Calibration of Sampling Clock Skew in High-Speed, High-Resolution Time-Interleaved ADCs.

by

Daniel Kumar

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering and Computer Science at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

© Massachusetts Institute of Technology 2015. All rights reserved.

Signature redacted

Department of Electrical Engineering and Computer Science March 27, 2015

Signature redacted

Hae-Seung Lee
Advanced Television and Signal Processing Professor of Electrical Engineering Thesis Supervisor

Signature redacted

Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses
Calibration of Sampling Clock Skew in High-Speed, High-Resolution Time-Interleaved ADCs.

by

Daniel Kumar

Submitted to the Department of Electrical Engineering and Computer Science on March 27, 2015, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

There is an ever-increasing demand for high-resolution and high-resolution ADCs. In order to raise the sampling rates of ADCs in a power efficient manner, time-interleaving is an essential technique, whereby \( N \) ADC channels, each operating at a sampling frequency of \( f_s \), are used to achieve an effective conversion rate of \( N \cdot f_s \).

While time-interleaving enables higher conversion rates in a given technology, mismatch issues such as gain, offset, and sampling clock skew between channels degrade the overall time-interleaved ADC performance. Of these issues, sampling clock skew between channels is the biggest problem in high-speed and high-resolution, time-interleaved ADCs as errors due to sampling clock skew become more severe for higher input frequencies. There are a few sources of sampling clock skew between channels. Mismatches in the sampling clock path and logic delays are the most obvious ones. Input signal routing mismatch and \( RC \) mismatch of the input sampling circuits also cause sampling clock skew.

In this thesis, we developed two new methods to mitigate the effects of sampling clock skew in time-interleaved ADCs. The first is the rapid consecutive sampling method, whereby each interleaved channel is implemented using two sub-channel ADCs. Two consecutive samples of the input are taken with a short time delay between them. This allows for a straight-forward linear interpolation between the consecutive samples in order to recover the de-skewed sample. The second method entails introducing a programmable delay in the input signal path, instead of delaying the sampling clock, in order to calibrate out sampling clock skew. The design and implementation of a proof-of-concept, time-interleaved ADC that implements the input signal delay method is detailed. Finally, measurement results to show the efficacy of the proposed method in mitigating the effects of sampling clock skew is also presented.

Thesis Supervisor: Hae-Seung Lee
Title: Advanced Television and Signal Processing Professor of Electrical Engineering
Acknowledgments

I would like to first thank my advisor, Professor Hae-Seung Lee. It has been a terrific opportunity to work with him. His technical guidance has been stellar and I have learned much about circuit design from him. I have always been amazed by the breadth and depth of his knowledge. However, what inspires me the most is his unwavering support and ability to see the best way forward when problems seem insurmountable, both with regards to research and life in general. For that, I will be ever grateful.

I would also like to thank the members of my thesis committee, Professor Anantha Chandrakasan and Professor Charles Sodini. I have gained a lot from their teaching, advice, feedback, and support over the years. I would also like to thank Professor Ayman Shabra for his valuable technical feedback.

Another highlight of graduate school was the people I had a chance to work with. Bruno DoValle, Kohei Onizuka, Eric Winokur, Sabino Pietrangelo, Do Yeon Yoon, Xi Yang, SungWon Chung, Mariana Markova, Albert Chang, Sunghyuk Lee, Kailiang Chen, Hyun Boo, Grant Anderson, Joohyun Seo, Harneet Khurana, and Maggie Delano have been great colleagues. Beyond the doors of 38-265, my thanks goes to Lorenzo Turicchia, Wei Li, Woradorn Wattanapanitch, Scott Arfin, Arun Paidimarri, Soumyajit Mandal, Micah O'Halloran, and Shirin Farrahi too. Their generosity in time and thoughtful advice (both technical and otherwise) has been humbling. Graduate school is a team sport, and I’m grateful to have had them on my team!

My time at MIT would also have been much less enjoyable if not for the close friends that have continually brought joy and companionship into my life. Omair Saadat, Siddharth Bhardwaj, Himanshu Dhamankar, Somani Patnaik, Johann John, Edward Boenig, Margo Liptsin, Victor Li, Paul Wilson, and Marc Biel. I cherish your friendship.

Finally, it would be remiss of me if I did not thank my family. My parents and siblings have cheered me on through the highlights and supported me through the
tough times. For that, I am eternally grateful. To Jaisree, you're probably my best 'discovery' at MIT - thank you for everything.
# Contents

1 Introduction  \hspace{1cm} 23
   1.1 Time-Interleaved ADCs  \hspace{1cm} 25
   1.2 Thesis Organization  \hspace{1cm} 29

2 Operation of Time-Interleaved ADCs  \hspace{1cm} 31
   2.1 Single Channel Sampling  \hspace{1cm} 32
      2.1.1 Single Channel Sampling in Time-Domain  \hspace{1cm} 33
      2.1.2 Single Channel Sampling in Frequency-Domain  \hspace{1cm} 34
   2.2 Time Interleaved Sampling  \hspace{1cm} 35
      2.2.1 Multi-Channel, Time-Interleaved Sampling in Time Domain  \hspace{1cm} 36
      2.2.2 Multi-Channel, Time-Interleaved Sampling in Frequency-Domain  \hspace{1cm} 38
   2.3 The Effect of Channel Errors on the Performance of Time-Interleaved ADC Systems  \hspace{1cm} 40
      2.3.1 Offset Mismatch Between Channels  \hspace{1cm} 42
      2.3.2 Gain Mismatch Between Channels  \hspace{1cm} 44
      2.3.3 Sampling Clock Skew Between Channels  \hspace{1cm} 49
   2.4 Comparison of Channel Mismatches  \hspace{1cm} 54

3 Sampling Clock Skew Errors in Time-Interleaved ADCs  \hspace{1cm} 57
   3.1 Major Sources of Sampling Clock Skew  \hspace{1cm} 57
      3.1.1 Skew Due to Transistor Variations  \hspace{1cm} 58
      3.1.2 Skew Due to Variations in Clock and Signal Routing  \hspace{1cm} 61
   3.2 Sampling Clock Skew Insensitive Time-Interleaved Architectures  \hspace{1cm} 63
3.2.1 Two-Rank Track-and-Hold .................................. 63
3.2.2 Channel Randomization ......................................... 65

3.3 Calibration and Correction Methods to Mitigate Effects of Sampling Clock Skew .................................................. 65
3.3.1 Sampling Clock Skew Detection .................................. 67
3.3.2 Calibration and Correction of Sampling Clock Skew .......... 73

3.4 Summary .............................................................. 80

4 Proposed Sampling Clock Skew Calibration Methods ................. 81
4.1 Rapid Consecutive Sampling (RCS) Method ......................... 82
4.1.1 Bound on Delay Between Samples ................................. 85
4.1.2 Behavioral Simulation Results of the RCS Technique ........... 87
4.1.3 Challenges in Implementing the RCS Technique ............... 88
4.2 Input Signal Delay Control Method .................................. 91
4.2.1 Fine Input Signal Delay Control Using Low-Pass Filters ...... 93
4.2.2 Behavioral Simulation Results of the Input Signal Delay Con-
trol Technique ...................................................... 96
4.2.3 Input Sampling Bandwidth Requirements ....................... 98
4.2.4 Effect of Residual Gain Mismatch ................................. 100
4.3 Summary .............................................................. 102

5 Prototype Time-Interleaved SAR ADC Circuit Design ............... 105
5.1 Interleaved Channel Converter Architecture ....................... 106
5.2 SAR ADC Block Diagram ............................................ 108
5.3 DAC Design ............................................................ 109
5.4 Input Sampling Network .............................................. 111
5.4.1 Bottom Plate Sampling ............................................ 111
5.4.2 Bootstrapped Sampling Switch ................................... 113
5.4.3 Input Signal Delay Control ...................................... 116
5.5 Latch Comparator ...................................................... 118
5.6 Asynchronous Clock Generation .................................... 126
5.7 Multi-Phase Clock Generator ................................................. 128
5.8 Off-Chip References .......................................................... 131
5.9 Layout ................................................................................. 133
5.10 Summary ............................................................................. 133

6 Test Setup and Measurement Results ............................................ 137
  6.1 Test Setup ........................................................................... 137
    6.1.1 Prototype Chip ............................................................ 137
    6.1.2 Printed Circuit Board (PCB) ............................................ 137
    6.1.3 Measurement Setup ...................................................... 139
  6.2 Measurement Results ........................................................... 139
    6.2.1 Static Performance ......................................................... 145
  6.3 Performance Summary ......................................................... 146
  6.4 Summary ............................................................................. 148

7 Conclusions .............................................................................. 151
  7.1 Summary ............................................................................. 151
  7.2 Future Work ....................................................................... 153
List of Figures

1-1 Block diagram of a typical signal-chain for electronic systems. .......... 24

1-2 Plot of power consumption vs. input signal Nyquist bandwidth. The curved line shows the expected power growth if a single channel ADC is used to achieve increasing bandwidths. The linear line shows the expected power consumption of an ideal time-interleaved ADC with increasing bandwidths (with power of overhead circuitry to implement time-interleaving excluded). ........................................... 25

1-3 Block diagram of a time-interleaved ADC. .................................. 27

1-4 Plot of Schreier FOM vs Nyquist sampling frequencies for all ADC papers presented at ISSCC and VLSI from 1997-2014. .................... 28

2-1 Block diagram of a typical Nyquist-rate ADC. ............................... 32

2-2 Block diagram of a sampling system. ........................................... 33

2-3 Time-domain representation of a continuous signal, \( x(t) \), sampled using a delta train function, \( d(t) \). .................................................. 34

2-4 Frequency-domain representation of a continuous signal, \( X(j\omega) \), sampled using a delta train function, \( D(j\omega) \). ................................. 35

2-5 Block diagram of a N-way time-interleaved sampler. The delta trains, \( d_i(t) \), are used to sample the \( i \)-th channel and are delayed by \( T/N \) from each other. .................................................... 36
Time-domain representation of a 4-way time-interleaved system. The sampling period of each delta train is $T$, but the delta train for each channel is shifted by $T/N$ from each other. The sampled output, $x_s(t)$ appears to be sampled with a sampling period of $T/4$.

Frequency-domain representation of a 2-way time-interleaved system. While the spectrum of each of the channels show aliasing in the frequency spectrum, the summed output does not suffer from aliasing. The images centered at $\pm \pi, \pm 3\pi, ...$ from the odd and even channels (denoted using a shaded image) cancel each other out, leaving the resulting spectrum free of aliasing artifacts.

Block diagram of a time-interleaved ADC with offset mismatch between channels modeled as an additive voltage source, $V_{o,i}$, at the input of the $i$-th channel.

Behavioral simulation results of the time domain error due to offset mismatches between channels. The plot in panel (a) shows a 12-bit output of a 4-way time-interleaved ADC with offset mismatch between channels. The graph in panel (b) shows the error in LSBs of the time-interleaved system with offset mismatch from the ideal, expected output code when no offset mismatch between channels is present.

A frequency spectrum of the output of a behavioral simulation of a 4-way time-interleaved ADC with offset mismatches between channels. The spurs at $f_s/2$ and $f_s/4$ are due to the offset mismatches between channels in the time interleaved ADC.

Block diagram of a time-interleaved ADC with gain mismatch between channels modeled as an gain block with gains, $G_i$, before the input of the $i$-th channel.
2-12 Behavioral simulation results of the time domain error due to gain mismatches between channels. The plot in panel (a) shows a 12b output of a 4-way time-interleaved ADC with gain mismatch between channels. The graph in panel (b) shows the error in LSBs of the time-interleaved system with offset mismatch from the ideal, expected output code.

2-13 A frequency spectrum of the output of a behavioral simulation of a 4-way time-interleaved ADC with gain mismatches between channels. The pair of spurs spaced by ±f_m around f_s/4 and f_s/2 are due to the gain mismatches between the channels in the time interleaved ADC.

2-14 Block diagram of a time-interleaved ADC with sampling clock skew between channels modeled as a time delay Δt_i added to the clock signal of the i-th channel.

2-15 Behavioral simulation results of the time domain error due to sampling clock skew between channels. The plot in panel (a) shows a 12-bit output of a 4-way time-interleaved ADC with sampling clock skew between channels. The graph in panel (b) shows the error in LSBs between output of the time-interleaved system with sampling clock skew and an the expected output code for an ideal, skew-free, time-interleaved system.

2-16 The same sampling clock skew results in a voltage error proportional to the gradient of the input error during the sampling instant.

2-17 A frequency spectrum of the output of a behavioral simulation of a 4-way time-interleaved ADC with sampling clock skew between channels. The pair of spurs spaced by ±f_m around f_s/4 and f_s/2 are due to the sampling clock skew between channels in the time interleaved ADC.

2-18 SNDR vs input signal frequency for a 4-way, 12-bit, time-interleaved ADC with offset mismatch, gain mismatch, and sampling clock skew between the interleaved channels.
3-1 Sampling clock for a time-interleaved ADC. $CLK_{FS}$ is the full-speed clock. The $i$-th interleaved channel is sampled on the falling edge of $Clk_i$. 

3-2 Circuit schematic of a multi-phase clock generator. The output of each AND gate is buffered to drive the clock routing interconnects to the individual channels. 

3-3 Circuit schematic of the input sampling network during the track phase. 

3-4 Possible routing schemes for the input signal and clock in a time-interleaved ADC: (a) input signal and clocks are routed linearly, (b) input signal and clocks are routed in a H-tree fashion. 

3-5 Time-interleaved ADC with a global T/H that samples the input signal. The T/H within each interleaved channel re-samples the output of the global T/H. 

3-6 Time-interleaved ADC with channel randomization to mitigate sampling clock skew. $P$-channels are used to implement an $N$-way time-interleaved ADC, where $P/gtN$. 

3-7 Sampling-clock skew calibration or correction process. 

3-8 Sampling clock skew estimation using out-of-energy calculation for bandlimited input signals. 

3-9 Zero-crossing for a 2-way interleaved ADC without skew (a), and with timing skew (b). 

3-10 Estimation of sampling clock skew between the ADC under test and the calibration (CAL) ADC. $x(t)$ is the input signal and $\phi$ and $\phi_{cal}$ are the clocks for the ADC channel under test and the CAL ADC respectively. Source:[21] 

3-11 Popular circuit implementations of variable delay lines. In (a), a variable load capacitance is used to control the delay of $Clk_d$, while in (b) a current-starved inverter is used to control $Clk_d$ by varying $I_{cal}$. 
3-12 Jitter addition from variable delay lines. When the clock edge is slowed down, it becomes more susceptible to noise near threshold of the retiming inverter, leading to increased jitter in the output clock.

3-13 The effect of increasing slew rates and inverter noise on \( t_{\text{thresh}} \).

3-14 Sampling clock skew correction using a fractional delay filter, \( H(z) \).

4-1 For small sampling clock skews, linear extrapolation can be used to estimate a skew-free sample.

4-2 Using linear interpolation to estimate the skew free sample using two samples of the input signal \( \text{Sample}_a \) and \( \text{Sample}_b \) of the input taken before and after the ideal sampling instant.

4-3 Block diagram of a time-interleaved ADC with RCS.

4-4 Timing diagram of a time-interleaved ADC with RCS.

4-5 Error between the estimate of the input signal and the actual value when linear interpolation applied using two samples separated by \( \Delta T_{\text{r}} \) for various input signal frequencies.

4-6 Output spectrum of a behavioral simulation for a four-way time interleaved ADC with sampling clock skew. In (a), the output spectrum of the uncalibrated ADC is shown, and in (b) the output spectrum after calibration for an ADC utilizing the RCS technique is shown.

4-7 Input disturbance after sub-channel sampling for the \( i \)-th channel in time-interleaved ADC with RCS.

4-8 Sampling noise of the linearly interpolated sample using the output of two sub-channels.
4-9 Controlling the delay of the input signal as a method to calibrate out sampling clock skew. In (a), the usual method for sampling clock skew calibration is presented. Here the sampling clock edge is shifted such that the effects of sampling clock skew is removed. In (b), sampling clock skew is reduced by delaying/advancing the input signal instead. The square marker indicates the sample when the input signal is delayed/advanced, showing that the effect of sampling clock skew has been mitigated. ................................. 92

4-10 Block diagram of a time-interleaved ADC with RCS. ................ 93

4-11 Sampling voltage error caused by sampling clock skew. ............ 94

4-12 Maximum allowable sampling clock skew, $t_{skew}$ vs. input signal frequency, $f_{in}$, plotted for various ADC resolutions. ............... 95

4-13 Block diagram of a time-interleaved ADC with RCS. ............... 95

4-14 Mapping the low-pass $RC$ network to the input sampling network of an ADC. .......................................................... 97

4-15 Output spectrum of a behavioral simulation for a four-way time interleaved ADC with sampling clock skew. In (a), the output spectrum of the uncalibrated ADC is shown, while in (b) the output spectrum of a calibrated ADC utilizing the input signal delay control technique is shown. ......................................................... 97

4-16 Output settling of the T/H circuit. In (a), the schematic that models the input T/H during the track phase is shown and in (b) the output waveforms for the worst case settling requirement is illustrated. .... 99

4-17 The maximum variance in gain mismatch between channels when the entire calibration range is used for sampling clock skew calibration. The maximum variance, $\sigma_{gain,skewCal}^2$, is normalized by $\sigma_{gain}^2$. ............ 102

4-18 The maximum variance in gain mismatch between channels when the calibration range of one standard deviation is used for sampling clock skew calibration. The maximum variance, $\sigma_{gain,skewCal}^2$, is normalized by $\sigma_{gain}^2$. ................................................. 103
5-1 T/H implementation for sampling with variable delay of the input signal. In (a), a simplified model of the sampling network that needs to be implemented to introduce controllable delay to the input signal $v_{in}$ is shown. In (b), one possible implementation to change the sampling switch resistance, $R_{sw}$, is shown. More than one sampling switch can be used by turning on multiple $Trk_i$ during the sampling phase.

5-2 Input sampling network for a SAR ADC with the auxiliary sampling switches to vary the delay of the sampled input. Only the input tracking switches are shown. The reference switches are omitted in this figure for simplicity.

5-3 Block diagram of the prototype 4-way time-interleaved ADC.

5-4 Schematic of a split-capacitor DAC.

5-5 Schematic of a capacitive-resistive hybrid DAC. In this schematic only one half of the DAC is shown for simplicity. A fully differential DAC was implemented in the prototype ADC.

5-6 Schematic of the T/H with bottom-plate sampling. CLKd is the delayed version of the sampling clock, CLK.

5-7 Schematic of the differential T/H with bottom-plate sampling switch $M1B$. The device $Mx$ is used to shuffle charge between the two half circuits rather than from the outputs.

5-8 Schematic of a simplified input tracking network. $R_{sw}$ models the switch resistance when it is on.

5-9 Schematic of the tracking switch clock bootstrap circuit.

5-10 Input signal and the gate voltage, $V_G$, of the tracking switch. During each track phase, $V_G$ is always VDD above the input signal.

5-11 Schematic of the input sampling network with auxiliary switches $M_{IB}$ for sampling clock skew calibration. Only one half of the capacitive DAC is shown for simplicity. The actual implementation is fully differential. The reference switches are also omitted for simplicity.

5-12 Schematic of a StrongArm comparator commonly used in SAR ADCs.
5-13 Relevant waveforms of the StrongArm comparator commonly used in SAR ADCs. .................................................. 120
5-14 Schematic of a Schinkel latch. .................................................. 121
5-15 Relevant waveforms of the StrongArm comparator commonly used in SAR ADCs. .................................................. 123
5-16 Schematic of a Miyahara comparator. The capacitor banks on D+ and D- are used for offset calibration. .................................................. 124
5-17 Schematic of the asynchronous SAR clock generation circuit block. . 126
5-18 Timing diagram illustrating the relevant signals of the SAR asynchronous timing scheme. .................................................. 127
5-19 Relevant waveforms of the StrongArm comparator commonly used in SAR ADCs. .................................................. 128
5-20 Block diagram of the time-interleaved clocking network. .............. 129
5-21 Schematic of the differential to single-ended converter used to generate a CMOS level square clock from the sinusoidal inputs. The bias current is supplied from off-chip. .................................................. 130
5-22 Schematic of the multi-phase clock generator used to generate the sampling clocks for each SAR ADC. .................. 130
5-23 Schematic used to model the effect of bondwire parasitics on the ADC references. .................................................. 132
5-24 Schematic used to model the interleaved bondwire parasitics on the ADC references. .................................................. 132
5-25 Layout of the core of the prototype time-interleaved ADC. The layout is shown in (a), and a legend indicating the relevant blocks is shown in (b). .................................................. 134
5-26 Layout of the SAR ADC used in the individual channels. The layout is shown in (a), and a legend indicating the relevant blocks is shown in (b). .................................................. 135

6-1 Die photo of the prototype ADC. .................................................. 138
6-2 Test setup used to characterize the prototype ADC. ............... 140
6-3 Prototype ADC output spectra for a low frequency input signal before and after offset calibration. The spectrum of the interleaved output before offset calibration is shown in (a) while the spectrum of the interleaved output after offset calibration is shown in (b) ............... 141
6-4 Prototype ADC output spectra for a low frequency input signal before sampling clock skew calibration. The single channel output spectrum is shown in (a) and the four channel output spectrum is shown in (b). 142
6-5 Prototype ADC output spectra for a high frequency input signal near its Nyquist rate before sampling clock skew calibration. The single channel output spectrum is shown in (a) and the four channel output spectrum is shown in (b). ......................... 143
6-6 Prototype ADC output spectra for a low frequency input signal after sampling clock skew calibration. The single channel output spectrum is shown in (a) and the four channel output spectrum is shown in (b). 144
6-7 Prototype ADC output spectra for a high frequency input signal near its Nyquist rate after sampling clock skew calibration. The single channel output spectrum is shown in (a) and the four channel output spectrum is shown in (b). ......................... 145
6-8 SNDR of the time-interleaved ADC output with various input signal frequencies before and after sampling clock skew calibration. ........ 146
6-9 INL and DNL performance of the ADC. Single channel results are shown on the left column and time-interleaved results are shown in the right column. ................................. 147
6-10 Breakdown of the power consumed by the time-interleaved ADC. ... 148
6-11 Energy versus SNDR plot for time-interleaved ADCs presented at ISSCC and VLSI from 1997-2014. ................................. 149
List of Tables

4.1 Number of settling time-constants, $M$, required to settle to within half LSB for various ADC resolutions. 100

6.1 Performance summary of the prototype time-interleaved ADC. 147
Chapter 1

Introduction

The need for computing elements to process real world signals is ever increasing. While signals in the real-world are analog, the predominant computing paradigm today is in the digital domain. Thus, analog-to-digital converters (ADCs) are required to sample and digitize real-world analog signals so that the digitized signals can be processed further in the digital domain.

The signal chain of a typical electronic system with a real-world, analog input signal is illustrated in Fig. 1-1. The analog signal is sensed and transduced into an electrical signal by a sensor. Next, the analog, electrical signal is conditioned and processed (amplified, filtered, etc.) in the analog domain before being sampled and converted into a stream of bits by an ADC. The stream of bits is then sent to a digital back-end or a computer to undergo further signal processing. Over the past few decades, we have observed that the prevailing trend in the field is to minimize the amount of processing that is done in the analog domain, and to move as much processing as possible to the digital domain. This shift is driven by two main forces, one due to technology scaling, and the other is the drive to manage complexity as systems scale in size and scope. From the technological standpoint, device scaling has benefited digital circuits tremendously. Scaling has increased the speed and power efficiency of digital circuits significantly. However, with scaling, the supply voltage and intrinsic gain of transistors keep reducing from one process node to another, which affects the performance of the analog blocks adversely. Thus, the trend has
been to minimize the signal conditioning and processing in the analog domain, and to shift the bulk of signal processing tasks to the digital domain.

The second motivation for minimizing the amount of signal processing that is done in the analog domain is the necessity to manage design complexity and to increase design reuse. As systems keep getting more and more complex, it is valuable to have access to raw data that can use the most up to date advancements in new digital processing techniques or software updates. In many applications, such as medical or communication systems, it may be difficult or cost prohibitive to physically upgrade the analog hardware, but updating the digital processing algorithms (e.g. by updating an FPGA or microcontroller) is feasible. One example of such a system is the cochlear implant - the analog blocks in the cochlear implant are usually made to be simple as possible, and most of the processing is done in the digital blocks that are easily re-programmable and that can updated as new speech processing techniques are invented. From a design re-use perspective, analog systems are highly application specific and cannot be ported or scaled easily from one application to another. Thus, for every application, significant design time needs to be dedicated for the design of the analog processing blocks. Digital systems, on the other hand, due to the advancements in hardware description languages (HDLs), have enabled increasing amounts of design reuse from one application to another.

Both these reasons outlined above keep moving the boundary between the digital and analog further into the analog domain (i.e. minimized the processing done in the
Figure 1-2: Plot of power consumption vs. input signal Nyquist bandwidth. The curved line shows the expected power growth if a single channel ADC is used to achieve increasing bandwidths. The linear line shows the expected power consumption of an ideal time-interleaved ADC with increasing bandwidths (with power of overhead circuitry to implement time-interleaving excluded).

analog domain, and move as much processing as is feasible into the digital domain). This places an increased burden on the ADC. ADCs need to be faster (i.e. convert more samples per second) in order to be able to digitize signals at higher frequencies, be very energy efficient, and operate at higher resolutions (i.e. encode more information per sample).

1.1 Time-Interleaved ADCs

As more and more signal processing is moved to the digital domain, the need for high-accuracy, and high-speed ADCs that operate in an energy efficient manner becomes increasingly important. One metric to compare the performance of different ADCs is to use the energy-per-conversion-step figure-of-merit (FoM) [1]. Typically, as the number of conversions per second or the resolution of the ADC (i.e. the number of distinct digital levels) increases, so does the energy consumed per conversion. However, the increase in the power consumption of an ADC is usually super-linear with increasing sampling speeds or bandwidth of a single-channel ADC as illustrated.
in Fig. 1-2. As the bandwidth of the input signal is increased, the corresponding power consumption of the converter increases much more quickly [2,3]. For example, consider a SAR ADC, where the total conversion time per sample is given by,

\[ t_{\text{clk}} = t_{\text{samp}} + N(t_{\text{settle}} + t_{\text{Comparator}} + t_{\text{logic}}), \]  

(1.1)

where \( N \) is the number of bits of the converter. In order to increase the speed of the ADC, a few or all of the variables in Eq. 1.1 above need to be reduced. In order to reduce the DAC settling time, faster references are needed that consume significantly more power. Low power, static-current free latch based comparators may need to be replaced by power hungry, current-mode comparators. Devices within the logic circuits will need to be upsized in order to reduce logic delays. Fundamentally, as the various components in an ADC approaches the limits of a particular technology, demanding an increase in operation speed comes at a disproportionately greater increase in power consumption. As such, it is becoming more energetically impractical to build ADCs with increasing input signal bandwidths and with higher resolutions.

One effective method to increase the sampling speed of an ADC without incurring a large energy penalty is to use more than one ADC in parallel. In Fig. 1-2, if 2 converters were used instead of 1, the increase in power consumption (denoted by the horizontal dotted lines), can be linear, if the extra power overhead to implement time-interleaving is excluded. This architecture is commonly known as time-interleaving and was first presented in [4]. A block diagram on a time-interleaved ADC is illustrated in Fig. 1-3. In a time-interleaved ADC, \( N \) ADC channels, each operating at a sampling frequency \( f_s \) are used to achieve an effective conversion speed of \( N \cdot f_s \) as illustrated in below. Therefore, in a time-interleaved ADC, each channel may operate at a slower, but more energy efficient conversion rate, but together achieve a higher effective conversion rate.

The importance of time-interleaving in achieving state-of-the-art performance can be seen by observing the ADC papers presented at the International Solid State Circuits Conference (ISSCC) and at the Symposia on VLSI Technology and Circuits
(VLSI) over the years. Fig. 1-4 shows a graph with the Schreier FoM vs Nyquist sampling frequency for all ADCs presented at ISSCC and VLSI from 1997-2014 [5]. The Schreier FoM is defined as,

$$F_{oM_{\text{Schreier}}} = \text{SNDR}_{dB} + 10 \cdot \log_{10} \left( \frac{BW}{P} \right),$$

(1.2)

where $BW$ is the maximum input signal bandwidth and $P$ is the power consumption of the converter. The filled markers indicate the ADCs that utilize time-interleaving in their design. It is clear from this graph that in order to achieve very high sampling speeds in an energy efficient manner, time-interleaving is an increasingly important design paradigm.

While time-interleaving enables higher conversion rates in a given technology, it is not without challenges. First, there is an area and energy overhead that comes with utilizing a time-interleaved architecture. Having more channels requires more die area. Having more channels also requires some auxiliary circuitry such as a multi-phase sampling clock generation circuits, which adds an energy overhead.

Beyond the issues of extra die area and energy overhead, mismatch between chan-
Figure 1-4: Plot of Schreier FOM vs Nyquist sampling frequencies for all ADC papers presented at ISSCC and VLSI from 1997-2014.

Channels further complicate the implementation of time-interleaved ADCs. Mismatch between channels, such as gain mismatch, offset mismatch, and sampling clock skew between channels can degrade the overall ADC performance [6]. Of these issues, sampling clock skew between channels is the most challenging problem in time-interleaved ADCs with high resolution and high sampling rates. There are a few sources of sampling clock skew between channels. Mismatches in the sampling clock paths and logic delays are the most obvious. Input signal routing mismatch and resistive and capacitive (RC) mismatches of the input sampling circuits also cause sampling clock skew.

The sampling clock skew can be mitigated by various calibration techniques. Previous calibration techniques employ either sampling clock delay adjustment techniques [7] or by digital correction of output data [8,9]. The timing adjustment requires adjustable delays that can result in increased sampling jitter [7]. The increased jitter degrades the SNDR at high input frequencies and cannot be compensated by calibration. On the other hand, the digital calibration of output data requires complex
interpolation. It is useful to note that many of the sampling clock skew calibration techniques have been published in recent years, indicating that sampling clock skew calibration is an important current research area in ADC design.

In this thesis, we have developed a much simpler calibration circuitry and algorithms for sampling time skew compared to previous techniques that have been published in the literature. We explore two proposed techniques to mitigate sampling clock skew: i) a rapid consecutive sampling technique whereby two samples of the input are acquired with a short delay between them for each channel in at time-interleaved ADC, and ii) and input signal programmable delay technique whereby the delay of the input signal is controlled such that it is realigned with the sampling edge of the sampling clock with time-skew. We explore both techniques using behavioral simulations. In both methods, the main goals are to avoid the increase in jitter and to minimize the complexity of the calibration method such that the impact on the total noise and power consumption in the analog circuits can be made negligible.

1.2 Thesis Organization

This thesis is organized as follows. In Chapter 2 we theoretically analyze the various errors that are present in time-interleaved ADC systems. In Chapter 3 we discuss the major sources of sampling clock skew in time-interleaved ADCs. In this chapter we also explore previous work in the literature that propose methods for the detection and mitigation of sampling clock skew in time-interleaved ADCs. Chapter 4 describes the calibration methods proposed in this research. Chapter 5 describes the circuit design of a prototype ADC with our proposed sampling clock skew calibration scheme. Chapter 6 summarizes the results from the prototype ADC. Future research directions are discussed and conclusion are drawn in Chapter 7.
Chapter 2

Operation of Time-Interleaved ADCs

In Chapter 1, we described the importance of time-interleaving architectures in order to build ADCs with higher resolutions and that operate at continually increasing speeds. In this chapter, we explore the operation of time-interleaved ADCs in more detail. Specifically, we begin by analyzing the operation of a Nyquist sampling rate converter in the time and frequency domains. This analysis will set the stage for the theoretical analysis of a time-interleaved system in which each channel is sampling at below the Nyquist sampling frequency of the combined time-interleaved system. More importantly, the analysis presented in this chapter will allow us to understand the effects of mismatches between channels on the operation of the time-interleaved system. We conclude this chapter by exploring the various errors that plague time-interleaved systems and their respective effects on the overall performance of the ADC system.

All Nyquist-rate ADCs can be reduced to a model with three main blocks: an anti-aliasing filter block, followed by a sampling block and a quantizer block, as illustrated in Fig. 2-1. The anti-aliasing filter block ensures that the input signal is band-limited to below the Nyquist frequency, which is half of the sampling frequency of the ADC; in order to prevent aliasing that can degrade the performance of the ADC. The sampling block samples the continuous time input signal periodically. The
output of the sampling block is then input into the quantizer that quantizes the sampled continuous time signal in order to generate a digital output corresponding to the sampled continuous time level.

2.1 Single Channel Sampling

Before analyzing the operation of a time-interleaved ADC, we begin by analyzing an single channel ADC. In a single channel architecture, the continuous time input signal is sampled uniformly in time. A simplified version of the sampling process is illustrated in Fig. 2-2, where $x(t)$ is the continuous time input signal, $d(t)$ denotes the continuous time delta train, $x_s(t)$ is the sampled continuous time signal, and $x_s[n]$ is the discrete-time sampled signal. For the rest of the analysis in this section, we make two simplifications:

1. The sampling operation is performed using a delta train.

In real implementations, the input signal sampling is carried out by a track-and-hold circuit block that is more accurately modeled using a square wave or a periodic rectangular pulse signal. The analysis of such a sampling block, modeled using a zero-order-hold (ZOH) filter is described in [10] but is omitted in the analysis in this chapter for algebraic simplicity. It is important to note that this simplification does not affect the analysis or insights obtained in any substantial way.
2. The sampling operation is modeled as a continuous time block.

In reality, the output of the sampling block is a discrete time signal. This is illustrated in Fig. 2-2 where the continuous-to-discrete time block converts the continuous time signal to a discrete time signal. In the following analysis, we analyze the system only in continuous time, i.e. at the output of the multiplier. Mathematically converting a continuous time sample to a discrete time sample, \( x_s[n] \), is easily done via a substitution given by,

\[
x_s[n] = x_s(nT),
\]

where \( n \) is an integer. However, in order to not be weighed down by notation, we will perform all the analysis in the following section in continuous time and take the output of the sampler to be \( x_s(t) \).

### 2.1.1 Single Channel Sampling in Time-Domain

The continuous time signal \( x(t) \) is sampled uniformly in time with a sampling period of \( T \) using the delta train, \( d(t) \) which is given by:

\[
d(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT).
\]

The sampled signal \( x_s(t) \), which is the multiplication of \( x(t) \) and \( d(t) \) can be
written as,

\[ x_s(t) = x(t) \cdot d(t) \]
\[ = \sum_{k=-\infty}^{\infty} x(kT)\delta(t - kT), \quad (2.3) \]

which is the delta train weighted by the samples of \( x(kT) \). The sampling process in time-domain is illustrated in Fig. 2-3.

### 2.1.2 Single Channel Sampling in Frequency-Domain

The sampling operation described in the time-domain above can also be analyzed in the frequency domain (and is often quite instructive). From the multiplication property of Fourier transforms, the multiplication of \( x(t) \) and \( d(t) \) in the time-domain results in the convolution of \( X(j\omega) \) and \( D(j\omega) \) in the frequency domain, whereby \( X(j\omega) \) and \( D(j\omega) \) are the Fourier transforms of \( x(t) \) and \( d(t) \) respectively. Hence, \( X_s(j\omega) \) can be expressed as,

\[ X_s(j\omega) = X(j\omega) \ast D(j\omega) \quad (2.4) \]

Furthermore, we know that the delta train in the frequency domain is just a scaled version (in frequency and amplitude) of the delta train in the time domain, and can
be written as,

\[ D(j\omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta \left( \omega - \frac{2\pi k}{T} \right) \]  

(2.5)

Since the convolution of a signal with an impulse train results in a shifted and summed version of the signal, the frequency response of the sampled signal, \( X_s(j\omega) \), is given by,

\[ X_s(j\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X \left( j(\omega + \frac{2\pi k}{T}) \right) \]  

(2.6)

The spectrum of the sampled signal is illustrated in Fig. 2-4. As long as the input signal is band-limited to the Nyquist frequency, there will not be aliasing in the frequency response of the sampled output.

### 2.2 Time Interleaved Sampling

As described in Chapter 1, a time-interleaved ADC is made up of a number of \( N \) channels or sub-ADCs. Each of the channels is used to sample the input signal in sequential manner such that the effective sampling rate of the system is \( N \) times greater than the sampling rate of each individual channel. In this section, we look at the sampling operation of a time-interleaved ADC and describe how by summing the sampled outputs of each of the channels, we can indeed achieve a higher effective sampling rate than that of each channel individually.
2.2.1 Multi-Channel, Time-Interleaved Sampling in Time Domain

A higher effective sampling rate can be achieved by ensuring that the sampling instance of each channel is distributed equally in time. As in the previous section, assuming that sampling of the input signal is performed using delta trains, the block diagram of an $N$-channel, time-interleaved sampler is illustrated in Fig. 2-5. In this block diagram, each channel is sampled with a shifted delta train, $d_i(t)$. The sampled signals are then summed to produce the output, $y(t)$. The delta train associated with the $i$-th channel in an $N$-way time-interleaved ADC can be expressed as,

$$d_i(t) = \sum_{k=-\infty}^{\infty} \delta \left( t - kT - \frac{T}{N}i \right),$$  \hspace{1cm} (2.7)

where $T$ denotes the sampling period of each channel.
The sampled signals $x_{s,i}(t)$ for each channel is then given by,

$$x_{s,i}(t) = x(t) \cdot d_i(t)$$

$$= \sum_{k=-\infty}^{\infty} x \left( kT + \frac{T}{N}i \right) \cdot \delta \left( t - kT - \frac{T}{N}i \right).$$  \hspace{1cm} (2.8)

The output of the time-interleaved sampling system is $y(t)$, and can be expressed as,

$$y(t) = \sum_{i=1}^{N} x_{s,i}(t)$$

$$= \sum_{i=1}^{N} \sum_{k=-\infty}^{\infty} x \left( kT + \frac{T}{N}i \right) \cdot \delta \left( t - kT - \frac{T}{N}i \right).$$ \hspace{1cm} (2.9)

By changing the variables of the summation and some algebraic manipulation, the equation above can be rewritten as,

$$y(t) = \sum_{m=-\infty}^{\infty} x \left( m \frac{T}{N} \right) \cdot \delta \left( t - m \frac{T}{N} \right).$$ \hspace{1cm} (2.10)

By comparing Eq. (2.10) above to that of Eq. (2.3), we can see that the input signal is sampled with a sampling period of $\frac{T}{N}$, which implies that the effective sampling rate of the time-interleaved system is $N$ times higher than that of the individual channel. The increased effective sampling rate of the time-interleaved system is more easily observed visually in Fig. 2-6. In this illustration, a 4-way time-interleaved system is shown. The delta train $d_{s,i}(t)$ is the sum of the delta trains for each of the channels. The delta train of each of the channels is separated by a delay of $T/4$ from each other, where $T$ is the sampling period of each channel. The time domain output, $x_s(t)$ shows the summed output of the time-interleaved sampler, in which the input signal appears to be sampled with a sampling period of $T/4$ rather than $T$, showing that time-interleaving can effectively increase the sampling rate by the interleaving-factor of the system.
Figure 2-6: Time-domain representation of a 4-way time-interleaved system. The sampling period of each delta train is $T$, but the delta train for each channel is shifted by $T/N$ from each other. The sampled output, $x_s(t)$ appears to be sampled with a sampling period of $T/4$.

2.2.2 Multi-Channel, Time-Interleaved Sampling in Frequency-Domain

While the time-domain analysis presented above is useful in illustrating the operation of an ideal time-interleaved ADC, understanding the issues in real time-interleaved ADCs is more readily achievable by analyzing the operation of the time-interleaved ADC system in the frequency domain.

As expressed in Eq. 2.4, the Fourier transform of $x_{s,i}(t)$, which is the sampled output of each channel, can be expressed as,

$$X_{s,i}(j\omega) = X(j\omega) \ast D_i(j\omega),$$  \hspace{1cm} (2.11)

where $D_i(j\omega)$ is the time-shifted delta train signal. Using the time-shifting property of Fourier transforms, $D_i(j\omega)$ is given by,

$$D_i(j\omega) = D(j\omega) \cdot e^{-j\omega \frac{T}{N}},$$  \hspace{1cm} (2.12)
Given the above result, the Fourier transform of the sampled signal for each channel, \( x_{s,i} \), is given by,

\[
X_{s,i}(j\omega) = X(j\omega) \ast D(j\omega) \cdot e^{-j\omega \frac{2\pi i}{T}} \\
= \frac{1}{T} \sum_{k=-\infty}^{\infty} X \left( j(\omega - \frac{2\pi k}{T}) \right) \cdot e^{-j\omega \frac{2\pi i}{T} k}
\]  

(2.13)

The Fourier transform of the multiplexed output of the time-interleaved system, \( Y(j\omega) \) is then given by,

\[
Y(j\omega) = \sum_{i=1}^{N} X_{s,i}(j\omega) \\
= \frac{1}{T} \sum_{i=1}^{N} \sum_{k=-\infty}^{\infty} X \left( j(\omega - \frac{2\pi k}{T}) \right) \cdot e^{-j\omega \frac{2\pi i}{T} k}.
\]  

(2.14)

Eq. 2.14 can be recast as,

\[
Y(j\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X \left( j(\omega - \frac{2\pi k}{T}) \right) \cdot E(j\omega),
\]  

(2.15)

where \( E(j\omega) \) is defined as,

\[
E(j\omega) = \sum_{i=1}^{N} e^{-j\omega \frac{2\pi i}{T} k} \\
= \sum_{i=1}^{N} e^{-j2\pi \frac{i}{N} k}.
\]  

(2.16)

From Eq. (2.16), we can observe that for integer values \( k/N \), \( E(j\omega) \) evaluates to \( N \), and for non-integer values of \( k/N \), \( E(j\omega) \) evaluates to zero. Given the result above, we can rewrite \( Y(j\omega) \) as follows,

\[
Y(j\omega) = \frac{N}{T} \sum_{m=-\infty}^{\infty} X \left( j(\omega - 2\pi m \frac{N}{T}) \right), \text{ where } m = \frac{k}{N}.
\]  

(2.17)
By comparing Eq. (2.17) to Eq. (2.6) above, we can see that time-interleaved sampling produces the same output as a single-channel system with a sampling period of $T/N$ instead of $T$. Thus the time-interleaved system has an effective sampling period of $T/N$ rather than $T$, which corresponds to an increase in the sampling frequency by a factor of $N$, the number of channels in the time-interleaved system.

The way in which time-interleaving results in an effectively higher-sampling system is more intuitively explained using diagrams. In Fig. 2-7, the frequency representation of a two-way time-interleaved systems is shown. In a two-way time interleaved system, the output of each channel is,

$$X_{s,0}(j\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X \left(j(\omega - \frac{2\pi k}{T})\right)$$

$$X_{s,1}(j\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X \left(j(\omega - \frac{2\pi k}{T})\right) \cdot e^{-j\omega \frac{T}{2}}$$

$$= \frac{1}{T} \sum_{k=-\infty}^{\infty} X \left(j(\omega - \frac{2\pi k}{T})\right) \cdot e^{-j\pi k}. \quad (2.18)$$

In the equation for $X_{s,1}(j\omega)$, for all odd values of $k$, the exponential $e^{-j\pi k}$ evaluates to $-1$, while for all other values of $k$, the exponential $e^{-j\pi k}$ evaluates to $1$. From Fig. 2-7, we can clearly observe, when the output of both channels are summed, in an ideal situation, the odd images of the spectrum of $X(j\omega)$ cancel each other and we are only left with the spectrum of $X(j\omega)$ spaced by $T/2$. As a result, $Y(j\omega)$, which is the sum of $X_{s,0}(j\omega) + X_{s,1}(j\omega)$, results in a spectrum with no aliasing.

### 2.3 The Effect of Channel Errors on the Performance of Time-Interleaved ADC Systems

In the previous section we described how a time-interleaved system can be used to achieve a higher sampling speed for the entire system. In other words, we saw how $N$ channels each sampling at $f_s$ can be used to mimic a single channel running at $N \cdot f_s$. This increase in performance is achieved by employing parallelism and requires the
Figure 2-7: Frequency-domain representation of a 2-way time-interleaved system. While the spectrum of each of the channels show aliasing in the frequency spectrum, the summed output does not suffer from aliasing. The images centered at $\pm \pi, \pm 3\pi, ...$ from the odd and even channels (denoted using a shaded image) cancel each other out, leaving the resulting spectrum free of aliasing artifacts.
use of four separate ADCs. In real implementations, there will always be systematic errors between the channels. As a result, in real time-interleaved systems, errors such as offset mismatch, gain mismatch, and sampling clock skew between ADC channels will result in the performance degradation of the overall time-interleaved system. In this section, we look at the effect of ADC channel errors on the overall time-interleaved ADC performance.

### 2.3.1 Offset Mismatch Between Channels

The effect of offset mismatch between channels in a time-interleaved ADC can be modeled as an additive voltage source, $V_{o,i}$. A block diagram that includes offset mismatch between channels is illustrated in Fig. 2-8. The presence of offset in itself does not pose a major problem for the ADC. If the offset for all channels are the same and input signal independent, the offset error can be easily corrected in the digital domain or even ignored. However, when the offsets between channels are mismatched, it causes a time-varying error in the time-interleaved system, which leads to the presence of spurious tones in the frequency spectrum of the output of the time-interleaved system.

The block diagram in Fig. 2-8 is used to model the output of each channel. By assuming a different offset voltage, $V_{o,i}$, of up to $\pm 10$ LSB for each 12-bit resolution channel, the time domain error resulting from offset mismatches is shown in Fig. 2-9. If we further assume that the variation in the offset of each ADC is independent of the input signal characteristics, the errors introduced by offset mismatch between channels is constant over the signal period, i.e. the offset mismatch produces fixed amplitude and frequency noise that is independent of the sampling frequency or the input amplitude of the time-interleaved ADC.

We can evaluate the effect of offset mismatches in the frequency domain by evaluating the Fourier transform of $x_{s,i}(t)$. With offset error, $x_{s,i}(t)$ now becomes,

$$x_{s,i}(t) = (x(t) + V_{o,i}) \cdot d_i(t)$$  \hspace{1cm} (2.19)
Figure 2-8: Block diagram of a time-interleaved ADC with offset mismatch between channels modeled as an additive voltage source, $V_{o,i}$, at the input of the $i$-th channel.

Figure 2-9: Behavioral simulation results of the time domain error due to offset mismatches between channels. The plot in panel (a) shows a 12-bit output of a 4-way time-interleaved ADC with offset mismatch between channels. The graph in panel (b) shows the error in LSBs of the time-interleaved system with offset mismatch from the ideal, expected output code when no offset mismatch between channels is present.
The Fourier transform of $x_{s,i}$ with offset is then given by,

$$X_{s,i}(j\omega) = (X(j\omega) + V_{o,i}) \ast D_i(j\omega)$$

$$= \frac{1}{T} \sum_{k=-\infty}^{\infty} X \left(j(\omega - \frac{2\pi k}{T})\right) e^{-j\omega \frac{k}{T}} + V_{o,i} \cdot D_i(j\omega). \quad (2.20)$$

Using Eq. 2.17, the sum of the output of all the channels can then be expressed as,

$$Y(j\omega) = \frac{N}{T} \sum_{m=-\infty}^{\infty} X \left(j(\omega - 2\pi m \frac{N}{T})\right) + \sum_{i=1}^{N} V_{o,i} \cdot D_i(j\omega) \quad (2.21)$$

From equation 2.21, we can expect that the offset error shows up as tones in the frequency domain at intervals of $1/T$ (since $D_i$ is periodic in $1/T$), where $T$ is the sampling rate of each channel. The location of these spurs in the frequency domain are at,

$$f_{n,o} = i \cdot f_s \quad \text{where } i = 1, 2, 3, ... \quad (2.22)$$

where $f_s$ is the sampling frequency of each interleaved channel.

An example of the effect of offset mismatch on the time-interleaved system using behavioral simulations is shown in Fig. 2-10. Here, a frequency spectrum of a simulated 4-way time-interleaved ADC is shown with each channel having a different offset. The spurious tones appear at multiples of the channel sampling frequency as expected from the theory.

Again, it is important to note that the tones caused by the offset mismatch between channels is independent of the input signal and only dependent on the channel sampling period and the offset mismatch between channels.

### 2.3.2 Gain Mismatch Between Channels

The effect of gain mismatch between channels in a time-interleaved system can be modeled by using a gain element with gain values, $G_i$, for each channel. A block diagram of a time-interleaved system that includes the gain mismatch between channels...
Figure 2-10: A frequency spectrum of the output of a behavioral simulation of a 4-way time-interleaved ADC with offset mismatches between channels. The spurs at $f_s/2$ and $f_s/4$ are due to the offset mismatches between channels in the time interleaved ADC.
Figure 2-11: Block diagram of a time-interleaved ADC with gain mismatch between channels modeled as an gain block with gains, $G_i$, before the input of the $i$-th channel.

is illustrated in Fig. 2-11.

The block diagram in Fig. 2-11 is used to model the output of each channel. To simulate the effects of gain mismatches between channels in a time-interleaved ADC, behavioral simulations are run for a 4-way time-interleaved ADC with different gains, $G_i$, of up to $\pm 5\%$ for each channel. The time domain error resulting in the gain mismatches between channels is shown in Fig. 2-12. The time-domain error plot shows that the error is greatest at the peaks and troughs of the output signal. This makes intuitive sense as the error between a channel with a non-unity gain and an ideal channel with unity gain is greater as the input signal amplitude increases.

As in the case with offset mismatches between channels, it is instructive to analyze the effect of gain mismatches between channels in the frequency domain. This can be done by writing the Fourier transform of $x_{s,i}$ with the gain elements included.

$$ x_{s,i}(t) = G_i \hat{x}(t) \cdot d_i(t) \quad (2.23) $$
Figure 2-12: Behavioral simulation results of the time domain error due to gain mismatches between channels. The plot in panel (a) shows a 12b output of a 4-way time-interleaved ADC with gain mismatch between channels. The graph in panel (b) shows the error in LSBs of the time-interleaved system with offset mismatch from the ideal, expected output code.

The Fourier transform of $x_{a,i}$ with gain elements included is then given by,

$$X_{s,i}(j\omega) = G_i X(j\omega) * D_i(j\omega)$$

$$= \frac{1}{T} \sum_{k=-\infty}^{\infty} G_i \cdot X(\omega - \frac{2\pi k}{T}) \cdot e^{-j\omega \frac{2\pi k}{T}}.$$  \hspace{1cm} (2.24)

The sum of the output of the channels can then be expressed as,

$$Y(j\omega) = \sum_{i=1}^{N} X_{s,i}(j\omega)$$

$$= \frac{1}{T} \sum_{k=-\infty}^{\infty} G_{tot}(j\omega) \cdot X(\omega - \frac{2\pi k}{T})$$  \hspace{1cm} (2.25)

where $G_{tot}$ is defined as,

$$G_{tot}(j\omega) = \sum_{i=1}^{N} G_i \cdot e^{-j\omega \frac{2\pi k}{T}}.$$  \hspace{1cm} (2.26)
Eq. 2.26 will reduce to Eq. 2.16 if all $G_i$'s are unity. However, when the gains are mismatched, this results in the spectral images of the interleaved channels to not cancel out perfectly for non-integer values of $k/N$.

When the input signal is sinusoidal, gain mismatches between the channels result in tones in the frequency domain. The location of these spurious tones, $f_{n,g}$, are at,

$$f_{n,g} = i \cdot f_s \pm f_{in} \quad \text{where } i = 1, 2, 3, ...$$

(2.27)

where $f_{in}$ is the frequency of the input sinusoidal signal and $f_s$ is the sampling frequency of each of the interleaved channels in the time-interleaved system. An example of the presence of these tones in the frequency domain of a 4-way time-interleaved ADC is shown in Fig. 2-13.
2.3.3 Sampling Clock Skew Between Channels

Besides offset and gain mismatches between channels, time-interleaved systems also suffer from the negative effects of sampling time skew between the interleaved channels, which can limit the performance of time-interleaved ADC systems. Ideally, each of the ADC channels in a time-interleaved ADC sample the input signal sequentially with an interval of $T/N$ between successive samples. However, if sampling time skew is present, the sampling edges of each channel is altered slightly by $\Delta t_i$ as shown in the block diagram in Fig. 2-14. The presence of sampling skew can be modeled by altering the sampling delta train function, $d_i(t)$, such that,

$$\hat{d}_i(t) = \sum_{k=\infty}^{\infty} \delta \left( t - kT - \frac{T}{N}i - \Delta t_i \right), \quad (2.28)$$

The effect of sampling time skew between channels in the time domain is illustrated in Fig. 2-15, which is a behavioral simulation output of a 4-way time-interleaved ADC system operating at a combined sampling frequency of 200 Ms/s with sampling clock skew. The sampling clock skew between channels was randomly chosen to be between $\pm 5$ps. From the amplitude error plot in Fig. 2-15(b), we can observe that the error is
Figure 2-15: Behavioral simulation results of the time domain error due to sampling clock skew between channels. The plot in panel (a) shows a 12-bit output of a 4-way time-interleaved ADC with sampling clock skew between channels. The graph in panel (b) shows the error in LSBs between output of the time-interleaved system with sampling clock skew and an the expected output code for an ideal, skew-free, time-interleaved system.
Figure 2-16: The same sampling clock skew results in a voltage error proportional to the gradient of the input error during the sampling instant.

greatest during the zero-crossing regions of the sinusoidal input signal. The diagram in Fig. 2-16 illustrates this more clearly. When the gradient of the input signal is low, the error due to timing skew is small, and when the gradient of the input signal is high, the error due to sampling time skew is large.

For a sinusoidal input, $x(t)$, where,

$$x(t) = A \sin(2\pi ft),$$

and assuming that $\Delta t_i \ll T$, (i.e. the sampling clock skew is much smaller than the sampling period of the time-interleaved converter), then the voltage error due to timing skew can be approximated by,

$$\Delta V(t) = \frac{dx}{dt} \cdot \Delta t_i$$

$$= 2\pi f \Delta t_i \cdot A \cos(2\pi ft)$$

The expression for the error due to sampling time skew above is dependent not only on the magnitude of the sampling skew, but also on the frequency of the input signal. Consequently, the errors due to sampling time skew increases with the input
signal bandwidth of the time-interleaved system.

In the frequency domain, the effect of sampling time skew can be analyzed by writing the Fourier transform of each channel's output. First, with timing skew, \( x_{s,i} \) can be expressed as,

\[
x_{s,i} = x(t) \cdot \hat{d}_i(t)
\]  

(2.31)

The Fourier transform of \( x_{s,i} \) with timing skew is then given by,

\[
X_{s,i}(j\omega) = X(j\omega) \cdot \hat{D}_i(j\omega)
\]

\[
= \frac{1}{T} \sum_{k=-\infty}^{\infty} X \left( j(\omega - \frac{2\pi k}{T}) \right) \cdot e^{j\omega \frac{T}{N} tk} \cdot e^{-j(\omega - \frac{2\pi k}{N})\Delta t_i}.
\]  

(2.32)

The sum of the output of the interleaved channels can then be expressed as,

\[
Y(j\omega) = \sum_{i=1}^{N} X_{s,i}(j\omega)
\]

\[
= \frac{1}{T} \sum_{k=-\infty}^{\infty} \hat{E}(j\omega) \cdot X \left( j(\omega - \frac{2\pi k}{T}) \right)
\]  

(2.33)

where \( \hat{E}(j\omega) \) is defined as,

\[
\hat{E}(j\omega) = \sum_{i=1}^{N} e^{j\omega \frac{T}{N} tk} \cdot e^{-j(\omega - \frac{2\pi k}{N})\Delta t_i}.
\]  

(2.34)

We can observe that when \( \Delta t_i \) is zero for all the channels (the case where there is no timing skew), Eq. 2.34 reduces to Eq. 2.16, resulting in the images of \( X(j\omega) \) to cancel when \( k/N \) is non-integer. However, when timing skew is present, the images of \( X(j\omega) \) when \( k/N \) is non-integer do not cancel completely, resulting in frequency artifacts that degrade the performance of the time-interleaved system.

As before, for time-interleaved ADC characterization, it is instructive to see the output spectrum when the input is a sinusoid. When a sinusoidal input is applied to a time-interleaved system with sampling time skew, the spurious tones will appear in
Figure 2-17: A frequency spectrum of the output of a behavioral simulation of a 4-way time-interleaved ADC with sampling clock skew between channels. The pair of spurs spaced by $\pm f_{in}$ around $f_s/4$ and $f_s/2$ are due to the sampling clock skew between channels in the time interleaved ADC.
the frequency spectrum at,

\[ f_{n,\text{skew}} = i \cdot f_s \pm f_{\text{in}} \quad \text{where } i = 1, 2, 3, \ldots \] (2.35)

where \( f_{\text{in}} \) is the frequency of the input sinusoidal signal and \( f_s \) is the sampling frequency of each of the channels in the time-interleaved system. An example of the presence of these tones in the frequency domain of a 4-way time-interleaved ADC is shown in Fig. 2-17. Here a behavioral simulation is performed to model a 4-way time-interleaved ADC. Randomized sampling clock skew is added to the individual channels and the spectral output of the converter is computed. The spur pairs around \( f_s/4 \) and \( f_s/2 \) are due to the effects of sampling clock skew between channels.

### 2.4 Comparison of Channel Mismatches

All three mismatch effects discussed above introduce spurs in the frequency domain that degrade the overall SNDR of the time-interleaved ADC system. In Fig. 2-18, behavioral simulation results for a 12-bit, 4-channel, time-interleaved ADC at a combined sampling frequency of 200 MS/s shows the impact of the various mismatches between the interleaved channels on the SNDR of the overall ADC. At a fixed input amplitude, the SNDR degradation due to gain and offset mismatch between channels is independent of the input frequency of the overall time-interleaved system. However, in the case for sampling time skew, the overall SNDR keeps degrading with increasing input frequency.

Thus, in order to increase sampling rates of high resolution ADC systems by utilizing time-interleaving architectures, the effects of timing skew must be mitigated. While gain and offset mismatch should be minimized to improve the overall ADC performance, they can be removed by relatively straightforward mixed-signal and digital calibration techniques. In this research project, the main focus is to mitigate the sampling time skew in time-interleaved ADCs as it poses the main challenge in achieving high speed and high-resolution time-interleaved ADCs.
Figure 2-18: SNDR vs input signal frequency for a 4-way, 12-bit, time-interleaved ADC with offset mismatch, gain mismatch, and sampling clock skew between the interleaved channels.
Chapter 3

Sampling Clock Skew Errors in Time-Interleaved ADCs

In Chapter 2, we mathematically analyzed the operation of time-interleaved ADCs, and explored the effects of mismatches between channels and their contribution to the performance degradation of time-interleaved ADCs. In this chapter, we turn our attention to errors due to the effects of sampling clock skew between channels and the challenges they pose to the design of high-speed and high-resolution time-interleaved ADCs.

We begin this chapter by looking at some of the major sources of sampling clock skew errors in time-interleaved ADC designs. We then discuss current methods for the detection of sampling clock skew between channels in a time-interleaved system. We end this chapter by surveying the state-of-the-art methods that have been proposed in the literature thus far for the correction or calibration of sampling clock skew in time-interleaved ADCs.

3.1 Major Sources of Sampling Clock Skew

The sources that contribute to sampling clock skew between channels in a time-interleaved ADC can be broadly divided into skew caused by the active elements (i.e. transistor variations) and on-chip clock and signal routing (e.g. wiring resistances
Figure 3-1: Sampling clock for a time-interleaved ADC. $CLK_{FS}$ is the full-speed clock. The $i$-th interleaved channel is sampled on the falling edge of $Clk_i$.

3.1.1 Skew Due to Transistor Variations

The operation of a time-interleaved ADC relies on a clocking scheme whereby each of the channels samples the input signal in a sequential manner. As we have seen in Chapter 2, if each channel is operating with a sampling period of $T$, an effective sampling rate of $N/T$ can be achieved by using each channel to sample the input signal at intervals of $T/N$ apart. This sampling scheme is illustrated in Fig. 3-1 and requires the generation of multi-phase clocks that are separated by a delay of $T/N$. These multi-phase clocks are often times derived from a high-speed clock operating with a period of $T/N$. A simplified version of a multi-phase clock generator is shown in Fig. 3-2. The enable signals, $EN_i$'s, are activated in a token-passing fashion whereby only a single enable signal is ON at any one time and is activated sequentially. After the multi-phase clocks are generated, they are usually buffered before being routed throughout the chip to each of the individual channels as illustrated in 3-2. The buffers at the output of the multi-phase clock generation block are used to drive the long interconnects between the multi-phase clock generation circuits and the
Figure 3-2: Circuit schematic of a multi-phase clock generator. The output of each AND gate is buffered to drive the clock routing interconnects to the individual channels.

individual channels in a time-interleaved ADC.

Process variations (e.g. variations in gate oxide thickness and doping concentrations) result in shifts in the transistor’s threshold voltage. In addition, process variations can also lead to variations in the dimensions (W/L) of the transistor. Variations in the threshold voltages and transistor dimensions, in turn, result in varying ON-resistances of the transistors used in the multi-phase clock generation circuit and the following clock buffers. Consequently, this variation in transistor ON-resistances causes variations in the propagation delays of the logic gates and contributes to sampling clock skew.

One method to reduce variation in the sampling clocks is to increase the size of the individual transistors used in the multi-phase clock generation circuits since. This is because from [11], we know that the standard deviation in the threshold-voltage, $V_T$, of a transistor is given by,

$$\sigma_{\Delta V_T} = \frac{A_{V_T}}{\sqrt{WL}}, \quad (3.1)$$

where $A_{V_T}$ is a process dependent constant, and $W$ and $L$ are the width and length of the transistor respectively.

Therefore, from Eq. 3.1, the variation in the threshold voltage of a transistor is inversely proportional to the transistor area. However, just increasing the size of the
active devices in order to minimize transistor threshold variations, and as a result, reduce sampling clock skew comes at the cost of increased power consumption since larger transistors have larger associated parasitic capacitances that need to be charged and discharged every clock cycle.

Besides the skew from transistor variations in the clock path, variations in the sampling switch of each of the interleaved channels is another source of sampling clock skew as shown in Fig. 3-3. The ON-resistance of the sampling switch during the track phase is given by,

\[ R_{ON} = \frac{1}{\mu_n C_{ox} W L (V_{DD} - V_{in} - V_T)}, \]  

(3.2)

where \( \mu_n \) is the electron mobility in silicon, and \( C_{ox} \) is the gate oxide capacitance per unit area.

Threshold voltage variations in the sampling transistor results in changes in the ON-resistance during the track phase of the track-and-hold circuit. The sampling transistor ON-resistance in conjunction with the sampling capacitor, \( C_S \), form a low-pass filter network that introduces an group delay of \( R_{ON} C_S \) delay between \( V_{out} \) and \( V_{in} \). Therefore as \( R_{ON} \) varies, the delay also varies, resulting in sampling clock skew between channels in a time-interleaved ADC. In addition to sampling clock skew, variations in \( R_{ON} C_S \) also contributes to amplitude mismatch between channels as the magnitude gain of the low-pass input sampling network is dependent on the corner frequency and the input signal frequency. However, if the corner frequency of the

Figure 3-3: Circuit schematic of the input sampling network during the track phase.
input sampling network is significantly larger than input signal bandwidth, as would be expected in Nyquist rate ADCs, the amplitude mismatch can be expected to be negligible. A detailed analysis of the effect of magnitude mismatches introduced by variations in the input network bandwidth is presented in Chapter 4.

3.1.2 Skew Due to Variations in Clock and Signal Routing

Besides the contribution of transistor process variations on sampling clock skew in time-interleaved ADCs, variations in the on-chip clock and signal routing is another contributing factor to sampling clock skew in time-interleaved ADCs. Two different possibilities for clock and signal routing in a time-interleaved converter (among many others that have been proposed in the literature) is shown in Fig. 3-4. While the linear routing diagram shown in Fig. 3-4(a) can result in a more efficient space usage, the clock and signal wiring lengths between channels are not equal. Consequently, the wiring length mismatches result in variations in the capacitive loading and interconnect resistances for the various clock and signal paths. This leads to propagation delays and sampling clock skew between channels. An improvement to the linear routing scheme is the H-tree routing scheme illustrated in Fig. 3-4(b). In the H-routing scheme, the signal and clock lengths between channels are matched. The trade-off in this routing scheme is in area as extra area needs to be dedicated for the H-tree clock routing network.

Even when highly symmetric routing schemes (e.g. using H-tree routing schemes) are used to route the individual clocks to each converter in a time-interleaved ADC, variations in the on-chip interconnects can contribute to sampling clock skew. The variations in the interconnect widths, thicknesses, lengths, height above the substrate, and via resistances all result in variations of the wiring resistances and capacitances. These variations in the resistance and capacitance of the interconnects result in variations in the propagation delay of the clock and input signals even when symmetric routing schemes are utilized.

In this section, we explored some of the major causes of sampling clock skew in time-interleaved ADCs. In the rest of this chapter, we will explore some of the
Figure 3-4: Possible routing schemes for the input signal and clock in a time-interleaved ADC: (a) input signal and clocks are routed linearly, (b) input signal and clocks are routed in a H-tree fashion.
methods available in the literature to mitigate the effects of sampling clock skew that are available to time-interleaved ADC designers of today. These include time-interleaved ADC architectures that are insensitive to sampling clock skew and circuits for the calibration and correction of sampling clock skew.

3.2 Sampling Clock Skew Insensitive Time-Interleaved Architectures

3.2.1 Two-Rank Track-and-Hold

It is possible to make a time-interleaved ADCs insensitive to sampling clock skew between channels by using a global front-end track-and-hold (T/H) sampling block as shown in Fig. 3-5(a). The global T/H samples the input signal at a sampling frequency of \( N \cdot f_s \) and its output is fed into all the sub-channel ADCs [12]. Each interleaved channel has a second T/H stage that is used to sample the output of the global T/H at the sub-channel sampling frequency of \( f_s \). The associated timing diagram of a time-interleaved ADC with a global T/H is illustrated in Fig. 3-5(b). As before, the sampling edges of each sub-channel is distributed in time such that the effective conversion rate is still maintained at \( N \cdot f_s \). Such architectures are also commonly known as two-rank T/H architectures. Since the second T/H within each sub-channel ADC is essentially sampling a constant signal which is being held by the first T/H, this architecture is insensitive to sampling clock skew as all the sampling edges are defined by the front-end T/H exclusively.

The major drawback of this architecture is the inability of this architecture to scale with respect to increasing number of channels, increasing input signal frequency, and increasing resolution of the converter. All these issues arise due to the limitation of the the front-end T/H block. The front end T/H needs to operate at the highest sampling rate of the time-interleaved system. As the number of interleaved channels increase, the loading on the front-end sampler increases, thus limiting its bandwidth. As a result, the required settling time constraint becomes harder to achieve, especially
Figure 3-5: Time-interleaved ADC with a global T/H that samples the input signal. The T/H within each interleaved channel re-samples the output of the global T/H.
in high-resolution ADCs. This requirement of the front-end T/H having to operate at the full speed of the time-interleaved converter defeats some of the purpose of interleaving in the first place, that is to simplify the design of high-speed and high-precision converters by relaxing the maximum operating speed of any one circuit block.

3.2.2 Channel Randomization

Another method to reduce the effect of sampling time skew is to randomize the time-interleaved ADC channels as shown in Fig. 3-6. By having more channels than necessary to achieve the required sampling frequency, channels can be swapped in and out at random. This technique “smears” the spurs produced by sampling time mismatch between channels. The randomization de-correlates the errors of each individual channel. Therefore, an improvement in SFDR can be expected. However, no improvement in SNDR should be expected as the total noise generated due to sampling time skews is not reduced, but is spread out across the frequency spectrum such that it appears to look more like noise [13]. If improvement of the SNDR is also required, then spectral shaping and filtering techniques can be applied in conjunction with a channel randomization strategy to increase the SNDR of the time-interleaved ADC [14].

3.3 Calibration and Correction Methods to Mitigate Effects of Sampling Clock Skew

While in the previous section we looked at architectures that can make a time-interleaved ADC less sensitive to sampling clock skew, these techniques often come at a significant performance, power, and area cost. Therefore, the canonical architecture as shown in Fig. 1-3 is often used, which can be affected by sampling clock skew. In high-speed and high-resolution applications, where the effects of sampling clock skew can severely degrade the performance of the time-interleaved ADC and need to be
Figure 3-6: Time-interleaved ADC with channel randomization to mitigate sampling clock skew. $P$-channels are used to implement an $N$-way time-interleaved ADC, where $P > gt N$. 
mitigated, calibration and correction techniques can be employed. In this section, we present some of the techniques available in the literature today to mitigate the effects of sampling clock skew in time-interleaved ADCs.

The process of mitigating the effects of sampling clock skew can be broken down into two separate phases, namely the detection phase and the correction phase as shown in Fig. 3-7. In the detection phase, the sampling clock skew between channels is estimated. This is usually done by processing the output data stream from the time-interleaved ADC system to extract the relative clock skews between channels. In the calibration or correction phase, the effect of sampling clock skew is mitigated either by calibrating out the sampling clock skew in the analog domain, or by correcting the output data stream to reduce the effect of sampling clock skew.

3.3.1 Sampling Clock Skew Detection

In Chapter 2, we observed how sampling clock skew introduces spurs in the frequency response of a time-interleaved ADC. In order to calibrate or correct the effects of sampling clock skew, the relative skew between channels needs to be measured first. There are two ways in which sampling clock skew detection can be carried out: in the foreground or in the background.

Foreground sampling clock skew detection techniques require that the ADC be
taken off-line for sampling clock skew detection and calibration [15–18]. The converter is then returned to normal operation after detection and calibration. When the converter is off-line, a known test signal is input into the time-interleaved ADC and the sampling clock skew detection algorithm is run. While the requirement that the converter be taken out of operation momentarily may not be feasible in some applications, the benefit of foreground sampling clock skew detection and calibration is that the efficacy of the detection algorithm does not depend on the specific use case of the time-interleaved converter. In particular, since a known test input is used during the skew detection phase, there needs to be no constraints on the input signal to guaranteed the efficacy of the skew detection algorithm.

Background sampling clock skew detection methods, on the other hand, do not require that the converter be taken offline during the detection phase. The skew detection algorithm runs while the converter is in normal operation. This quality is highly desirable in systems that cannot be easily taken off-line without severely inhibiting the quality of service (e.g. communication links). The drawback of background sampling clock skew detection schemes is that the detection algorithm usually places some requirements on the characteristics of the input signal during the detection and calibration phase [7,9,18]. Therefore, care has to be taken when choosing a particular detection algorithm to ensure that skew detection can be guaranteed based on the application in which the converter will be used in.

Detection Using Digital Estimators

Sampling clock skew between channels can be estimated using digital post-processing techniques. One such technique is a background sampling detection scheme presented in [8]. The basis of this detection technique lies in the fact that sampling clock skew results in spurs at frequencies of \( i \cdot f_s \pm f_{in} \), where \( f_s \) is the sampling frequency of each channel in the time-interleaved ADC and \( f_{in} \) is the input frequency to the converter. If the bandwidth of the input signal is bandlimited to less than the Nyquist bandwidth, for a sinusoidal input, a DC value proportional to the amount of sampling clock skew between two channels can be generated by mixing the output of the time-interleaved
ADC with the sampling clock, and mixing the resultant signal again with the output of the time-interleaved converter. By minimizing this DC signal, sampling clock skew between channels can be calibrated out. The main drawback of this method is the limitation on the input signal bandwidth, which is below the Nyquist sampling bandwidth of the converter [8].

Another sampling clock skew detection technique using digital post-processing is described in [9]. While the previous example works well for two or four interleaved channels, the technique developed by [9] can be extended to time-interleaved converters with more than four channels. This technique again limits the input signal to below the Nyquist sampling frequency. However rather than down-converting the output of the time-interleaved ADC, a least-mean-square (LMS) engine to estimate the sampling clock skew by estimating the energy of the out-of-band images introduced by sampling clock skew as shown in Fig. 3-8. In Fig. 3-8(a), when no sampling clock skew is present, there is no out-of-band energy. In Fig. 3-8(b), when sampling clock skew is present, the out-of-band energy increases because the images from the interleaved channels do not cancel out exactly. While this approach can also be run in the background, it places some limitations on the input signal bandwidth. In order for this detection scheme to work well, the input signal needs to be bandlimited to below the Nyquist frequency such that no input signal is present in the frequency
Detection Using Zero-Crossings

Another method to measure sampling clock skew between channels is to count the zero-crossings between the output of the different channels in a time-interleaved ADC [19]. Let us take a 2-way time-interleaved ADC as an example. Let us assume that the input to this time-interleaved ADC is a single tone with a frequency that is an irrational ratio with respect to the sampling clock frequency. In an environment where there is no sampling clock skew, the number of zero-crossings between the sample from Channel 1 to Channel 2 will be equal to the number of zero-crossings between the output of Channel 2 to Channel 1. This is because the elapsed time between when Channel 1 sampled the input and when Channel 2 samples the input is equal to the period between when Channel 2 samples the input followed by Channel 1, as is illustrated in Fig. 3-9 and denoted by $T_{1,2}$ and $T_{2,1}$. However, in the case when Channel 2 suffers from sampling clock skew, and assuming that the skew

Figure 3-9: Zero-crossing for a 2-way interleaved ADC without skew (a), and with timing skew (b).
Figure 3-10: Estimation of sampling clock skew between the ADC under test and the calibration (CAL) ADC. $x(t)$ is the input signal and $\phi$ and $\phi_{\text{cal}}$ are the clocks for the ADC channel under test and the CAL ADC respectively. Source: [21]

is positive, the elapsed time between when Channel 1 samples the input signal and when Channel 2 samples the input signal increases as denoted by $T_{1,2}$ in Fig. 3-9(b). On the contrary the elapsed time between when Channel 2 samples the signal and Channel 1 subsequently samples the input signal decreases. As such, the number of zero-crossings in the first case will increase, while the number of zero-crossings in the latter case will decrease. Therefore, by counting the number of zero-crossings between Channel 1 and Channel 2 and vice versa, the sampling clock skew between channels can be measured.

The main drawback of this detection method is that it requires that the input signal be narrowband and asynchronous to the sampling clock [19]. This requirement on the input signal limits this technique to use in foreground calibration systems where a known input can be used during the sampling clock skew detection and calibration phase. The zero-crossing sampling clock skew detection method can be implemented in the background if a dedicated zero-crossing is added to each channel [20], but comes at the cost of increased design complexity.

Detection Using Cross-Correlation and Variance

In [7], an interesting cross-correlation based technique to detect sampling clock skew in the background was presented and is illustrated in Fig. 3-10. In this scheme, each of
the channels is chosen one at a time for sampling clock skew detection and calibration. An auxiliary “gold-standard” channel is used to sample the input signal in at the same instant as the channel under evaluation. The cross-correlation between the outputs of both the channels is computed. The cross-correlation is maximum when there is no sampling clock skew between the channel under test and the auxiliary channel. Correction methods in the mixed signal domain can then be applied such that the cross-correlation is maximized. The calibration is then performed on the all channels, one-by-one, with the output of the auxiliary ADC used as the “gold standard”. The auxiliary “gold-standard” ADC does not have to operate with the same resolution or speed as the interleaved channels. The only requirement is that the sampling instant of the auxiliary ”gold-standard” ADC is the same as that of the channel being calibrated.

A similar method to [21] of using the signal statistics to estimate sampling clock skew between channels is developed in [22]. In this work, a flash ADC operating at the full speed of the interleaved system is used in conjunction with the interleaved channels. The variance of the output of the flash ADC and the output of the individual channels is computed to estimate the sampling clock skew between. If sampling clock skew is absent, the variance between the output of the flash and the individual channels is minimized.

One of the limitations of the techniques presented in [21, 22] is that the input signal needs to cross at least one of thresholds of the auxiliary ADC in [21] or the flash ADC [22] in order for the calibration system to work. Threshold crossings may also not be detected if the input signal is a multiple of \( f_s \), where \( f_s \) is the sampling frequency of each channel in the time-interleaved ADC. This requirement places limitations on the amplitude and frequency of the input signals during calibration.

Detection Using Signal Mixing

Another method for sampling clock skew detection is presented in [23], which is most easily explained by considering a 2-way time-interleaved ADC. In this detection scheme, the authors recognized that for a two channel time-interleaved ADC, the delay
between samples of two consecutive channels is asymmetrical when sampling clock skew is present, but equal when sampling clock skew is absent. Therefore, by mixing the delayed output of each channel with the output of the other channel, two products will be obtained that contain information about the skew in either direction (positive and negative). The difference in the two products will tend to zero when there is no skew present. Therefore, when used in conjunction with a sampling clock skew calibration scheme, calibration can be achieved by minimizing the magnitude of the difference of the two mixing products. This skew detection method can be extended to more channels by calibration pairs of channels at a time. The main drawback of this approach is that it requires a large number of multiplications to generate the two digitally mixed outputs, which can incur a large area penalty if implemented entirely on chip. The additional digital circuitry can also consume significant power if the detection algorithm is run at full speed. However, in most cases, the detection algorithm can be run at much lower speeds than the converter itself to reduce the power overhead.

3.3.2 Calibration and Correction of Sampling Clock Skew

In the previous section, we discussed a number of methods used to estimate the clock skew between channels in a time-interleaved ADC. Given the estimates of the sampling clock skew, the next step is to calibrate out or correct the sampling clock skew. As illustrated in Fig. 3-7, there are two main methods employed in the literature for sampling clock skew calibration/correction: i) calibrating out the sampling clock skew in the mixed-signal domain, and ii) correcting the effects of sampling clock skew via further processing in the digital domain.

Calibration Using Variable Delay Lines

Clock delay calibration is the most common method used to reduce sampling clock skew in time-interleaved ADC systems [21,24]. The idea is relatively straightforward: a method to introduce a variable delay in the clock signal for each of the channels
Figure 3-11: Popular circuit implementations of variable delay lines. In (a), a variable load capacitance is used to control the delay of $\text{Clkd}$, while in (b) a current-starved inverter is used to control $\text{Clkd}$ by varying $I_{cal}$.

is implemented. Once the sampling clock skew between channels has been detected using any of the sampling clock skew estimation methods presented in the previous section, a delay is introduced in the clock paths with the fastest relative edges in order to minimize the sampling clock skew between channels [21,22].

The two main methods that are commonly used in the literature to implement variable delay lines are illustrated in Fig. 3-11. In Fig. 3-11(a), a variable capacitance is placed at the output of the first inverter. By increasing this variable load capacitance, the clock edge at node $\text{Clk}$ is slowed down. This increases the time when the output of the first inverter crosses the threshold of the second inverter, thus introducing a delay in $\text{Clkd}$. Alternatively, a delay of the clock signal can be realized by controlling the driving strength of the inverter by using a current-starved inverter as shown in Fig. 3-11(b). Here, by reducing $I_{cal}$, the rise and fall times of $\text{Clk}$ can be increased. As a result, the time for $\text{Clk}$ to reach the switching threshold of the second inverter is increased, which introduces a delay in the output of $\text{Clkd}$.

In high-speed and high-resolution ADCs, the jitter performance of the sampling clock is pivotal in ensuring that the ADC performance is not limited by sampling clock jitter. The maximum allowable SNDR for an ADC that is jitter limited is given
by,

\[ SNDR_{\text{max}} = 20 \log \frac{1}{2\pi f_{\text{in}} t_j}, \]  

(3.3)

where \( f_{\text{in}} \) is the input frequency and \( t_j \) is the rms jitter of the sampling clock.

For example, to achieve 12-bit performance for a 100 MHz input signal bandwidth, the cumulative clock jitter needs to be limited to <350fs. As such, care has to be taken in the design of variable delay cells that are placed in the path of the sampling clock. However, almost all the implementations of variable delay lines in the literature results in the slowing down of the sampling clock edge as shown in Fig. 3-12. By slowing down the clock edge, the output of the second inverter becomes more susceptible to jitter due to noise (e.g. thermal noise of the inverter or disturbance of the power supply). Consider the two cases presented in Fig. 3-12. When the input is close to the threshold of the following inverter, for a slowly rising clock edge, the signal spends more time near the threshold of the second inverter. Therefore, any noise close to this point will result in the second inverter tripping sooner or later than expected. By increasing the clock edge slew rate, the amount of time the input of the second inverter is close to the the threshold is minimized, thus reducing the impact of noise and reducing jitter. The increase of clocking jitter with decreasing clocking slew rates has been well documented in application notes from industry [25, 26]. However, one simple way to quantify the increase of clocking jitter with decreasing slew rates is to analyze the amount of time the clock at \( V_x \) spends around the threshold of the second inverter in Fig. 3-12 where it is most susceptible to noise, and is a good proxy for the clock jitter. This time, \( t_{\text{thresh}} \), is given by,

\[ t_{\text{thresh}} = \frac{v_n}{SR}, \]  

(3.4)

where \( v_n \) is the 6\( \sigma \) input-referred rms noise of the second inverter, and \( SR \) is the slew rate of the clock edge. As can be observed from Fig. 3-13, where \( t_{\text{thresh}} \) is plotted for various values of \( SR \) and \( v_n \), where the inverse relationship between slew rate and \( t_{\text{thresh}} \) can be observed. Thus, slowing down the clock edge (lower slew rates) will result in increased jitter.
Figure 3-12: Jitter addition from variable delay lines. When the clock edge is slowed down, it becomes more susceptible to noise near threshold of the re-timing inverter, leading to increased jitter in the output clock.
A secondary drawback of variable delay line based sampling clock skew calibration is the achievable calibration resolution. As the resolution of the time-interleaved converter increases, the maximum allowable sampling clock skew decreases. Variable delay line techniques usually have difficulty in achieving calibration resolutions of less than 100 fs in a power efficient manner. For example, if the variable capacitive load delay line is used, to achieve very small delays, a very small capacitances or large inverters are needed. The minimum switchable capacitance is usually limited by the process (e.g. the minimum gate-to-source and gate-to-drain capacitances). Thus, larger inverters that consume more power are needed to ensure that the required resolution can be achieved.

**Correction Using Digital Filters**

Another technique to mitigate the effects of sampling clock skew is by correcting the output of the time-interleaved converter to remove the effects of sampling clock skew. The main advantage of this techniques is that it operates exclusively in the digital
Figure 3-14: Sampling clock skew correction using a fractional delay filter, $H(z)$

domain, after the conversion is done. As such, no additional circuitry in the analog domain is required, thus relaxing the analog design complexity of each channel. These techniques can also take advantage of the speed and power efficiency gains of digital circuits obtained via technology scaling. The main drawback of digital correction techniques is that they usually have to run all the time, at the full sampling speed of the time-interleaved system, thus requiring a significant power budget. In addition, digital correction techniques can also incur a significant area penalty to accommodate the complex digital circuitry to implement sampling clock skew correction algorithms. For example, in [27], where digital correction is used to mitigate the gain, offset, and sampling clock skew effects between channels, the digital correction block consumes 53% of the total power and 40% of the total active area.

There are two main digital correction techniques that have been presented in the literature, namely correction techniques using i) fractional delay filters, and ii) differentiating filters. In this section, we will discuss briefly the implementation of these correction techniques.

- **Correction Using Fractional Delay Filters:**

  A sampling clock skew correction technique using fractional delay filters is presented in [8]. The main idea behind this technique is that digital processing can be employed to estimate a skew-free sample using the sample with clock skew.

  A block diagram showing such a correction technique is illustrated in Fig. 3-14. The output of the second channel is passed thorough an FIR filter with transfer function $H(z)$. Ideally, the filter $H(z)$ is designed to introduce a delay that is
a fraction of the sampling period, such that,

\[ H(z) = z^{\frac{M}{T}}. \]  

(3.5)

By choosing the coefficients of the fractional delay filter appropriately, the small delay introduced by the fractional delay filter can be used to cancel out the sampling clock skew from that channel.

The main drawback for this technique is the complexity of the fractional delay filters required. In particular, as the sampling frequency and the resolution of time-interleaved converters increases, the complexity of the FIR filters required to implement this technique (i.e. the number of filter taps required) becomes significantly more costly from a power and complexity point of view. For example, in [8] a 21-tap filter is required to sufficiently mitigate the effects of sampling clock skews in a 2-way, 10 bit time-interleaved with input signal frequencies up to 54 MHz.

- **Correction Using Differentiating Digital Filters:**

Another viable sampling clock skew correction technique is presented in [9,27]. For small sampling clock skews between channels (as can be expected if careful design, layout, and on-chip routing is used), the input signal can be expected to be linear within the region of the ideal sampling time and the actual sampling time with clock skew. Therefore, if i) the sampling clock skew, and ii) the slope of the input signal during sampling are known, then the skew-free sample can be estimated using,

\[ \hat{x}[n] = x[n] - t_{s,i} \cdot \left( \frac{dx}{dt} \right), \]  

(3.6)

where \( \hat{x}[n] \) is the skew-free estimate of sample \( x[n] \), and \( t_{s,i} \) is the estimated sampling clock skew for channel \( i \).

The sampling clock skew can be estimated using one of the techniques presented in the previous section. The derivative of the input signal can be estimated using differentiating FIR filters. The differentiating FIR filters need to be designed
with sufficient number of coefficients such that the frequency response of the differentiating filter is accurate for the given Nyquist sampling bandwidth. As in the previous technique, this method can also be costly from a power point of view. For example in the implementation in [27], more than a third of the total time-interleaved ADC power is dedicated to sampling clock skew correction.

3.4 Summary

In this chapter, we looked at the main sources of sampling clock skew in time-interleaved ADC systems. We then discussed the various methods to detect sampling clock skew that are available in the literature today. We concluded the chapter by exploring the various techniques that have been proposed in the literature to correct sampling clock skew either in the mixed-signal or digital domains.

While many different techniques for the detection of sampling clock skew have been published, only a few techniques for the calibration or correction of sampling clock skew can be found in the literature. In the next chapter, we will turn our attention to novel methods for the calibration of sampling clock skew. We will focus on low-complexity sampling clock skew calibration methods that minimize the addition of jitter into the multi-phase clock distribution network, which can limit the performance of high-speed and high-resolution time-interleaved ADCs.
Chapter 4

Proposed Sampling Clock Skew Calibration Methods

In Chapter 2, we analyzed the negative effects of sampling clock skew on the performance of time-interleaved converters. Subsequently, in Chapter 3, we described some of the methods that have been proposed in the literature for the detection and calibration/correction of sampling clock skew between channels. On the calibration front, most techniques that have been published in the literature almost always utilize variable delay lines to control the delay of the sampling clock. Most circuit implementations of variable delay lines invariably add jitter to the clock signal, which can limit the performance of high-speed or high-resolution time-interleaved converters. On the other hand, correction algorithms that post-process the digital output of the converter to correct the sampling clock skew between channels often require complicated additional hardware and can consume significant amounts of power.

As such, there is a need for improved calibration and correction techniques to mitigate the effects of sampling clock skew in time-interleaved converters. Preferably, these techniques should not add significant additional jitter to the sampling clock for each channel. Furthermore, it is preferable that the calibration or correction techniques proposed require minimal overhead in terms of additional circuitry and consuming minimal additional power.

In this chapter, we explore two new methods to mitigate sampling clock skew in
time-interleaved ADCs. The first is called the Rapid Consecutive Sampling method, whereby, for each interleaved channel, two consecutive samples of the input are taken with a short time delay between them. This allows for a straightforward linear interpolation between the consecutive samples in order to recover the de-skewed sample. The second method entails introducing a programmable delay in the input signal path, instead of delaying the clock signal, in order to calibrate out sampling clock skew. Both these techniques are described in more detail in the following sections.

4.1 Rapid Consecutive Sampling (RCS) Method

When all reasonable measures have been taken in the design and layout of the input sampling and clock routing networks, we can expect that the sampling clock skew to be very small when compared to the sampling period of the time interleaved system as shown in Fig. 4-1. Furthermore, in Nyquist-rate converters, the input signal is bandlimited to half the sampling frequency, which limits the maximum gradient of the input signal. Give these two observations, the input signal can be reasonably...
approximated using a linear function between the actual and ideal sampling times as shown in Fig. 4-1. Thus, if the sampling clock skew is known, along with the gradient of the signal when the signal is sampled, linear extrapolation can be used to estimate the skew-free sample.

This correction method can be extended by using two converters to sample the input with a small time delay between each sample as illustrated in Fig. 4-2. The two samples of the input signal are taken such that the first sample is taken before the ideal sampling instant while the second sample is taken after the ideal sampling instant. Given these two samples of the input signal, linear interpolation can be used to estimate a skew-free sample, if the sampling clock skew between channels is known. The delay between the two samples is $\Delta T_{cr}$ and is chosen based on two requirements: i) the maximum calibration range needed, and ii) limiting the maximum error that can be introduced by linear interpolation.
Figure 4-3: Block diagram of a time-interleaved ADC with RCS.

Figure 4-4: Timing diagram of a time-interleaved ADC with RCS.
In order to implement this double sampling architecture to correct for sampling clock skew, each time-interleaved ADC channel is split into two sub-channels as shown in the block diagram in Fig. 4-3 and the corresponding timing diagram for the RCS based time-interleaved ADC is shown in Fig. 4-4. The first sub-channel acquires the first sample of the input signal and the second sub-channel acquires the second, time delayed, sample of the input signal. Both samples are digitized, and linear interpolation is then used to calculate a more accurate estimate of the actual sample at the ideal sampling point. The skew-free sample estimate is then given by:

\[
y_i[n] = \alpha \cdot y_{ia}[n] + (1 - \alpha) \cdot y_{ib}[n], \quad \text{where } 0 < \alpha < 1,
\]

where \(y_i[n]\) is the linearly-interpolated estimate of the \(i\)-th channel in a time-interleaved ADC and \(y_{ia}[n]\) and \(y_{ib}[n]\) are the outputs of the two sub-channels. The variable \(\alpha\) is the interpolation factor between the two samples.

### 4.1.1 Bound on Delay Between Samples

In order to be able to use linear interpolation to mitigate sampling clock skew, the consecutive samples need to be close enough to each other in time, such that the maximum error from linear interpolation is sufficiently small. Higher order interpolation methods would relax this condition, but such techniques would require additional digital circuitry, more processing power, and more information about the input signal (e.g. by using historical samples, signal statistics, etc.).

Numerically, the error due to linear interpolation between two samples is bounded by [28]:

\[
|\Delta V_E| \leq \frac{\Delta T_{cr}^2}{8} \cdot \max \left| \left( \frac{d^2V_{in}}{dt^2} \right) \right|,
\]

where \(V_{in}\) is the input signal and the second derivative of \(V_{in}\) is evaluated in the interval of \(\Delta T_{cr}\). This equation implies that the bound of the error due to linear interpolation is proportional to the second derivative of the input signal. An intuitive way of looking at this is that if the input signal is more “curvy”, then the estimate
Figure 4-5: Error between the estimate of the input signal and the actual value when linear interpolation applied using two samples separated by $\Delta T_{cr}$ for various input signal frequencies.

from linear interpolation can be expected to have a larger error.

Fig. 4-5 shows the maximum interpolation error for various values of $\Delta T_{cr}$. These maximum error bounds were obtained via numerical simulation of a 12-bit ADC with a sinusoidal input signal. The simulation was repeated for various input frequencies. As can be observed from Fig. 4-5, the linear interpolation error increases for larger values of $\Delta T_{cr}$ and with increasing input frequencies. Similarly, the interpolation error can be expected to increase with increasing resolution of the ADC. Ideally, the interpolation error should not contribute more than 0.5 LSB error in order to limit performance degradation from the linear interpolation technique. Therefore, based on the graph in Fig. 4-5, for a 12-bit converter with an input signal bandwidth of 100 MHz, $\Delta T_{cr}$ should be limited to $<50$ps.
4.1.2 Behavioral Simulation Results of the RCS Technique

In order to verify the efficacy of the RCS method to mitigate the effects of sampling clock skew, behavioral simulations were performed in MATLAB. The results of the behavioral simulations are presented in Fig. 4-6. In these simulations, a four-way, 12-bit time-interleaved ADC operating at 200 MS/s with randomly generated sampling clock skews between ±5 ps was modeled. The performance of the ADC was modeled with a 97 MHz input signal. The output spectrum of the time-interleaved ADC before calibration is illustrated in Fig 4-6(a). The time-interleaving spurs are clearly visible and limit the performance of the ADC. The output spectrum of the modeled ADC with the RCS technique used for sampling clock skew calibration is shown in Fig. 4-6(b). A sampling delay of 25 ps was used between the consecutive samples and the interpolation was limited to 8 bits. As can be observed from the output spectrum of the two simulated ADCs, the RCS method is able to greatly mitigate the effects of sampling clock skew with over 15 dB reduction in the spurs due to sampling clock skew between the interleaved channels.
4.1.3 Challenges in Implementing the RCS Technique

There are a number of challenges in implementing the RCS method in order to mitigate the effects of sampling clock skew in a time-interleaved converter. For example, we have seen that the time delay between consecutive samples needs to be quite small to ensure that the interpolation error is minimized. In this section, we explore some of the possible challenges that the designer should be wary of in the implementation of the RCS method described above.

**Input Network Settling**

In the RCS method, the input signal will need to be sampled twice within a very short duration. A simplified circuit model of the input sampling network is illustrated in Fig. 4-7(a) and the input and sub-channel sampling clock waveforms are illustrated in Fig. 4-7(b). The sampling switch is implemented using a transistor. The input source sees two different input impedances during the track and hold phases. As such, the input waveform at ‘X’ will be disturbed right after the sampling switch opens, and will need to settle before the second sub-channel samples the input.
As was shown in Fig. 4-5, in order to limit the interpolation error to less than 0.5 LSB for a 12-bit converter with a maximum input signal frequency of 100 MHz, a maximum delay between samples of 50 ps is allowable. This constraint becomes more stringent as the maximum input signal frequency increases. Thus, the T/H of each sub-channel needs to be carefully designed such that the input signal is not severely disturbed by the opening of the first sampling switch.

If the T/H is designed to have a very fast fall time (on the order of 10-20 ps) in order to allow for as much time as possible for the input to settle, large and power hungry switch drivers will be required. In addition, very fast sampling edges can lead to clock leakage via the parasitic gate-to-drain and gate-to-source capacitances on the input sampling switch. Clock leakage and charge injection from the sampling switch can perturb the input network to corrupt the second consecutive sample. This effect may be exacerbated if the ADC input source has finite input impedance or is limited by parasitics (e.g. bondwire inductances). On the other hand, if a slow switch is used instead (on the order of 50ps) to reduce clock feedthrough and charge injection, the input may not have sufficient time to settle after the first sample is taken. Therefore, special care has to be taken in the design of not only the T/H of each sub-channel but also the input signal network and the ADC driver to ensure that the input signal is able to settle to the required accuracy between the consecutive samples.

**Input Sampling and Comparator Noise**

The input sampling noise can be calculated by modeling the input sampling switch as a resistor, with thermal noise power spectral density of $4kTR$. The rms sampling noise on the input sampling capacitor, $C$, is then given by $\sqrt{kT/C}$ [29].

By using both the converters to sample the input signal, and using a linear interpolation technique, it might appear that the sampling noise will be "averaged out" between the samples. By denoting the sampling noise $\sigma_{sampNoise}$ and assuming that the sampling noise of each channel is equal and uncorrelated, the variance of the sampling noise of the linearly interpolated signal as described in Section 4.1 can be
written as,

\[
\sigma_{\text{sampNoise, total}}^2 = \alpha^2 \cdot \sigma_{\text{sampNoise, 1}}^2 + (1 - \alpha)^2 \cdot \sigma_{\text{sampNoise, 2}}^2 \\
= (2\alpha^2 - 2\alpha + 1) \cdot \sigma_{\text{sampNoise, 1}}^2 \\
= \beta \cdot \sigma_{\text{sampNoise, 1}}^2.
\]

(4.3)

When plotted against \( \alpha \) in Fig. 4-8, we see that the variance in the sampling noise is minimized when \( \alpha \) is 0.5. This is expected because \( \alpha = 0.5 \) corresponds to the arithmetic mean of both samples. However, as the value of \( \alpha \) tends to 0 or 1, the variance of the linearly interpolated output increases. Given this observation, one might conclude that using two sub-channels does not necessarily guarantee noise averaging, thus not allowing a reduction in the capacitance budget per channel. However, the curve in Fig. 4-8 is a parabola and is therefore, flat at the minimum. For small sampling clock skews, the noise averaging between the channels is able to reduce the sampling noise significantly. Only at the extreme ends of the calibration range (which occurs very rarely) will we incur a significant noise penalty. Thus, for a sampling skew calibration range designed to correct a sampling skew distribution of 3\( \sigma \), skews within one sigma of the mean leads to \( \beta = 0.57 \). This indicates that the linearly interpolated noise will increase by only 14% compared to the minimum noise level achievable.

The same argument can be used to analyze the comparator noise. The power and noise budgets for the comparator can be split between the two sub-channels. For nominal sampling clock skews, the comparator noise will be “averaged out” between the two channels. Another option with regards to comparators is to use comparator designs with reconfigurable noise characteristics, whereby the noise can be increased or decreased depending on the interpolation factor being used [30].
4.2 Input Signal Delay Control Method

In the previous section, we explored the RCS method where for each sub-channel, the input is sampled twice with a short delay between them. Linear interpolation is then used to estimate a sampling clock skew free sample of the input. The RCS method greatly simplifies the digital post-processing required to correct for the sampling clock skew. However, as we have seen in the previous section, the T/H for each of the sub-channels needs to be carefully designed in order to ensure that the input signal settles to the required accuracy of the ADC between consecutive samples.

In this section, we explore an alternative method for sampling clock skew calibration by controlling the delay of the input signal instead. Fig. 4-9 shows two situations, one where the clock signal is delayed, and the other where the input signal is delayed instead. In either case, it can be clearly seen that controlling the delay of either the sampling clock signal or the input signal results in the removal of sampling clock skew. Therefore, an alternative way to calibrate out sampling clock skew in a time-
Figure 4-9: Controlling the delay of the input signal as a method to calibrate out sampling clock skew. In (a), the usual method for sampling clock skew calibration is presented. Here the sampling clock edge is shifted such that the effects of sampling clock skew is removed. In (b), sampling clock skew is reduced by delaying/advancing the input signal instead. The square marker indicates the sample when the input signal is delayed/advanced, showing that the effect of sampling clock skew has been mitigated.
interleaved ADC is to not control the delay the sampling clock, but to control the delay of the input signal itself.

The block diagram of such a system is shown in Fig. 4-10. The control bits CalBits_i are chosen appropriately such that the input signal is delayed with the same delay as the skew for Channel_i. By controlling the delay of the input signal appropriately, it can be sampled at a point closest to the ideal sampling instant in a sampling clock skew free system as illustrated in Fig. 4-9. By repeating this process for all channels in a time-interleaved converter, the effect of sampling clock skew on the time-interleaved system can be mitigated.

4.2.1 Fine Input Signal Delay Control Using Low-Pass Filters

In order to be able to employ the input signal delay control method to calibrate out sampling clock skew, we will have to be able to control the delay the input signal with fine enough resolution such that the remaining skew is within the accuracy limit for the speed and resolution of the time-interleaved ADC.

Fig. 4-11 shows the voltage error that results when sampling clock skew is present.
Preferably, the voltage error induced by sampling clock skew, $\Delta V_e$, should be less than 0.5 LSB, which can be expressed as,

$$\Delta V_e < \frac{1}{2} \text{LSB}$$

$$\max \left| \left( \frac{dV_{in}}{dt} \right) \right| \cdot t_{skew} < \frac{1}{2} \text{LSB}. \tag{4.4}$$

For a sinusoidal input, $V_{in} = A \sin(2\pi f_{in}t)$, the maximum allowable skew such that the ensuing voltage error is less than half LSB can be written as,

$$2\pi f_{in} A \cdot t_{skew} < \frac{2A}{2^{N+1}}$$

$$t_{skew} < \frac{1}{2^N (2\pi f_{in})}, \tag{4.5}$$

where $f_{in}$ is the maximum input frequency of the time-interleaved ADC.

In Fig. 4-12, we graph maximum allowable sampling clock skew against input frequencies for time-interleaved ADCs with varying resolutions. Using this plot, we can see that for a converter with resolutions above 12-bits and input frequencies of higher than 100 MHz, it is necessary to be able to control the delay the input signal with a sub-picosecond resolution.

In order to achieve such fine delay control, we can utilize an $RC$ line as shown in
Figure 4-12: Maximum allowable sampling clock skew, $t_{skew}$ vs. input signal frequency, $f_{in}$, plotted for various ADC resolutions.

Fig. 4-13. The output voltage, $v_{out}$, is given by,

$$v_{out}(s) = \frac{v_{in}(s)}{(1 + RCs)}$$

Further, given that the sampling speed is much higher than input signal frequen-

\[ v_{out}(s) = \frac{v_{in}(s)}{(1 + j2\pi f_{in}RC)} \]  

(4.6)

Figure 4-13: Block diagram of a time-interleaved ADC with RCS.
cies, we can assume that $f_{in} \ll \frac{1}{2\pi RC}$. Given this condition, and by setting,

$$v_{in}(t) = A \cdot \sin(2\pi f_{in} t),$$

the following expression for $v_{out}(t)$ can be derived:

$$v_{out}(t) \approx A \cdot \sin(2\pi f_{in} (t - t_D)),$$

where $t_D = RC$, is the effective time delay of the output signal.

This derivation basically says that if the input signal frequency range is significantly lower in frequency than the corner frequency of a low-pass $RC$ network, then the input signals will have a group delay corresponding to $t_D = RC$. Hence, the delay of the input signal can be controlled by varying the product of $RC$. We can now map the $RC$ network to the input sampling network of an ADC, which can be well approximated as a low-pass network. In this mapping shown in Fig. 4-14, $C$ would correspond to the sampling capacitance, $C_s$ of the sampling network, while $R$ would correspond to the ON-resistance of the sampling switch, $M_{samp}$. The sampling capacitance of an ADC is usually carefully designed to meet noise and matching requirements, thus cannot be varied easily. However, the ON-resistance of the input sampling switch can be varied (e.g. by varying the size of $M_{samp}$) such that a well defined time delay, $t_D$, can be added to the input signal. The resolution of the time delay can also be controlled quite finely. For example, given a sampling capacitance of 100 fF, by varying the sampling switch ON-resistance by 1 Ω, a delay of 100 fs can be added to the input signal.

4.2.2 Behavioral Simulation Results of the Input Signal Delay Control Technique

In order to verify the efficacy of the input signal delay control method to mitigate the effects of sampling clock skew, behavioral simulations were performed in MATLAB. The results of the behavioral simulations are presented in Fig. 4-15. In these simula-
Figure 4-14: Mapping the low-pass $RC$ network to the input sampling network of an ADC.

Figure 4-15: Output spectrum of a behavioral simulation for a four-way time interleaved ADC with sampling clock skew. In (a), the output spectrum of the uncalibrated ADC is shown, while in (b) the output spectrum of a calibrated ADC utilizing the input signal delay control technique is shown.
tions, a four-way, 12-bit time-interleaved ADC operating at 200 MS/s with randomly generated sampling clock skews between ±5 ps was modeled. The performance of the ADC was simulated with a 97 MHz input signal. The output spectrum of the time-interleaved ADC before calibration is illustrated in Fig 4-15(a). The time-interleaving spurs are clearly visible and limit the performance of the ADC. The output spectrum of the modeled ADC with the input signal delay control technique used for sampling clock skew calibration is shown in Fig. 4-15(b). A calibration range of 15 ps with a calibration resolution of 6-bits was used. As can be observed from the output spectrum of the two simulated ADCs, the input signal delay control method is able to greatly mitigate the effects of sampling clock skew with over 15 dB reduction in the spurs due to sampling clock skews.

4.2.3 Input Sampling Bandwidth Requirements

Fig. 4-16(a) shows a circuit model of a track-and-hold (T/H) circuit used to sample the input onto a sampling capacitor. The associated waveforms are shown in Fig. 4-16(b). Ideally, as soon as the track switch closes, the output should track the input signal. However, in reality, there is a lag between when the sampling switch closes and when the output actually tracks the input signal due to the finite bandwidth of the T/H. To first order, the bandwidth of the T/H circuit is limited by the sampling switch ON-resistance and the sampling capacitance.

The finite time-constant, $\tau$, of the sampling network imposes a constraint on the maximum input signal frequency and the achievable accuracy for the T/H circuit. From Fig. 4-16, let us assume that the output, $v_{out}$, is initially at ground, and the input is at $V_{FS}$, which is the maximum input voltage for the converter. This is the worst case settling condition whereby the output needs to change by the largest amount in order to track the input. This condition occurs in real converters when the input signal is full-scale and close to the Nyquist sampling frequency of the converter. This occurs when the current sample is completely out-of-phase with the previous sample. The settling time of the T/H can be modeled as an $RC$ circuit as shown in
Figure 4-16: Output settling of the T/H circuit. In (a), the schematic that models the input T/H during the track phase is shown and in (b) the output waveforms for the worst case settling requirement is illustrated.

Fig. 4-16 and the output voltage is described by,

\[ v_{out} = V_{FS} \left( 1 - \exp\left( \frac{-\alpha T_s}{\tau} \right) \right). \]  

(4.9)

The equation above can be rewritten as,

\[ V_{FS} \cdot \exp\left( \frac{-\alpha T_s}{\tau} \right) = V_{FS} - v_{out}. \]  

(4.10)

Consequently, if we expect that the output of the T/H settles to a fraction of an LSB, \( \epsilon \cdot V_{LSB} \), for a converter with a resolution of B-bits, the required settling time is given by,

\[ V_{FS} \cdot \exp\left( \frac{-T_{trk}}{\tau} \right) < \epsilon \cdot V_{LSB}, \]

\[ 2^B V_{LSB} \cdot \exp\left( \frac{-T_{trk}}{\tau} \right) < \epsilon \cdot V_{LSB}, \]  

(4.11)

\[ \frac{T_{trk}}{\tau} > \ln\left( \frac{2^B}{\epsilon} \right). \]

Usually, the T/H output is expected to settle to within 0.5 LSB (i.e \( \epsilon = 0.5 \)) by
Table 4.1: Number of settling time-constants, $M$, required to settle to within half LSB for various ADC resolutions.

<table>
<thead>
<tr>
<th>Bits (B)</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>6.24</td>
</tr>
<tr>
<td>10</td>
<td>7.62</td>
</tr>
<tr>
<td>12</td>
<td>9.01</td>
</tr>
<tr>
<td>14</td>
<td>10.40</td>
</tr>
</tbody>
</table>

the end of the track phase. Thus, the expression above can be reduced to,

$$\frac{T_{trk}}{\tau} > \ln \left(2^{(B+1)}\right) = M.$$  

Eq. 4.12 above gives us the number of settling time constants, $M$, required for the output to settle to 0.5 LSB of the input within the tracking period, $T_{trk}$. The number of required settling time constants is dependent on the number of bits of the converter, $B$. The number of settling time constants for various ADC resolutions is given in Table 4.1. This table shows that in order to achieve high accuracy with high speed operation (short tracking time), the input network time constant needs to be significantly higher than the sampling time.

4.2.4 Effect of Residual Gain Mismatch

The input signal delay control method to calibrate out sampling clock skew relies on changing the input sampling bandwidth in order to vary the delay of input signal as it propagates through the input sampling network. However, one side effect of this method is that varying the input sampling bandwidth, also varies the magnitude of the sampled signal. To see this, consider an input signal of the form $v_{in}(t) = A \cdot \sin(2\pi f_{in}t)$. The output of the sampling network for this signal is given by,

$$v_{out}(t) = \frac{A \cdot \sin(2\pi f_{in}t + \phi)}{\sqrt{1 + (2\pi f_{in})^2 \cdot \tau^2}},$$  

(4.13)
where \( \tau \) is the bandwidth of the input sampling network and \( \phi \) is the phase shift experienced by the input signal. The phase shift is the variable that is being used to calibrate out sampling clock skew. However, as the bandwidth of the sampling network changes, so does the magnitude gain of the sampling network, as can be seen by the presence of \( \tau \) in the denominator of the output signal. Thus, after sampling clock skew calibration, each of the channels can be expected to have slightly varying magnitude gains that appear as gain mismatch between channels.

One method to quantify the effects of these varying magnitudes on the performance of the time-interleaved system, one method is to ensure that SNR of a time-interleaved system with the gain mismatch described above is higher than the SNR due to quantization. The SNR due to quantization, \( SNR_{\text{quant}} \), for an ADC with a resolution of \( B \) bits is given by,

\[
SNR_{\text{quant}} = \frac{3}{2} \cdot (2^{2B}). \tag{4.14}
\]

Thus, we would like to have the SNR of a system with gain mismatches between channels, \( SNR_{\text{gain}} \), to be

\[
SNR_{\text{gain}} \geq SNR_{\text{quant}}. \tag{4.15}
\]

The equation above, can be used to provide a statistical bound on the variance of gains between channels as described in [21], and can be expressed as:

\[
\sigma^2_{\text{gain}} = \left( \frac{N}{N-1} \right) \cdot \left( \frac{2}{3 \cdot 2^{2B}} \right). \tag{4.16}
\]

We can utilize Eq. 4.16 above to compare the variance of a time-interleaved system with different residual gains after sampling clock skew calibration. In Fig. 4-17 the maximum gain variance for a 12-bit, four channel time-interleaved ADC with a maximum input signal frequency of 100 MHz is shown. The graph in Fig. 4-17 is generated in Matlab as follows: (i) A minimum sampling bandwidth, \( f_{lo} \) is chosen. (ii) A maximum sampling bandwidth, \( f_{hi} \), is calculated, assuming that a sampling clock skew calibration range of 15 ps is required. (iii) The maximum variance in gain,
Figure 4-17: The maximum variance in gain mismatch between channels when the entire calibration range is used for sampling clock skew calibration. The maximum variance, $\sigma^2_{gain,skewCal}$, is normalized by $\sigma^2_{gain}$.

$\sigma^2_{gain,skewCal}$ is calculated by calculating the gains at $f_{lo}$ and $f_{hi}$ and calculating the maximum variance of in a four channel system. As can be seen from Fig. 4-17, if a minimum sampling bandwidth of $\sim 3.5$ GHz is used, then the effect of residual gain mismatch can be expected to be smaller than the effect of quantization errors on the ADC performance. However, as was described in Section 4.1.3, the maximum calibration range is designed to cover a range of $3\sigma$ in skew distribution. In Fig. 4-18, we plot the maximum variance of the gain mismatch after sampling clock skew calibration for a sampling clock skew distribution of one standard deviation from the mean. As can be seen from Fig. 4-18, this relaxes the required bandwidth of the input sampling network.

4.3 Summary

In this chapter we have seen two methods for sampling clock skew calibration without using a variable delay line to shift the sampling clock. The first method presented was a linear interpolation method that uses two ADCs per sub-channel to sample
Figure 4-18: The maximum variance in gain mismatch between channels when the calibration range of one standard deviation is used for sampling clock skew calibration. The maximum variance, $\sigma_{gain,skewCal}^2$, is normalized by $\sigma_{gain}^2$.

the input signal with a small time interval between them. The sampling clock skew correction is performed completely in the digital domain with minimal additional circuitry. We also presented a novel calibration technique whereby the input signal delay is controlled rather than the sampling clock as is usually done. We also presented some of the limitations of each of these proposed methods.
Chapter 5

Prototype Time-Interleaved SAR ADC Circuit Design

In Chapter 4, we proposed two methods to mitigate the effects of sampling clock skew in a time-interleaved ADC that does not involve the use of variable delay lines. From the behavioral simulations results presented in Chapter 4, both techniques (RCS and input signal delay) show promise in correcting the sampling clock skew between channels in a time-interleaved ADC without increasing the sampling clock jitter. In this chapter, we describe the implementation of a prototype IC utilizing the input signal delay control method to mitigate the effects of sampling clock skew. The main reason that motivates this choice is that in the RCS method, the input signal is sampled twice with a very small delay between them. This means great care has to be taken to ensure that the input signal that is disturbed by the first sample is able to settle rapidly to within the accuracy requirement of the ADC in the short time between the sub-channel samples. This requirement is affected not only by the on-chip design, but also by the off-chip input network and packaging parasitics, which increases the complexity in the design of high-resolution and high-speed converters. For this reason, the input signal delay control technique described in Section 4.2 was chosen over the RCS method to be implemented in our prototype IC. In this chapter we detail the design of each interleaved ADC, and the multi-phase clock generator circuit block needed to time-interleave the individual converters.
Figure 5-1: T/H implementation for sampling with variable delay of the input signal. In (a), a simplified model of the sampling network that needs to be implemented to introduce controllable delay to the input signal $v_{in}$ is shown. In (b), one possible implementation to change the sampling switch resistance, $R_{sw}$, is shown. More than one sampling switch can be used by turning on multiple $Trk_i$ during the sampling phase.

5.1 Interleaved Channel Converter Architecture

Fig. 5-1(a) shows a model of the T/H needed to implement the input signal delay control method for a generic ADC. In the previous chapter, we have seen that the delay of the input signal can be controlled by varying the resistance of the input sampling switch. The effective resistance of the input sampling switch during the track phase can be made variable by using multiple switches in parallel and choosing to turn on as many as necessary to reduce or increase the delay of the input signal as illustrated in Fig. 5-1(b). Thus, the input signal delay control block can be incorporated into the T/H circuit, allowing for flexibility in choosing the architecture of the converter itself. The only requirement is that one should be able to vary the switch resistance with a resolution fine enough that the effects of sampling clock skew can be mitigated for a given time-interleaved converter resolution and bandwidth as shown in Fig. 4-12.

The ADC architecture that was chosen to test the programmable input signal
delay control technique is the successive approximation register (SAR) architecture. In SAR-based ADC converters, since the T/H is integrated into the sampling DAC, controlling the sampling switch resistance can be integrated into the SAR DAC itself, and each bit line provides a binary control of the input signal delay equal to the resolution of the capacitive DAC. Fig. 5-2 illustrates this idea in more detail. In this implementation, each capacitor in the main capacitive DAC has two sampling lines to sample the input signal, \( v_{in} \). We can either sample with both switches (low-R, when \( Ctr1 \) is high) or sample just using the larger switch alone (high-R, when \( Ctr1 \) is low) when the Trk signal is high, i.e. during the tracking phase of the T/H. Using both switches, the effective resistance of the parallel switches is reduced, thus reducing the overall \( RC \) delay. When only the larger switch is used, the delay is larger. By sizing the smaller switch accordingly, the delay difference between the two sampling scenarios can be designed for. In this scheme, full binary control with a wide range and small delay steps can be achieved by choosing which capacitors are sampled with a high or low-R setting. For example, for a half-way interpolation between both samples, the most significant bit, \( C_8 \) can be sampled using the high-R setting and all other capacitors can be sampled using the low-R setting.
5.2 SAR ADC Block Diagram

The block diagram of the prototype time-interleaved SAR ADC is shown in Fig. 5-3. The prototype time-interleaved ADC is composed of four interleaved SAR channels. Each channel SAR converter is designed to have a resolution of 12 bits. A fully differential architecture is used to minimize common-mode disturbances in the converter. A capacitive DAC is used for the eight MSBs and a resistive string DAC is used for the four LSBs. Asynchronous, self-timed SAR logic is used to avoid the need for a high-frequency bit-cycling clock.
5.3 DAC Design

In SAR ADCs, the capacitive DAC is used for two main purposes: i) to sample the input signal and ii) to generate voltage steps required for the binary search process during conversion. As such, the DAC noise during sampling and linearity during conversion are the main concerns in the design of the DAC. At room temperature, 1 pF of capacitance results in 64μV of voltage noise, which is significantly smaller than the LSB steps of a 12-bit converter operating on a 1V reference. As such, capacitor matching requirements often dictate the size of the DAC unit capacitance in a SAR ADC.

In a canonical binary-weighted SAR DAC, capacitive matching in the chosen process technology often dictates the minimum unit capacitance size for a given ADC resolution. This in turn dictates the size of the entire capacitive DAC, which increases exponentially with increasing resolution of the converter. For example, with a unit capacitance of 10fF, a 12-bit converter will have a DAC with a total capacitance of 20pF. Such a large DAC capacitance value is impractical for three reasons: i) it would not be economically feasible to manufacture such large capacitor arrays, ii) driving this large capacitance during sampling will require power-hungry buffer, and iii) the required DAC settling time during bit-cycling would severely limit its conversion speed.

As such, alternative designs have been proposed to reduce the size of the capacitive array. The most commonly used technique is to employ the use of a split capacitor array as shown in Fig. 5-4. Using this architecture, the total DAC capacitance can be reduced significantly depending on the division between the main- and sub-DACs. For an \( M + N \) bit split capacitor DAC with an \( M \) bit main-DAC and an \( N \) bit sub-DAC, the total capacitance of the DAC as seen by the input is \( 2^M \cdot C_u \), where \( C_u \) is the unit capacitance of the split-capacitor DAC. For example, for a 12-bit DAC with an eight bit main DAC and a four bit sub-DAC, the total capacitance seen by the input is \( 2^8 \cdot C_u \) compared to \( 2^{12} \cdot C_u \) in a full binary array. The main drawback of this architecture lies in the effects of parasitic capacitances at the top-plate of the sub-
A DAC, which can cause sub-DAC gain and linearity errors that need to be calibrated out [31].

An alternative method is to use a hybrid DAC as shown in Fig. 5-5. In this implementation, the MSB DAC is a capacitive DAC and the sub-DAC is implemented using a resistor ladder to interpolate between the smallest step size of the MSB DAC. The benefit of using a resistive string for the sub DAC is the lack of sub-DAC gain error. The drawback, however, is that static current will be consumed in the resistive string, which may be undesirable for ultra-low-power applications.

In our prototype, we use a hybrid C-R DAC as it does not require additional DAC calibration circuitry to correct for sub-DAC gain error. This allowed us to simplify the DAC design and focus on the skew correction circuit design. The DAC is partitioned with the eight MSBs implemented using a capacitive DAC and the four LSBs implemented using a resistive string. The capacitive DAC was implemented using MiM capacitors. Based on foundry data, it was determined that a 6 fF unit capacitance was sufficient to achieve 12-bit matching. The resistive string was implemented using poly resistors which has the best matching characteristics based on the data provided by the foundry.
5.4 Input Sampling Network

In SAR converters, there is no need for a dedicated T/H circuit as the input signal is sampled directly onto the capacitive charge-sharing DAC. Furthermore, the sampling clock skew correction technique presented in Chapter 4 by adjusting the delay of the input signal relies on our ability to control the sampling switch resistance. As such, care has to be taken in the design of the sampling network to minimize signal dependent effects.

5.4.1 Bottom Plate Sampling

A source of input amplitude dependent sampling errors is the effect of the channel charge injection in the sampling switch. During the track phase, the charge in the channel of a MOS transistor, to first order, is given by,

\[
Q_{ch} = C_{ox}WL (V_{GS} - V_T) = C_{ox}WL (V_{DD} - V_{in} - V_T).
\]
Given that $V_T$ is non-linearly related to $V_{in}$, the channel charge is also non-linearly dependent on the input amplitude. Therefore, as the sampling switch turns off and the T/H transitions from the track mode to the hold mode, the charge stored in the channel is cleared with a fraction of the charge moving to the sampling capacitor, thus erroneously affecting the voltage of the sampling capacitor.

The effect of channel charge injection is exacerbated in high-speed converters that require high-speed T/H circuits. This is because to achieve a high tracking bandwidth, the resistance of the sampling switch needs to be reduced, and is done by increasing the width of the sampling switch. Thus, as the size of the sampling switch increases, from Equation 5.1, the amount of charge stored in the channel also increases resulting in increased sampling error.

To mitigate this effect, bottom plate sampling is used. Fig. 5-6 shows a schematic of the sampling network with the bottom plate sampling switch. In this T/H, M1 operates as the tracking switch while M1B operates as the sampling switch. At the end of the track phase, switch M1B is opened first before switch M1. In this configuration, when switch M1 opens, switch M1B is already off, thus almost all of the charge in the tracking switch will return to the input, and not be transferred to the sampling capacitor.

When using bottom plate sampling, there is still charge injection from the sam-
Figure 5-7: Schematic of the differential T/H with bottom-plate sampling switch M1B. The device Mx is used to shuffle charge between the two half circuits rather than from the outputs.

pling switch. However, since the sampling switch is connected to a fixed potential, charge injection from the sampling switch is constant, and can be modeled as a constant offset. When used differentially, as show in Fig. 5-7, the effect of the constant charge injection of the sampling switch is largely canceled out.

One last consideration in using bottom plate sampling is to ensure that the additional resistance due to the addition of the bottom plate switch does not adversely affect the sampling bandwidth. One way of ensuring this is to increase the size of M_{PB} and M_{NB} in Fig. 5-7. Another way is to use an additional switch Mx can also be used to shuffle charge between both differential sampling switches without having to go through the supply V_{CM}. This switch will have an effective length of L_x/2 for differential signals.

5.4.2 Bootstrapped Sampling Switch

A simplified model of the input tracking circuit is shown in Fig. 5-8. In the track phase, when CLK is high, the resistance of the switch when it is turned on is given
Given this equation for the input tracking switch resistance during the track phase, as the input signal increases, the overdrive voltage, $V_{DD} - V_{in}$, keeps decreasing. Therefore, the switch resistance keeps increasing as the input signal amplitude increases. Furthermore, as the input signal amplitude increases, $V_{SB}$ also keeps increasing, resulting in the threshold voltage, $V_T$ to increase. Eq. 5.2 shows that increasing $V_T$ also results in an increase in the sampling switch resistance. Both these effects make the sampling switch resistance a strong function of the input signal amplitude, which will result in harmonic distortion in the output of the T/H circuit for dynamic signals, as well as input dependent input signal delay.

A widely used method in T/H design to reduce the signal dependence of the input tracking switch ON-resistance on the input signal amplitude is the use of a bootstrap circuit [32]. The circuit schematic of the bootstrap circuit is presented in Fig. 5-9. The fundamental characteristic of a bootstrap circuit is that it ensures that the $V_{GS}$ of the input sampling transistor is constant and independent of the amplitude of the input signal. The associated waveforms of the bootstrap circuit is shown in Fig. 5-10. Compared to the waveforms for the simple T/H, the bootstrap circuit ensures that the overdrive voltage of the tracking transistor is constant and independent of the input signal amplitude.
Figure 5-9: Schematic of the tracking switch clock bootstrap circuit.

Figure 5-10: Input signal and the gate voltage, $V_G$, of the tracking switch. During each track phase, $V_G$ is always $V_{DD}$ above the input signal.
The bootstrap switch shown in Fig. 5-9 operates by pre-charging the capacitor C3 to VDD, during the hold phase, and then connecting capacitor C3 between the gate and source of the tracking switch during the track phase. Transistors M7 and M10 are used to discharge $V_G$ back to ground when the the clock, CLK, goes low. Transistor M9 reduces the drain-to-source and gate-to-drain voltages experienced by M11. M13 on the other hand ensures that the gate-to-source voltage of transistor M6 does not exceed VDD.

5.4.3 Input Signal Delay Control

In order to implement the signal delay adjustment method, we need to be able to control the tracking switch resistance of the input tracking circuit. The circuit implementation of the sampling switch resistance control is shown in Fig. 5-11. The actual implementation is differential, but only one half of the circuit is shown in this figure for simplicity. As described in Section 5.1, the binary nature of the SAR capacitive DAC can be leveraged to provide the finely controllable delay that is required for sampling clock skew calibration.

As can be observed from Fig. 5-11, the $i$-th sampling capacitor is connected to two tracking switches, $M_{ia}$ and $M_{ib}$. All $M_{ia}$ switches are always on during the track phase. Switches $M_{ib}$, on the other hand, can also be used to sample the input signal onto the $i$-th capacitor in the DAC during sampling by setting the $i$-th control bit, $Ctrl_{<i>}$, high. CLK indicates the sampling CLK, while CLKd denotes the delayed version of the sampling clock as shown in Fig. 5-6. This delay is imposed to ensure that the sampling switches, MPB and MX are turned off first before the tracking switches turn off for bottom plate sampling. This is required to ensure that the switch charge injection is input-independent.

The AND logic gate in the path of the auxiliary sampling switches is used to control if one or both switches are used during the sampling phase. It might seem that the propagation delay of the AND gate will delay when switches $M_{ib}$ are turned on and off. This is not the case because the sampling instant of the input is set by when the sampling switches, MPB and MX, on the common mode side are turned off.
Figure 5-11: Schematic of the input sampling network with auxiliary switches $M_{ib}$ for sampling clock skew calibration. Only one half of the capacitive DAC is shown for simplicity. The actual implementation is fully differential. The reference switches are also omitted for simplicity.
and is unaffected by the extra logic in the auxiliary switch control path. However, the extra delay of the logic in the control path of the auxiliary tracking switches will reduce the amount of settling time in the track phase. The logic delay for the AND gate in this technology is nominally in the order of 15-25ps and is negligible given that the sampling period for the prototype time-interleaved ADC is greater than 2ns.

The main design constraint here is the sizing of the auxiliary switch. The auxiliary switch is sized by choosing a tracking switch size such that the change in the delay between when both switches are on and only one switch is on corresponds to the maximum calibration range required. For example, in order to implement a calibration range of ~15ps, the size of switches $M_{ih}$ are chosen such that when both switches are on, the signal delay is reduced by 15ps. Therefore, in the two extreme cases, i.e. when all auxiliary switches are on or off, the input signal will be delayed by 15ps between them. By choosing which of the auxiliary switches are on or off, smaller delays of the input signal can be achieved. The step delay size is given by,

$$t_{\text{delay}, \text{LSB}} = \frac{t_{\text{max}}}{2^B},$$

where $t_{\text{max}}$ is the maximum delay between when all auxiliary switches are on and off, and $B$ is the number of bits of the capacitive DAC. Assuming a calibration range of 15ps and an 8-bit capacitive DAC, calibration ranges of ~60fs is achievable.

### 5.5 Latch Comparator

Besides the design of the DAC and the T/H, the comparator is also another crucial circuit block and care has to be taken in its design. There are four main considerations in the design of comparators for ADCs, namely input referred noise, input referred offset, regeneration speed, and power consumption. Input referred noise is one of the most crucial design criteria because noise can affect the accuracy of the ADC and cannot be calibrated or corrected easily. Unlike the input referred noise, the effect of input referred offset can be calibrated out in the analog domain or corrected in the
Figure 5-12: Schematic of a StrongArm comparator commonly used in SAR ADCs.

digital domain.

StrongArm latch based comparators have found widespread use in SAR ADCs because of its low power operation (no static current consumption) and high attainable speeds due to the use of positive feedback. Fig. 5-12 shows the schematic of a StrongArm latch, which is the most commonly used comparator in SAR ADCs [33]. The main strengths of the StrongArm comparator is that it consumes no static power and can drive the outputs at full CMOS logic levels.

The transient voltage waveform for relevant nodes of a StrongArm latch is shown in Fig. 5-13. The operation of the StrongArm latch can be broken down into four phases: reset, preamplification, regeneration, and decision phases. During the reset phase, CLK is low, and both the outputs, Out+ and Out-, are pulled high. The preamplification phase begins when CLK, goes high. When CLK transitions from low to high, M1 and M2 start to discharge nodes X and Y toward ground. When nodes X and Y have discharged sufficiently, such that the NMOS transistors M3 and M4 begin to turn on, transistors M3 and M4 then start discharging the output nodes Out+ and Out- toward ground. The regeneration phase begins when the PMOS transistors M5 and M6 start conducting. During the regeneration phase, the
Figure 5-13: Relevant waveforms of the StrongArm comparator commonly used in SAR ADCs.
cross-coupled inverters amplify the sampled signal using positive feedback to generate CMOS logic level output signals.

The StrongArm comparator described above is well suited for ADCs due to its low-power (it does not consume any static power) and high speed operation. However, this architecture does have a few drawbacks. In the StrongArm, the main tradeoff is between optimizing the comparator for speed versus noise and offset. In order to reduce noise and offset, M9 should be sized such that the input transistors are close to the edge of saturation during the preamplification phase. Furthermore, the current in M9 should also be reduced in order to minimize offset [34]. Reducing the current in M9 however, limits the current in the latch during the regeneration stage and as a result hurts the speed performance.

The Schinkel comparator attempts to address the drawbacks in the StrongArm comparator by splitting the input and latch stages [35]. The schematic of a Schinkel comparator is illustrated in Fig. 5-14. In this comparator, one tail current source,
M11, is used for the input stage and another tail current source, M12, for the latching stage. The operation of the Schinkel comparator can also be split into the reset, preamplification, regeneration and decision phases. The relevant waveforms of the Schinkel comparator is illustrated in Fig. 5-15. During the reset phase, when the clock, CLK, is low, M3 and M4 precharge nodes Di+ and Di- to VDD. As a result, M5 and M6 pull the outputs, Out+ and Out-, to ground. When CLK goes high, M3 and M4 are turned off and the comparator enters the preamplification phase. In this phase, the input transistors M1 and M2 start to discharge node Di+ and Di-. At this moment, the tail current in the latch half of the circuit, M11, is also turned on. As soon as the common-mode voltage between Di+ and Di- falls such that M5 and M6 cannot pin the outputs to ground, the comparator enters the regeneration phase. In the regeneration phase, the difference between Di+ and Di- is amplified via positive feedback and a decision is generated.

The benefit of the Schinkel comparator is that it provides further isolation between the latch outputs and the inputs, thus reducing the effect of kickback noise. As the outputs are latched to the rails, the kickback noise needs to traverse the $C_{gd}$ of both M5/M6 and M1/M2 rather than $C_{gd}$ of M1 alone.

The main drawback of the Schinkel comparator is the requirement of having both phases of the clock, CLK and CLK. The skew in the signal CLK needs to be controlled well because skew in the signal CLK can be detrimental in the operation of the latch. For example, if CLK falls too late, when Di+ and Di- have fallen well below the threshold voltage of M5 and M6, then the latch will not be able to detect the difference between Di+ and Di- and the latch will cease to operate as expected.

The Miyahara comparator [36] illustrated in Fig. 5-16 attempts to improve upon the performance of the Schinkel comparator in Fig. 5-14. The operation of the Miyahara comparator is very similar to the operation of the Schinkel comparator. The pertinent signal behavior of the Miyahara comparator is essentially the same as that of the Schinkel comparator shown in Fig. 5-15. Again, the operation of the comparator can be split into four phases. During the reset phase, M3 and M4 pull up nodes Di+ and Di-, which in turn pull the outputs low. During the preamplification phase,
Figure 5-15: Relevant waveforms of the StrongArm comparator commonly used in SAR ADCs.
Figure 5.16: Schematic of a Miyahara comparator. The capacitor banks on D+ and D- are used for offset calibration.
the clock, CLK, goes high, which turns off the precharge transistors M3 and M4, while simultaneously turning on the tail current source M5. During this phase, the voltages at nodes Di+ and Di- start to fall with a voltage difference of $\Delta V_{Di}$ building between the nodes Di+ and Di-. As the nodes Di+ and Di- fall, they start to turn off transistors M8 and M9, which were initially pinning the outputs to ground as in the Schinkel inverter. The main difference is that the tail currents M13 and M14 of the latch stage in the Miyahara comparator is also turned on automatically using Di+ and Di-, and does not require the use of CLK.

The noise of the Miyahara comparator is dominated by the first stage [34]. When the tail current source is turned on, the r.m.s. noise at the drains of M1 and M2 is proportional to $\sqrt{g_m \cdot t}$ [37], where $g_m$ is the transconductance of the input transistors and $t$ is the integration time. However, the signal component at the drains of M1 and M2 is proportional to $g_m \cdot t$ [38]. Thus, the input referred noise power is inversely proportional to $\sqrt{g_m \cdot t}$. Therefore, in order to reduce the noise of the comparator, the input pair should be operated with the minimum overdrive voltage allowable in order to maximize its $g_m$. This can be achieved by reducing the size of M5 to reduce the discharging current, increasing the size of the input transistor pair, or by decreasing the input common-mode voltage [34]. Decreasing the common mode voltage is not preferable as it may lead to a reduced full-scale input range in SAR ADCs. Therefore, the noise of comparator was designed by sizing the tail current source and the input pairs appropriately. In this prototype, the standard deviation of the comparator noise was simulated to be $\sim 120 \mu V$, which corresponds to about 0.2 LSB.

The input-referred offset of the input transistor pair, M1 and M2 was simulated using a Monte Carlo method to have a standard deviation of 2 mV. However, in order to ensure that the offset was below 1LSB, offset calibration was included in the comparator via a capacitor bank at the output of D+ and D-. The offset calibration capacitor bank consists of 31 switches. The parasitic capacitors of the switches were sufficient to obtain fine enough calibration steps over the required calibration range. The offset can be calibrated between -8 mV and +8 mV with a step size of 250$\mu V$, which corresponds to about 0.4 LSB.
5.6 Asynchronous Clock Generation

In some SAR converters, a high-speed clock is often used to clock each of the steps in the conversion. During the conversion phase, three main events take place per bit cycle: i) the SAR logic sets the appropriate bits at the DAC, ii) the DAC is allowed to settle, and iii) the comparator is strobed. The period for each conversion is then given by,

\[ t_{clk} = t_{Logic} + t_{Settle} + t_{Comparator}, \]  

(5.4)

where \( t_{clk} \) is the period of the SAR clock, \( t_{Logic} \) is the delay in the SAR logic, \( t_{Settle} \) is the time required for the capacitive DAC to settle, and \( t_{Comparator} \) is the comparator delay. If a high-speed clock is used, then \( t_{clk} \) has to be chosen for the worst case scenario delays of each of the phases (e.g. when the comparator hits a metastable state because the comparator input difference is very small, etc.).

In order to enable high speed operation, asynchronous clocking is used such that the bit cycling can begin as soon as the comparator outputs are valid [39]. An asynchronous clocking scheme was used in the prototype ADC. The schematic of the asynchronous timing circuit block is shown in Fig. 5-17. The Start signal goes high at
the end of the sampling phase of the SAR ADC. The Comp Done signal goes from low to high when the comparator outputs are valid, and the Conv Done signal indicates that the last bit of the conversion is complete, and is used to gate the comparator clock (Comp Clk) at the end of each conversion.

The timing diagram of the asynchronous clocking scheme is shown in Fig. 5-18. As soon as the input has been sampled, Start goes high. A set time delay later (which is programmable), when the capacitive DAC has settled, the comparator is strobed. As soon as the comparator is done, which is indicated by the Comp Done signal, the comparator clock is reset after some delay to ensure that the output has been stored by the SAR logic and the bit-cycling has begun. After a further delay to ensure that the capacitive DAC has settled after bit-cycling, the comparator is strobed again. Each delay block is programmable between 200 ps to 1 ns with 100 ps steps to allow for slower operation when required during testing. The delay for the settling of the capacitive DAC for the MSB and LSB sections are individually programmable too. This is to allow the flexibility for longer DAC settling times for the MSBs as larger capacitors are charged and discharged to VDD and GND.

For the SAR logic, a static logic implementation was chosen over dynamic logic in order to allow for slow operation during testing, if required. The schematic of the

Figure 5-18: Timing diagram illustrating the relevant signals of the SAR asynchronous timing scheme.
Figure 5-19: Relevant waveforms of the StrongArm comparator commonly used in SAR ADCs.

SAR logic is shown in Fig. 5-19.

5.7 Multi-Phase Clock Generator

In an N-way time-interleaved ADC system, N clocks offset by T/N is required, where T is the sampling period of a single channel in the time-interleaved ADC. Each of these clocks need to meet the jitter requirement for the targeted conversion speed and resolution. For a jitter limited system, the ideal SNR is given by,

$$ SNR_{ideal} = 20 \log_{10} \left( \frac{1}{2 \pi f_{int} \text{jitter}} \right). $$

Therefore, for a 12-bit, 200 MS/s ADC time-interleaved converter with a maximum Nyquist input frequency of 100 MHz, the clocking network needs to have a clock jitter of less than ~300 fs. In order to meet this tight clock jitter specification, a differential clock input is used as it is difficult to find square clock sources that have such low
jitter. The block diagram of the clock signal chain is illustrated in Fig. 5-20. An RF-source is used to provide a low-jitter, single-ended sinusoidal signal, which is then converted into differential signals, CLK_p and CLK_n, using an off-chip center-tapped transformer or balun. The differential signals are then input to the chip where it is converted into a single-ended, high speed, square-edged clock on chip using the differential-to-single ended converter shown in Fig. 5-21 [40]. This differential to single-ended converter operates similarly to a seven-transistor differential to single-ended OTA, with its benefits of common-mode and power-supply rejection. Since this circuit operates predominantly in the large-signal regime, the extra cross-coupled NMOS pair provides positive feedback to increase the gain and speed of the differential to single-ended converter.

The high speed clock is then fed into a multiphase clock generating circuit illustrated in Fig. 5-22. Shift registers have been shown to add less jitter than a DLL [41] for multi-phase clock generation, which are used to generate the required multi-phase clocks for this prototype ADC. A transient noise simulation using Spectre was performed to estimate the standard deviation of the jitter of the clock signal. The simulated jitter was 202 fs, below the required 300 fs limit for a 12 bit converter operating with a maximum input frequency of 100 MS/s. This allows for a margin of ~100 fs to accommodate the jitter in the off-chip, sinusoidal clock source. The
Figure 5-21: Schematic of the differential to single-ended converter used to generate a CMOS level square clock from the sinusoidal inputs. The bias current is supplied from off-chip.

Figure 5-22: Schematic of the multi-phase clock generator used to generate the sampling clocks for each SAR ADC.
clocks for each channel are then routed throughout the chip to the respective SAR ADCs using a H-tree routing network to minimize routing length mismatches between channels, which can introduce sampling clock skew. The clocks are routed on metal layer M3 with ground shields above and below (implemented using metal layers M1 and M5) to minimized coupling to the input signals that are routed above (on metal layer M9).

5.8 Off-Chip References

Decoupling caps for all references were included on chip. Of particular importance is the decoupling between the positive and negative references for the ADC. During full-chip simulation with bond-wire models on the references as shown in Fig. 5-23, it was observed that the on-chip decoupling capacitors had a tendency to ring when the capacitive DAC was being charged and discharged, in particular when the large MSB capacitors were charged and discharged. In order to prevent these oscillations that can result in significant increase of the DAC settling times, three measures were taken. First, chip-on-board assembly was used rather than a package to minimize bondwire lengths and associated inductances. Second, to further reduce the bondwire inductance, for each reference, two sets of interleaved bondwires were used for the ADC references as illustrated in Fig. 5-24. The mutual inductance in the interleaved bondwires further reduces the effective inductance in the loop between the off chip positive and negative references. The effect of these interleaved bondwires were simulated in Agilent ADS using Philips/TuDelft bondwire models [42]. Finally, small damping resistors were added to the on-chip decoupling capacitors as shown in Fig. 5-24 to ensure that the bondwire inductance and on-chip decoupling capacitors do not ring when large amounts of charge are shuffled in and out of the capacitive DACs.
Figure 5-23: Schematic used to model the effect of bondwire parasitics on the ADC references.

Figure 5-24: Schematic used to model the interleaved bondwire parasitics on the ADC references.
5.9 Layout

Floorplanning is essential in any high-speed and high-accuracy ADC design. In time-interleaved ADCs, floorplanning becomes even more crucial in order to ensure that sampling clock skew effects are not exacerbated by poor layout. The layout of the chip core is shown in Fig. 5-25. The input signal and clocks are brought on chip at the bottom edge. They are routed in an H-tree fashion in order to minimize wiring length mismatches that can contribute to sampling clock skew. The clock signals are routed beneath the input signal, but are shielded to avoid capacitive coupling between the signals. The digital output of each channel is collected at the outer edges of each channel (on the left and right edges of the chip core) and are routed to the top-edge of the chip, where the digital output pins are placed. The digital output pads are separated from the ADC core using deep N-wells to minimize noise coupling from the digital output pins and the sensitive mixed-signal core.

The layout of a single channel SAR ADC is shown in Fig. 5-26. Significant attention was devoted to ensuring that i) symmetric layout between the positive and negative halves were achieved to maximize the benefits of a differential architecture in rejecting common mode disturbances, ii) minimizing the wiring lengths between the reference and tracking switches and the capacitor array and iii) seperating as much as possible the noisy digital circuits from the sensitive analog side of the ADC.

5.10 Summary

In this chapter, we have outlined the circuit design of the prototype time-interleaved ADC with sampling clock skew correction using the input signal delay control method. The design of important components of each SAR channel such as the input sampling network, comparator, asynchronous timing block, and SAR logic was also detailed. In addition, supporting circuits that are needed to implement time-interleaving such as a multi-phase clock generator, and the design of the off-chip reference inputs were presented.
Figure 5-25: Layout of the core of the prototype time-interleaved ADC. The layout is shown in (a), and a legend indicating the relevant blocks is shown in (b)
Figure 5-26: Layout of the SAR ADC used in the individual channels. The layout is shown in (a), and a legend indicating the relevant blocks is shown in (b).
Chapter 6

Test Setup and Measurement Results

In Chapter 5 we described the circuit design of the prototype time-interleaved ADC with sampling clock skew correction using the input signal delay control method. In this chapter we present the test setup and measurement results of the fabricated prototype time-interleaved ADC described in the previous chapter.

6.1 Test Setup

6.1.1 Prototype Chip

The prototype time-interleaved ADC was implemented in a 65 nm LP CMOS process with 9 Metal layers. The ADC occupies an active area of 0.9 mm$^2$ and operates using a 1.2 V supply. A die photo of the fabricated device is shown in Fig. 6-1. The location of each interleaved channel is highlighted in the die photo.

6.1.2 Printed Circuit Board (PCB)

The prototype chip was mounted on a PCB using chip-on-board (COB) assembly technology whereby the chip is directly wirebonded to landing pads on the PCB. This assembly method minimizes bondwire lengths and reduces parasitics associated with
Figure 6-1: Die photo of the prototype ADC.
IC packages. Linear regulators (Linear Technology, LT3021) are used to provide the power, reference, and common mode voltages to the ADC. The input and clock signals are provided from off chip using two single ended synthesized RF-sources (Agilent 8644B for the input signal and HP 83732B for the clock signal). The single ended sources are converted to fully differential signals on the PCB using surface-mount transformers (Mini Circuits, TC1-1T+). For power consumption measurements, a 1Ω resistor was used in series with the regulator outputs to measure the current drawn by the various supplies and references.

6.1.3 Measurement Setup

The block diagram of the test setup used to measure the performance of the prototype ADC is shown in Fig. 6-2. High performance (low-noise and low-jitter) synthesized signal generators are used to supply the input and clock signals. While synthesized signal generators have low noise and jitter, their harmonic performance may be insufficient for ADC characterization. As such, band-pass filters are used in order to filter out harmonics as much as possible in the signal and clock inputs.

The output bits of the ADC are captured using a logic analyzer (Tektronix TLA715). The output bits are then processed using MATLAB on an external computer. A digital I/O controller (National Instruments USB-6501) was used to program the chip and for power-on-reset control.

6.2 Measurement Results

In this section we present the measurement results of the prototype time-interleaved ADC that implements the input signal delay control method for sampling clock skew calibration. We first show the dynamic performance of the ADC before and after offset calibration between channels. We then show the dynamic performance of the prototype ADC before and after sampling clock skew calibration. We then present the ADC SNDR as a function of varying input frequencies. Finally, the static measurements of the ADC are presented along with the power breakdown of the chip.
Offset Calibration

Before sampling clock skew calibration is performed, offset calibration is first carried out to calibrate the offsets between channels. The offset of each channel is calculated in MATLAB using a low-frequency input signal with a small amplitude. The capacitor banks at the output of each channel is programmed to calibrate out the offsets between channels. The output spectra of the 4-way time interleaved ADC before and after offset calibration is shown in Fig. 6-3 offset. As can be seen from Fig. 6-3(a), before offset calibration, the tones due offset mismatch limit the SNDR and SFDR of the time-interleaved ADC. After offset calibration, the tones in the frequency spectrum due to offset mismatch between channels is reduced by over 20 dB as shown in Fig. 6-3.

Before Sampling Clock Skew Calibration

The output spectra of the prototype ADC for low input frequencies (2.1 MHz) is shown in Fig. 6-4. The single channel output spectrum is illustrated in Fig. 6-4(a). A typical single channel output achieves 60.9 dB SNDR and 71.8 dB SFDR. The time-interleaved output spectrum is shown in Fig. 6-4(b) after offset calibration between
channels was completed. The time-interleaved output achieves 59.8 dB SNDR and 72.7 dB SFDR. At low frequencies, as described in Section 2.3.3, we expect that small sampling time skews do not lead to large voltage errors. As such, time interleaving should not result in substantial degradation in performance when compared to the output of a single channel. This observation is confirmed by comparing the output of the single channel and the time-interleaved system. The degradation in the SNDR between the single channel and time-interleaved outputs is most likely due to residual offset and gain mismatches between channels.

The output spectra of the prototype ADC for high input frequencies (74.1 MHz) close to the Nyquist rate of the converter is shown in Fig. 6-5. A typical single channel output is shown in Fig. 6-5(a) and achieves 58.6 dB SNDR and 71.1 dB SFDR. The time-interleaved output spectrum is shown in Fig. 6-5(b). The time-interleaved output achieves 53.1 dB SNDR and 56.9 dB SFDR. At high input frequencies, there is a small degradation in the single channel output performance. This drop is mainly...
due to increased second harmonic content, which does not appear in in the single channel spectrum at low input frequencies. The second harmonic content may be due to phase imbalances in the off-chip transformer. However, spurs due to sampling clock skew between channels are clearly visible in the time-interleaved spectrum and degrades the SNDR and SFDR performance of the time-interleaved output.

After Sampling Clock Skew Calibration

For sampling clock skew calibration, we used the zero-crossing counting method described in Section 3.3.1 for sampling clock skew detection. The detection algorithm is carried out off-chip and was implemented in MATLAB using 500k samples/cycle. All sampling clock skew calibration was done with an input signal of 74.1 MHz. The sampling clock skew calibration bits obtained were then frozen for all other tests at other input frequencies.

The calibration process begins by choosing two channels to calibrate first (e.g. Channel 3 and Channel 1). The two channels are calibrated by ensuring that the
number of zero-crossings in the time period between when Channel 1 samples the input and Channel 3 samples the input is the same as the number of zero-crossings in the time period between Channel 3 and Channel 1. Once the sampling clock skew calibration between these two channels is complete, a third channel is calibrated against Channel 1 and Channel 3. For example, in order to calibrate Channel 2, the number of zero-crossings between the output of Channel 3 and Channel 2 is compared to the number of zero-crossings between Channel 2 and Channel 1. When no sampling clock skew is present, these two zero-crossing counts should be equal. Once Channel 2 has been calibrated, Channel 4 is calibrated against the pair of Channel 1 and Channel 3 (like it was done during the calibration of Channel 2), or against Channel 2 only.

The output spectra of the prototype ADC after sampling clock skew calibration for low input frequencies (2.1 MHz) is shown in Fig. 6-6. The single channel output spectrum is illustrated in Fig. 6-6(a). A typical single channel output achieves 61.1 dB SNDR and 73.1 dB SFDR. The time-interleaved output spectrum is shown in Fig.
The time-interleaved output achieves 60.2 dB SNDR and 74.3 dB SFDR. As expected, sampling clock skew calibration yields only slight improvement in the performance of the ADC at low input frequencies.

The output spectra of the prototype ADC for high input frequencies (74.1 MHz) close to the Nyquist rate of the converter is shown in Fig. 6-7. A typical single channel output is shown in Fig. 6-7(a) and achieves 58.7 dB SNDR and 70.9 dB SFDR. The time-interleaved output spectrum is shown in Fig. 6-7(b). The tones due to sampling clock skew between channels has been significantly reduced. After calibration, the time-interleaved output achieves 58.2 dB SNDR and 68.0 dB SFDR. After sampling clock skew calibration, the SNDR of the converter has improved by 5.1 dB, or close to 1 bit. The SFDR has improved by over 11 dB.

The SNDR for the time-interleaved output for various input signal frequencies is plotted in Fig. 6-8. From this plot, it can be seen that the SNDR of the time-interleaved converter keeps decreasing with increasing input signal frequency due to the effects of sampling clock skew. With sampling clock skew calibration enabled, the
Figure 6-7: Prototype ADC output spectra for a high frequency input signal near its Nyquist rate after sampling clock skew calibration. The single channel output spectrum is shown in (a) and the four channel output spectrum is shown in (b).

SNDR curve “flattens out” with increasing input signal frequency and maintains less than 2 dB degradation throughout the entire Nyquist bandwidth.

6.2.1 Static Performance

The INL and DNL performance of the prototype ADC is shown in Fig. 6-9. The INL and DNL for a typical single channel are shown in the left column, while the INL and DNL of the time-interleaved ADC is shown in the right column. The maximum INL and DNL for a single channel are +1.37/-1.34 LSBs and +0.85/-0.91 LSBs respectively. For the time-interleaved converter, the INL and DNL are +1.07/-1.17 LSBs and +0.60/-0.69 LSBs respectively. The main sources of INL and DNL is the due to the capacitive DAC. The jumps in the INL curve can be explained due to extra parasitic capacitances between the top plate and the bottom plate of the DAC capacitors due to layout. Going from the 3rd MSB to the 4th MSB, the unit capacitors do not occupy full rows in the capacitive DAC. Thus, there is longer wiring to reach the bottom plates of the LSBs for the capacitive DAC, which introduces extra parasitic
6.3 Performance Summary

In Table 6.1, we summarize the main performance metrics of the prototype time-interleaved ADC. The power breakdown of the time-interleaved ADC is shown in Fig. 6-10. The ADC operates with a sampling rate of 150 MS/s. After sampling clock skew calibration, the ADC achieves an SNDR of 60.2 dB for low input frequencies and 58.2 dB for input frequencies near Nyquist.

Fig. 6-11 shows a plot of ADC energy versus the achieved SNDR for all time-interleaved ADCs published in ISSCC and VLSI from 1997-2014 [5]. The ADC energy is calculated using the formula $P/f_s$, where $P$ is the power consumption and $f_s$ is the sampling rate of the converter. ADCs that are located towards the bottom right of the graph are more efficient. This work achieves one of the lowest energy values for time-interleaved ADCs with SNDR >55 dB. The main goal of this work has been to show a proof-of-concept that controlling the delay of the input signal is a viable method for sampling clock skew calibration in time-interleaved ADCs and was not
Figure 6-9: INL and DNL performance of the ADC. Single channel results are shown on the left column and time-interleaved results are shown in the right column.

Table 6.1: Performance summary of the prototype time-interleaved ADC.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>TI SAR</td>
</tr>
<tr>
<td>Technology</td>
<td>65 nm</td>
</tr>
<tr>
<td>Supply</td>
<td>1.2V</td>
</tr>
<tr>
<td>Sampling Rate ($f_s$)</td>
<td>150 MS/s</td>
</tr>
<tr>
<td>Resolution</td>
<td>12 bit</td>
</tr>
<tr>
<td>Power</td>
<td>12.4 mW</td>
</tr>
<tr>
<td>Area</td>
<td>0.9 mm$^2$</td>
</tr>
<tr>
<td>SNDR @ 2.1 MHz</td>
<td>60.2 dB</td>
</tr>
<tr>
<td>FoM$\text{Walden}$</td>
<td>98 fJ/step</td>
</tr>
<tr>
<td>SNDR @ 74.1 MHz</td>
<td>58.2 dB</td>
</tr>
<tr>
<td>FoM$\text{Walden}$</td>
<td>124 fJ/step</td>
</tr>
</tbody>
</table>
In this chapter, we described the test setup and measurement results used to characterize the prototype time-interleaved ADC. The prototype ADC implemented an input signal delay control method for sampling clock skew calibration. We described the equipment and off-the-shelf components that were used in testing. We also presented the dynamic performance of the ADC before and after sampling clock skew calibration. The INL and DNL performance and power consumption of the ADC was also presented.

6.4 Summary

In this chapter, we described the test setup and measurement results used to characterize the prototype time-interleaved ADC. The prototype ADC implemented an input signal delay control method for sampling clock skew calibration. We described the equipment and off-the-shelf components that were used in testing. We also presented the dynamic performance of the ADC before and after sampling clock skew calibration. The INL and DNL performance and power consumption of the ADC was also presented.
Figure 6-11: Energy versus SNDR plot for time-interleaved ADCs presented at ISSCC and VLSI from 1997-2014.
Chapter 7

Conclusions

7.1 Summary

Time-interleaving is an increasingly important technique to achieve higher speed and higher resolution ADCs. As the performance of a single-channel ADC begins to reach the limits of a given process technology, time-interleaving enables the realization of faster converters without incurring a disproportionate increase in power consumption. However, with time-interleaving, comes a number of challenges such as offset mismatch, gain mismatch, and sampling clock skew between channels that can degrade the performance of the time-interleaved converter. Of the three, sampling clock skew errors have proven to be most challenging to overcome as its effect keeps growing with increasing input frequency and cannot be easily corrected with digital post-processing.

In this thesis, we have developed two novel techniques to mitigate sampling clock skew errors in time-interleaved ADCs. The first method utilized a rapid consecutive sampling method whereby two samples of the input are acquired with a short delay between them. A simple linear interpolation is then used to recover a sampling clock skew free estimate of the input signal. The time delay between the two samples is determined by the maximum calibration range and the estimation error that can be tolerated.

In the second method, we proposed a sampling clock skew calibration technique
that controls the delay inherent in the input sampling network instead of the clock signal. The input sampling network can be modeled as an LPF. For input frequencies much lower than the corner frequency of the input network, a delay corresponding to $RC$ is introduced, where $R$ is the sampling switch ON-resistance and $C$ is the sampling capacitance. By varying the sampling switch resistance, the delay can be controlled in order to compensate for the effects of sampling clock skew.

The main benefit of the input signal delay control technique over techniques that control the delay of the sampling clock is that the input signal delay control method does not cause an increase of clock jitter. As the ADC resolution and sampling speeds increase, the amount of allowable jitter such that the ADC is not jitter limited keeps decreasing. For example, for a 12-bit converter with input frequencies up to 100 MHz, the allowable jitter is $\sim300$ fs. Almost all methods to control the delay of the sampling clock result in increased clock jitter. This is because in order to delay the sampling clock, the clock edge has to first be slowed down before being re-timed. Reducing the slew-rate of the clock edge increases jitter as shown in Fig. 3-12 and Fig. 3-13. In the input signal delay control technique, however, since the delay of input signal is adjusted rather than the clock, there is no increase in the jitter of the sampling clock. This quality is beneficial in realizing high-speed and high-resolution time-interleaved converters that are not jitter limited.

A circuit level design and implementation of the input signal delay control technique was presented. A prototype four-way time-interleaved ADC implementing this method was fabricated in a 65 nm LP CMOS process. Measurement results of the proof-of-concept ADC were presented. To the best knowledge of the author, this is the first ever circuit implementation of sampling clock skew calibration by controlling the delay of the input signal rather than the sampling clock. The prototype ADC operates at an effective sampling rate of 150 MS/s. The ADC achieves 60.2 dB SNDR and 74.3 dB SFDR for low input frequencies and 58.2 dB SNDR and 68.0 dB SFDR at Nyquist. The single channel INL and DNL measurements of the ADC are $\pm1.37$ LSBs and $\pm0.91$ LSBs, respectively. The prototype ADC consumes 12.4 mW of power from a 1.2 V supply, which corresponds to an FoM of 98 and 124 fJ/step for low input
frequencies and at Nyquist, respectively.

The relatively high FoM of the converter is mainly due to the use of a hybrid C-R DAC in the SAR ADC implementation. As described in Section 5.3, a hybrid DAC was chosen in the proof-of-concept chip to simplify the design of the interleaved ADC. The main drawback of the C-R DAC is the static power consumed in the resistor string. As was shown in the ADC power breakdown in Fig. 6-10, almost half of the ADC power was dissipated in the references. Of the power consumed by the references, about 60% was dissipated in the resistive string. If a split capacitor DAC with calibration is utilized in conjunction with energetically-efficient switching schemes such as the merged capacitor switching (MCS) scheme [43], the power drawn from the references can be significantly reduced. For example, by employing a split DAC in conjunction with the MCS switching scheme in this ADC, the power consumption of the references is estimated from [43] to be 1.1 mW. This would translate to an FoM of 73 fJ/step for input frequencies at Nyquist.

7.2 Future Work

Based on the ideas and results presented in this thesis, there are a number of areas that warrant further investigation.

1. Extending the input signal delay control technique to other ADC architectures.

In the prototype described in Chapter 5, the input delay control method was implemented using a SAR ADC because the SAR architecture provides a natural means for binary-weighted delay adjustment via the capacitor array. However, this technique is not limited to SAR ADCs only and can be extended to other architectures that utilize a T/H. For example, in time-interleaved pipeline ADCs, methods to vary the ON-resistance of the first stage T/H can be investigated to implement the input signal delay control technique.

2. Implementing with more energy efficient SAR ADCs.
The prototype SAR ADC utilized a resistor string for the LSB DAC in order to minimize the complexity of the design in the proof-of-concept IC. However, the resistor string consumes static power and increases the power consumption of the ADC. The energy efficiency of the converter can be greatly improved by utilizing a fully-capacitive split-capacitor based DAC with calibration or by employing redundancy (e.g. a sub-binary radix DAC) to calibrate out gain and linearity mismatches in the sub DAC. Energy efficiency of the interleaved ADCs can also be further increased by by utilizing low-power switching techniques such as MCS [43].

3. Adding on-board sampling clock skew detection.

In both the techniques presented in this thesis, sampling clock skew estimation between channels was planned to be performed off-chip using a foreground detection method. The calibration time would be greatly reduced if the sampling clock skew estimation can be performed on-chip. For example, in the zero-counting skew detection method from [19], a large number of samples are required when the skew estimation is performed off-chip. Large number of samples are needed so that a sufficient number of zero-crossings are present in the data stream in order to extract the skew information to the required resolution, and is limited by the memory of the logic analyzer or data acquisition equipment. However, if performed on-chip, only the number of zero-crossings need to be stored allowing larger number of samples to be processed to extract skew information. In addition, exploring novel ways to incorporate background sampling clock skew estimation would be very beneficial. For example, for the rapid consecutive sampling method, if a common flash ADC similar to [22] was used in conjunction with the sub-channels, background sampling clock skew detection can be integrated.

4. Exploring limits of joint clock and input signal delay control calibration

It would be worthwhile to investigate the limits of a joint calibration system utilizing both clock and signal delay methods. Both clock and input signal delay
control can be integrated into a time-interleaved ADC for sampling clock skew calibration. Techniques to delay the sampling clock usually adds jitter to the sampling clock, which degrades the SNDR but usually does not degrade SFDR of the converter. Residual gain mismatch after sampling clock skew calibration using the input signal delay control method can lead to SNDR and SFDR degradation if the spurs due to gain mismatch keep increasing. It would be worthwhile to investigate the limits of a joint calibration system utilizing both clock and signal delay methods for different applications. For example when SFDR performance is important, once the limits of the input signal delay control is reached, the clock delay control method can be applied to further reduce the spurs due to sampling clock skew in order to improve SFDR performance.
Bibliography


