A Sampling Jitter Tolerant Continuous-Time Pipeline ADC

by

Rishabh Mittal

B.Tech., Indian Institute of Technology, Delhi (2017)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2020

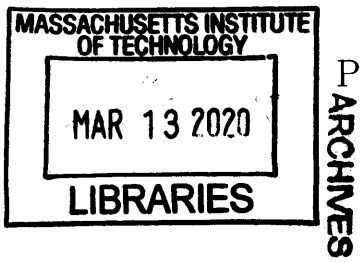
Signature redacted

Author .....
Department of Electrical Engineering and Computer Science
January 27, 2020

Certified by ..... Signature redacted .....
Anantha P. Chandrakasan
Vannevar Bush Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by .... Signature redacted .....
Hae-Seung Lee
ATSP Professor of Electrical Engineering
Thesis Supervisor

Accepted by ..... Signature redacted .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students



A Sampling Jitter Tolerant Continuous-Time Pipeline ADC

by

Rishabh Mittal

Submitted to the Department of Electrical Engineering and Computer Science
on January 27, 2020, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

A sampling jitter tolerant continuous-time (CT) pipeline ADC has been presented in this thesis. In conventional discrete-time (DT) pipeline ADCs, the input is sampled upfront. The improvements in the bandwidth and sampling speed due to CMOS scaling have brought the deleterious effects of sampling clock jitter to the forefront. Any jitter in the sampling clock edge adds a random error to the input signal thereby limiting the maximum achievable signal-to-noise ratio (SNR), and hence the effective resolution of the ADC. The effect of sampling clock jitter has been considered fundamental. In the proposed ADC, we do not sample the input upfront. Rather, we sample the residue from the first stage. Since the residue is bandlimited and has a small magnitude, therefore it will have a smaller derivative. Hence, the sensitivity to the clock jitter will be greatly reduced.

Thesis Supervisor: Anantha P. Chandrakasan
Title: Vannevar Bush Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Hae-Seung Lee
Title: ATSP Professor of Electrical Engineering

Acknowledgments

I would like to thank my advisors Prof. Hae-Seung (Harry) Lee, Prof. Anantha Chandrakasan, and my industry supervisor Dr. Gabriele Manganaro for their guidance and insightful discussions all along. I consider myself very fortunate to have gotten the opportunity to work with them and learn from their immense experience in the field of circuit design. They have been very patient in guiding me, and extremely encouraging of my endeavors. Their support has been indispensable in the completion of this thesis.

I am also very grateful to my colleagues at MIT, especially Xi Yang, Harneet Singh Khuraana, Preet Garcha, Taehoon Jeong and Vipasha Mittal. Being the senior-most student in our group, Xi had always been very helpful with circuit design as well as simulator related issues. Harneet and Preet are always enthusiastic about discussions on a variety of topics, not just related to ADC design, which makes me think about the problem at hand from different perspectives. Taehoon has been an incredible mentor since my day one at MIT. Be it research work, courses or navigating life as a grad student, Taehoon has always been there to guide me. And finally, I would like to thank Vipasha for being an incredible friend, whom I can always count on for any kind of problem, be it related to research or life in general.

I would also like to thank Janet Fischer, Kathy McCoy, Alicia Duarte and Leslie Kolodziejski from the EECS department at MIT, and Emily Cheng from MIT's International Student Office to facilitate my collaboration with Analog Devices (ADI) which was absolutely essential for the completion of this work.

While collaborating with ADI, I got the opportunity to interact with Hajime Shibata and Sharvil Patil. Their technical discussions were quintessential in shaping the direction of this project. Also, I would like to thank my colleague and friend Athanasios (Thanos) Ramkaj. He took keen interest in my project and was always ready for technical discussions.

I would like to thank all my friends, especially Devansh, Sidharth, Anant, Ritesh and Aditi for being there with me both in happy and tough times, and for always

believing in me. Last but not the least, I would like to thank my family for their unconditional love, support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Almost all real-world signals are analog (continuous-time and continuous-amplitude), for example, temperature, pressure, sound, light, etc. But due to advances in the integrated circuit (IC) technology, most of the data is stored and processed digitally (discrete-time and quantized amplitude). Therefore, analog-to-digital converters (ADCs) are an essential part of any electronic system.

The advances in modern communication systems including the 5G mobile networks and baseband processors necessitate the ADCs to have a large dynamic range with high digitization bandwidth (about 1 GHz). Because of innovative architectures, improved technologies and continual optimization, there have been steady improvements in the performance of ADCs [1]. However, the speed-resolution product is limited by the sampling clock jitter [2–4]. Fig. 1-1 shows the performance trends of ADCs [5] elucidating the SNDR limitation due to sampling clock jitter.

In conventional pipeline ADCs, the analog input is sampled upfront [6–12]. If the sampling clock has any jitter, that directly affects the sampled voltage by introducing a random error in the sampled input voltage given by

$$\Delta v = \frac{dv}{dt}\Delta t \tag{1.1}$$

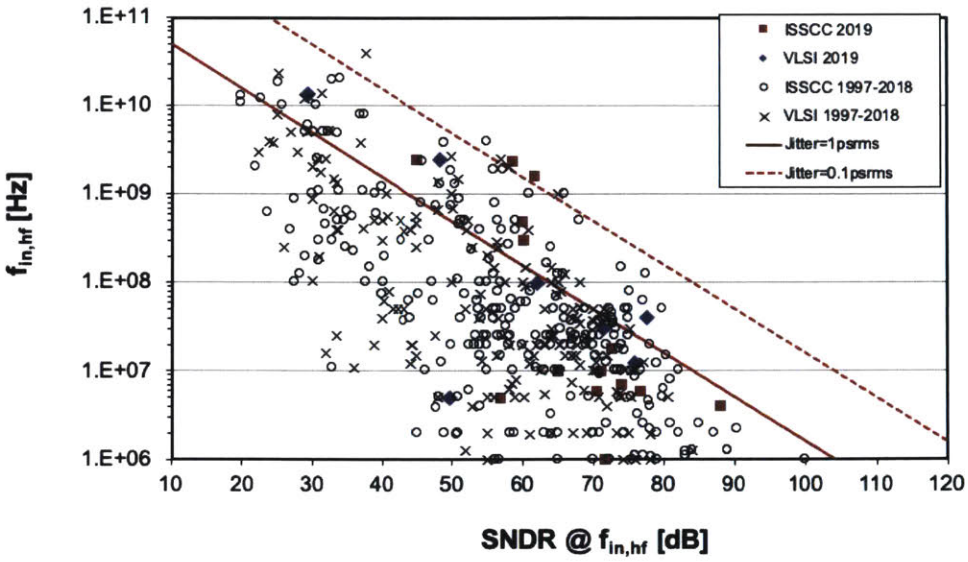


Figure 1-1: Performance survey for published ADCs (ISSCC 1997-2019 and VLSI 1997-2019) [5].

where dv/dt is the time-derivative of the input signal and Δt is the error in the sampling instant compared to a fictitious ideal sampling clock. Note that the error due to clock jitter is independent of the clock frequency. Since the jitter in the sampling clock edge is random, the error in the sampled voltage is also random. This introduces a random noise in the sampled signal and limits the signal-to-noise ratio (SNR). For a sinusoidal input signal, the maximum achievable SNR is given by [2, 4]:

$$\mathrm{SNR}_{\mathrm{max}} = \frac{1}{2\pi f_{in}\sigma_t} \tag{1.2}$$

where f_{in} is the input signal frequency and σ_t is the RMS jitter in the sampling clock. Typically, it is difficult to reduce the RMS jitter below 100 fs [13]. This limits the maximum achievable SNR to just 44 dB (which is equivalent to 7 bits) for a 10 GHz signal. Therefore, unless the effect of sampling jitter is reduced, the performance of an ADC would be greatly limited for high frequency input signals.

The dynamic range limitation due to the sampling clock jitter has been considered fundamental. However, some continuous-time delta-sigma modulators (CT-DSMs) have been shown to be tolerant to the sampling jitter [14–16]. But the requirement of

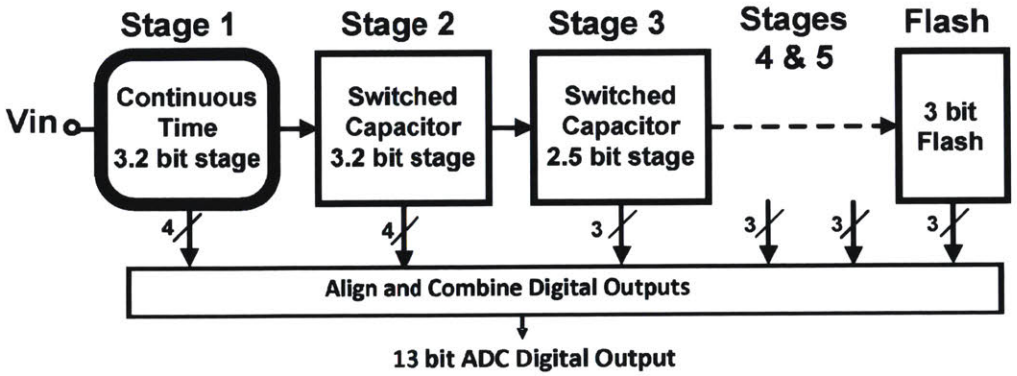


Figure 1-2: A continuous-time input pipeline ADC presented by Gubbins et al [18].

a relatively high oversampling ratio (OSR) [17], usually greater than 8, renders them unsuitable for high frequency applications. Therefore, Nyquist-rate ADCs or ADCs with low OSR are advantageous for high speed data conversion.

1.2 Literature Survey

A continuous-time input pipeline ADC was first introduced in [18]. This work showed some of the key benefits of using a continuous-time pipeline architecture like inherent anti-aliasing and relaxed non-linearity requirements. Fig. 1-2 shows the ADC architecture consisting of a continuous-time first stage followed by several switched-capacitor stages (as in a regular pipeline ADC). To match the timing of the signal path and the feedforward (ADC-DAC) path, a prediction filter has been used in the feedforward path. However, the prediction filter needs to be accurate within an error budget to have a small residue and to avoid saturating the back-end stages of the pipeline ADC. But even with a 9-level quantizer, the first stage gain is just 4/3 as shown in Fig. 1-3 because of the accuracy requirements for the prediction filter. Since the input-referred noise and distortion of the back-end stages is reduced by the gain of the first stage, a low interstage gain lessens the benefit of having a pipeline.

An alternate solution to using prediction filter for mitigating the timing mismatch in the signal path and the feedforward (Flash-DAC) path has been presented in [19]. A delay is introduced in the signal path by using an RC-lattice delay line as shown in Fig. 1-4. Although the cancellation is not perfect over the entire signal bandwidth

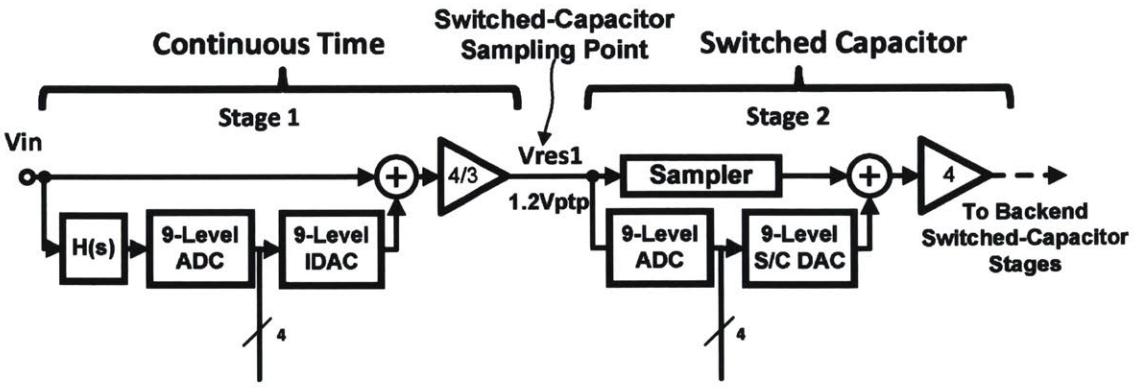


Figure 1-3: The first two stages of the continuous-time ADC presented by Gubbins et al [18] having a 9-level quantizer and interstage gain of 4/3.

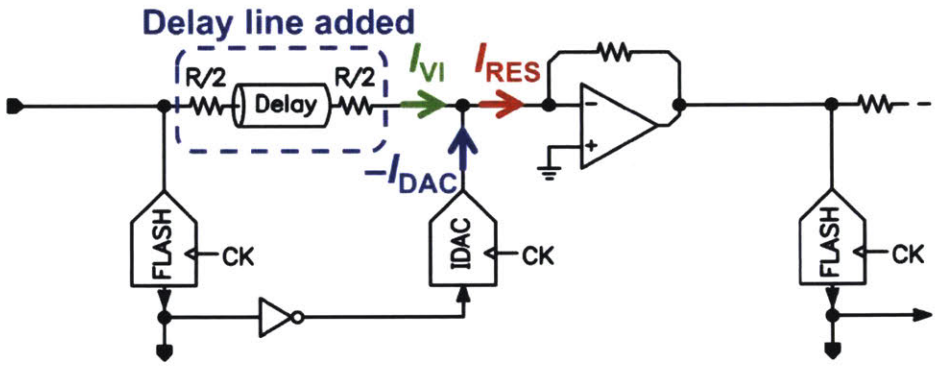


Figure 1-4: An alternate continuous-time pipeline ADC with a delay in the signal path presented by Shibata et al [19].

owing to the phase mismatch in the signal path and the Flash-DAC path, the residue magnitudes are acceptable for the 4-bit quantizer that has been used in their design. However, the back-end ADC consists of six identical continuous-time stages which is responsible for the massive power consumption of 2.3 W. Also, the transimpedance amplifier uses a simple RC filter to low-pass-filter the residue. This can in fact be modified to achieve tolerance to sampling jitter in the back-end ADC as described in more detail in section 3.2.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 gives an overview of the conventional pipelined ADCs. First, the working of the discrete-time (DT) pipelined ADCs has

been described, followed by highlighting some of the limitations of that architecture. Continuous-time (CT) pipelined ADCs have been proposed as a solution to overcome the drawbacks of the DT pipeline architecture and have been discussed in Chapter 3. The system architecture of the CT pipeline ADC has been described. The effect of DAC jitter and the clock jitter in the back-end sampler have also been discussed. Later, in Chapter 4, the behavioral model for the CT pipeline ADC has been described. The details of the various sub-blocks – Flash-DAC path, sample-and-hold, quantizer and DAC-jitter-generator have been presented in this chapter. Then, the implementation of the digital reconstruction filter has been described, followed by a discussion on the simulation results showing the jitter tolerance of the proposed CT pipelined ADC architecture. Finally, the thesis has been summarized in Chapter 5.

Chapter 2

Concept of Pipeline ADCs

There are several popular ADC topologies for Nyquist-rate A/D conversion like flash ADC, single/dual-slope ADCs, successive-approximation ADCs and pipeline ADCs [20–22]. But pipeline ADCs are preferred for high resolution and high bandwidth applications. The concept of a pipeline ADC is an old idea first published in the 1950s [23]. Various low resolution A/D stages are cascaded and the bits from all the stages are combined appropriately to give a high resolution digital output. This chapter describes the working of conventional discrete-time pipelined ADCs and analyzes the limitations with this architecture. Continuous-time pipeline ADCs have been proposed as a possible solution.

2.1 Discrete-Time Pipeline ADCs

To illustrate the concept of discrete-time pipeline ADC, a simplified schematic is shown in Fig 2-1. The analog input signal x is sampled upfront by a sample-and-hold circuit. Then, this sampled signal is processed by a cascade of pipeline stages. In order to avoid aliasing due to sampling, an anti-alias filter precedes the sample and hold. In the first stage, the sampled signal is quantized using a low resolution (typically 2-4 bits) A/D converter. This sub-ADC adds a quantization error e_1 to the

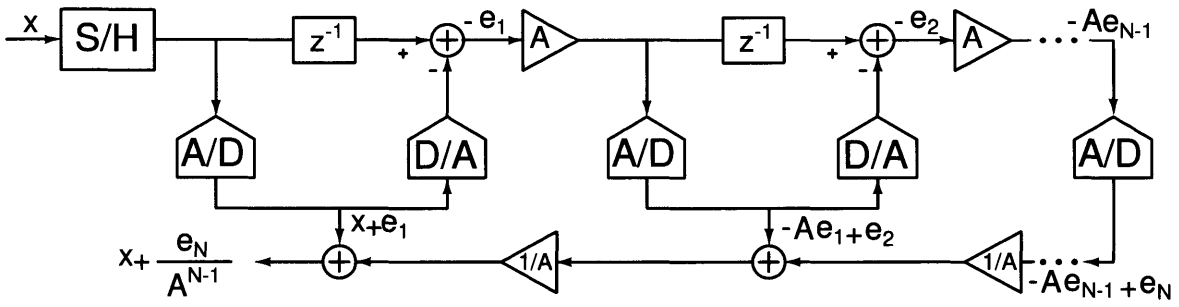


Figure 2-1: A simplified schematic of a discrete-time pipeline ADC.

sampled signal. The output from the first stage is:

$$y_1 = x + e_1 \tag{2.1}$$

The sampled signal along with the error e_1 is passed through a D/A converter which gives an analog representation of $x + e_1$. This signal is subtracted from the sampled input signal (delayed appropriately so that it is in phase with $x + e_1$) and a residue $-e_1$ is obtained. To rescale the output to the full scale, the residue is amplified by a factor A. This amplified residue is passed through the second stage sub-ADC which adds an error e_2 to give the stage 2 output as:

$$y_2 = -Ae_1 + e_2 \tag{2.2}$$

Therefore, if there are N stages in the pipeline and the stage i sub-ADC adds an error e_i, then the stage i output will be given by:

$$y_i = -Ae_{i-1} + e_i \tag{2.3}$$

for $i = 2, 3, ..., N$. To reconstruct the final digital output, the bits from all the stages are appropriately weighted and combined in order to account for the inter-stage gain. Note that in this architecture, the delay and residue amplification blocks

are implemented in discrete-time. The final digital output is given by

$$y = y_1 + y_2 + y_3 + ... + y_{N-1} + y_N \tag{2.4}$$

$$= (x + e_1 + \frac{1}{A}(-Ae_1 + e_2 + \frac{1}{A}(-Ae_2 + e_3 + ... \\ ... + \frac{1}{A}(-Ae_{N-2} + e_{N-1} + \frac{1}{A}(-Ae_{N-1} + e_N))...))) \tag{2.5}$$

$$= x + \frac{e_N}{A^{N-1}} \tag{2.6}$$

The final output is a combination of the input signal x and the last pipeline stage error e_N attenuated by the product of the inter-stage gains. The errors due to the sub-ADCs get cancelled, and therefore, we achieve a very high resolution digital output using low-resolution ADCs in every stage of the pipeline ADC.

2.2 Discrete-Time to Continuous-Time Transformation

The above mentioned DT pipeline ADC can be transformed into a CT pipeline ADC by removing the sample-and-hold (S/H) upfront, and replacing the discrete-time blocks (delay and residue amplifier) with their continuous-time counterparts. A simplified schematic of the transformed CT pipeline ADC is shown in Fig. 2-2. Since there is no S/H upfront, the discrete-time delay block z^{-1} has been replaced by its continuous-time counterpart e^{-sT}. The delay T is chosen to match the delay in the ADC-DAC path. Similarly, the discrete-time residue amplifier A is replaced with a continuous-time amplifier $H(s)$ with a DC gain of A and sufficient bandwidth to accommodate the input signal.

For digital recombination of the bits from every stage, the weighing factor $1/A$ is replaced with $H^{-1}(z)$, the inverse of the residue amplifier transfer function [1]. The

[1] Strictly speaking, we have to apply an impulse-invariant transformation on $H(s)$ before taking its inverse to get $H^{-1}(z)$. This has been discussed in detail later in section 4.3

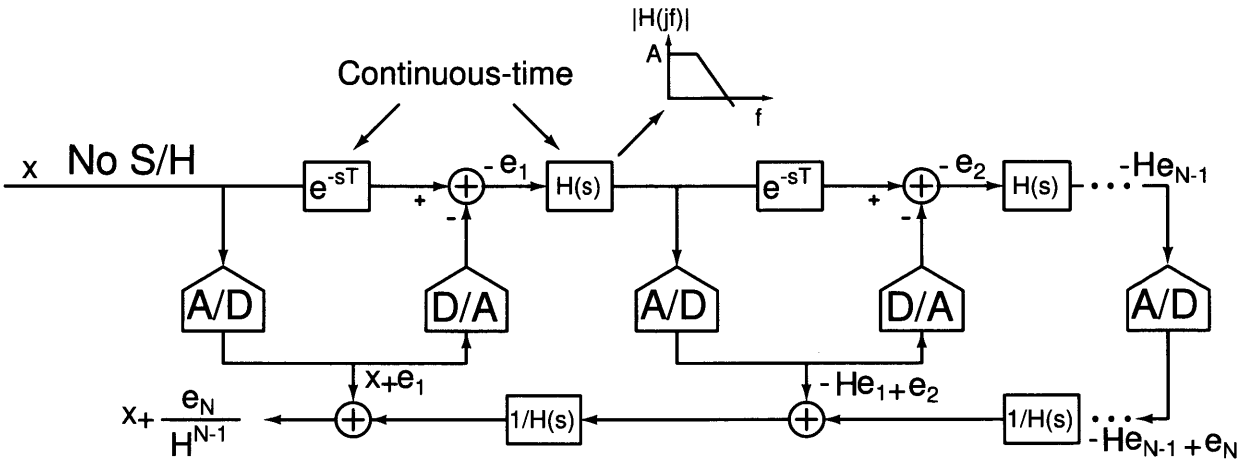


Figure 2-2: Transformation of a discrete-time pipeline ADC to a continuous-time pipeline ADC.

final digital output in case of CT pipeline ADC is:

$$y = x + \frac{e_N}{H^{N-1}} \tag{2.7}$$

Again, similar to DT pipeline ADC, only the error in the last stage sub-ADC ends up reaching the output. All of the other intermediate sub-ADC's errors get cancelled to the degree that $H^{-1}(z)$ is matched to $H(s)$. Also, since $H(s)$ has a low-pass response, the error in the final digital output is high-pass shaped.

2.3 Inherent Anti-Aliasing in CT Pipeline ADC

As discussed previously in section 2.1, in a DT pipeline ADC, the analog input is sampled upfront which causes aliasing. Since aliasing happens right at the input, there is no way to distinguish the signal from the aliased components. Therefore, it is impossible to get rid of the aliased components anywhere in the signal chain during the discrete-time processing.

This necessitates the use of an anti-alias filter (AAF) before the signal is given to the DT pipeline for A/D conversion. Since the AAF is processing the signal directly, any distortion in the AAF will directly impact the signal. Therefore, a high linearity specification is needed for the AAF. Moreover, since the AAF needs to drive the

capacitors in the S/H block, it has to have a high peak driving current. Hence the power consumption of the AAF is expected to be quite high.

On the other hand, a CT pipeline ADC provides inherent anti-aliasing. Even though aliasing happens at all the sub-ADCs, the errors in stages $1, ..., N-1$ ($e_1, ..., e_{N-1}$) get cancelled and do not reach the output. Only the error in the last sub-ADC (e_N) reaches the output, and therefore, aliasing matters only in the last sub-ADC. The residue goes through a cascade of $H(s)$ ($N-1$ times) which acts as an inherent anti-aliasing filter.

2.4 Summary

In this chapter, the concept behind pipelined A/D converters was explained. First, the working of the conventional DT pipeline ADCs was discussed, followed by how it can be transformed into a CT pipeline ADC. A CT pipeline ADC inherits the wide bandwidth benefit from the DT pipeline architecture while having inherent anti-aliasing due to a continuous-time front end. A qualitative comparison of CT-$\Delta\Sigma$ ADC, DT pipelined ADC and CT pipelined ADC is shown in Table 2.1. Although coming with the cost of a more complex digital reconstruction, CT pipeline looks to be a promising architecture for high speed and high resolution A/D conversion.

	CT-$\Delta\Sigma$ ADC	DT Pipeline ADC	CT Pipeline ADC
Inherent Anti-Aliasing	Yes	No	Yes
Driving Current	Small	Large	Small
Stability	Can go unstable	Always stable	Always stable
Bandwidth	Low (OSR $>$ 8 or 16)	High (Nyquist-rate)	High (OSR $<$ 4)
Digital Recons-truction	Simple	Simple	Complex

Table 2.1: Qualitative comparison of continuous-time $\Delta\Sigma$ ADC, discrete-time pipelined ADC and continuous-time pipelined ADC.

Chapter 3

Proposed Continuous-Time Pipeline ADC

As discussed at the end of the previous chapter, the SNR limitation due to clock jitter is fundamental in conventional discrete-time pipelined ADCs because of the sample and hold upfront. Therefore, to eliminate this error, we remove the sample-and-hold from the input of the first stage and place it at the input of the second stage as shown in Fig. 3-2. This opens room for the use various innovative techniques throughout the signal chain as will be discussed in the rest of this chapter.

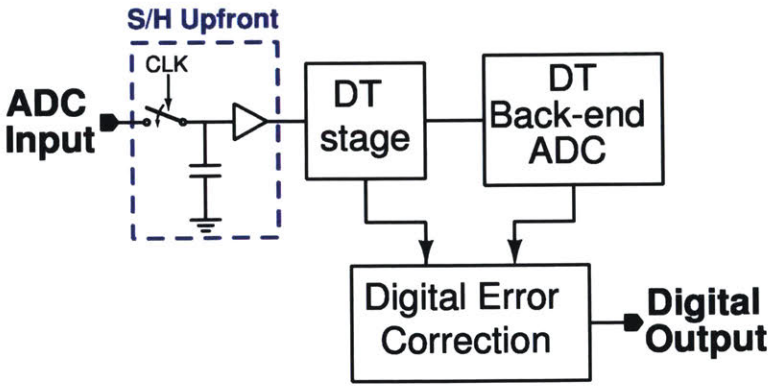


Figure 3-1: A conventional discrete-time pipelined ADC with a sample-and-hold upfront.

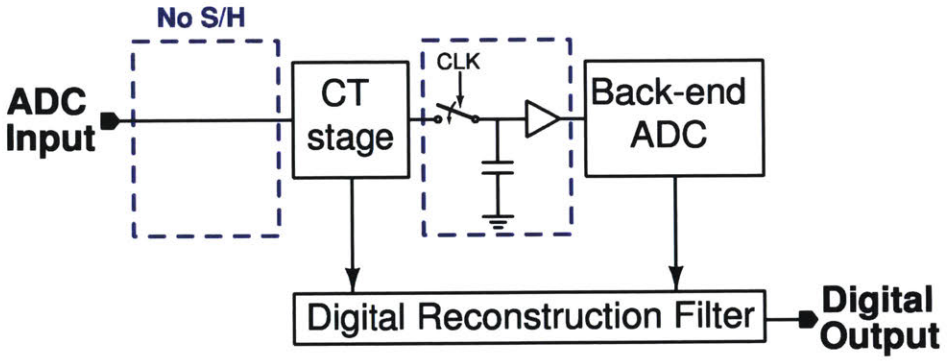


Figure 3-2: Eliminating the sample-and-hold upfront and using a continuous-time first stage.

3.1 System Architecture

Fig. 3-3 shows the overall system architecture which comprises two stages. The analog input V_{in} is applied to the first stage. The sub-ADC in the first stage is typically a 3-4 bit flash ADC which gives a digital estimate of the input (discrete-time and discrete/quantized-amplitude):

$$D_1 = V_{in} + E_{ADC1} \tag{3.1}$$

where E_{ADC1} includes all the errors from the stage 1 sub-ADC. Error from the stage 1 sub-ADC is dominated primarily by the quantization noise. The stage 1 sub-ADC output goes to the DAC which converts the incoming digital input to an analog

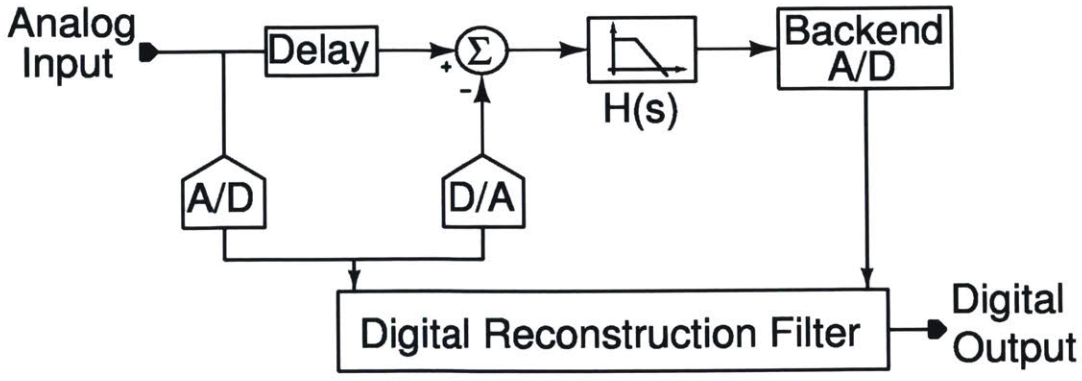


Figure 3-3: The overall system architecture for the proposed jitter tolerant continuous-time pipeline ADC.

output:

$$V_{in,est} = V_{in} + E_{ADC1} + E_{DAC} \tag{3.2}$$

where E_{DAC} is the DAC error due to any non-idealities. The sub-ADC-DAC path (which we will also refer to as the Flash-DAC path) introduces a delay of half clock cycle. Therefore, we need a delay block in the signal path to match its delay to the Flash-DAC path [1]. Then, the estimate of the input is subtracted from the delayed input to give the residue:

$$V_{res} = V_{in,del} - V_{in,est} = -(E_{ADC1} + E_{DAC}) \tag{3.3}$$

This residue goes through a low-pass filter with cut-off frequency equal to the bandwidth of the input signal to filter out the out-of-band noise. Then the low-pass filtered residue is amplified to full-scale. In conventional discrete-time pipeline ADCs, the residue is amplified by a factor of 2^{N_1-1} where N_1 is the resolution of stage 1 sub-ADC. The interstage gain is chosen to be 2x smaller to avoid overrange due to errors from the stage 1 ADC [11, 22]. In a continuous-time pipeline ADC, the interstage gain doesn't have to be 2^{N_1-1} necessarily. Any value of interstage gain that prevents overloading the stage 2 ADC works fine in practice [2]. The input to the second stage ADC or the backend ADC is

$$V_2 = H_{LPF} \cdot V_{res} = H_{LPF} \cdot (-E_{ADC1} - E_{DAC}) \tag{3.4}$$

where H_{LPF} is the combined transfer function for low-pass filter and amplifier. The backend ADC can either be regular discrete-time pipelined ADC [18], a continuous-time pipeline ADC [19], or any high-resolution ADC that can digitize the residue.

[1] Having a delay with a sinc magnitude response will be even better because then the transfer functions from both paths will be perfectly matched, and it will eliminate any undesirable components at multiples of clock frequency, essentially providing better anti-aliasing. Also, this will prevent any undesirable frequency components at f_{CLK} and its multiples from over-loading/saturating the backend ADC.

[2] Although from MATLAB simulations it turns out that an interstage gain of 2^{N_1-1} is a pretty good choice for an oversampling ratio of 4 and filter order = 1 to 6. This will be elaborated further in section 3.2.

The backend ADC digitizes the residue from stage 1 to give

$$D_2 = H_{LPF} \cdot (-E_{ADC1} - E_{DAC}) + E_{STG2} \tag{3.5}$$

where E_{STG2} includes all the errors from the stage 2 ADC. Now, the digital reconstruction filter (DRF) combines the stage 1 and stage 2 outputs D_1 and D_2 respectively to give the final digital output:

$$D_{out} = D_1 + H_{DRF} \cdot D_2 \tag{3.6}$$

$$= V_{IN} + E_{ADC1} + H_{DRF} \cdot [H_{LPF} \cdot (-E_{ADC1} - E_{DAC}) + E_{STG2}] \tag{3.7}$$

If the DRF transfer function is chosen to be H_{LPF}^{-1}, then the error due to the stage 1 ADC will get cancelled and the final digital output will be [3]:

$$D_{out} = V_{IN} + E_{ADC1} \cdot [1 - H_{DRF} \cdot H_{LPF}] + H_{DRF} \cdot H_{LPF} \cdot (-E_{DAC}) + H_{DRF} \cdot E_{STG2} \tag{3.8}$$

$$D_{out} = V_{IN} - E_{DAC} + H_{DRF} \cdot E_{STG2} \tag{3.9}$$

Eq. 3.8 tells us that the any error due to the stage 1 Flash ADC will get cancelled to the degree that H_{DRF} is matched to H_{LPF}. This is similar to the regular pipeline ADC, albeit slightly different from the discrete-time pipeline ADC because of a more complex digital recombination of the bits from stage 1 and stage 2. In case of a discrete-time pipelined ADC, the digital recombination is extremely simple – the bits from each stage are shifted and added (with one-bit overlap to account for the 2x smaller interstage gain) to give the final digital code. But in case of continuous-time pipeline, the digital reconstruction filter has to match the inverse of the analog low-pass-filter between stage 1 and stage 2 to effectively cancel the errors from the stage 1 sub-ADC.

[3]Notice that we have been tacitly avoiding either a Laplace-Transform or a Z-Transform notation to represent transfer function for the low-pass filter transfer function and the DRF. This issue has been addressed and resolved with great clarity throughout Chapter 4 and specifically in section 4.3. But for now, it suffices to focus on the working of the continuous-time pipeline ADC.

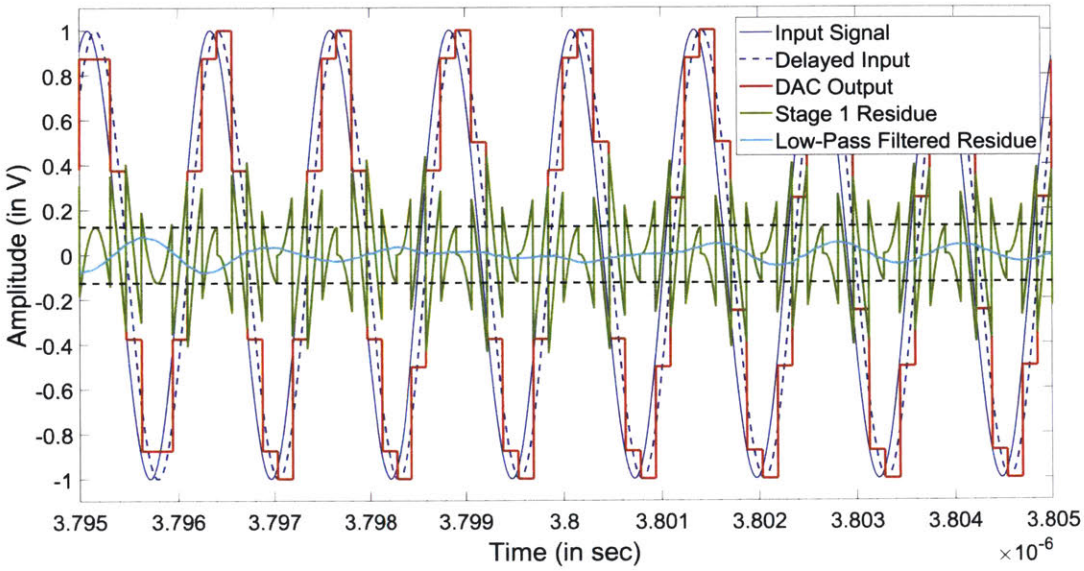


Figure 3-4: A plot showing the analog input, DAC output and the residue signal at stage 1 before and after low-pass filtering.

The digital reconstruction filter has a high-pass response, the exact opposite of H_{LPF}. From Eq. 3.9 we can see that the errors due to stage 2 ADC will be shaped (high-pass). Eq. 3.9 also tells us about the effect of the DAC errors and the stage 2 ADC errors which are discussed in section 3.2-3.3.

3.2 Effect of Stage 2 Jitter

As shown in Fig. 3-2, there is a sample and hold at the input of stage 2. A random error will be introduced in the sampled residue voltage if there is any jitter in the sampling clock. This error is proportional to the derivative of the signal V_2 (the input to the second stage) as evident from Eq. 1.1. The residue voltage from stage 1 before and after low-pass filtering are shown in Fig. 3-4. Before low-pass filtering, the residue has the exact same derivative as the input signal [4]. Therefore, if it were sampled, there would have been no improvement in jitter sensitivity.

But we can see that the low-pass filtered residue has a much smaller amplitude, and since it is a bandlimited signal, it has a much smaller derivative as well. Since

[4]Ignoring the transition points which occur near the clock edges.

after amplification, the differential residue's amplitude goes up to at most the full-scale voltage V_{FS} (to avoid overloading the backend ADC), therefore we can be certain that the amplitude of the low-pass-filtered residue is less than $V_{FS}/2^{N_1-1}$ where N_1 is the resolution of the stage 1 ADC.

We assume here that the clock jitter in the sampling clock Δt is a random variable with zero mean and Gaussian distribution. Let the RMS jitter be $\sigma_{\Delta t}$. Since the clock jitter is independent of the input signal, it is also independent of the residue and its derivative. Therefore, the power in the error voltage Δv can be written as:

$$P_{\Delta v} = \sigma_{\Delta t}^2 \cdot \left(\frac{dv}{dt}\right)_{RMS}^2 \tag{3.10}$$

where $\left(\frac{dv}{dt}\right)_{RMS}^2$ is the power in the derivative of the low-pass-filtered residue. The RMS value of the derivative of the residue reduces by approximately the gain of the first stage. Therefore, the RMS noise voltage due to clock jitter reduces (at least [5]) by about 2^{N_1-1}. Therefore, if we use a 4-bit Flash-ADC in stage 1, then, in absence of any other non-idealities, we get an improvement of about $20\log(2^{4-1}) = 18$-dB in the SNR limit due to clock jitter. This means that if we have an 800 MHz input signal sampled with a clock having 200 fs RMS jitter, then the SNR limitation due to clock jitter will be 78 dB instead of 60 dB (as calculated from Eq. 1.2) in absence of any other non-idealities.

3.3 Effect of DAC Jitter

From Eq. 3.9 we see that the DAC error shows up in the final digital output. Intuitively, this happens because the DAC error doesn't show up in the stage 1 digital output D_1 while it goes through the low-pass-filter, stage 2 and the DRF to show up in the stage 2 output D_2. Any clock jitter in the DAC will show up directly at the output and thereby limit the maximum achievable SNR. Therefore the DAC design

[5]The amplitude of the residue, and therefore the RMS of the derivative of the residue, depend on the stage 1 Flash ADC and DAC errors. So it can either increase or decrease based on how well the Flash-DAC path has been designed. But in any case, the residue amplitude will be less than $V_{FS}/2^{N_1-1}$.

is very crucial in this continuous-time pipelined ADC.

Some jitter tolerant DAC architectures using switch capacitor DACs already exist and have been used in the design of continuous-time delta-sigma ADCs [14–16,24–26]. Using a raised sinewave DAC is another possibility [27]. A similar DAC architecture can be used in this design as well.

Chapter 4

Behavioural Model

4.1 MATLAB Model Description

Our continuous-time pipeline ADC architecture consists of analog, digital and mixed-signal blocks as shown in Fig. 4-1. We have modelled all the blocks by writing our own custom functions for sample-and-hold, quantizer and low-pass-filter which constitute the ADCs, DAC and the digital reconstruction filter. Although a bit cumbersome, this was done to make sure that none of the essential properties of continuous-time signals were compromised given that we modelled the system in MATLAB which is inherently discrete-time. In our ADC model, the clock frequency F_{CLK} is 6.4 GHz ($\sim$ 156 ps) with a signal bandwidth of 800 MHz (OSR = 4). But to model the continuous-time signals we have used numerical sampling frequencies of $128 \cdot F_{CLK}$ ($\sim$ 1.2 ps), $512 \cdot F_{CLK}$ ($\sim$ 305 fs) or $4096 \cdot F_{CLK}$ ($\sim$ 38 fs) depending on how fast or how accurate we want the simulations. The description of the various sub-blocks has been given in the following sections.

4.2 Flash-DAC Path

The Flash-DAC path in stage 1 has been modelled as a sample-and-hold followed by a quantizer since it achieves the same functionality (described in section 4.2.1-4.2.2). This is true only when there is no clock jitter in the sub-ADC and the DAC.

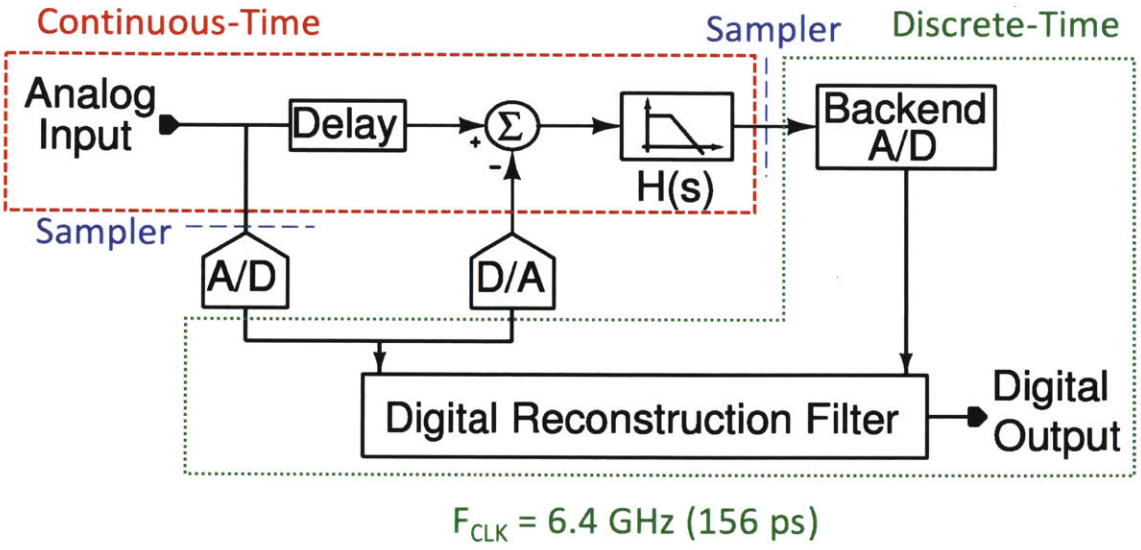


Figure 4-1: A behavioural model for the continuous-time pipeline ADC highlighting the continuous-time and discrete-time portions of the system.

In presence of clock jitter, the Flash-ADC will sample an erroneous value at the its input and the DAC output edges will have some timing error. Clearly, the ways in which the clock jitter affects the ADC and DAC are quite different. Also, in general, the values for the RMS jitter in both the clocks can be different. Therefore, it is imperative to model the effect of clock jitter on the ADC and the DAC separately.

Given any continuous-time input, the purpose of the Flash-DAC path is to digitize the input, extract N_1 bits and then produce an analog output which can only take one out of the 2^{N_1} quantized values. We can model the continuous-time signals as sampled signals (discrete-time) but with a very high numerical sampling frequency, say F_s, where $F_s \gg F_{CLK}$.

We can imagine a single system equivalent to the Flash-DAC path which has one input (the analog input signal), several input parameters (RMS jitter for sub-ADC clock, the DAC jitter, resolution of the sub-ADC N_1, sampling frequency for *continuous-time* signals F_s and clock frequency F_{CLK}), and two outputs (digital output D_1 from the sub-ADC and the continuous-time DAC output V_{DAC}). This system consists of three components (implemented as functions in MATLAB): sample-and-hold, quantizer and a DAC-jitter-generator.

4.2.1 Sample-and-Hold

The sample-and-hold function looks at the analog input signal at multiples of the clock period $T_{CLK} = 1/F_{CLK}$. Therefore,

$$V_{S/H} = V_{IN}(n \cdot T_{CLK}) \tag{4.1}$$

for $n = 1, 2, 3, ..., N$ such that $N \cdot T_{CLK} < T_{SIM}$ where T_{SIM} is the simulation time[1]. In presence of finite clock jitter in the ADC clock, the sample-and-held voltage becomes

$$V_{S/H} = V_{IN}(n \cdot T_{CLK} + \Delta t_{n,ADC}) \tag{4.2}$$

where $\Delta t_{n,ADC}$ is the random jitter in the ADC clock edge at the nth sample[2]. The RMS value of this jitter is $\sigma_{\Delta t_{ADC}}$ which is a parameter in the sample-and-hold function. Although technically the output of a sample-and-hold is a discrete-time signal, we have modelled $V_{S/H}$ as a *continuous-time* signal (i.e. with a sampling rate of F_s and not F_{CLK}) such that

$$\begin{aligned} V_{S/H}(nT_{CLK} + \Delta t_{n,ADC} : (n+1)T_{CLK} + \Delta t_{n+1,ADC}) \\ = V_{IN}(nT_{CLK} + \Delta t_{n,ADC}) \end{aligned} \tag{4.3}$$

with the understanding that downsampling $V_{S/H}$ by a factor of F_s/F_{CLK} will give us the discrete-time sample-and-hold output at the ADC clock rate F_{CLK}.

4.2.2 Quantizer

The quantizer function simply takes $V_{S/H}$ (at sample rate F_s) and N_1 (the resolution of stage 1) as inputs and quantizes the amplitude to one of the $2^{N_1} + 1$ levels[3] depending on the current value of $V_{S/H}$. If the output of the quantizer V_Q is downsampled by a

[1]The end case from $t = N \cdot T_{CLK}$ to T_{SIM} has been handled separately.

[2]The value of sampling period for the *continuous-time* signals $1/F_s$ has to be sufficiently smaller than $\Delta t_{n,ADC}$ else the effect of the clock jitter will never show up in $V_{S/H}$.

[3]One extra level has been added to cover the entire full-scale from V_{SS} to V_{DD} in the quantizer output.

factor of $M = F_s/F_{CLK}$, then we will get the stage 1 digital output

$$D_1[n] = V_Q[n \cdot M] \tag{4.4}$$

which can be combined with the stage 2 digital output D_2 to get the final digital output D_{OUT} (discussed in section 4.3). Note that in the absence of any clock jitter, the DAC output V_{DAC} will be same as the quantizer output V_Q since both are *continuous-time* signals (sampled at F_s). But in the presence of clock jitter, the DAC output will differ from the quantizer output as discussed in the following section.

4.2.3 DAC-Jitter-Generator

The DAC output gets affected by the jitter in the sub-ADC clock as well as the DAC clock. If there is any jitter in the sub-ADC clock, the quantizer output might have an error[4]. This means that the *digital input* (discrete-amplitude) to the DAC itself has an error. Therefore, even if there is no other non-ideality in the DAC, its output will still have an *amplitude error*. Now if the DAC clock has some jitter, then the timing of the DAC output transition will be slightly off (as compared to a fictitious ideal clock) and this will lead to a *timing error* in the DAC output.

The primary requirement of the DAC-jitter-generator function is to incorporate the DAC timing error while simultaneously taking into account any amplitude errors due to sub-ADC clock jitter. This can be done as follows:

$$\begin{aligned} V_{DAC}(nT_{CLK} + \Delta t_{n,DAC} : (n+1)T_{CLK} + \Delta t_{n+1,DAC}) \\ = V_Q(nT_{CLK} + \Delta t_{n,ADC}) \end{aligned} \tag{4.5}$$

where $\Delta t_{n,DAC}$ is a random jitter in the DAC clock affecting the timing of the n^{th} DAC output. The argument (time index) of V_Q captures any amplitude error due to the sub-ADC clock jitter while the argument of V_{DAC} captures the timing error due to the DAC clock jitter. Similar to the sample-and-hold function, the DAC-jitter-

[4]This error probability due to clock jitter will be very small since V_Q will be limited by quantization noise and not the jitter noise because the stage 1 ADC is only resolving 4 bits.

generator function also takes the RMS DAC jitter $\sigma_{\Delta t_{DAC}}$ as an input parameter[5]. In general, $\sigma_{\Delta t_{ADC}}$ and $\sigma_{\Delta t_{DAC}}$ can have different values depending on the clock path design. But V_{DAC} captures all the effects and can be used as the DAC output for further processing in our continuous-time pipelined ADC model.

4.3 Digital Reconstruction Filter

The role of the digital reconstruction filter (DRF) is to combine the stage 1 and stage 2 outputs in such way so that the errors due to the stage 1 sub-ADC get cancelled as shown by Eq. 3.8. This entails the DRF transfer function to match the inverse transfer function of the analog low-pass filter between stage 1 and stage 2. Since the inputs to the reconstruction filter are digital, the DRF has to be a digital filter. The DRF transfer function can be obtained by applying impulse-invarint transformation, $\mathcal{I}$, on the analog low-pass filter transfer function [19, 28]

$$H_{DRF}(z) = \mathcal{I}\{H_{LPF}^{-1}(s)\} \tag{4.6}$$

where $H_{LPF}(s)$ is the low-pass filter transfer function.

The analog low-pass filter has a cut-off frequency of 800 MHz (equal to the input bandwidth BW). In our MATLAB model, we modelled the analog low-pass filter as a digital filter with a cut-off frequency

$$\omega_c = \frac{BW}{F_s/2} \tag{4.7}$$

The stage 1 and 2 digital outputs are coming at the rate of F_{CLK}. Using the decimation identity as shown in Fig. 4-2, the filtering and decimation operations can be interchanged [28]. Therefore, the cut-off frequency for the DRF becomes $BW/(F_{CLK}/2)$ as shown in Fig. 4-3.

[5]The DAC-jitter-generator does not take $\sigma_{\Delta t_{ADC}}$ as an input. Rather, the sample-and-hold function saves the timing information, which is fed to the DAC-jitter-generator to be able to account for any amplitude errors.

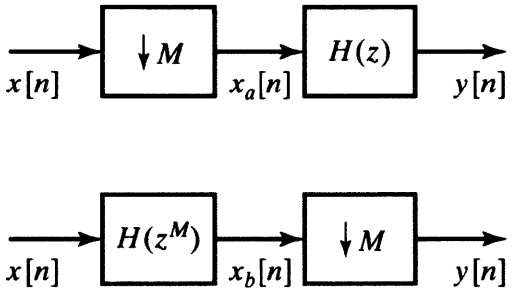


Figure 4-2: The decimation identity: the filtering and decimation operation can be interchanged if the filter transfer function is transformed as shown.

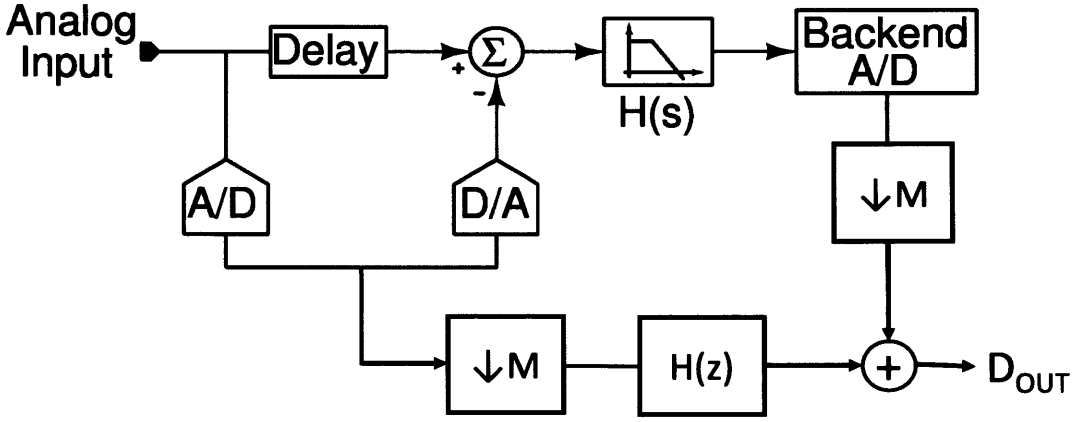


Figure 4-3: The digital reconstruction filter operating at F_{CLK}.

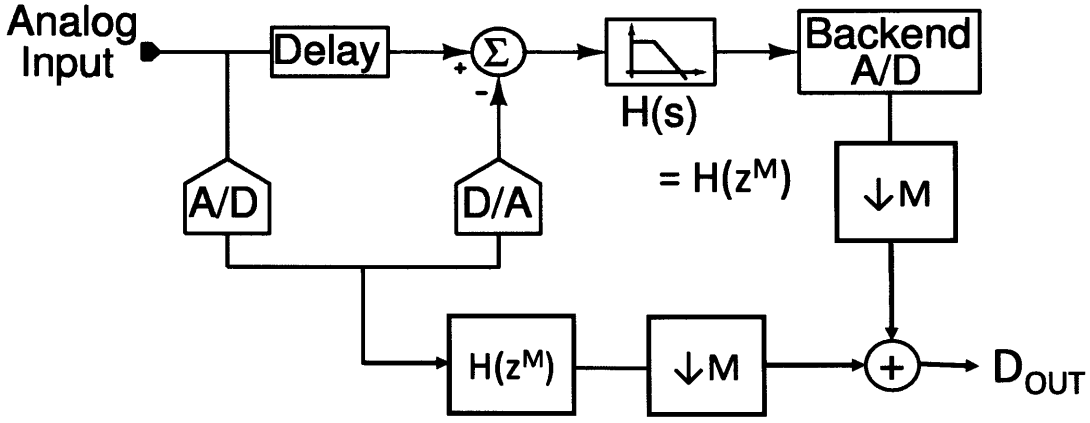


Figure 4-4: The digital reconstruction filter becomes an exact replica of the *analog* low-pass filter operating at F_s.

Alternatively, instead of applying H_{DRF} to the stage 2 digital output D_2, we can apply H_{LPF} to the stage 1 output D_1 at the rate F_{CLK}. Again, by using the

decimation identity, the decimation and filtering can be interchanged. Therefore, the *continuous-time* sub-ADC output V_Q (at rate F_s) can be passed through the same low-pass filter (modelled as a digital filter with cut-off frequency ω_c) as the residue as shown in Fig. 4-4. This is equivalent to having an inverse filter at stage 2.

4.4 Simulation Results

The simulation results based on the MATLAB behavioural model have been presented in this section. A sinusoid signal at 800 MHz is applied at the input of our continuous-time pipeline ADC. The sampling clock frequency is 6.4 GHz. The resolution of the stage 1 sub-ADC is 4 bits. The analog low-pass filter has been modelled as a 4^{th} order elliptic filter with 0.1 dB passband ripple and 40 dB stopband attenuation. The interstage gain is 8 and the stage 2 has 8-bit resolution, therefore the effective resolution of the ADC is 11 bits. A 65536-point Hann-windowed FFT of the stage 1 digital output, stage 2 digital output and the reconstructed digital output is shown in Fig. 4-5. With 1 ps RMS jitter in the stage 2 clock, we get an SNR of 62.2 dB (with 200 fs DAC jitter). In the case of upfront sampling, the SNR is limited to 46 dB for an 800 MHz input signal sampled with a clock having 1 ps RMS jitter (from Eq. 1.2). This shows that we get an improvement of about 16 dB in the maximum achievable SNR.

To demonstrate the jitter tolerance of this architecture, we calculate the SNR of the final reconstructed digital output for various RMS values of stage 2 clock jitter and compare it with the SNR from the conventional pipeline ADC architecture with upfront sampling. Fig. 4-6 shows the effect of stage 2 jitter on the overall SNR (in absence of sub-ADC and DAC jitter).

For small values of RMS jitter, the error due to clock jitter is smaller than the quantization error. Therefore, the overall SNR is limited by the quantization noise. As the RMS jitter increases, the overall SNR starts decreasing, but still, it is much higher compared to those of conventional pipeline ADCs.

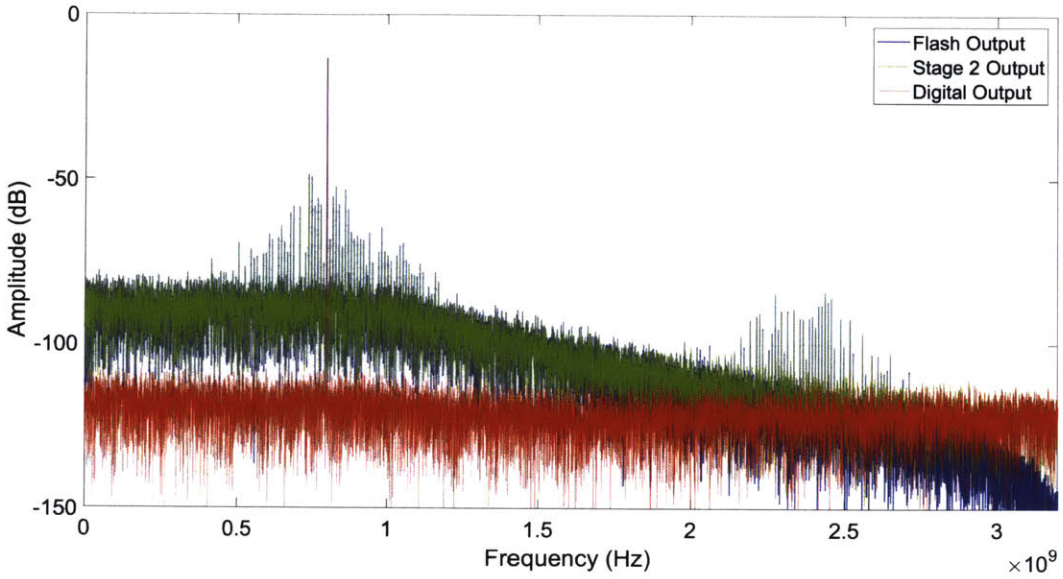


Figure 4-5: Spectrum of the stage 1 output, stage 2 output and the final reconstructed output.

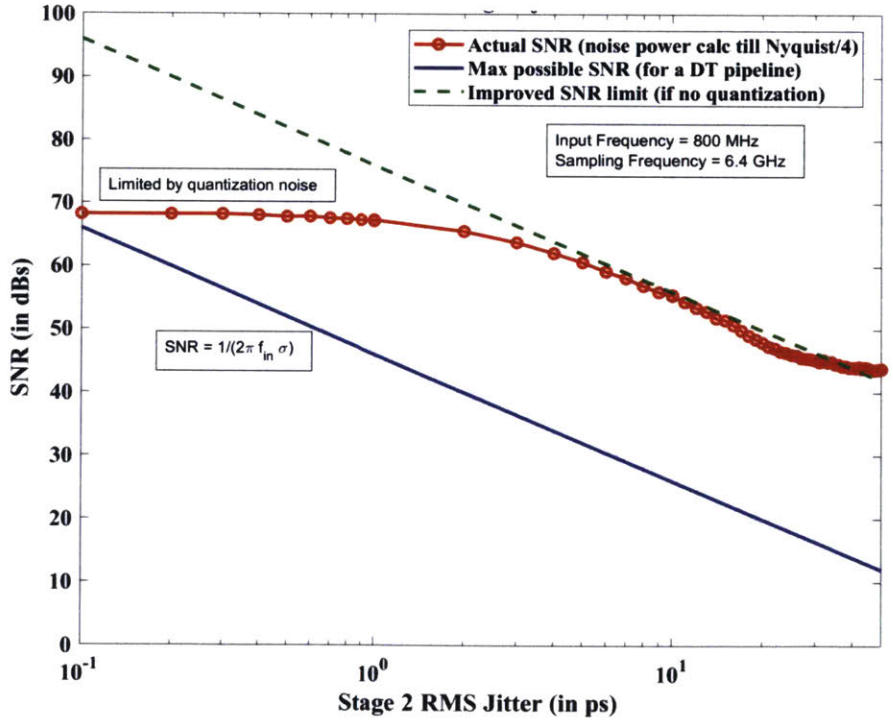


Figure 4-6: Variation of SNR with the stage 2 clock jitter.

Chapter 5

Conclusion

5.1 Thesis contribution

In this thesis, we have presented a novel continuous-time pipeline ADC architecture that is tolerant to the sampling jitter. In conventional pipeline ADCs, the analog input is sampled upfront which introduces a random error due to uncertainty in the sampling clock edge. This is akin to adding a random noise voltage to the signal thereby limiting its SNR. In the proposed ADC, we remove the sample-and-hold from the first stage, process the analog input signal in continuous-time and sample the voltage residue at the input of the backend ADC. The residue that goes to the backend ADC has a small magnitude and therefore a smaller derivative since it is a bandlimited signal. Therefore, the effect of the clock jitter will be greatly reduced.

We designed a system level behavioral model in MATLAB to verify the concept of continuous-time pipeline ADCs in general, and jitter tolerance in particular. The design of various sub-blocks like the sub-ADC, DAC, low-pass filter, and digital reconstruction filter have been discussed in detail. Finally, from the system-level behavioral simulations, we have shown an increase of at least 16 dB in the maximum achievable SNR if a jitter-tolerant DAC is used in the first stage Flash-DAC path.

5.2 Future Work

One can imagine converting the problem of amplitude quantization to time quantization by using non-uniformly sampled (NUS) A/D converters [29–34]. Technology scaling also favors this since with smaller devices, the time accuracy generally improves. Therefore, we are exploring the possibility of using this idea of NUS ADCs in a continuous-time pipeline ADC to get even better tolerance to the sampling clock jitter. Also, we are working towards estimating the energy efficiency gains by comparing the power consumption of our topology with that of the case where a more precise clock with lower jitter is used.

Bibliography

[1] B. Murmann. The Race for the Extra Decibel: A Brief Review of Current ADC Performance Trajectories. *IEEE Solid-State Circuits Magazine*, 7(3):58–66, 2015.

[2] Mitsuru Shinagawa, Yukio Akazawa, and Tsutomu Wakimoto. Jitter analysis of high-speed sampling systems. *IEEE Journal of Solid-State Circuits*, 25(1):220–224, 1990.

[3] Robert H Walden. Analog-to-digital converter survey and analysis. *IEEE Journal on selected areas in communications*, 17(4):539–550, 1999.

[4] Nicola Da Dalt, Moritz Harteneck, Christoph Sandner, and Andreas Wiesbauer. On the jitter requirements of the sampling clock for analog-to-digital converters. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(9):1354–1360, 2002.

[5] Boris Murmann et al. Adc performance survey 1997-2019. *Online] http://www. stanford. edu/murmann/adcsurvey. html*, 2019.

[6] Stephen H Lewis, H Scott Fetterman, George F Gross, R Ramachandran, and TR Viswanathan. A 10-b 20-Msample/s analog-to-digital converter. *IEEE Journal of Solid-State Circuits*, 27(3):351–358, 1992.

[7] Andrew N Karanicolas, Hae-Seung Lee, and KL Barcrania. A 15-b 1-Msample/s digitally self-calibrated pipeline ADC. *IEEE Journal of Solid-State Circuits*, 28(12):1207–1215, 1993.

[8] Thomas Byunghak Cho and Paul R Gray. A 10 b, 20 Msample/s, 35 mW pipeline A/D converter. *IEEE journal of solid-state circuits*, 30(3):166–172, 1995.

[9] Paul C Yu and Hae-Seung Lee. A 2.5-v, 12-b, 5-msample/s pipelined cmos adc. *IEEE Journal of Solid-state circuits*, 31(12):1854–1861, 1996.

[10] Boris Murmann and Bernhard E Boser. A 12-bit 75-MS/s pipelined ADC using open-loop residue amplification. *IEEE Journal of Solid-State Circuits*, 38(12):2040–2050, 2003.

[11] David A Johns and Ken Martin. *Analog integrated circuit design*. John Wiley & Sons, 2008.

[12] Ashutosh Verma and Behzad Razavi. A 10-Bit 500-MS/s 55-mW CMOS ADC. *IEEE Journal of Solid-State Circuits*, 44(11):3039–3050, 2009.

[13] Bengt E Jonsson. A/D-converter performance evolution. *Converter Passion*, 2012.

[14] Robert van Veldhoven. A tri-mode continuous-time/spl sigma//spl delta/modulator with switched-capacitor feedback dac for a gsm-edge/cdma2000/umts receiver. In *2003 IEEE International Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC.*, pages 60–477. IEEE, 2003.

[15] M Ortmanns, F Gerfers, and Y Manoli. Jitter insensitive feedback dac for continuous-time ea modulators. In *International Conference on Electronics, Circuits and Systems*, 2001.

[16] Maurits Ortmanns, Yiannos Manoli, and Friedel Gerfers. A continuous-time sigma-delta modulator with reduced jitter sensitivity. In *Proceedings of the 28th European Solid-State Circuits Conference*, pages 287–290. IEEE, 2002.

[17] Richard Schreier, Gabor C Temes, et al. *Understanding delta-sigma data converters*, volume 74. IEEE press Piscataway, NJ, 2005.

[18] D. Gubbins, B. Lee, P. K. Hanumolu, and U. Moon. Continuous-Time Input Pipeline ADCs. *IEEE Journal of Solid-State Circuits*, 45(8):1456–1468, August 2010.

[19] H. Shibata, V. Kozlov, Z. Ji, A. Ganesan, H. Zhu, D. Paterson, J. Zhao, S. Patil, and S. Pavan. A 9-GS/s 1.125-GHz BW Oversampling Continuous-Time Pipeline ADC Achieving 164-dBFS/Hz NSD. *IEEE Journal of Solid-State Circuits*, 52(12):3219–3234, December 2017.

[20] Marcel JM Pelgrom. Analog-to-digital conversion. In *Analog-to-Digital Conversion*, pages 325–418. Springer, 2013.

[21] Mikael Gustavsson, J Jacob Wikner, and Nianxiong Tan. *CMOS data converters for communications*, volume 543. Springer Science & Business Media, 2000.

[22] Behzad Razavi. *Principles of data conversion system design*, volume 126. IEEE press New York, 1995.

[23] Blanchard D Smith. An unusual electronic analog-digital conversion method. *IRE Transactions on Instrumentation*, pages 155–160, 1956.

[24] Davide Adami, Christian Callegari, Stefano Giordano, Michele Pagano, and Teresa Pepe. A real-time algorithm for skype traffic detection and classification. In *Smart Spaces and Next Generation Wired/Wireless Networking*, pages 168–179. Springer, 2009.

[25] Donghyun Kim, Tatsuji Matsuura, and Boris Murmann. A continuous-time, jitter insensitive σ δ modulator using a digitally linearized g m-c integrator with embedded sc feedback dac. In *2011 Symposium on VLSI Circuits-Digest of Technical Papers*, pages 38–39. IEEE, 2011.

[26] R. Adams, K. Nguyen, and K. Sweetland. A 113 dB SNR oversampling DAC with segmented noise-shaped scrambling. In *1998 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, ISSCC. First Edition (Cat. No.98CH36156)*, pages 62–63, February 1998.

[27] Susan Luschas and Hae-Seung Lee. High-speed/spl sigma//spl delta/modulators with reduced timing jitter sensitivity. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 49(11):712–720, 2002.

[28] Alan V Oppenheim. *Discrete-time signal processing.* Pearson Education India, 1999.

[29] T. Wu, S. Dey, and M. S. Chen. A Nonuniform Sampling ADC Architecture With Reconfigurable Digital Anti-Aliasing Filter. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(10):1639–1651, October 2016.

[30] T. Wu, C. Ho, and M. S. Chen. A Flash-Based Non-Uniform Sampling ADC With Hybrid Quantization Enabling Digital Anti-Aliasing Filter. *IEEE Journal of Solid-State Circuits*, 52(9):2335–2349, September 2017.

[31] T. Wu and M. S. Chen. A Subranging-Based Nonuniform Sampling ADC With Sampling Event Filtering. *IEEE Solid-State Circuits Letters*, 1(4):78–81, April 2018.

[32] M. Trakimas, R. D'Angelo, S. Aeron, T. Hancock, and S. Sonkusale. A Compressed Sensing Analog-to-Information Converter With Edge-Triggered SAR ADC Core. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(5):1135–1148, May 2013.

[33] M. Trakimas and S. R. Sonkusale. An Adaptive Resolution Asynchronous ADC Architecture for Data Compression in Energy Constrained Sensing Applications. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(5):921–934, May 2011.

[34] B. Schell and Y. Tsividis. A Continuous-Time ADC/DSP/DAC System With No Clock and With Activity-Dependent Power Dissipation. *IEEE Journal of Solid-State Circuits*, 43(11):2472–2481, November 2008.