The moment the large capacitance of the interstage amplifier is connected to the output of the first buffer (V_{OUT1}), this node will show a large spike. To prevent that this affects the output of the second buffer, switch MP_4 is used to disconnect the second buffer. The parasitic capacitance at the input of the second buffer is used to store the signal value, and dummy MP_5 minimizes the pedestal step.

4.4 Sub-ADC

In this section the sub-ADC is described, that is used 16 times in the design. An overview is shown in Fig. 4.21. It consists of a first 6 bits SA-ADC (SA-ADC1), a digital-to-analog converter (DAC), an interstage amplifier with T&H function, and a second 6 bits SA-ADC (SA-ADC2).

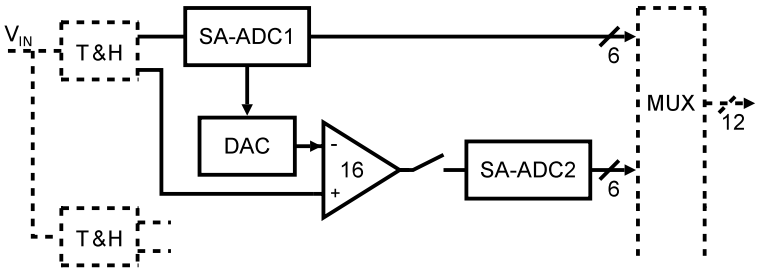

Fig. 4.21 Overview of the sub-ADC

Fig. 4.22 Due to the amplifier gain of 16, the bits from SA-ADC2 have to be shifted 4 positions to the right

SA-ADC1	b_5	b_4	b_3	b_2	b_1	b_0					
SA-ADC2					b_5	b_4	b_3	b_2	b_1	b_0	
OUTPUT	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	

To simplify debugging, the output data of the two SA-ADCs in a channel is made available off-chip without combining it into a 10 bits code. All signal processing is pseudo-differential, for simplicity however, single-ended schematics are usually shown.

The interstage amplifier has an implicit T&H, such that both SA-ADCs have a full period for their conversion.

In Chap. 3, the SA-ADC architecture was recommended for its high power efficiency. Due to an early design choice to use 16 channels, a sample-rate of 2 GS/s requires a sub-ADC sample-rate of 125 MS/s. During the SA-ADC design this did not appear to be feasible. To aim for the highest sample-rate nevertheless, the architecture described above was chosen as a compromise between power efficiency and sample-rate.

Compared to a single 10 bits SA-ADC, the requirements of the two 6 bits SA-ADCs are relaxed: the required accuracy is less and fewer steps are needed, resulting in more time per step. This enables higher sample-rates than achievable with a single 10 bits SA-ADC. Since the SA-ADC resolution is only 6 bits, the use of different comparators for different conversion steps (see Sect. 3.2.5) does not save a significant amount of power, and therefore it is not implemented.

The gain of the amplifier is 16, such that the data bits from SA-ADC2 have to be shifted $\log_2(16) = 4$ positions to the right in order to get the same weights for both ADCs. This is indicated in Fig. 4.22, and the resolution of the sub-ADC is thus 10 bits. The large overrange of 4 LSBs of SA-ADC1, relaxes the requirements on the interstage amplifier significantly, because after amplification by 16, the residue signal is nominally only a quarter of the full range.

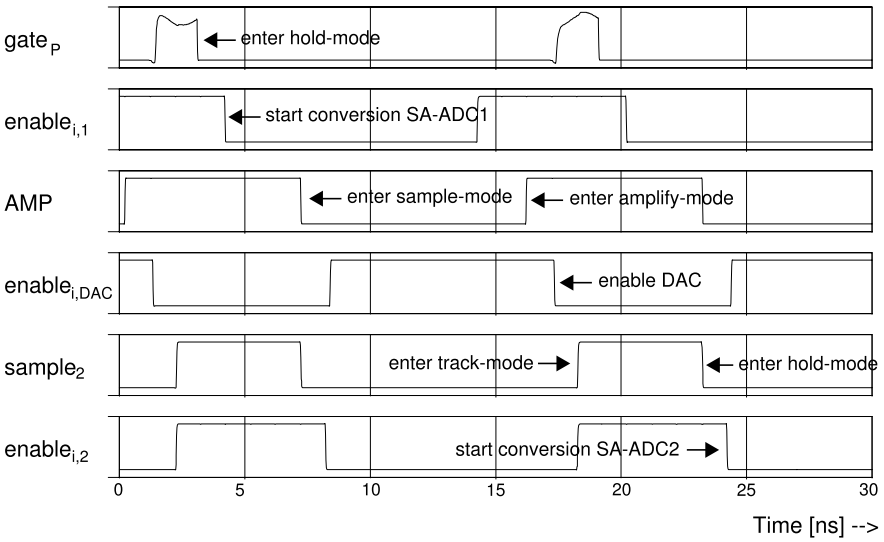

Fig. 4.23 Important timing signals of the sub-ADC, showing the order of operation

In pipeline converters with a high resolution, the number of bits in the first stage is increased to lower the requirements on the Multiplying DAC[11] (MDAC). In such converters usually a flash ADC is used, but its high input capacitance limits the resolution. In this design, an SA-ADC with a low input capacitance is used, and which allows using 6 bits in the first stage, significantly reducing the requirements on the MDAC. The implementation of the SA-ADC is described in Sect. 4.4.2.

The DAC is implemented as a resistor ladder with switches and this ladder is shared between all channels, to avoid differences between channels. It is described in Sect. 4.4.3. The amplifier is implemented as a switched-capacitor opamp and is described in Sect. 4.4.4. Before describing the implementation of the various blocks, the channel timing is described.

4.4.1 Channel Timing

Figure 4.23 shows the important timing signals and events of a sub-ADC channel. A conversion starts when the T&H enters track-mode, indicated by rising $gate_P$

[11] An MDAC is usually implemented as a switched-capacitor opamp configuration with two functions: (1) sample the residue signal of the previous stage, and (2) amplify the residue signal while subtracting the DAC signal.

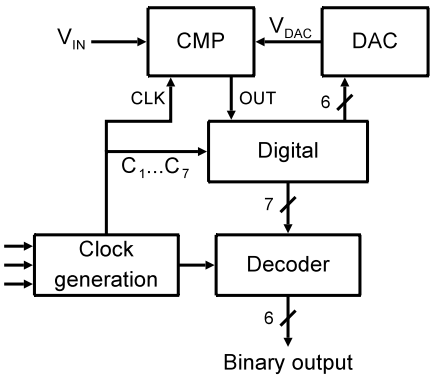

Fig. 4.24 Overview of the SA-ADC

signal.[12] After the T&H enters hold-mode, $enable_{i,1}$ becomes low and SA-ADC1 starts converting the input sample.

A moment later, *AMP* becomes low and the amplifier is put into sample-mode. It re-samples the input sample on the capacitors of the amplifier. After the conversion of SA-ADC1 is finished, the amplifier is put into amplify-mode (*AMP* is high) and when $enable_{i,DAC}$ becomes low, the DAC is enabled with the digital value found by SA-ADC1.

Next, the re-sampler after the amplifier enters track-mode ($sample_2$ goes high) and at the end of the amplify-phase, the re-sampler enters hold-mode, and SA-ADC2 starts converting the residue signal, indicted by the falling edge of $enable_{i,2}$. When SA-ADC2 is finished, the conversion of the input sample is complete. In the meantime, SA-ADC1 is already working on the next sample.

4.4.2 SA-ADC

This section describes the SA-ADC, see Fig. 4.24 for an overview. It uses the single-sided overrange technique described in Sect. 3.2.2 on p. 46. The input signal V_{IN} is a sampled signal and is compared to the initial DAC value. On basis of the comparator's decision, the digital control selects the next DAC value, the comparator makes a new decision, and so on. After seven steps the digital code is known and the seven comparator decisions are translated into a 6 bits binary code by the decoder. The clock-generation block times the other blocks. The various blocks in the SA-ADC will now be described.

[12]This signal is not digital, but it is the actual signal on the gate of one of the two sample switches. In track-mode, this signal follows the input signal with a DC shift as bootstrapping is used. In track-mode the signal is equal to ground.

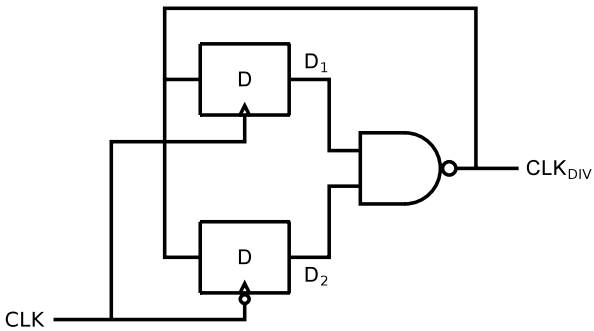

Fig. 4.25 Simplified schematic of the clock-divider with ratio 1.5

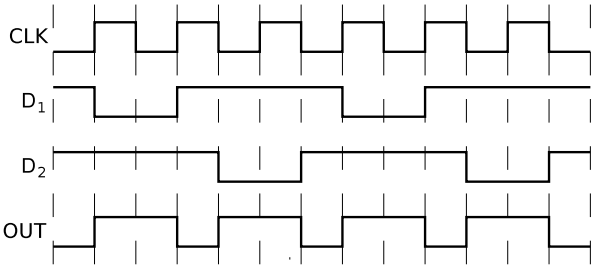

Fig. 4.26 Waveforms of the simplified clock-divider

Clock Generation

The clock generation block in Fig. 4.24 generates the clocks for the SA-ADC. The SA-ADC needs to perform 7 steps in 11 periods of the master-clock. To fully use the available time, a clock-divider with a ratio of 1.5 is required to perform the 7 steps in 10.5 clock periods. Moreover, the duty-cycle should be around one third, to give the comparator just enough time and maximize the DAC settling time.

A simplified schematic and the waveforms are shown in Figs. 4.25 and 4.26 respectively. The flipflops alternately set and reset the NAND gate.

The actual implementation is shown in Fig. 4.27. The upper part is the divider as in the previous figure, showing the simple implementation of the flipflops. The NOR gates are added to switch the circuit on and off when required.

The lower part synchronizes the enable signal with the clock and is used to start and stop the divider. The $\text{disable}_{\text{IN}}$ signal is delayed by one clock period and it serves as $\text{disable}_{\text{IN}}$ signal for the next channel. The accompanying waveforms are shown in Fig. 4.28; for this example the clock frequency is 1 GHz. As intended, 7 pulses are generated for one conversion.

To save some logic and power, the circuit relies on parasitic capacitance at the input of the inverters and NOR gates to operate. For the targeted range of sample-rates, the time between charge and discharge events is short enough, so that leakage does not cause problems. Gate delay causes some spikes, but this is not problematic.

The divider generates a 2-out-of-3 signal on S1 in sync with the rising edge of CPI, and another 2-out-of-3 signal on S2 which is in sync with the rising edge of

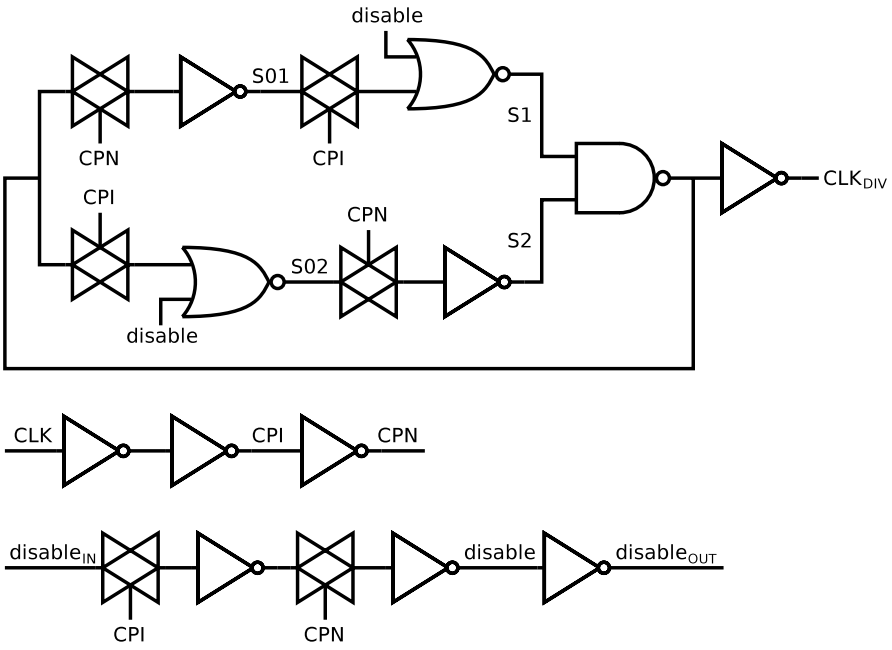

Fig. 4.27 Actual schematic of the clock-divider, with enable functionality

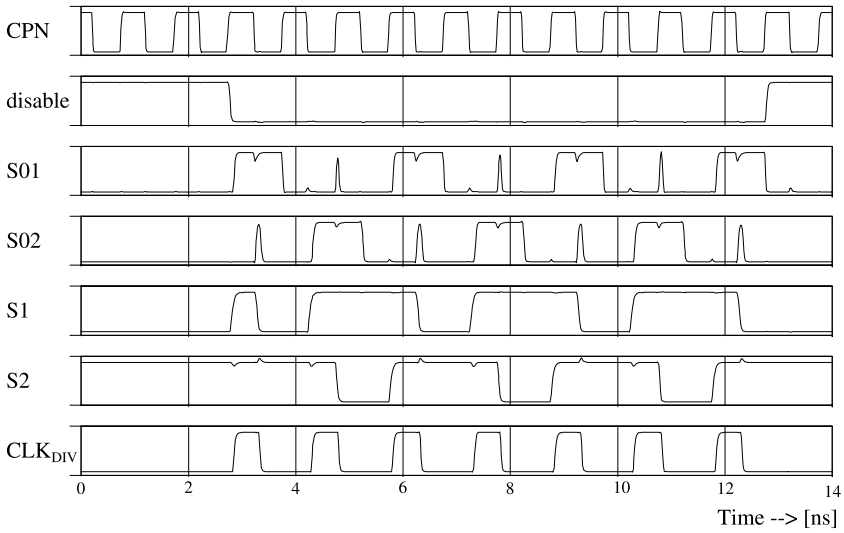

Fig. 4.28 Waveforms of the clock-divider

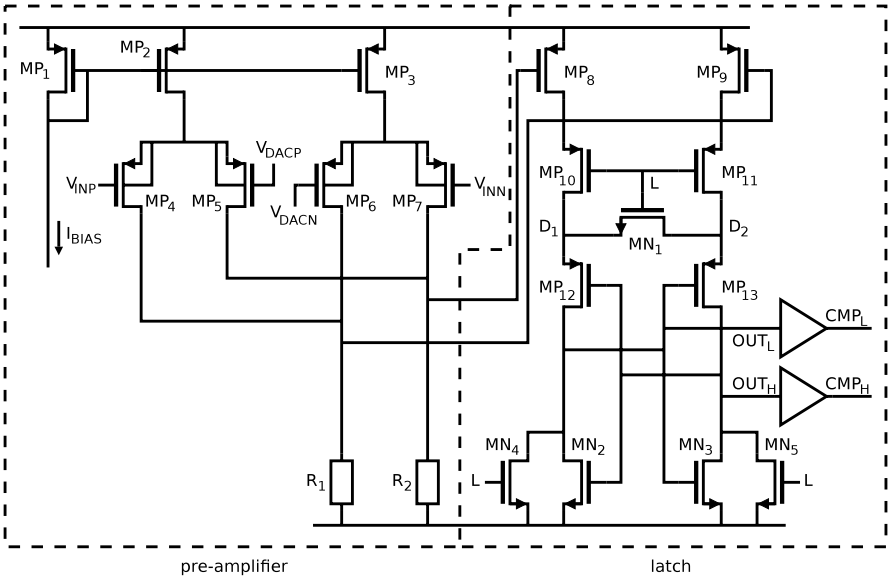

Fig. 4.29 Comparator schematic with pre-amplifier and latching stage

CPN. The phases of these signals are such that the AND of S1 and S2 results in the desired $^1/_2$-out-of-$1^1/_2$ signal.

The generated signal is used as comparator clock signal, and it is also used to clock a shift-register within the clock generation block. Together with an appropriate initialization signal, a '1' ripples through the shift-register indicating the current step number of the SA-ADC. The outputs are called $C_1 \cdots C_7$.

Comparator

The comparator (CMP) in the SA-ADC (Fig. 4.24) compares the differential input signal with the differential DAC signal. The clock-period of the SA-ADC is 1.5 times that of the ADC, so 750 ps at a sample-rate of 2 GS/s, and about $^1/_3$ of this time is reserved for the comparator. The required accuracy is about $^1/_2$ LSB at 6 bits level with a signal swing of 0.4 V.

The schematic of the comparator is shown in Fig. 4.29. It has a differential-difference pre-amplifier to subtract the positive DAC signal from the positive input signal and to subtract the negative DAC signal from the negative input signal, and a latch to perform the actual comparison.

The amplifier is loaded with resistors instead of current sources to minimize the capacitance at those nodes. Moreover, for a large difference between the input signal and the DAC signal, one of the branches becomes current-less and a current source would need a relatively long time to recover from this.

For critical decisions, when

$$V_{\mathrm{INP}} - V_{\mathrm{DACP}} \approx V_{\mathrm{INN}} - V_{\mathrm{DACN}} \tag{4.10}$$

the currents from MP_2 and MP_3 are equally split between R_1 and R_2, so that the common-mode level for the comparator latch is well defined. This is advantageous, since the speed [42], offset and noise of the comparator all depend on the input common-mode level.

To lower the input referred offset and noise of the comparator latch, a second function of the pre-amplifier is to provide gain. The bandwidth of the pre-amplifier needs to be quite large, since only a small amount of time is available for amplification. To maintain a reasonable power consumption, a gain of three was chosen.

Comparators or latch-type voltage sense-amplifiers are popular [36, 61, 63] because of their high input impedance, full-swing output and absence of static power consumption [42].

The used latching comparator stage is similar to others and is shown in the right part of Fig. 4.29. MP_8 and MP_9 are the input transistors of the comparator. In the reset phase, latch signal L is high and MP_{10} and MP_{11} are not conducting, such that the drains of the input transistors are charged to V_{DD}, and no current is drawn from the supply. Reset transistor MN_1 forces the differential signal between nodes D_1 and D_2 to clear the memory of the previous decision to minimize hysteresis. The regenerative stage consists of MP_{12}, MP_{13}, MN_2 and MN_3, and switches MN_4 and MN_5 reset the output nodes of this stage to ground.

When the latch signal L becomes low, the comparator is switched into latch-mode. The reset transistors MN_1, MN_4 and MN_5 are inactive and MP_{10} and MP_{11} are conducting. Nodes D_1 and D_2 start to rise and the differential input signal on the gates of MP_8 and MP_9 is integrated on these nodes. When these nodes reach a value of about one threshold voltage, the regenerative stage starts to operate. MP_{12} and MP_{13} become active first and when nodes D_1 and D_2 reach about two threshold voltages, also MN_2 and MN_3 become active. Finally, a full-swing signal develops at the comparator outputs.

The most important waveforms are shown in Fig. 4.30, for a small differential input signal. At $t = 0$, $\mathrm{OUT_H}$, $\mathrm{OUT_L}$, $\mathrm{CMP_H}$ and $\mathrm{CMP_L}$ are at ground level, and D_1 and D_2 are approximately one V_{T} above ground. After the latch signal L becomes low, nodes D_1 and D_2 start to rise and a differential voltage builds up between them. Since the input signal is small in this example, the differential voltage on the D-nodes is small as well. The nodes $\mathrm{OUT_H}$ and $\mathrm{OUT_L}$ also start to rise and due to regeneration, a large differential voltage is generated between these nodes.

Output buffers are used to increase the drive capability of the comparator. In the case of a very small input signal, both outputs of the regenerative stage $\mathrm{OUT_H}$ and $\mathrm{OUT_L}$ come close to the threshold voltage of an NMOST for a short period of time. This is a problem, since the logic in the system relies on the fact that only one of the two outputs becomes active. To prevent false positives, the buffers are dimensioned such that their trip-levels are increased sufficiently to prevent these false positives.

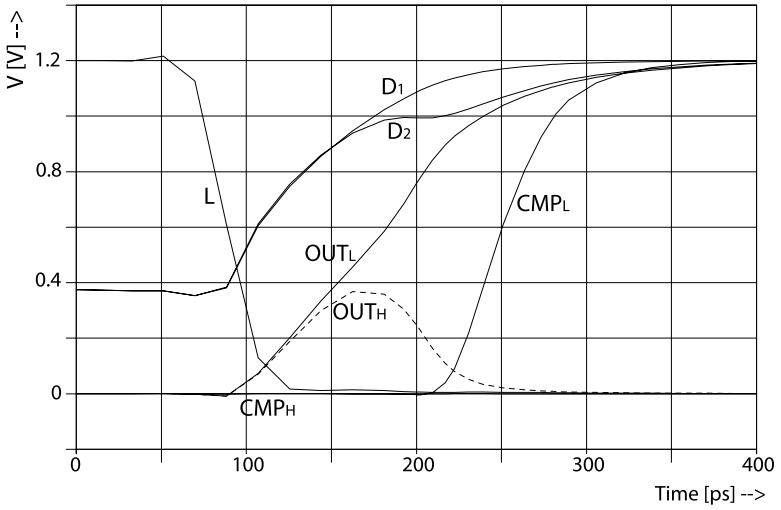

Fig. 4.30 Waveforms of the latching comparator stage for a small differential input-signal

Digital Control Logic Implementing the Single-sided Overrange Technique and the Look-ahead Functionality

The digital control block of Fig. 4.24 on p. 94 sets the DAC level based on the decisions of the comparator. It implements the energy-efficient single-sided overrange technique described in Sect. 3.2.2 on p. 46, and the look-ahead [22] functionality described in Sect. 3.2.4 on p. 53. Look-ahead logic minimizes the time between the comparator decision and the updating of the DAC, to increase the maximum sample-rate.

The digital control block consists of a register, a ladder control block, DAC logic and a decoder, see Fig. 4.31. The register is described first. It stores the decisions of the comparator and consists of seven cells like the one shown in Fig. 4.32. A cell has three inputs: $\mathrm{CMP_{DIG}}$ is the digital comparator signal, $\mathrm{CMP_{HL}}$ indicates if the comparator has made a decision, and $\mathrm{CLK_N}$ indicates the $\mathrm{N^{th}}$ step of the sub-ADC, which is used to activate the proper register cell. So, each step one of the $\mathrm{CLK_N}$ signals is active, and when the comparator makes its decision, it is directly clocked into the flipflop. Note that both $\mathrm{CMP_{HL}}$ and $\mathrm{CLK_N}$ are active low signals.

The register cell has two outputs: R_N and RR_N. The first signal goes to the ladder-control block and the latter signal is a buffered version and goes to the decoder. The logic on top of the figure is common to all cells.

The ladder-control block has the following inputs: The R_N signals from the register, the C_N signals from the clock generator, a reset signal and the comparator signals. The outputs (named $\mathrm{LEFT_N}$ and $\mathrm{RIGHT_N}$) are controls for the DAC with integrated logic.

The controller is a binary tree built up of blocks with a function as indicated in Table 4.1: If the PREV input is logical 0, all three outputs are 0 and the branch is

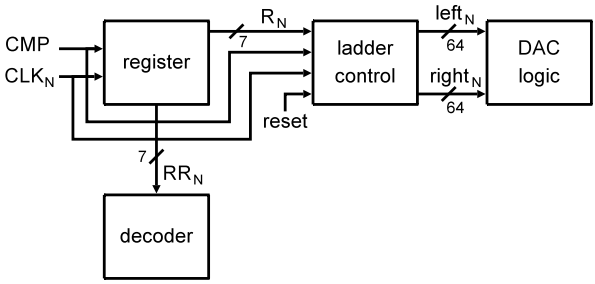

Fig. 4.31 Overview of the digital control block of the SA-ADC

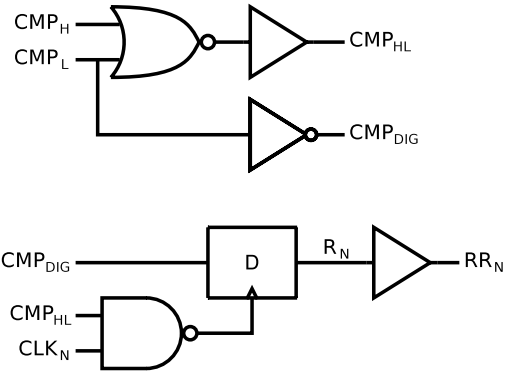

Fig. 4.32 Schematic of the common register-logic (*top*) and one cell of the register (*bottom*)

Table 4.1 Function of a cell of the binary tree

PREV	R_N	*CLK*	NEW_0	NEW_1	NEW_{CLK}
0	–	–	0	0	0
1	0	0	1	0	0
1	1	0	0	1	0
1	0	1	1	0	1
1	1	1	0	1	1

dead. If the PREV input is logical 1, the 1 is propagated to either NEW_0 or NEW_1, depending on R_N. The additional output NEW_{CLK} is active when both PREV and CLK are 1.

Part of the binary tree is shown in Fig. 4.33. It is in fact a binary-to-pointer decoder, with additional CMP_{NNN} outputs. These outputs indicate the finalized comparator decisions. For example, CMP indicates that no comparator decisions have been made yet, and that the logic is waiting for the first one; CMP_0 means the first comparator decision was 0 and the logic is waiting for the second decision, CMP_{10} means the first comparator decision was 1, the second was 0 and the logic is waiting for the third decision, and so on.

Each CMP_{NNN} signal (with a variable number of N's) is connected to one of the 64 $LEFT_N$ signals and to one of the 64 $RIGHT_N$ signals to implement the single-

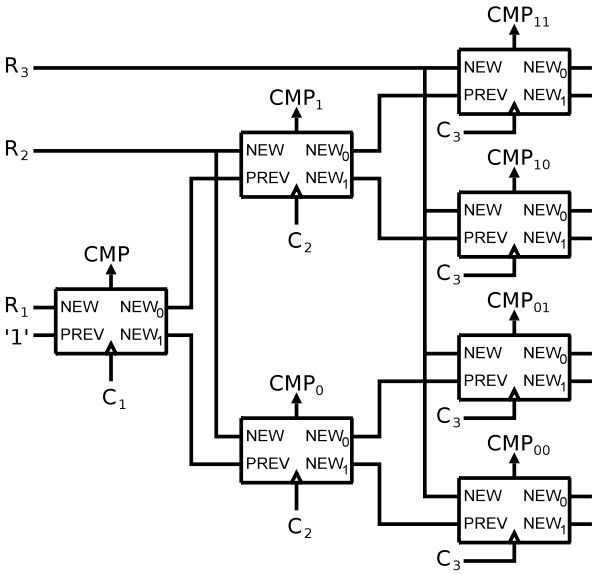

Fig. 4.33 Part of the binary tree implementing a binary to pointer decoder, with additional CMP_{NNN} outputs to control the DAC

sided overrange technique. The number of connections to a $LEFT_N$ or $RIGHT_N$ signal varies between zero and four, therefore NOR-gates are used to enable multiple inputs.

The reset signal is used to set the initial DAC value by activating the proper $LEFT_N$ and $RIGHT_N$ signals.

Each tap of the resistor-ladder DAC is surrounded by some logic, the schematic is shown in Fig. 4.34. The $LEFT_N$ and $RIGHT_N$ signals from the ladder control block are connected to the CONi signals. The comparator signals CMP_H and CMP_L are the other inputs of this logic and the resistor is part of the resistor-ladder DAC.

The purpose of this block is to connect the ladder-tap to either the positive DAC output, or to the negative output, or to leave it floating, as the two DAC outputs should be connected to only one of the ladder taps at a time. This blocks combines the functionality of the look-ahead logic and the DAC.

The DAC switches MN_2 and MN_4 can connect the ladder tap to the positive, respectively the negative DAC output. If both CON_{H1} and CON_{L1} are inactive (logical 1 in this case), switch MN_1 is active and the DAC switch MN_2 is disabled. If either CON_{H1} or CON_{L1} is active (logical 0), the transmission gate connects the corresponding comparator output to the DAC switch. The same holds for the negative part of the DAC.

The look-ahead function works as follows: Initially, both comparator outputs CMP_H and CMP_L are inactive and before each comparator decision, the transmission gates are set such, that one of the switches (e.g. MN_2) between the ladder and the positive DAC output V_{DACP} is controlled by CMP_H and another switch between the ladder and the positive DAC output is controlled by CMP_L. The same holds for the N-side of the DAC. Now, the moment the comparator makes a decision, either one of the comparator outputs CMP_H or CMP_L becomes active and directly

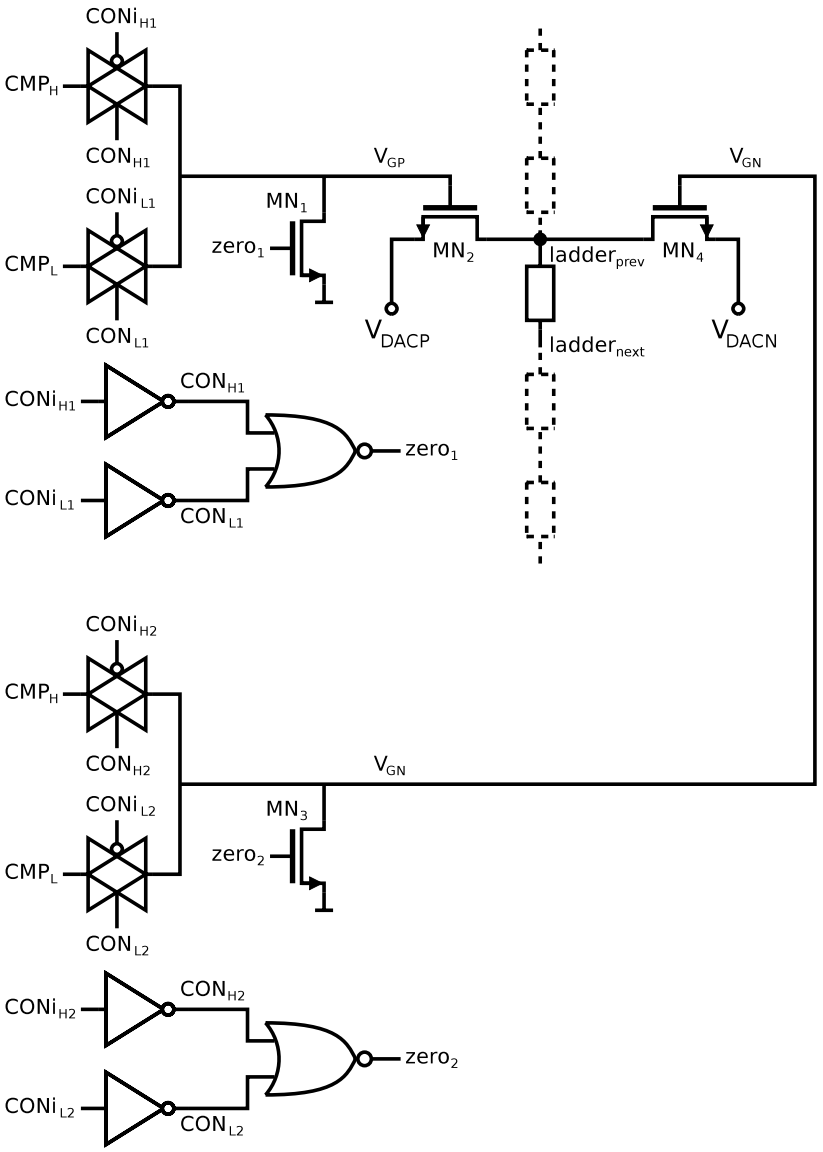

Fig. 4.34 Logic connected to each tap of the ladder for controlling the DAC. This is part of the look-ahead logic

activates the proper DAC switches; one connected to V_{DACP} and one connected to V_{DACN}. The differential DAC starts settling towards its new value with almost no delay with regard to the moment the comparator took its decision. This increases the maximum sample-rate of the SA-ADC.

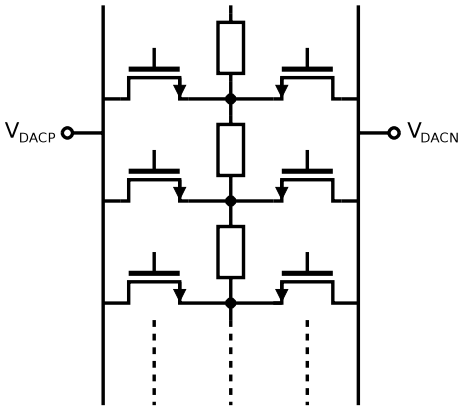

Fig. 4.35 Schematic of the differential resistor-ladder DAC

Before each comparator decision, the two possible next DAC values are calculated in advance, therefore this is called look-ahead logic.

DAC of the SA-ADC

The DAC used in the SA-ADC is a resistor-ladder DAC, as shown in Fig. 4.35. The ladder taps serve as reference voltages and on both sides of the differential DAC, only one switch is conducting. When the DAC is fully settled, the resistance of the switches does not affect the DAC voltage, as the DAC load is capacitive.

The resolution of the DAC is 6 bits, so normally 64 switches would be required on each side. Each switch has parasitic capacitance and the sum of these is significant and the RC-time limits the maximum sample-rate. To decrease the RC-time, the resistor ladder can be scaled-up (smaller resistors, more current) but this would increase the power consumption.

Here, this parasitic capacitance is limited by halving the number of switches on each side of the DAC and allowing a small common-mode increase of $^1/_2$ LSB for odd codes $(1, 3, 5, \ldots)$. The conventional configuration is shown in Fig. 4.36 on the left-hand side. Signed codes are used to indicate the levels, and 0 is the middle code. The numbers indicate which two switches are conducting for a given code. If the DAC code is increased by one, the P-side is increased by one step and the N-side decreased by one step, so the differential voltage increases with two steps.

The new configuration is shown on the right-hand side of the same figure. The numbers on the bottom and the top are the same,[13] so that both configurations have the same range $(-4 \ldots +4)$. If the DAC code is increased by one, alternatingly the P-side is increased by two steps, or the N-side is decreased by two steps. In both cases the differential voltage increases with two steps, which is in accordance with

[13]The -5 and $+5$ at the bottom can be neglected; to select a certain code, the code must be selected on both sides of the ladder (left and right). Since the second -5 and $+5$ are not within the indicated range, they cannot be selected.

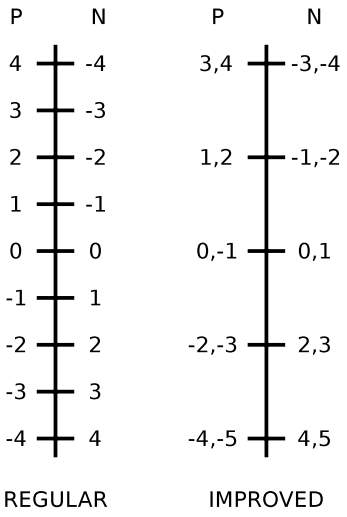

Fig. 4.36 DAC configurations: regular (*left-side*) and improved using only half the number of taps (*right-side*)

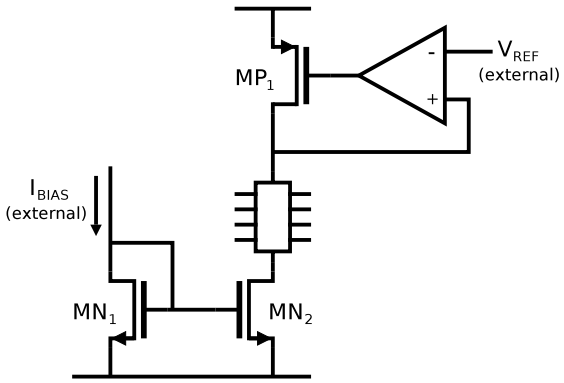

Fig. 4.37 Bias circuit for the resistor-ladder

the conventional configuration. Since only half the number of switches and ladder taps are used, the time-constant is roughly halved.

To bias the resistor-ladder, the circuit of Fig. 4.37 is used. The current is controlled by current source MN_2 and is derived from an externally applied bias current. The current and the resistance of the ladder determine the full-scale voltage of the DAC.

The common-mode voltage of the ladder is controlled by MP_1, which is controlled by the opamp. The opamp aims to make the voltage at the top of the ladder equal to the externally applied V_{REF}. The common-mode level of the ladder is not critical, since the pre-amplifier of the comparator suppresses common-mode signals.

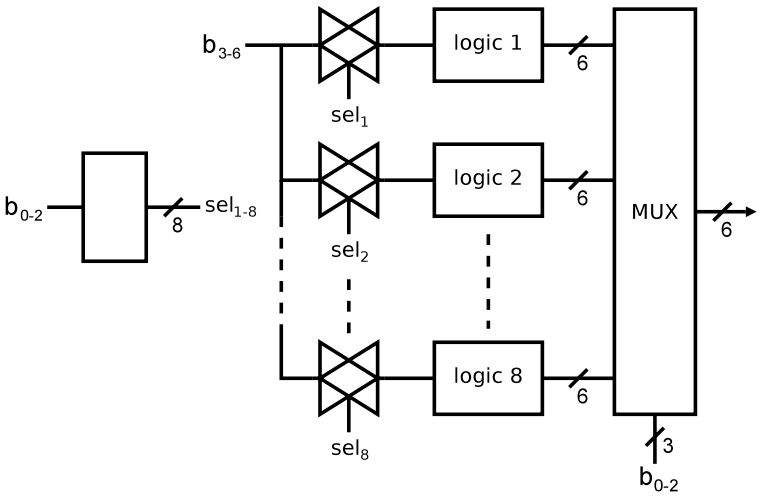

Fig. 4.38 Overview of the decoder logic for converting the 7 comparator decisions into a 6 bits binary code

Decoder

The decoder converts the seven comparator decisions into a 6 bits binary code. It is important that the delay of the decoder is small, since it is part of the SA-ADC loop and limits the maximum sample-rate.

An overview of the decoder schematic is shown in Fig. 4.38. The first three comparator decisions (b_0–b_2) are converted into a 1-out-of-8 signal (sel_1–sel_8). After the 7^{th} comparator decision is made, the remaining 4 decisions (b_3–b_6) are connected to one of the eight logic blocks, based on b_0–b_2. The active logic block generates the 6 bits binary code and the proper output is selected by a MUX, which is already set after the third comparator decision.

Advantages of this implementation are that the delay is small, and since only the necessary logic is activated, the power consumption is low.

4.4.3 DAC of the Sub-ADC

The sub-ADC DAC drives the switched-capacitor interstage amplifier, see Fig. 4.21 on p. 92. The DAC signal is subtracted from the input signal, before the amplification is performed. So, errors made in the DAC directly appear in the output signal, and a DAC accuracy of 10 bits is required. The number of levels is however only 64, since the resolution of SA-ADC1 is 6 bits.

The DAC is a resistor-ladder type DAC and the same technique as described in Sect. 4.4.2 on p. 103 is used to halve the number of ladder taps and switches. Also the biasing of the ladder is identical to the one described in that section.

The impedance of this ladder needs to be low, since it needs to drive the switched capacitor load with sufficient speed. The parasitics of the DAC switches are therefore large. To lower the parasitic capacitance at the DAC output, four additional switches in series with the DAC switches are used, and each connected to 8 DAC switches. Since only one of the four additional switches is conducting at a time, the capacitive load of the DAC is roughly decreased by a factor four.

Connections Between DAC Ladders

Each channel needs a resistor ladder DAC. The ladder-taps of all these ladders could be connected together, since they have the same potential. An advantage of this is that mismatch between ladders is completely avoided and that the matching of the resistance between taps is improved, as the total area of the resistors is increased by a factor of 16. The impedance at each tap is also 16 times lower.

By connecting the taps however, the requirements change. If the ladders are not connected together the following requirement holds: At the beginning of the amplify phase of the interstage amplifier, the DAC is connected to the amplifier, which causes a step in the voltage on the ladder-taps. Before the end of the amplify phase (8 clock periods later), the voltage on the ladder taps should return to a value within $^1/_4$ LSB of its nominal value.

If the ladders are now connected together, each clock-period one of the channels enters the amplify mode, so each clock period a ladder tap is connected to the amplifier causing a step on the ladder taps. Also, each clock-period, one of the channels is at the end of the amplify-mode, and the DAC value needs to be accurate. Therefore, the resistor taps should recover in only one clock period.

Since the impedance of the connected ladder is 16 times lower, while the settling time requirement is 8 times higher, it is attractive to connect the ladders together. Although there are more factors that favor one or the other option (e.g. cross-talk between channels), the ladders are connected together. To limit the number of wires only 8 of the 32 taps are connected. Compared to fully connected ladders, the differences are only minor.

Binary to 1-out-of-32 Decoder

The 6 bits DAC in the SA-ADC uses the configuration with half the number of taps as shown in Fig. 4.36 on p. 104. To set the binary output code of SA-ADC1 on this DAC, a conversion is needed. The five most significant bits are converted into a 1-out-of-32 signal, suitable for the N-side of the DAC. Adding the LSB information yields another 1-out-of-32 signal, which is suitable for the P-side of the DAC.

The 1-out-of-32 decoder is built up with cells shown in Fig. 4.39. These cells are connected to each other as indicated in Fig. 4.40. Each row forms a NAND gate, with three inputs in this example: the NMOSTs are connected in series between ground and the output and the PMOSTs are connected in parallel from V_{DD} to the

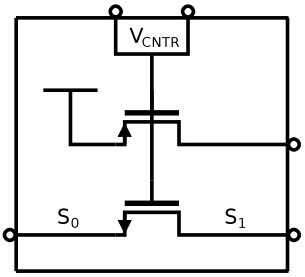

Fig. 4.39 Cell of the binary to 1-out-of-32 decoder

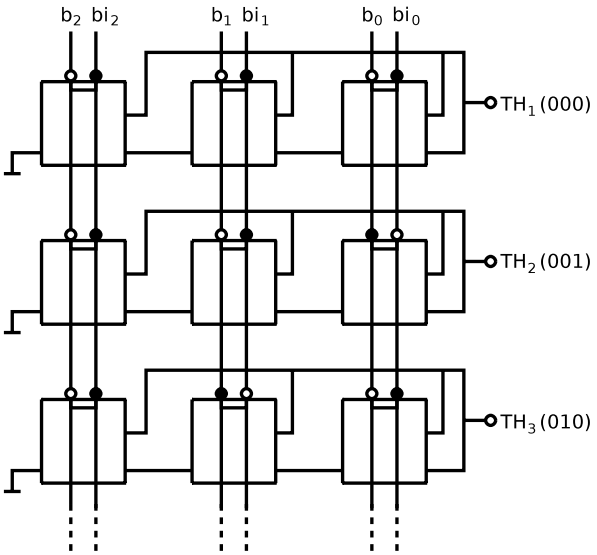

Fig. 4.40 Binary to 1-out-of-32 decoder

output. If one of the inputs is '0', the output is '1' (inactive) and for all inputs equal to '1' the output is '0' (active).

The inputs of the cells (V_{CNTR}) are connected such, that all binary combinations are present in the appropriate order. In the figure, a closed dot means connected, and an open dot means not connected. The inputs are named b_2, b_1 and b_0 and the inverted inputs are named bi_2, bi_1 and bi_0. bi_2, bi_1 and bi_0 are connected to the first NAND, such that a binary value of '000' is converted to '0' on TH_1; bi_2, bi_1 and b_0 are connected to the second NAND, such that a binary '001' is converted to '0' on TH_2; and so on. The output is now the desired 1-out-of-32 signal, suitable for the N-side of the DAC.

To make the signal for the P-side of the DAC and to implement an enable function, some additional logic is needed, which is shown in Fig. 4.41. The TH_N signals are the 1-out-of-32 signals and for the N-side this can directly be combined with the ENABLE signal. For the P-side, the LSB information is added to determine whether the current P_N or that of the next block should be activated.

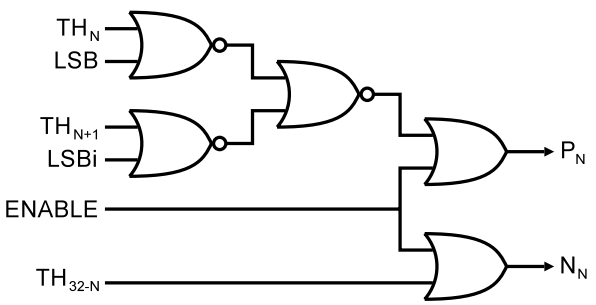

Fig. 4.41 Additional logic to make the 1-out-of-32 signal suitable for the DAC with half the number of taps

The enable function is needed, because in sample-mode of the amplifier, the DAC signal should not be connected to the opamp.

This specific implementation of the 1-out-of-32 decoder is mainly chosen to enable a compact layout for minimal parasitic capacitance and maximum speed.

4.4.4 Interstage Amplifier

The interstage amplifier is implemented as a two-stage opamp with a switched-capacitor network. It has two modes: sample-mode and amplify-mode, which are shown in Fig. 4.42. In sample-mode, the input signal is sampled on the capacitors with respect to the common-mode voltage. In amplify-mode, the connection of the feedback capacitors (value 1 C) causes an output voltage of V_{IN}, and the transfer of the charge from the 15 C capacitors to the 1 C feedback capacitors adds another $15{\cdot}V_{\mathrm{IN}}$. In amplify-mode, the gain from the DAC nodes to the output is 15, so the output voltage in amplify-mode is:

$$V_{\mathrm{OUT,AMPL}} = 16 \cdot V_{\mathrm{IN}} - 15 \cdot V_{\mathrm{DAC}} \tag{4.11}$$

where V_{IN} is the differential input signal and V_{DAC} is the differential DAC signal. Advantages of this well known configuration are that:

- The gain is only determined by the capacitor ratio, which is accurate in modern CMOS processes.
- The input referred offset of the amplifier does not affect the output voltage.
- Parasitic capacitance at the input-node and output-node of the opamp does not affect the transfer function.

Errors in the sampling-action or the subtraction, directly corrupt the signal, so an accuracy of 10 bits is needed. Therefore, the open-loop gain of the amplifier should be at least 65 dB. As the amplifier-stage has a gain of 16, the required accuracy at the output is only 6 bits. The available time for the two phases is almost half the sample period of the sub-ADC, so 4 nS at a sample-rate of 2 GS/s. The unity capacitor is chosen 25 fF as a trade-off between noise, accuracy and power consumption.

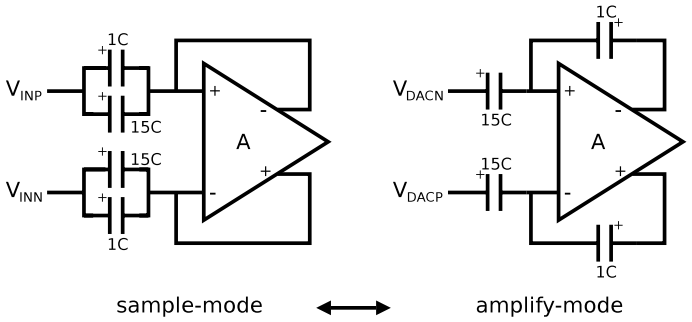

Fig. 4.42 The two operation modes of the interstage amplifier

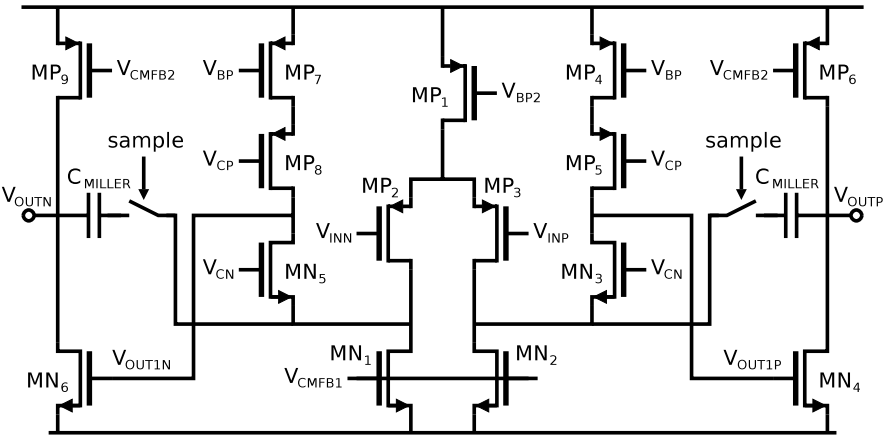

Fig. 4.43 Schematic of the two-stage opamp with frequency-compensation capacitor only connected in sample-mode

Frequency compensation is used to guarantee stability. In sample-mode the feedback factor is 1, requiring a large Miller capacitor for stability. In amplify-mode the feedback factor is only 1/16 and no frequency compensation is needed, but the required gain-bandwidth product is 16 times larger for the same time-constant. This conflict between bandwidth and stability is simply resolved by connecting the frequency compensation capacitor only in sample-mode.

For fast settling, cascode or Ahuja compensation [3] is used, so the frequency compensation capacitor is connected below the cascode transistor, instead of above the cascode as in regular Miller frequency compensation. This kind of compensation creates an inherently higher bandwidth, three pole system [1]. The schematic of the opamp is shown in Fig. 4.43.

To stabilize the common-mode signal, switched-capacitor common-mode feedback (CMFB) is added to both stages. Now, the CMFB circuit for the second stage is described, but for the first stage the same circuit is used. The schematic is shown in Fig. 4.44. V_{OUTN} and V_{OUTP} are the differential outputs of the second stage, V_{CMFB2}

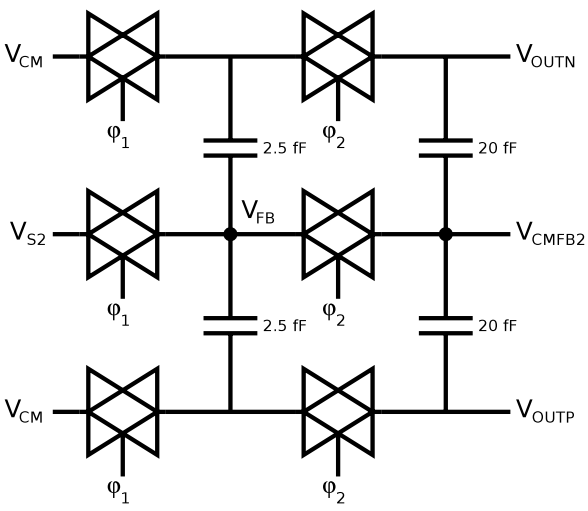

Fig. 4.44 Switched capacitor common-mode feedback (CMFB) circuit, which is used around both stages of the amplifier

is the common-mode feedback voltage for the second stage (connected to MP_6 and MP_9), V_{CM} is the desired common-mode output voltage and V_{S2} is the nominal node voltage of V_{CMFB2}. V_{S2} has such a value that if V_{CMFB2} would have the same value and mismatch and input common-mode variation would be absent, the output common-mode voltage would be at the desired level. The two capacitors on the right-hand side arrange common-mode feedback for non-DC frequencies. The transfer function of the common-mode output signal to the common-mode feedback point is close to one, while the differential gain is negligible.

A major drawback of conventional common-mode feedback circuits using active circuitry is the phase-shift between the common-mode output signal and the feedback signal. Using the above implementation, there is (almost) no phase-shift up to very high frequencies, which is a very desirable characteristic for these kinds of applications.

The output node (V_{CMFB2}) is only connected to capacitors and therefore it needs to be biased. The rest of the circuitry arranges this bias voltage: During the first phase ($\varphi_1 = 1$), the small capacitors on the left side are charged to the difference between the desired common-mode voltage and the nominal value of V_{CMFB2} present on V_{S2}. In the second phase ($\varphi_2 = 1$), the small capacitors are connected in parallel to the large ones on the right. charge redistribution causes node V_{CMFB2} to slowly reach the desired value of V_{S2}.

This can be understood as follows: suppose the switches connected to φ_1 are opened, and the small capacitors are charged. Next, the upper and lower switches connected to φ_2 are closed. The voltage in between the two small capacitors V_{FB} is a function of the output voltages V_{OUTP} and V_{OUTN}. If the output voltages are equal to V_{CM}, V_{FB} is equal to V_{S2} and when the middle switch is closed, nothing would happen as V_{CMFB2} is already at the desired value.

If V_{CMFB2} is too low and consequently the common-mode of the output voltage is higher than the desired V_{CM}, V_{FB} gets a value larger than V_{S2}. Now, when closing

the middle switch, charge redistribution causes the voltage on V_{CMFB2} to rise, and the common-mode voltage of the output to drop. This also works when a differential signal is present, as a differential signal does not affect the potential of V_{FB}. Care has to be taken of the stability of the CMFB network, otherwise it could show ringing or even oscillation.

Apart from the advantages already mentioned, the power consumption of the circuit is negligible and it has a very large input range. Disadvantages are the increased capacitive load and the practical aspect that simulation of the circuit is more complex compared to circuits with continuous-time common-mode feedback.

4.4.5 Re-sampler

When the output signal of the interstage amplifier is fully settled, there is only a short time left before the amplifier needs to sample the next input sample. To give SA-ADC2 sufficient time for its conversion, a re-sampler or T&H is needed. This circuit should sample the output of the amplifier and hold the signal for the rest of the period, so SA-ADC2 can perform its conversion.

The simplest implementation of such a circuit consists of a switch and a capacitor. The re-sampler is however loaded with a comparator: The output of the re-sampler is connected to one side of the differential pair of the comparator preamplifier, while the DAC is connected to the other side of the differential pair, see the top part of Fig. 4.45. For simplicity, a single-ended system is used here, but the principle is the same for a differential system.

The settled residue signal is sampled on C_{S} by opening switch S_{RS}. At the sample moment, the DAC is at its initial value of 9/16 of the full range. During the conversion of SA-ADC2, the DAC value moves towards the value of the re-sampled residue signal, and at the end of the conversion it should be within $^1/_2$ LSB of the residue signal. Due to feed-through via the parasitic capacitors C_{P}, a change in the DAC voltage results in a change of the sampled value on C_{S}, as shown on the bottom part of Fig. 4.45. Feed-through distorts the sampled value as indicated by the left arrow, and causes a wrong decision of the comparator, as indicated by the right arrow.

The amount of charge redistribution is a function of the difference between the initial DAC voltage and the final DAC voltage, and therefore it is a function of the input voltage. This causes offset and gain errors, and if the parasitic capacitance is nonlinear, distortion as well. Note that kT/C noise is not a big issue here, since the required resolution is only 6 bits.

To avoid these problems, sample-capacitor C_{S} could be made very large compared to C_{P}. This would have large implications for the amplifier driving it and would increase the power consumption significantly.

Another solution is to include a buffer to drive the comparator, similar as in the frontend T&H. Apart from additional power consumption, mismatch in gain and offset could be an issue.

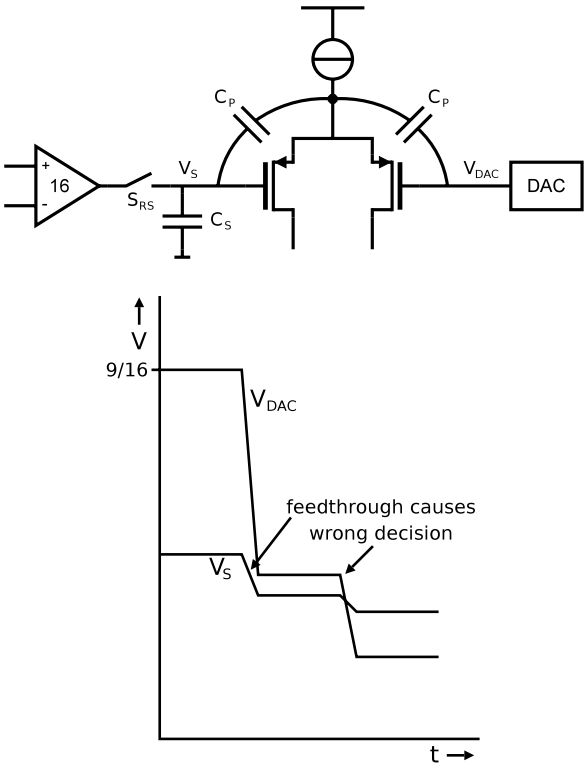

Fig. 4.45 Straightforward implementation of the re-sampler loaded with the pre-amplifier of the comparator (*top*) and corresponding waveforms (*bottom*). Feed-through at the *left arrow* causes a wrong decision of the comparator indicated by the *right arrow*

A much simpler solution is introduced here. In fact it only consists of two switches. One switch is connected between the inputs of the comparator pre-amplifier, and one switch is connected between the pre-amplifier and the DAC. In the top part of Fig. 4.46 the switches are added to the circuit.

When the re-sampler is in track-mode, switches S_{RS} and S_1 are conducting, while switch S_2 is not conducting. At the end of the amplify-phase of the amplifier, switches S_{RS} and S_1 are opened, so the residue signal is sampled on C_S and on both pre-amplifier inputs. Now, switch S_2 is closed and the pre-amplifier input settles to the first DAC value (9/16). Capacitive feed-through causes a change in V_S, as can be seen in the bottom part of Fig. 4.46. During the conversion however, V_{DAC} approximates the sampled value, and while this happens, the initial feed-through is undone. So, for critical comparator decisions, when the DAC value is close to the sampled value, the feed-through is completely undone and does not cause errors.

With this simple solution, only two switches with appropriate clock-signals are needed and issues with offset, gain or distortion are avoided. Moreover, the power consumption is negligible.

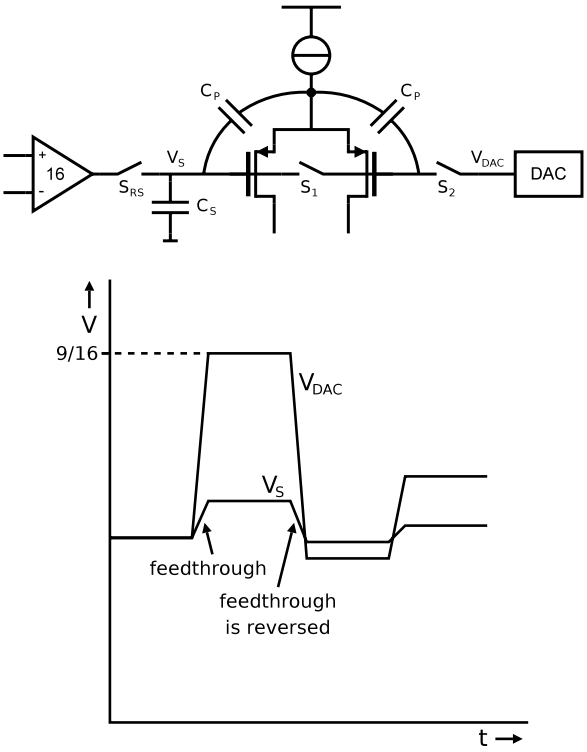

Fig. 4.46 The re-sampler and its load extended with two additional switches to avoid feed-through from the DAC (*top*) and corresponding waveforms (*bottom*). The feed-through is undone, while the DAC value approaches the input value

4.5 Calibration

Time-interleaved ADCs usually require calibration of gain and offset [16, 18, 40], and sometimes of timing as well [40]. In Sect. 2.7.3 it was explained that timing calibration is troublesome. To avoid it here, the aim is to make the timing alignment accurate by design by using the low-skew switch-driver, as explained in Sect. 4.3.3.

In this design the channel gain and offset, and the comparator offset is adjustable. The adjustments are implemented in the analog domain as discussed in Sect. 2.7 on p. 32. This has the advantage that the complete input window can be used and that high-speed power-hungry digital operations are avoided.

The adjustments for channel gain and offset are needed to correct for deviations caused by the use of small transistors in the T&H buffers [24], and for deviations in the sample process caused by mismatch as well.

In this implementation, the digital settings of the adjustments are controlled externally and no calibration algorithm is implemented. The calibration could be automated in a simple start-up calibration, which would only require quasi-DC input signals.

The adjustments for channel gain and offset are implemented by modifying analog bias settings of the T&H buffers, controlled by DACs.

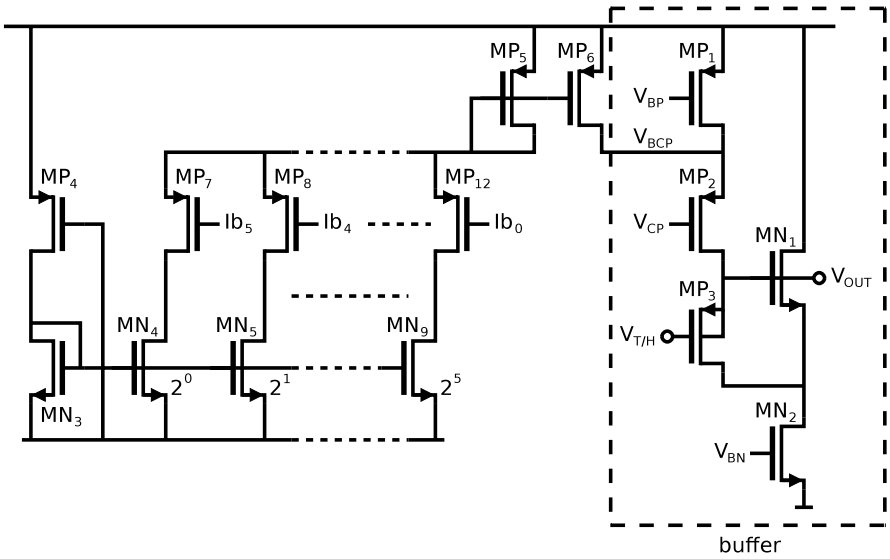

Fig. 4.47 Schematic of the offset adjustment of the T&H buffer

4.5.1 Offset Calibration

Offset calibration is performed with the circuit shown in Fig. 4.47. On the right-hand side, part of the T&H buffer can be seen (MP_1–MP_3 and MN_1–MN_2). For the complete schematic of the T&H buffer is referred to Fig. 4.20 on p. 91. The circuit operates as follows: MP_4 generates a current, which is converted into a bias voltage by diode MN_3. Transistors MN_4 to MN_9 generate binary scaled copies of this current, and depending on the digital calibration signals Ib_{0-5}, these currents either flow into diode MP_5 or they do not. The sum of the scaled currents is copied to MP_6 and added to the nominal bias current generated by MP_1.

The same circuitry is added to the other side of the quasi-differential buffer, so that the bias current of both halves of the buffer can be adjusted in 64 steps. A larger bias current results in a larger V_{GS} of the source-follower transistor MP_3 and thus in a larger voltage difference between the input and output of the buffer. By applying a differential digital setting to both halves of the DAC circuit, the differential offset can be controlled, while the common-mode offset is kept (almost) constant.

4.5.2 Gain Calibration

Gain calibration is implemented by connecting 7 binary scaled PMOST resistors between nodes V_{BCP} (Fig. 4.47) and V_{BCN} (the same node on the other half of the quasi differential circuit). The gates of these transistors are digitally controlled by an

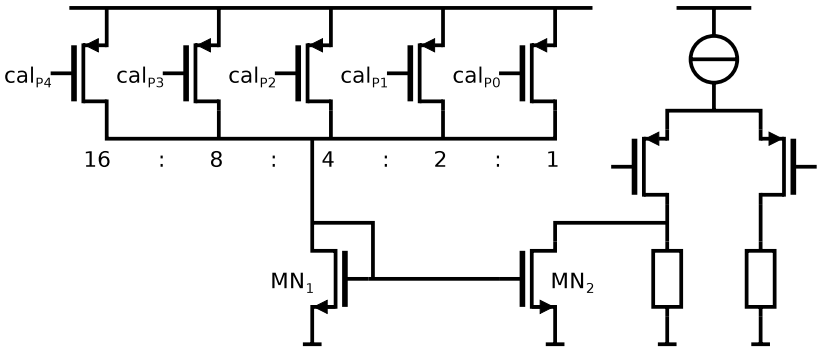

Fig. 4.48 Schematic of the offset adjustment of the comparator

external signal. If the gate is near the supply voltage, the transistor is off, and if the gate is near ground the transistor operates in triode region and forms an impedance between the two differential nodes. By lowering this impedance, the gain of the quasi-differential buffer is lowered.

The adjustable impedance is placed above cascode transistor MP_2 for two reasons: The impedance level below the cascode is quite high, and to make small steps in the gain possible (e.g. 5% in 128 steps $\Rightarrow$ 0.04% per step) the transistors would need a very high resistance. Very high resistances are hard to realize and costly with respect to area. This is also true for different types of resistors (e.g. poly or active resistors). Above the cascode, the impedance level is lower, and hence the resistance of the transistors can be lower as well.

The second reason is that below the cascodes, the signal swing can be as large as 0.4 V and the transistors would need to be very linear, to avoid distortion. The resistance in triode region shows a large dependance on V_{GS} and therefore this would not be a viable option. Above the cascode however, the signal swing is small and the linearity requirement is no longer an issue.

To allow the use of small differential pairs to keep the load for the DAC small and thus save power, comparator offset is also made adjustable. The calibration schematic is shown in Fig. 4.48, with the pre-amplifier on the right-hand side. Depending on the digital control signals cal_{P4}–cal_{P0}, the current from the 5 binary scaled PMOSTs flows into diode MN_1 or it does not. The sum of these currents is added to the current flowing in the resistor via the current mirror. The same circuit is used for the other resistor as well, so that the differential offset can be adjusted.

To ease the calibration of the comparator in SA-ADC2, a calibration mode is implemented. In this mode, the inputs of the comparator are shorted to a common-mode voltage. The comparator offset can now be adjusted independent on the input signal.

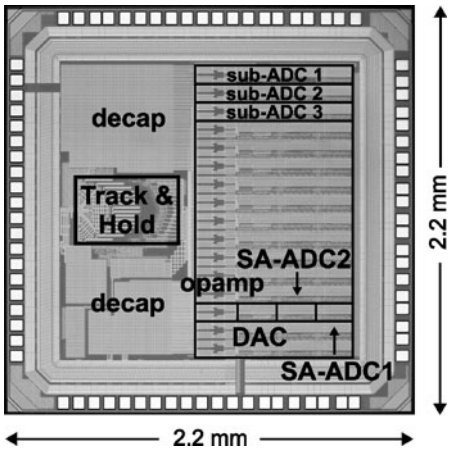

Fig. 4.49 Photograph of the time-interleaved ADC. The total active area is 1.6 mm^2

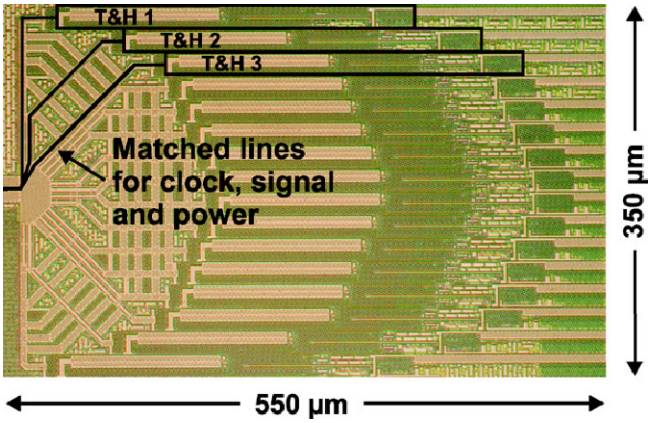

Fig. 4.50 Photograph of the time-interleaved T&H. Matched lines are used for the clock, signal and power lines. The area of the T&H is 0.2 mm^2

4.6 Layout

The complete design with the 16 channel time-interleaved T&H and 16 sub-ADCs, is fabricated in a 6-metal 0.13 µm CMOS process and a photograph of the chip is shown in Fig. 4.49. The chip measures 2.2 mm by 2.2 mm and the total active area is 1.6 mm^2.

Regarding timing, gain and bandwidth, the T&H circuits are most critical, and therefore they are not combined together with a sub-ADC in a channel, but instead all T&Hs are placed together in one compact block.

This interleaved T&H block is located on the left-hand side, and has an area of 0.2 mm^2. A close-up is shown in Fig. 4.50. The input-signal, the buffered clock-signal and the power lines enter the T&H in the middle on the left-hand side. From that point, the signals are routed to all the channels. To obtain good matching be-

tween these lines with respect to impedance and delay, they need to have an equal length. As discussed in Sect. 2.3.1 on p. 15, the preferred half circular layout is used.

From left to right a T&H channel contains a sample switch (MN_6 in Fig. 4.13 on p. 84), bootstrap switches and bootstrap capacitor MN_7, the clock generation for these switches, and to save space, the sample capacitor is placed on top of this circuitry in metals 4–6. This is followed by the T&H buffer and bias generation. A T&H channel is 20 µm high and 300 µm long.

The 16 sub-ADCs are located on the right-hand side of the chip. The lines from the T&Hs to the sub-ADCs have different lengths. This is not a problem, since these signals are time-discrete and the bandwidth is sufficiently large. From left to right, a sub-ADC channel consists of the interstage amplifier, the main DAC, SA-ADC1, SA-ADC2 and the multiplexer. The order of these blocks is chosen such that the capacitance of the critical lines is minimal.

The remaining area is filled with decoupling capacitance, consisting of both oxide capacitance and metal plate capacitance.

The chip has 80 pins and includes a pad-ring with ESD protections and buffers for the various IOs: standard, RF with low series resistance and low parasitic capacitance, supply, Schmitt triggers and LVDS drivers.

To minimize crosstalk from digital signals to analog signals, the pad-ring consists of two separate parts: one for analog signals and one for digital signals. This can be seen in the chip photograph by the two cuts in the padring.

4.7 Measurements

4.7.1 Measurement Setup

The measurement board is shown in Fig. 4.51 with the ADC chip under the large black chip holder. The board contains connections for the power supplies, reference

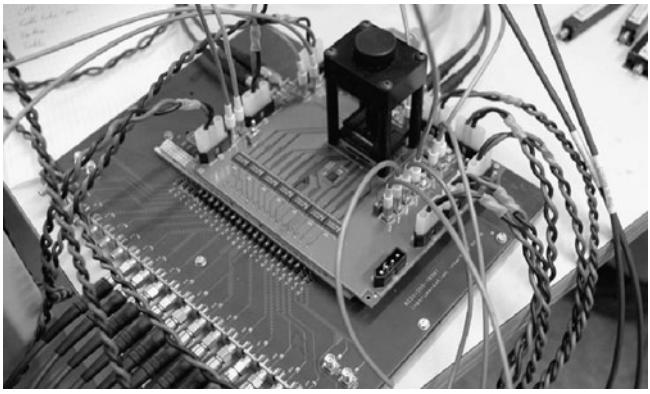

Fig. 4.51 Measurement board containing the ADC

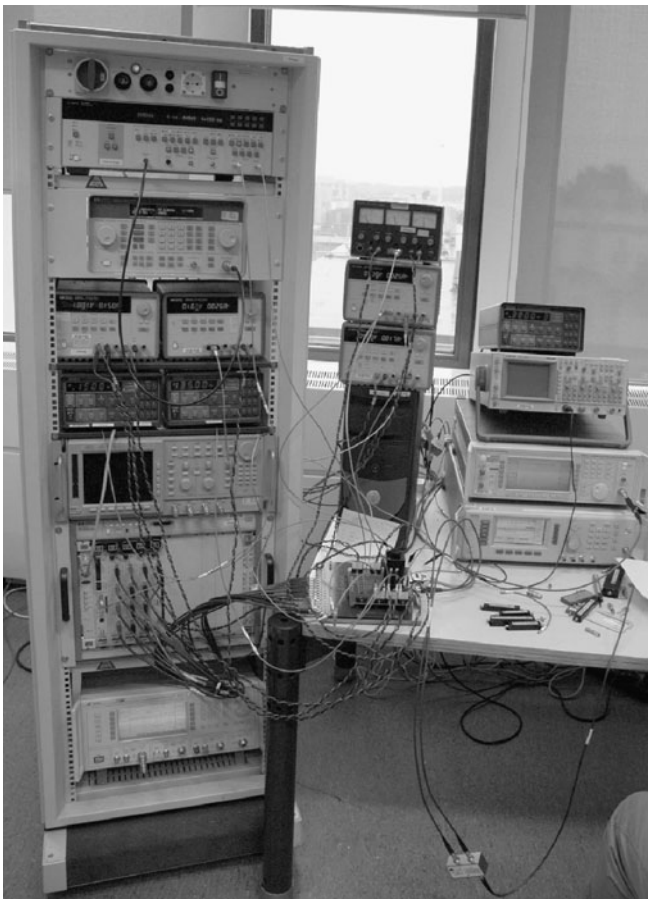

Fig. 4.52 The complete measurement setup

voltages, bias currents, the clock and signal inputs of the ADC, digital inputs to load the calibration settings in the chip and decoupling.

The seven ICs (Integrated Circuits) on the left side of the board contain flipflops to buffer and re-clock the digital output data of the ADC. These flipflops are clocked with the IC located in between the ADC and the flipflops. The buffered output data is transferred via the lower board to a data acquisition device. After this, the data stream is transferred offline[14] to a PC, where it is analyzed.

The complete measurement setup is shown in Fig. 4.52. The rack consists of the following devices: A device to make square clock signals for the re-clocking circuitry on the measurement board and for the acquisition device; a clock-generator

[14]That is, not in realtime.

that generates the clock for the previous device; supply sources; current sources and the digital acquisition device.

Next to the rack, there are more voltage sources and a PC, and next to that a current source, an always handy oscilloscope, the generator of the clock-signal[15] and the generator for the input signal.[16]

The three signal generators are locked to a common 10 MHz reference signal, to guarantee that the output bits are re-clocked at the right moment and that the phase between the clock signal and the input signal is fixed, which is needed for a proper performance analysis.

The signal generators are single-ended, while the ADC has differential inputs. Therefore, two baluns are used to perform the single-ended to differential conversion. One of the baluns can be seen at the bottom of the measurement setup, floating between three cables.

All voltage and current sources and the calibration settings are controlled by National Instruments Labview® running on the PC. Software has been written to examine both individual sub-ADC channels and the complete interleaved ADC with 16 channels.

4.7.2 Measurement Results

In this section the measurement results of a design running at 1.35 GS/s are presented. In Sect. 4.8 an improved design is presented with a sample-rate of 1.8 GS/s, and the performance is compared with other state-of-the-art designs.

Single Channel Performance

First, the measurement result of a single channel is discussed. During this measurement all channels are active, so that noise caused by other channels is still present. However, only the data from one channel is analyzed. The T&H is directly connected to a 50 Ω signal generator. The digital tester is limited to a few hundred megahertz, therefore on-chip decimation with a factor 9 is used. In Fig. 4.53, the measurement result is shown at a total sample-rate of 1350 MS/s resulting in $1350/16 \approx 84$ MS/s for a single channel. At low signal frequencies, the SNDR is 8.0 ENOB limited by amplifier noise and quantization imperfections. The THD (Total Harmonic Distortion) for low frequencies is −60 dB and the THD improvement at 8 GHz is due to a decrease in signal amplitude caused by losses in the test-bench signal path. THD at 4 GHz is −52 dB and THD at 8 GHz is −44 dB, which shows the excellent bandwidth and linearity of the T&H thanks to the use of the new circuit techniques applied in the T&H and the buffer. At 4 GHz signal frequency, the

[15] Marconi 2042 Signal Generator 10 kHz to 5.4 GHz.

[16] Anritsu 69177B Ultra Low Noise Synthesized Signal Generator 10 MHz to 50 GHz.

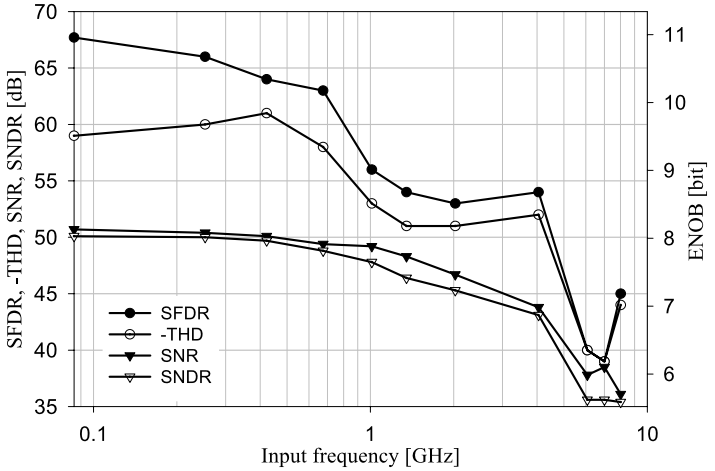

Fig. 4.53 Measurement result of a single channel with a sample-rate of 1350/16 MS/s ≈ 84 MS/s

SNDR is 6.9 ENOB and at 8 GHz the SNDR is 5.6 ENOB, which are higher than any values reported in literature at the time of writing of this book for these signal frequencies.

When increasing the signal frequency, thermal noise and quantization noise stay constant, while noise due to jitter increases linearly with the signal frequency. At very high signal frequencies the SNR is strongly dominated by jitter and a good (but worst-case) approximation of the RMS jitter is therefore given by:

$$\sigma(\Delta t) = \frac{10^{-\mathrm{SNR}/20}}{2\pi \cdot f_{\mathrm{IN}}} \tag{4.12}$$

Using this equation, the total jitter stemming from clock and signal generators and the circuit is only 0.2 ps RMS, which is better than any value published for a T&H or ADC in CMOS at the time of publication [25].

All Channel Performance

The complete 16-channel interleaved performance at 1.35 GS/s is shown in Fig. 4.54. The SNDR is 7.7 ENOB at low input frequencies and the ERBW is 1 GHz. Compared to the single channel case, the performance is only slightly degraded, showing that channel gain and offset are adjusted satisfactorily and the step-size of the adjustment DACs is sufficiently small.

It is possible to extract the timing misalignment from the measured data by determining the phase of the output signal for each channel by means of an FFT. This way, jitter is averaged out and only the timing offsets remain.

Bandwidth mismatch between channels also appears as timing offset. In Sect. 2.2.2 it was shown that for an input frequency of 1 GHz and a switch as

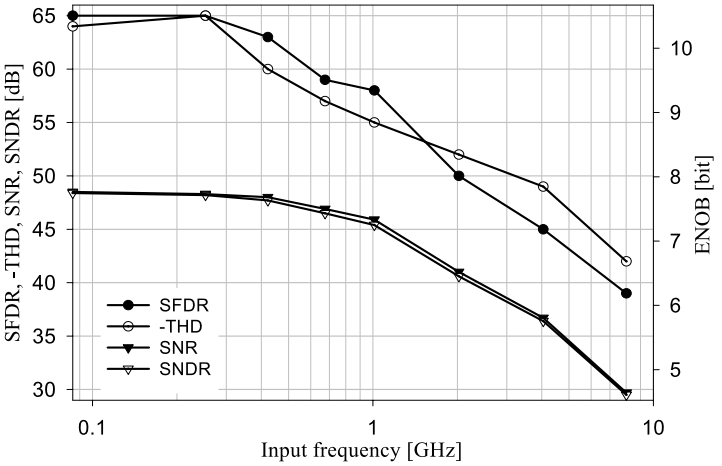

Fig. 4.54 Measurement result of the complete time-interleaved ADC at 1.35 GS/s

used in this design, 10 effective bits can be reached, if only bandwidth mismatch is considered. Using (4.12), this translates to a (worst-case[17]) RMS timing error of 0.13 ps. Compared to the total timing error shown next, this can be neglected.

The result of the timing misalignment extraction for two measurements is shown in Fig. 4.55. The extracted RMS timing misalignment is 0.6 ps RMS, which is close to the expected value of 0.45 ps RMS and which shows that the low skew technique (Sect. 4.3.3) works well and avoids the need for timing calibration. Due to the dominance of timing misalignment, total timing error across all channels including jitter is also 0.6 ps RMS. For ADCs with an ERBW over 500 MHz, this value is slightly better than the best reported in literature [40] where elaborate timing calibration is used. At 2 GHz, the SNDR is 6.5 ENOB and at 4 GHz the SNDR is 5.8 ENOB, limited by timing misalignment.

The input capacitance is about 1 pF and with a termination of 50 Ω on-chip and 50 Ω source impedance this results in an RC-limited analog input bandwidth of 6 GHz. The T&H buffers use a supply voltage of 1.6 V, however the potential between different device terminals does not exceed the nominal supply voltage. The rest of the chip uses the nominal supply voltage of 1.2 V.

The power consumption of the T&H including clock buffer and timing generation is 34 mW; the T&H buffers consume 40 mW and the 16 sub-ADCs together consume 100 mW. The FoM of the complete ADC calculated by

$$\text{FoM} = \frac{\text{power}}{2^{\text{ENOB}} \cdot \min(f_S, 2 \cdot \text{ERBW})} \tag{4.13}$$

[17]This is worst-case, since it assumes bandwidth mismatch only causes phase shifts. In reality it also causes gain errors, which do not result in timing errors. The approximation is however quite close, since for low signal frequencies compared to the channel bandwidth, phase errors strongly dominate, as explained in Sect. 2.2.2.

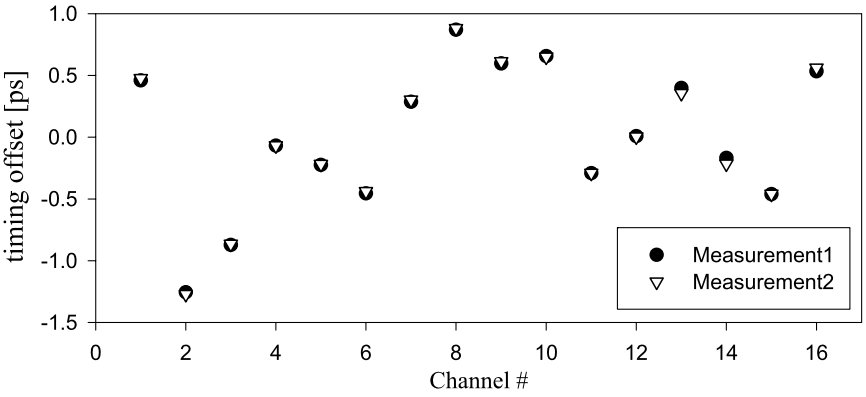

Fig. 4.55 Normalized extracted timing misalignment

is 0.6 pJ per conversion step.

4.8 Improved Design

In order to further increase the performance of the ADC, a second design was made. In this second design two aspects are improved: (1) SNR and (2) sample-rate. In the first design, SNR is mainly limited by (1a) thermal noise of the interstage amplifier and (1b) DNL of the SA-ADCs. This DNL was impaired by a parasitic capacitance asymmetry of 0.2 fF, causing crosstalk from the comparator output to the DAC, resulting in LSB errors. Shielding or increasing wire distance was insufficient. Instead, in the second design, the differential DAC outputs were twisted in the middle, making the crosstalk common-mode and easy to reject by the comparator. In the second design, noise from the interstage amplifier (1a) was decreased by circuit scaling.

By increasing the bias current of the CML clock generator in the second design, significantly improved timing alignment and a higher maximum sample-rate were achieved. Also, special care was taken with the dummy metal fill to avoid degradation of the maximum sample-rate.

4.8.1 Measurement Results of the Improved Design

At low sample-rates, the interleaved performance of the second design is 8.6 ENOB (8.8 ENOB for a single channel), proving that the noise of the interstage amplifier is lowered and the DNL of SA-ADCs is reduced.

At the nominal supply voltage, the T&Hs and SA-ADCs operate well up to 2 GS/s, however, the interstage amplifier only works well up to a sample-rate of 1.2 GS/s; for higher sample-rates the differential output signal of the amplifier is

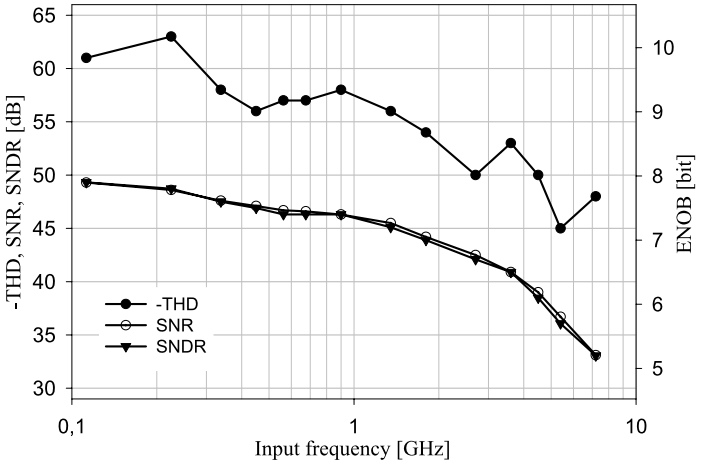

Fig. 4.56 Measurement result of the improved design at 1.8 GS/s

zero. Due to time limitations of the researchers this issue was not further investigated.

When the bias current of the amplifier is decreased, the amplifier is functional at higher sample-rates, however in this case its limited settling degrades the SNDR. In order to make the amplifier operate with nominal bias settings, the supply voltage had to be increased to 1.8 V, which slightly degrades the SA-ADCs performance. At a sample-rate of 1.8 GS/s, single-channel performance is 8.3 ENOB for low input frequencies, 7.4 ENOB at 3.6 GHz and 5.9 ENOB at 7.2 GHz.

Measurement results using all channels at 1.8 GS/s are shown in Fig. 4.56. At low input frequencies, the SNDR is 7.9 ENOB, limited by DAC settling and the negative effect of the high supply voltage. The ERBW is 1 GHz, the FoM is 1 pJ/conversion step and the power consumption is 416 mW, which has almost doubled due to the increased supply voltage. Total timing error due to jitter and misalignment between channels is 0.4 ps RMS. The timing alignment is improved by 30% due to the increased edge steepness of the CML clock.

Compared to the first design, the maximum sample-rate is increased significantly and at the nominal supply voltage (and reduced sample-rate) the ENOB is increased by almost 1 bit.

In Table 4.2 an overview of state-of-the-art time-interleaved ADCs is shown. Reference [40] has a very high sample-rate but it requires timing calibration and is not suitable for low-cost embedded application, due to its high power consumption and BiCMOS buffer. The design presented here reaches the same timing alignment without timing calibration and it has less jitter. Compared to [16] and [18], the design presented here has a much higher sample-rate and ERBW. For frequencies up to 1 GHz, [47] reaches a better accuracy at the cost of a significantly lower power efficiency. At signal frequencies above 1 GHz, our design achieves better performance than any previous publication.

Table 4.2 Performance overview of high-speed time-interleaved ADCs, a '–' means unspecified

Design	Poulton 2003 [40]	Gupta 2006 [16]	Hsu 2007 [18]	Louwsma 2007 [25]	Louwsma 2008 [27]	Taft 2009 [47]
Sample-rate [GS/s]	20	1.0	0.8	1.35	1.8	1.0
ENOB ($f_{IN} = DC$)	6.5	8.85	9.0	7.7	7.9	9.2
ENOB ($f_{IN} = 4$ GHz)	5.3	–	–	5.8	6.4	–
ERBW [GHz]	2.0	0.4	0.4	1.0	1.0	1.0
Input BW [GHz]	6.6	–	–	6	6	3
Power [W]	10	0.25	0.35	0.18	0.42	1.3
FoM [pJ/conv.step]	28	0.7	0.9	0.6	1	2.1
jitter [ps RMS]	0.6	–	0.43	0.2	0.2	0.2
Timing misalignment [ps RMS]	0.4 (calibr.)	–	–	0.6	0.4	–

4.9 Conclusions

In this chapter a time-interleaved ADC was demonstrated with 16 channels. By omitting a frontend sampler, using a new buffer and removing the load in track mode, the T&H reaches a high bandwidth and good linearity. For a single channel, THD is -52 dB at an input frequency of 4 GHz and SNDR is 43 dB, which is only limited by (best-in-class) jitter of 0.2 ps RMS. Low jitter is enabled by the use of CML clock generation in the T&H and by omitting circuitry to adjust timing alignment, as this increases the amount of jitter. Instead, a novel circuit design is used to achieve a timing alignment of 0.6 ps RMS, avoiding the need for timing calibration.

By pipelining two SA-ADCs, a combination of high sample-rate and good power efficiency is reached. The single-sided overrange architecture achieves a 22% higher power efficiency compared to the conventional overrange architecture and look-ahead logic minimizes logic delay in the SA-ADC. The FoM of the complete ADC including T&H is 0.6 pJ per conversion-step. The SNDR is 7.7 ENOB for low signal frequencies, while the ERBW is 1 GHz, showing broadband signal handling capability.

An improved design achieves an SNDR of 8.6 ENOB for low sample-rates and with a higher supply voltage it reaches a sample-rate of 1.8 GS/s with 7.9 ENOB at low signal frequencies and an ERBW of 1 GHz. At $f_{IN} = 3.6$ GHz, the SNDR is still 6.5 ENOB and total timing error including jitter is only 0.4 ps RMS, which is better than any previous publication for an ADC with a bandwidth larger than 500 MHz.

Chapter 5
Summary and Conclusions

5.1 Summary

We live in an analog world, whereas signal processing performed in the digital domain has many advantages regarding noise immunity, accuracy and flexibility. Moreover, its power consumption decreases rapidly as technology shrinks to smaller feature sizes. Together with the increasing system requirements, this creates a large demand for ADCs with a high sample-rate, high resolution and low power consumption.

This book investigates the feasibility of an analog-to-digital converter with a sample-rate of 1–2 GS/s, a resolution of 8–10 bits, and a state-of-the-art power efficiency of less than 1 pJ/conversion step. The time-interleaved architecture exploits parallelism to increase the sample-rate while maintaining good power efficiency, and therefore it is the most suitable architecture.

Chapter 2 describes the Track and Hold (T&H) architecture for such a time-interleaved ADC. Mismatch between channels, like difference in offset, gain and timing, degrades the performance. Regarding bandwidth mismatch it is shown that for a large bandwidth compared to the signal frequency, the effect of bandwidth mismatch is mitigated for the frequencies of interest, and bandwidth calibration is not needed.

Two T&H architectures are discussed, one with a frontend sampler and one without. The use of a frontend sampler has the advantage of good timing alignment between channels, the resistance of the switch is however a problem: it limits both the input bandwidth and the achievable resolution and the track-time has to be less than one sample period. Under the assumptions made, the use of a frontend sampler is not feasible for the target performance.

The gain-bandwidth product of T&H buffers using feedback is too small for application in high-frequency T&Hs. A new open loop buffer is introduced which has a large bandwidth and improved linearity compared to a conventional source follower. If the T&H buffer is loaded with a large capacitive load, the bandwidth is small and for high-frequency input signals distortion arises. By placing a switch between the buffer and the capacitive load, which is open in track-mode, the buffer

S.M. Louwsma et al., *Time-interleaved Analog-to-Digital Converters,*
Analog Circuits and Signal Processing,
DOI 10.1007/978-90-481-9716-3_5, © Springer Science+Business Media B.V. 2011

bandwidth is increased and the distortion is avoided without increasing the power consumption. Thanks to the relatively low sample-rate of the sub-ADCs, there is sufficient time available to let the output signal of the buffer settle after the switch is closed.

Some guidelines are given for determining the number of channels of a time-interleaved ADC. This depends on resolution, bandwidth, technology, and whether a frontend sampler is used or not. For the target specifications and process technology, and considering the current state-of-the-art, a sub-ADC sample-rate of about 100 MS/s can serve as a starting point, as literature shows that power efficient sub-ADCs with this sample-rate are currently feasible.

The topic of calibration is discussed and while offset and gain calibrations are relatively easy to implement, timing calibration is much harder to realize: it requires high-frequency test-signals and the required adjustable timing circuitry causes jitter by itself. It is therefore beneficial if the timing alignment is accurate by design, such that calibration is not needed. It is advantageous to compensate offset and gain errors in the analog domain, as this is power efficient, the complete analog input range can be used, and it allows for sub-LSB corrections.

The jitter requirements on the sample-clock are discussed. The traditional requirement assumes a full-scale sinusoid at the maximum frequency. For many applications this requirement is too strict and can lead to over-design. To make a realistic estimation for the allowed amount of jitter, the signal spectrum must be taken into account.

Chapter 3 discusses the architecture of the sub-ADCs, which are used in the time-interleaved ADC. A Successive Approximation ADC (SA-ADC) can have a very good power efficiency, its sample-rate is however limited. In a conventional SA-ADC, the sample-rate is mainly limited by settling of the DAC. The use of different DAC settling times for different steps in an SA-ADC can reduce the DAC settling time up to 42% for a 6 bits converter, the implementation of the required clocking scheme however is not trivial.

Overrange techniques can reduce the required DAC settling time even more. A new overrange technique is presented called the single-sided overrange technique. Compared to a conventional 6 bits SA-ADC, it saves 58% of the settling time, while compared to previous overrange architectures, it uses 16% less static energy per conversion and 22% less dynamic energy per conversion. For the single-sided overrange architecture the optimum number of conversion steps is determined. It turns out that the optimum is 7 steps for a 6 bits converter.

By using multiple comparators with different accuracies in an SA-ADC with overrange, power can be saved. For example, in a 10 bits converter, the first 8 conversion steps can be performed with a low power and low accuracy comparator, while the remaining 2 steps are performed with an accurate comparator. In this way, a factor of three in comparator power can be saved.

Look-ahead logic removes the delay of the logic out of the SA-ADC loop and increases the maximum sample-rate.

An efficiency comparison between an SA-ADC and a pipeline ADC is made, based on the power consumption of comparators and opamps. For the same spec-

ifications, an SA-ADC can use roughly 10 times less power. This conclusion is supported by literature.

Chapter 4 describes the actual implementation of a high-speed time-interleaved ADC based on the design choices described in this book. To maximize the input bandwidth, each channel has a dedicated T&H without frontend sampler. The sample switch is bootstrapped with a simple circuit, which is suitable for high sample-rates. The clock-generation circuitry for the T&Hs uses current mode logic (CML) to obtain sufficient power supply rejection to ensure low sampling jitter and to minimize crosstalk to the signal. Since timing calibration is hard to implement, a switch-driver circuit with low skew is introduced, such that timing calibration is not needed.

The T&H buffer is implemented as described in the second chapter. The sub-ADCs consist of two 6 bits SA-ADCs, a DAC and an amplifier to achieve a good power efficiency, while increasing the maximum sample-rate over that of a single 10 bits SA-ADC. The large overrange relaxes the requirements on the amplifier. The amplifier consists of a two-stage opamp with a switched capacitor network. To optimize settling, frequency compensation is only used in sample-mode.

Look-ahead logic and the single-sided overrange technique as described in the second chapter are implemented. The resistor ladder DACs use only half the number of taps of a regular DAC, minimizing the output capacitance and decreasing the power consumption. The ladders of the different channels are connected together to minimize settling time.

The re-sampler after the amplifier can be implemented by using only two additional switches, avoiding the need for an additional buffer and saving power.

Offset and gain calibrations are performed in the analog domain to avoid power consuming digital operations and to keep the full input range available. Gain adjustment is implemented by placing an adjustable impedance above the cascodes of the differential T&H buffer. This relaxes the requirements on the impedance: it does not need to be very linear (as would be the case if the impedance would be placed between the buffer outputs) and its value can be lower, which is advantageous since very high impedances are hard to realize in CMOS processes.

The chapter ends with a description of the measurement results: The ADC achieves a sample-rate of 1.8 GS/s with 7.9 ENOB and an ERBW of 1 GHz, while the power efficiency is 1 pJ/conversion-step. At $f_{\mathrm{IN}} = 3.6$ GHz, the SNDR is still 6.5 ENOB and total timing error including jitter is only 0.4 ps RMS. At a sample-rate of 1.35 GS/s and 7.7 ENOB, a FoM of 0.6 pJ/conversion step is achieved. This proves that the specifications stated in the beginning of this chapter are feasible and that the presented techniques are useful.

5.2 Conclusions

- It is advantageous if the T&H channels of an interleaved ADC have a large bandwidth, such that for the frequencies of interest the effect of bandwidth mismatch is mitigated.

- The use of a frontend sampler has the advantage of good timing alignment between channels, the resistance of the switch is however a problem: it limits both the input bandwidth and the achievable accuracy.
- If a T&H buffer is loaded with a large capacitive load, the bandwidth is only moderate and for high-frequency input signals distortion arises. By placing a switch between the buffer and the capacitive load, which is open in track-mode, the buffer bandwidth is increased and the distortion is avoided, without increasing the power consumption. This enables high speed sampling with low power consumption.
- Overrange techniques can reduce the required DAC settling time. A new overrange technique is presented called the single-sided overrange technique. Compared to a conventional 6 bits SA-ADC, it saves 58% of the settling time, while compared to previous overrange architectures, it uses 16% less static energy per conversion and 22% less dynamic energy per conversion.
- By using multiple comparators with different accuracies in an SA-ADC, power can be saved.
- Based on the power consumption of opamps and comparators, an SA-ADC can use roughly 10 times less power than a pipeline converter using opamps.
- By using a switch-driver circuit that has low skew, good timing alignment of 0.4 ps RMS can be achieved in 0.13 μm CMOS, making timing calibration unnecessary.
- It is feasible to make a time-interleaved ADC with a sample-rate of 1.8 GS/s, 7.9 ENOB and a FoM of 1 pJ/conversion-step. At $f_{IN} = 3.6$ GHz, the SNDR is still 6.5 ENOB and total timing error including jitter is only 0.4 ps RMS.

5.3 Original Contributions

- The analysis of the effects of bandwidth mismatch as a function of the nominal channel bandwidths.
- The comparison of two time-interleaved T&H architectures, one with and one without a frontend sampler.
- The analysis of distortion caused by a limited bandwidth of the T&H buffer, and the introduction of a switch between the buffer and the capacitive load, which is open in track-mode.
- The introduction of the single-sided overrange technique.
- The analysis of using comparators with different accuracies in an SA-ADC, to save power.
- A comparison between the power efficiency of an SA-ADC and a pipeline converter, based on comparators and opamp.
- The introduction of a low skew switch-driver circuit, avoiding the need for timing calibration.
- The experimental proof that the suggested concepts are feasible.
- A simple 1.5 clock-divider, with only two flipflops and a NAND-gate.

- The introduction of a new open loop buffer that has a large bandwidth and improved linearity compared to a source follower.
- The introduction of a resistor ladder DAC that uses only half the number of taps of a regular DAC, minimizing the output capacitance and decreasing the power consumption.
- The introduction of an adjustable impedance placed above the cascodes of a differential buffer to adjust the gain of the T&H buffer, without requiring a very linear or unpractically high ohmic impedance.
- The derivation of the optimum number of conversion steps for the single-sided overrange architecture.

5.4 Recommendations for Future Research

- The performance of the presented T&H is quite satisfactory. To further improve the design, one should focus on the sub-ADC design.
- The number of channels of the presented time-interleaved ADC was decided in an early stage of the project. With the gained knowledge during this project, it is recommended to increase the number of channels slightly, to decrease the sample-rate of the sub-ADCs. This makes their implementation easier and allows for a higher power efficiency. For example, the interstage amplifier could then probably be omitted.
- To enable a higher number of channels without decreasing the input bandwidth, the layout of the critical T&H circuits should be made very compact.
- The single-sided overrange technique minimizes the required DAC settling time, but requires more complex control logic and a decoder. Depending on the technology feature size, this requires quite some chip area and power consumption. It is recommended to investigate alternative solutions that require less control logic.
- A charge redistribution SA-ADC can have a very good power efficiency [54]. Therefore its application in a time-interleaved ADC should be investigated.
- In [54], the supply rails are used as a voltage reference, which is possible there since the sample-rate is low and no other functionality is integrated on the chip. In a high-speed time-interleaved ADC these criteria are not met, and another voltage reference is required. It is therefore recommended to investigate low power voltage reference buffers, especially aimed at charge redistribution DACs.
- The investigation of alternative architectures (e.g. binary search ADC [23, 52]) to increase the sub-ADC sample-rate.
- The investigation of an input buffer for the time-interleaved ADC that can meet the specifications, while consuming little power.

Bibliography

[1] A.M. Abo, Design for reliability of low-voltage, switched-capacity circuits. Ph.D. Thesis, University of California, Berkeley, 1999

[2] A.M. Abo, P.R. Gray, A 1.5-V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter. IEEE J. Solid-State Circuits **34**(5), 599–606 (1999)

[3] B.K. Ahuja, An improved frequency compensation technique for CMOS operational amplifiers. IEEE J. Solid-State Circuits **18**(6), 629–633 (1983)

[4] V.J. Arkesteijn, Analog front-ends for software-defined radio receivers. Ph.D. dissertation, University of Twente, 2007

[5] V.J. Arkesteijn, E.A.M. Klumperink, B. Nauta, Jitter requirements of the sampling clock in software radio receivers. IEEE Trans. Circuits Syst. (TCAS) II **53**(2), 90–94 (2006)

[6] C.W. Barbour, Simplified PCM analog to digital converter using capacity charge transfer, in *Proc. of the Telemetering Conf.* (1971), pp. 4.1–4.11

[7] A. Barna, D.I. Porat, *Integrated Circuits in Digital Electronics* (Wiley, New York, 1973), pp. 353–354

[8] W.C. Black, D.A. Hodges, Time interleaved converter arrays. IEEE J. Solid-State Circuits **15**(6), 1022–1029 (1980)

[9] M. Boulemnakher, E. Andre, J. Roux, F. Paillardet, A 1.2 V 4.5 mW 10 b 100 MS/s pipelined ADC in 65 nm CMOS, in *ISSCC Dig. Tech. Papers* (2008), pp. 250–251

[10] T.B. Cho, P.R. Gray, A 10 b, 20 Msample/s, 35 mW pipeline A/D converter. IEEE J. Solid-State Circuits **30**(3), 166–172 (1995)

[11] R.H. Dennard, F.H. Gaensslen, H.N. Yu, V.L. Rideovt, E. Bassous, A.R. LeBlanc, Design of ion-implanted MOSFET's with very small physical dimensions. IEEE J. Solid-State Circuits 256–268 (1974)

[12] S. Devarajan, L. Singer, D. Kelly, S. Decker, A. Kamath, P. Wilkins, A 16 b 125 MS/s 385 mW 78.7 dB SNR CMOS pipeline ADC, in *ISSCC Dig. Tech. Papers* (2009), pp. 86–87

[13] A.G.F. Dingwall, Monolithic expandable 6 bit 20 MHz CMOS/SOS A/D converter. IEEE J. Solid-State Circuits **14**(6), 926–932 (1979)

[14] M. El-Chammas, B. Murmann, General analysis on the impact of phase-skew in time-interleaved ADCs, in *IEEE International Symposium on Circuits and Systems (ISCAS)* (2008), pp. 17–20

[15] S.L.J. Gierkink, Control linearity and jitter of relaxation oscillators. Ph.D. dissertation, University of Twente, 1999

[16] S.K. Gupta, M.A. Inerfield, J. Wang, A 1-GS/s 11-bit ADC with 55-dB SNDR, 250-MAW power realized by a high bandwidth scalable time-interleaved architecture. IEEE J. Solid-State Circuits **41**(12), 2650–2657 (2006)

[17] K. Hadidi, A. Khoei, A highly linear cascode-driver CMOS source-follower buffer, in *IEEE Intl. Conf. on Electronics, Circuits and Systems* (1996), pp. 1243–1246

S.M. Louwsma et al., *Time-interleaved Analog-to-Digital Converters,*
Analog Circuits and Signal Processing,
DOI 10.1007/978-90-481-9716-3, © Springer Science+Business Media B.V. 2011

[18] C.-C. Hsu, F.-C. Huang, C.-Y. Shih, C.C. Huang, Y.-H. Lin, C.-C. Lee, B. Razavi, An 11 b 800 MS/s time-interleaved ADC with digital background calibration, in *ISSCC Dig. Tech. Papers* (2007), pp. 464–465

[19] E. Iroaga, B. Murmann, L. Nathawad, A background correction technique for timing errors in time-interleaved analog-to-digital converters, in *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 6 (2005), pp. 5557–5560

[20] E.A.M. Klumperink, B. Nauta, Systematic comparison of HF CMOS transconductors. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **50**(20), 728–741 (2003)

[21] N. Kurosawa, H. Kobayashi, K. Maruyama, H. Sugawara, K. Kobayashi, Explicit analysis of channel mismatch effects in time-interleaved ADC systems. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **48**(3), 261–271 (2001)

[22] F. Kuttner, A 1.2 V 10 b 20 MSample/s non-binary successive approximation ADC in 0.13 μm CMOS, in *ISSCC Dig. Tech. Papers* (2002), pp. 176–177

[23] Y.Z. Lin, S.J. Chang, Y.T. Liu, C.C. Liu, G.Y. Huang, A 5 b 800 MS/s 2 mW asynchronous binary-search ADC in 65 nm CMOS, in *ISSCC Dig. Tech. Papers* (2009), pp. 80–81

[24] S.M. Louwsma, E.J.M. van Tuijl, M. Vertregt, B. Nauta, A 1.6 GS/s, 16 times interleaved track & hold with 7.6 ENOB in 0.12 μm CMOS, in *Proc. ESSCIRC* (2004), pp. 343–346

[25] S.M. Louwsma, E.J.M. van Tuijl, M. Vertregt, B. Nauta, A 1.35 GS/s, 10 b, 175 mW time-interleaved AD converter in 0.13 μm CMOS, in *Proceedings of the Symposium on Very Large Scale Integration (VLSI) Circuits* (2007), pp. 62–63

[26] S.M. Louwsma, E.J.M. van Tuijl, M. Vertregt, B. Nauta, A time-interleaved track & hold in 0.13 μm CMOS sub-sampling a 4 GHz signal with 43 dB SNDR, in *Proceedings of the Custom Integrated Circuits Conference (CICC)* (2007), pp. 329–332

[27] S.M. Louwsma, A.J.M. van Tuijl, M. Vertregt, B. Nauta, A 1.35 GS/s, 10 b, 175 mW time-interleaved AD converter in 0.13 μm CMOS. IEEE J. Solid-State Circuits **43**(4), 778–786 (2008)

[28] J. McCreary, P. Gray, All-MOS charge redistribution analog-to-digital conversion techniques. IEEE J. Solid-State Circuits **10**(6), 371–379 (1975)

[29] E. Mensink, E.A.M. Klumperink, B. Nauta, Distortion cancellation by polyphase multipath circuits. IEEE Trans. Circuits Syst. I, Regul. Pap. **52**(9), 1785–1794 (2005)

[30] G.E. Moore, Cramming more components onto integrated circuits, in *Electronics* (1965), pp. 114–2117

[31] B. Murmann, A/D converter trends: power dissipation, scaling and digitally assisted architectures, in *Proceedings of the Custom Integrated Circuits Conference (CICC)*, (2008), pp. 751–758

[32] B. Murmann, ADC Performance Survey 1997–2009. Available online: http://www.stanford.edu/~murmann/adcsurvey.html (2009)

[33] K. Nagaraj, D.A. Martin, M. Wolfe, R. Chattopadhyay, S. Pavan, J. Cancio, T.R. Viswanathan, A dual-mode 700-Msamples/s 6-bit 200-Msamples/s 7-bit A/D converter in a 0.25-μm digital CMOS process. IEEE J. Solid-State Circuits **35**(12), 1760–1768 (2000)

[34] Y. Nakagome, H. Tanaka, K. Takeuchi, E. Kume, Y. Watanabe, T. Kaga, Y. Kawamoto, F. Murai, R. Izawa, D. Hisamoto, T. Kisu, T. Nishida, E. Takeda, K. Itoh, Experimental 1.5-V 64-Mb DRAM. IEEE J. Solid-State Circuits **26**(4), 465–472 (1991)

[35] B. Nauta, Analog CMOS low power design considerations, in *Low Power Workshop on ESSCIRC Conference* (1996)

[36] B. Nikolić, V.G. Oklobdžija, V. Stojanović, W. Jia, J.K.S. Chiu, M.M.T. Leung, Improved sense-amplifier-based flip-flop: design and measurements. IEEE J. Solid-State Circuits **35**(6), 876–884 (2000)

[37] H. Pan, M. Segame, M. Choi, J. Cao, A.A. Abidi, A 3.3-V 12-b 50-MS/s A/D converter in 0.6-μm CMOS with over 80-dB SFDR. IEEE J. Solid-State Circuits **35**, 1769–1780 (2000)

[38] M.J.M. Pelgrom, A.C.J. Duinmaijer, A.P.G. Welbers, Matching properties of MOS transistors. IEEE J. Solid-State Circuits **24**(5), 1433–1439 (1989)

[39] K. Poulton, J.J. Corcoran, T. Hornak, A 1-GHz 6-bit ADC System. IEEE J. Solid-State Circuits **22**(6), 962–970 (1987)
[40] K. Poulton, R. Neff, B. Setterberg, B. Wuppermann, T. Kopley, R. Jewett, J. Pernillo, C. Tan, A. Montijo, A 20 GS/s 8 b ADC with a 1 MB memory in 0.18 μm CMOS, in *ISSCC Dig. Tech. Papers* (2003), pp. 318–496
[41] B. Razavi, *Rf Microelectronics* (Prentice Hall, New York, 1998)
[42] D. Schinkel, E. Mensink, E.A.M. Klumperink, A.J.M. van Tuijl, B. Nauta, A double-tail latch-type voltage sense amplifier with 18 ps setup+hold time, in *ISSCC Dig. Tech. Papers* (2007), pp. 314–315
[43] D. Schinkel, E. Mensink, E.A.M. Klumperink, A.J.M. van Tuijl, B. Nauta, A low-offset double-tail latch-type voltage sense amplifier, in *Proceedings of the 18th ProRisc Workshop* (2007)
[44] H. Schmidt, *Analog-Digital Conversion* (Van Nostrand-Reinholt, New York, 1970)
[45] A.J. Scholten, G.D.J. Smit, B.A.D. Vries, L.F. Tiemeijer, J.A. Croon, D.B.M. Klaassen, R. van Langevelde, X. Li, W. Wu, G. Gildenblat, The new CMC standard compact MOS model PSP: advantages for RF applications, in *IEEE Radio Frequency Integrated Circuits Symposium* (2008), pp. 247–250
[46] T. Sepke, P. Holloway, C.G. Sodini, H.S. Lee, Noise analysis for comparator-based circuits. IEEE Trans. Circuits Syst. I, Regul Pap. **56**, 541–553 (2009)
[47] R.C. Taft, P.A. Francese, M.R. Tursi, O. Hidri, A. MacKenzie, T. Hoehn, P. Schmitz, H. Werker, A. Glenny, A 1.8 V 1.0 GS/s 10 b self-calibrating unified-folding-interpolating ADC with 9.1 ENOB at Nyquist frequency, in *ISSCC Dig. Tech. Papers* (2009), pp. 78–79
[48] H.P. Tuinhout, G. Hoogzaad, M. Vertregt, R.L.J. Roovers, C. Erdmann, Design and characterization of a high precision resistor ladder test structure, in *Proceedings of the IEEE International Conference on Microelectronic Test Structures (ICMTS)*, vol. 15 (2002), pp. 223–228
[49] R.C.H. van de Beek, High-speed low-jitter frequency multiplication in CMOS. Ph.D. dissertation, University of Twente, 2004
[50] R.J. van de Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters* (Kluwer Academic, Dordrecht, 1994)
[51] R.J. van de Plassche, R.E.J. van der Grift, A high-speed 7 bit A/D converter. IEEE J. Solid-State Circuits **14**(6), 938–943 (1979)
[52] G. van der Plas, B. Verbruggen, A 150 MS/s 133 μW 7 b ADC in 90 nm digital CMOS using a comparator-based asynchronous binary-search sub-ADC, in *ISSCC Dig. Tech. Papers* (2008), pp. 242–243
[53] H. van der Ploeg, Calibration techniques in two-step a/d converters. Ph.D. dissertation, University of Twente, 2005
[54] M. van Elzakker, A.J.M. van Tuijl, P.F.J. Geraedts, D. Schinkel, E.A.M. Klumperink, B. Nauta, A 1.9 μW 4.4 fJ/conversion-step 10 b 1 MS/s charge-redistribution ADC, in *ISSCC Dig. Tech. Papers* (2008), pp. 245–245
[55] A.J.M. van Tuijl, personal communication
[56] A. Verma, B. Razavi, A 10 b 500 MHz 55 mW CMOS ADC, in *ISSCC Dig. Tech. Papers*, (2009), pp. 84–85
[57] M. Vertregt, The analog challenge of nanometer CMOS, in *International Electron Devices Meeting (IEDM)* (2006), pp. 1–8
[58] M. Vertregt, H.P. Tuinhout, personal communication
[59] Video-transcript, Excerpts from a conversation with Gordon Moore: Moore's law. ftp://download.intel.com/museum/Moores_Law/Video-Transcripts/Excepts_A_Conversation_with_Gordon_Moore.pdf (2005)
[60] R.H. Walden, Analog-to-digital converter survey and analysis. IEEE J. Sel. Areas Commun. **17**(4), 539–550 (1999)
[61] B. Wicht, T. Nirschl, D. Schmitt-Landsiedel, Yield and speed optimization of a latch-type voltage sense amplifier. IEEE J. Solid-State Circuits **39**(7), 1148–1158 (2004)

[62] Wikipedia, Orthogonal frequency-division multiplexing. http://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing (2009)
[63] K.L.J. Wong, C.K.K. Yang, Offset compensation in comparators with minimum input-referred supply noise. IEEE J. Solid-State Circuits **39**(5), 837–840 (2004)
[64] W. Yang, D. Kelly, L. Mehr, M.T. Sayuk, L. Singer, A 3-V 340-mW 14-b 75-Msample/s CMOS ADC with 85-dB SFDR at Nyquist input. IEEE J. Solid-State Circuits **36**(12), 1931–1936 (2001)

Index

S.M. Louwsma et al., *Time-interleaved Analog-to-Digital Converters,*
Analog Circuits and Signal Processing,
DOI 10.1007/978-90-481-9716-3, © Springer Science+Business Media B.V. 2011