國立交通大學

電子工程學系電子研究所

博士論文

應用於類比數位轉換器之
時脈抖動量測與補償技術

Clock Jitter Measurement and Compensation for Analog-to-Digital Converters

研究生 ： 范啟威
指導教授 ： 吳介琮

中華民國九十九年九月

國立交通大學

電子工程學系電子研究所

博士論文

應用於類比數位轉換器之
時脈抖動量測與補償技術

Clock Jitter Measurement and Compensation for Analog-to-Digital Converters

研究生 ： 范啟威
指導教授 ： 吳介琮

中華民國九十九年九月

應用於類比數位轉換器之
時脈抖動量測與補償技術

Clock Jitter Measurement and Compensation for Analog-to-Digital Converters

研究生 ： 范啟威　　Student ： Chi-Wei Fan
指導教授 ： 吳介琮　　Advisor ： Jieh-Tsorng Wu

國立交通大學

電機學院　　電子工程學系　　電子研究所

博士論文

A Dissertation
Submitted to Department of Electronics Engineering
and Institute of Electronics
College of Electrical and Computer Engineering
National Chiao-Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Electronics Engineering
March 2010
Hsin-Chu, Taiwan, Republic of China

中華民國九十九年九月

應用於類比數位轉換器之
時脈抖動量測與補償技術

學生：范啟威　　　　指導教授：吳介琮

國立交通大學

電機學院　　電子工程學系　　電子研究所

摘要

在一個新式的通訊接收器中，接收到的連續類比訊號會先被一個類比數位轉換器量化成不連續的數位序列，以便在數位領域進行更複雜的訊號處理。類比數位轉換器需要一個週期性的時脈訊號做為基準以對輸入訊號進行取樣。如果取樣時脈發生抖動，類比數位轉換器會因為取樣的錯誤而造成輸出之訊號雜訊比性能下降。對於一個低速度低解析度的類比數位轉換器而言，取樣時脈抖動所造成的誤差並不是很重要。隨著通訊系統的進步，類比數位轉換器的操作速度以及解析度也隨之增加。對於一高速度高解析度的類比數位轉換器而言，準確的取樣時脈訊號是不可或缺的。

時脈訊號中之抖動可藉由一時間數位轉換器量測出來並量化成數位訊號。藉由適當的校正技術可將此數位訊號對應成時脈抖動之資訊。量測出來的抖動資訊，用在數位領域補償類比數位轉換器因取樣錯誤造成的誤差，改善輸出之訊號雜訊比。本篇論文描述一個應用於類比數位轉換器之時脈抖動量測及補償技術。

我們實現了一個65奈米互補式金氧半場效電晶體製成的7-bit、80-MS/s之時間數位轉換器。利用時間數位轉換器量測出類比數位轉換器取樣時脈之抖動。我們也提出了新的時間數位轉換器之數位校正技術。此校正技術可在背景執行，因此不會影響類比數位轉換器以及時間數位轉換器之正常工作。我們提出的技術也不會受到元件或繞線不匹配的影響，也不會對取樣時脈信號的波型敏感。我們實現之7-bit時間數位轉換器解析度為0.27 ps。此時間數位轉換器佔據0.1mm^{2}的晶片面積，在電源供應器為1.2 V下，消耗的功率為20 mW。

將此時間數位轉換器應用於一16-bit類比數位轉換器之取樣時脈抖動量測與補償上，我們提出了兩種方案: 1) 類比數位轉換器之取樣時脈為理想的，以及2:) 類比數位轉換器之取樣時脈訊號為主要的抖動來源。在第一種方案中，對於一個最佳化設計之鎖控延遲迴路，在類比數位轉換器之輸入為29 MHz正弦波下，16-bit的類比數位轉換器可將訊號雜訊比由71.2 dB改善為77.3 dB。對於一個設計不良的鎖控延遲迴路則可將訊號雜訊比由60.8 dB改善為74.4 dB。在第二種方案中，我們提出的時脈抖動補償技術可在輸入時脈抖動之均分根值為8.2 ps的情況下，達到等效於均方根值為4 ps抖動之取樣效果。

Clock Jitter Measurement and Compensation for Analog-to-Digital Converters

Student : Chi-Wei Fan Advisor : Jieh-Tsorng Wu

Department of Electronics Engineering

and Institute of Electronics

National Chiao-Tung University

Abstract

In a modern communication receiver, the received continuous-time analog signal is quantized into a discrete-time digital sequence by an analog-to-digital converter (ADC) so that the complex signal processing can be performed in the digital domain. The ADC requires a periodic clock as a timing reference for input sampling. If the sampling clock exhibits jitter, the ADC suffers from sampling errors and its signal-to-noise ratio (SNR) performance is degraded. For a low-speed low-resolution ADC, the sampling error due to clock jitter is not crucial. As the progress of advanced communication system, the operation speed and the resolution of the ADC are also increased. An accurate sampling clock is essential for a high-speed high-resolution ADC.

Clock jitter can be measured and digitized by a time-to-digital converter (TDC). With appropriate calibration technique, the output code of the TDC can be translated in to the corresponding jitter information. This jitter information is then used to compensate the

ADC ’ s sampling error in the digital domain, improving the ADC ’ s SNR performance. This thesis presents a clock jitter measurement and compensation scheme for analog-to-digital converters.

A 7-bit 80-MS/s TDC was fabricated using a 65 nm CMOS technology. The clock jitter of an ADC is measured by the TDC. We also demonstrate a new digital calibration technique for the TDC. The calibration can be performed in the background without interrupting the normal ADC and TDC operation. The proposed technique is immune to device and interconnection mismatches, and is not sensitive to the waveforms of the input clocks either. The resolution of the 7-bit TDC is 0.27 ps. The TDC occupies a die area of 0.1mm^2 while consuming 20 mW from a 1.2 V supply.

The TDC is applied to a 16-bit ADC for the clock jitter measurement and compensation. Two different system scenarios are covered: 1) an ADC with a clean external clock and 2) an ADC with an external clock as the main jitter source. For the first scenario, the SNR of the 16-bit ADC is improved from 71.2 dB to 77.3 dB for an optimized delay-locked loop (DLL) and 60.8 dB to 74.4 dB for an ill-conditioned DLL by the jitter correction at a sine wave input frequency of 29 MHz. For the second scenario, the proposed jitter correction technique achieves an equivalent sampling jitter root-mean-squared value (rms) of 4 ps when the jitter rms of the original sampling clock is 8.2 ps.

誌謝

首先我要對我的指導教授教吳介琮教授致上最誠摯的謝意與敬意。感謝他在我博士班期間不厭其煩地給我指導與協助，以及研究理念上的薰陶。同時也在老師身上學習到了做研究的態度和處理問題的方法。這些影響都使我受益良多並且永生難忘。

感謝實驗室的學長姊、同學以及學弟妹，他們在這幾年間給予我許多的指導與關照。在類比積體電路與系統實驗室的這幾年中，承蒙許多學長與同學在研究過程中適時的給予幫助，在此致上萬分的謝意。另外我還要感謝台積電邏輯設計發展處給予的協助，提供我製作晶片的機會，讓我得以將我的構想化為實際的成品。

此外，我還要特別感謝我的妻子雨潔、我的兒子世希、女兒芷甯以及我的家人，因為你們的支持與鼓勵，使的我有動力完成這份研究，在此衷心地感謝他們。最後僅將我的論文獻給我最摯愛的父母。

范啟威

國立交通大學
中華民國九十九年九月

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

In a modern communication receiver, an analog-to-digital converter (ADC) first samples the received continuous-time analog signal and then quantizes the sampled data into a discrete-time digital sequence, so that the complex signal processing can be performed in the digital domain. As shown in Figure 1.1, an input signal first passes through an analog signal processing block, the block is usually a low-pass filter or a band-pass filter to limit the bandwidth of the input signal. Then a programmable-gain amplifier (PGA) is placed in front of the ADC, adapting the loss of transmission to relax the dynamic range requirement of the ADC. The gain of the PGA is digitally controlled by an automatic gain control (AGC) loop. The ADC quantizes the amplified analog signal into a digital sequence. Finally, the required signal processing can be performed by the digital signal

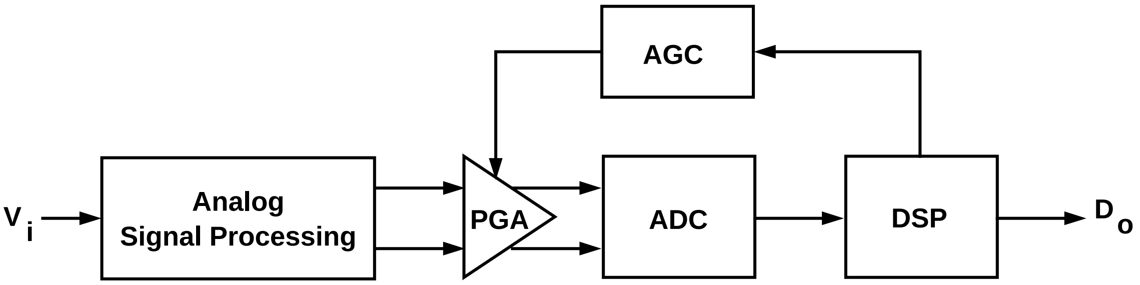


Figure 1.1: Block diagram of an analog front-end.

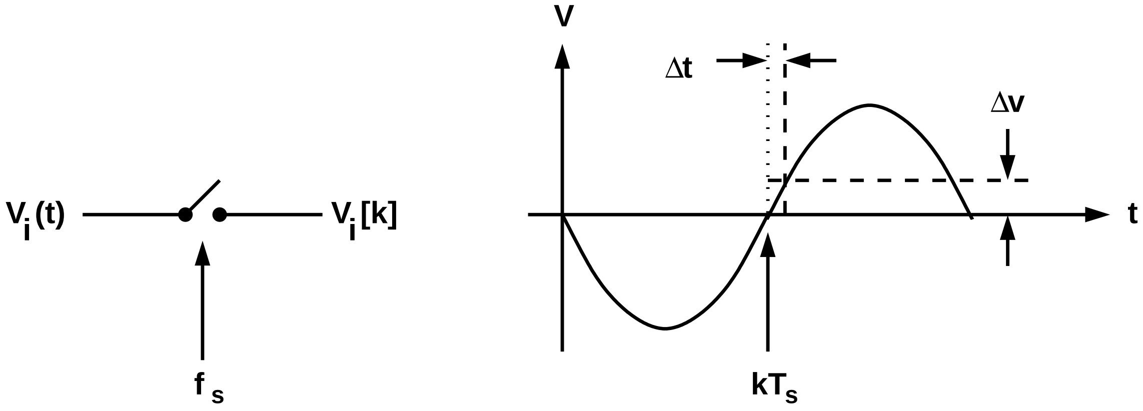


Figure 1.2: Sampling-time uncertainty (aperture jitter).

processing (DSP) block to generate the output signal.

There are many ways to qualify an ADC's performance such as spurious-free dynamic range (SFDR) and signal-to-noise ratio (SNR). The SFDR is usually dominated by the non-linearity of the input to output transfer curve and is solved by many calibration techniques. These techniques include linearizing the transfer curve of the multiplying digital-to-analog converter (MDAC) in pipelined ADCs [1, 2, 3, 4, 5, 6, 7, 8], trimming the input-referred offsets of the comparators in flash ADCs or two-step ADCs [9, 10, 11, 12, 13]. To further improve the conversion linearity, there are schemes which also calibrate the non-linearity of the amplifier [2, 5, 6, 8].

The contributions to SNR of these techniques are restricted since the SNR is usually dominated by the environment noise and the quality of the sampling clock. The former is usually solved by enlarge the signal swing to increase the signal power, or use larger sampling capacitors to decrease the thermal noise. The latter problem is usually ignored with the assumption that a pure sinusoid signal source is available and the noise introduced by the clock buffers is neglectable. The most commonly used method for a clock reference is a crystal oscillator followed by a narrow-band high order band-pass filter. However, as the conversion rate of the ADC is increased, a phase-locked-loop (PLL) based clock generator is inevitable which results in a noisier clock source.

A periodic clock is required to provide a reference for the sampling time. If the sampling clock jitters, sampling error occurs during the sampling process [14, 15]. Excess clock jitter can degrade the SNR performance of an ADC. Figure 1.2 shows the effect of clock jitter. The $V_i(t)$ signal is sampled by the ADC every T_s, where $T_s = 1/f_s$ and f_s is the sampling rate of the ADC system. Therefore, the k-th sampling time is kT_s ideally. If the sampling clock jitters, the signal is sampled at $t = kT_s + \Delta t$ instead of $t = kT_s$. Thus the sampled data is ΔV deviated from an ideal sample, which degrades the SNR performance.

If the clock jitter dominates the SNR of an ADC, the SNR can be approximated as [14]

$$\mathrm{SNR} = \frac{1}{\omega_i^2 \times (\Delta t_{rms})^2} \tag{1.1}$$

where Δt_{rms} is the rms of Δt. If the quantization error dominates the SNR of an B-bit ADC, the SNR can be approximated as [16]

$$\mathrm{SNR} = \frac{1}{(2/3)2^{-2B}} \tag{1.2}$$

Figure 1.3 shows the SNR for an ADC limited by aperture jitter for various jitter values (the sloped solid lines) and the quantization noise limited performance at various resolutions (the horizontal dashed lines). For an ADC with 120 MS/s sampling rate, if the analog input is a 60 MHz sine wave and the clock jitter is random, the root-mean-squared value (rms) of the clock jitter must be less than 0.5 ps to ensure a 12-bit resolution. If the rms value of the clock jitter deteriorates to 2 ps, the resolution of the ADC degrades to 10-bit. As the conversion rate or the resolution of the ADC increase, the jitter requirement also increased.

Clocks generated from PLL can hardly achieve this stringent jitter requirement [17]. Low-jitter clocks, mostly based on crystal oscillators, are inflexible and expensive. There are many researches of low noise PLL design. Such as optimizes the loop bandwidth of a PLL to compromise the noise of different frequency response [18], or lower the sensitivity to supply noise for a PLL [19] or for a voltage-controlled oscillator (VCO) [20, 21, 22, 23]. There are also investigations strive to overcome the difficulties of low noise VCOs designs since the VCO usually dominates the jitter performance of a PLL [24, 25, 26]. Although these methods reduce the jitter of a PLL dramatically, it is still hard to meet the specification for a high speed high resolution ADC.

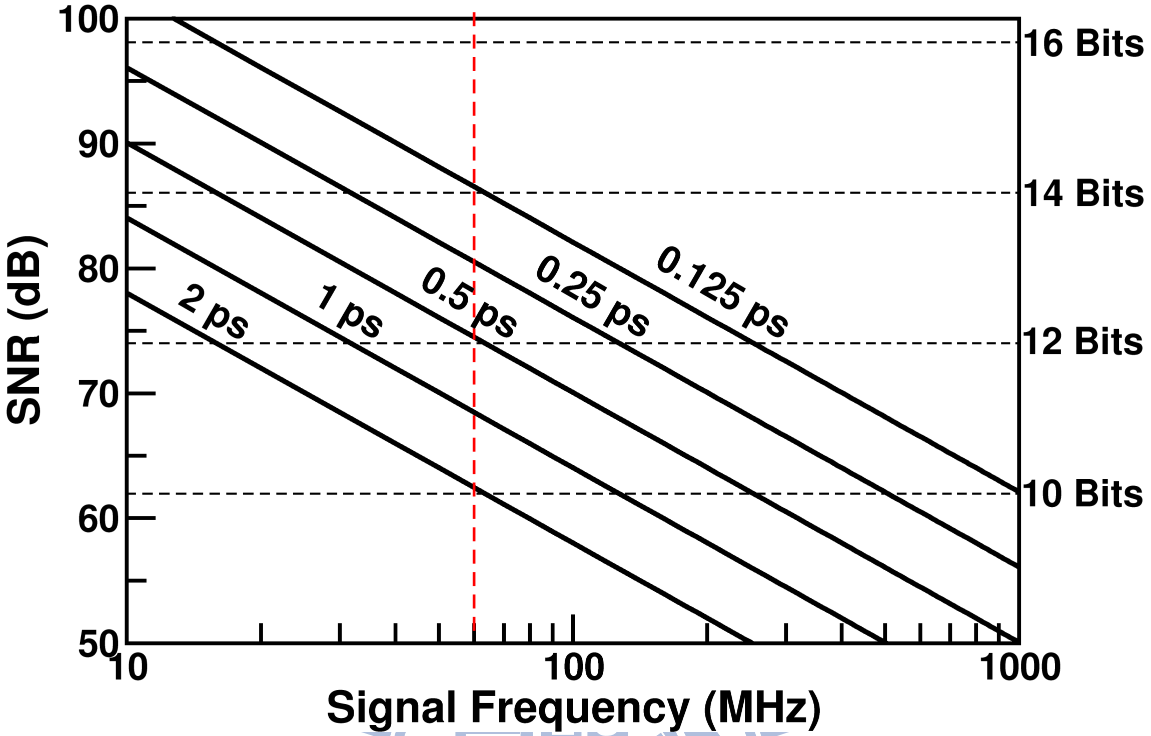


Figure 1.3: Signal-to-noise ratio due to aperture jitter.

The importance of having a low jitter clock source for an ADC is well-known. While most of the researches try to generate a low jitter clock source, it is possible to compensate the sampling error of an ADC thus relax the clock jitter requirement. To the best of the author's knowledge, there is only one other published work that tries to compensate the sampling error caused by clock jitter [27]. As shown in Figure 1.4, the scheme of Tourabaly and Osseiran modulates the analog input before the sampler so that the correct input signal is sampled at incorrect sampling time.

Although the simulation result in [27] shows that a first order approximation of jitter correction is sufficient in respect to produce an SNR improvement of about 15dB, this scheme is hard to implement for several reasons.

1. This scheme requires high-precision analog circuits which are difficult to implement. These circuits include analog multiplier, analog adder, and differentiator.

2. The use of differentiators makes it sensitive to high-frequency noises.

3. Analog calibrations are required for phase mismatches and gain error.

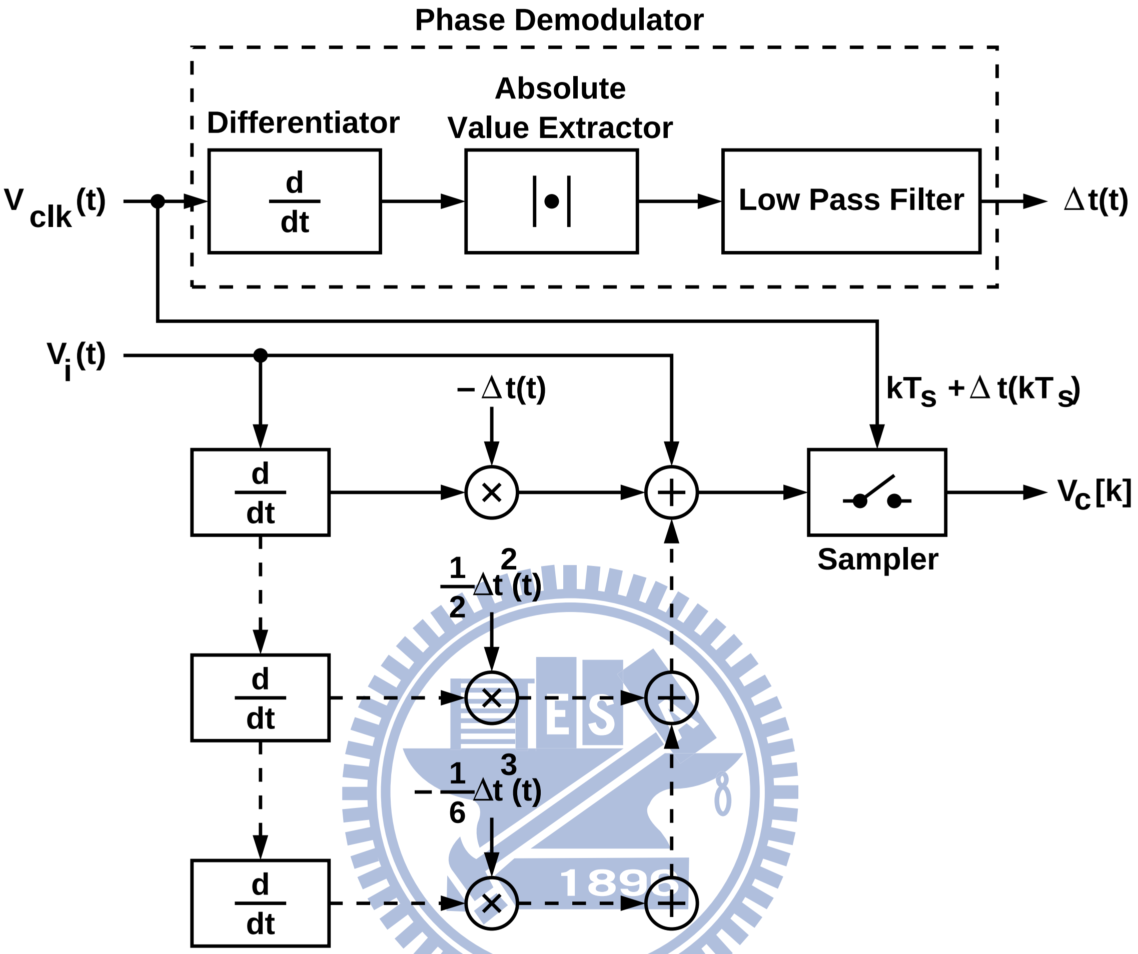


Figure 1.4: Jitter compensation scheme proposed by Tourabaly and Osseiran.

(a) Phase mismatch: There are mismatches of the phase delay between the phase demodulator, the jitter compensation circuit and the sampler. Any difference of the phase delay between them degrades the performance dramatically.

(b) Gain error: The gain of the phase demodulator and the gain of the mixer in the jitter compensation must equal to one to extract the jitter correctly.

For these reasons, only simulation results are given in [27], no experimental result is provided. Moreover, the simulation results show that the improvement on SNR is only 0.0015 dB if the phase delay and the attenuation caused by the differentiator and the mixer are not calibrated.

In this thesis, we propose a technique that employs digital signal processing to relax the clock jitter requirement. Clock jitter is measured and digitized by a stochastic time-to-digital converter (TDC) [28]. This jitter information is then used to compensate the ADC's sampling error in the digital domain, improving the ADC's SNR performance. We also propose techniques for TDC calibration. The calibration can be performed in the background without interrupting the normal ADC operation. Theoretical analyses, system simulations and silicon proved measurement results are provided to verify the proposed jitter compensation and TDC calibration techniques. A 16-bit 80 MS/s ADC system is discussed as a design example.

1.2 Organization

The organization of the thesis is described as follows:

Chapter 2 discusses the effect of sampling jitter on a signal. The sampling uncertainty caused by clock jitter makes the sampled signals deviate from their nominal values and the SNR performance is degraded. To improve the SNR when the clock is noisy, a novel jitter compensation technique is proposed. Theoretical analyses and system simulations are provided to verify the technique.

In order to measure the clock jitter, a TDC is required to digitize the timing difference between two clocks of identical frequency. Various TDCs are introduced in Chapter 3, such as the counter-based TDC, the time-to-amplitude TDC, the tapped delay line TDC and the stochastic TDC. The advantages of each kind of TDC and their limitations are also discussed.

Chapter 4 describes a jitter compensation scenario in which the external clock is clean. To estimate the clock jitter accurately, a background calibration technique for TDC is proposed to against the process, supply voltage, and temperature (PVT) variations. The proposed TDC background calibration is based on signal reconstruction. Theoretical analyses and simulation results are also provided. A 7-bit 80 MS/s TDC is fabricated in 65-nm CMOS technology to verify the jitter compensation and TDC background calibration configuration.

Chapter 5 describes an alternative ADC system in which the external clock is the main jitter source. This scheme can achieve high SNR performance even when a clean clock source is not available, thus mitigates the jitter requirement for the input clock. The jitter compensation can break the SNR limitation predicted by Equation (1.1).

Finally, conclusions and recommendations for future works will be given in Chapter 6.

Chapter 2

Jitter Compensation

2.1 Introduction

A Nyquist-rate analog-to-digital converter (ADC) periodically samples its continuous-time analog input, and converts it into a discrete-time digital data stream. The ADC requires a periodic clock as a timing reference for input sampling. If the sampling clock exhibits jitter, the ADC suffers from sampling errors, and its signal-to-noise ratio (SNR) performance is degraded [14, 15].

It is possible to relax the clock jitter requirement by introducing jitter compensation in the analog-to-digital signal path. The scheme of Tourabaly and Osseiran [27] modulates the analog input before the sampler so that the correct input signal is sampled. As mentioned in Chapter 1 this scheme requires high-precision analog circuits, which are difficult to implement. The use of differentiators also makes it sensitive to high-frequency noises.

In this chapter, we introduced a digital signal processing technique to relax the clock jitter requirement. Both intuitive interpretations and theoretical analyses will be given.

2.2 SNR of an ADC

As described in Chapter 1, the SNR performance of an ADC is usually dominated by the environment noise and the quality of the sampling clock. The SNR of an ADC can be measured by applying the following sine wave input:

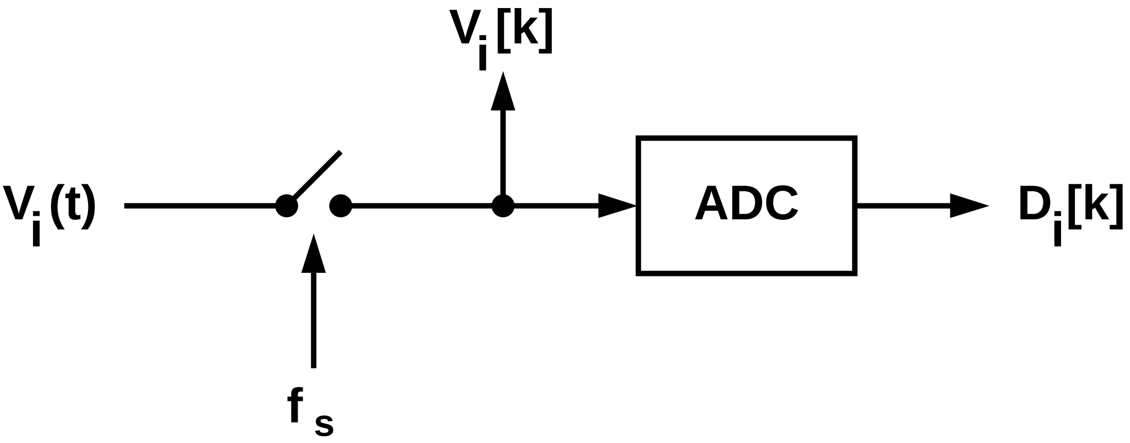


Figure 2.1: Continuous-to-Discrete Conversion of an ADC.

$$V_i(t) = A_i \times \sin(\omega_i t + \phi_i) \tag{2.1}$$

where A_i is the amplitude of the input sine wave signal, ω_i is the input frequency, and ϕ_i is the phase. The $V_i(t)$ signal is sampled at a sampling rate of f_s as shown in Figure 2.1, i.e., T_s is the sampling interval and $T_s = 1/f_s$. Thus, the k-th nominal sampled voltage is

$$V_i[k] = V_i(kT_s) = A_i \times \sin(k\omega_i T_s + \phi_i) \tag{2.2}$$

However, if the sampling clock jitters, the ADC suffers from sampling errors and Equation (2.2) becomes

$$V_i[k] = V_i(kT_s + \Delta t[k]) = A_i \times \sin(\Omega_i(k + \epsilon[k]) + \phi_i) \tag{2.3}$$

where $\Omega_i = \omega_i T_s$ is the normalized input frequency, $\Delta t[k]$ is k-th jitter of the sampling clock and $\epsilon[k] = \Delta t[k]/T_s$ is the normalized clock jitter. The corresponding output from the ADC at the k-th sampling can be expressed as

$$D_i[k] = V_i(kT_s) + q[k] + V_e[k] \tag{2.4}$$

The $D_i[k]$ signal consists of the following: 1) the desired input $V_i(kT_s)$; 2) the quantization noise $q[k]$; and 3) the sampling error $V_e[k]$ caused by the clock jitter. The SNR of the $D_i[k]$ signal is defined as

$$\mathrm{SNR} \equiv \frac{P_s}{P_q + P_e} \tag{2.5}$$

where $P_s = (1/2)A_i^2$ is the power of the $V_i(t)$ input signal, P_q is the signal power of the $q[k]$ sequence, and P_e is the signal power of the $V_e[k]$ sequence. Here we have ignored the thermal noise power and other environment noise for simplicity. The effect of these noises can be included in Equation (2.5) easily as

$$\mathrm{SNR} \equiv \frac{P_s}{P_q + P_e + P_t} \tag{2.6}$$

where P_t is the summation of thermal noise power and other environment noise.

Considering an ideal B-bit ADC with an input range of ±1, it has a uniform quantization step size of $s_q = 2/2^B$. The quantization noise power of an ideal quantization process can be approximated by [16]

$$P_q \equiv \overline{(q[k])^2} = \frac{1}{12} \times s_q^2 = \frac{1}{12} \times \left(\frac{2}{2^B}\right)^2 \tag{2.7}$$

The quantization noise $q[k]$ is assumed to be random and uniformly spreads between $\pm 1/2^B$.

If the clock jitter is small enough compared to T_s, the k-th sampling error $V_e[k]$ can be approximated by [14]

$$V_e[k] \approx \left.\frac{dV_i(t)}{dt}\right|_{t=kT_s} \times \Delta t[k] = A_i\omega_i \cos(\Omega_i k + \phi_i) \times \Delta t[k] \tag{2.8}$$

where $\Delta t[k] = \epsilon[k]T_s$ is the clock jitter. Therefore, the sampling error power can be expressed as

$$P_e \equiv \overline{(V_e[k])^2} = \frac{1}{2}A_i^2\omega_i^2 \times (\Delta t_{rms})^2 = \frac{1}{2}A_i^2\Omega_i^2 \times \epsilon_{rms}^2 \tag{2.9}$$

where Δt_{rms} is the root-mean-squared value (rms) of $\Delta t[k]$, and ϵ_{rms} is the rms of $\epsilon[k]$. The clock jitter is assumed to be random and has a mean of zero. Assume that the input is a full-range sine wave expressed as Equation (2.1) with $A_i = 1$. From Equation (2.5), Equation (2.7), and Equation (2.9), the SNR becomes

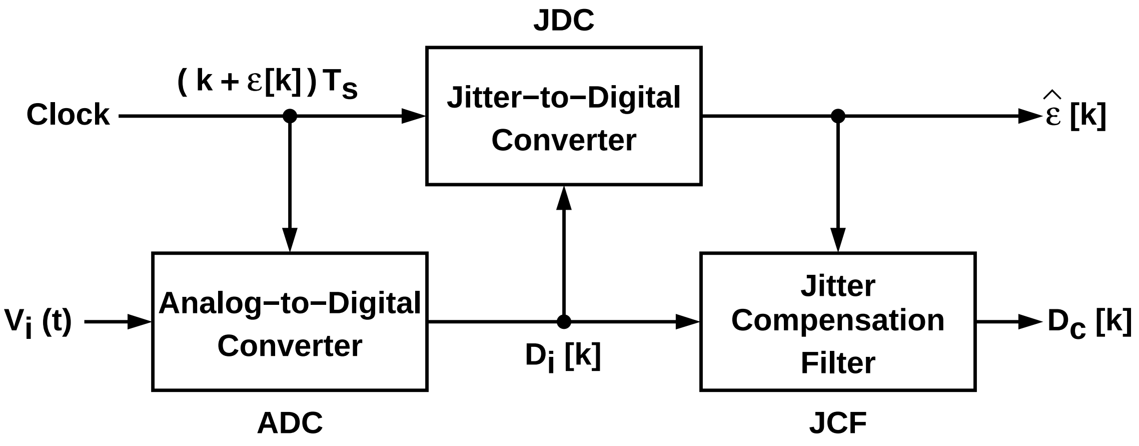

JDC
Clock
(k + ε[k]) T_S
Jitter−to−Digital Converter
ε̂ [k]
V_i (t)
Analog−to−Digital Converter
D_i [k]
Jitter Compensation Filter
D_C [k]
ADC
JCF

Figure 2.2: Jitter compensation block diagram.

$$\mathrm{SNR}_i = \frac{1}{(2/3)2^{-2B} + \omega_i^2(\Delta t_{rms})^2} = \frac{1}{(2/3)2^{-2B} + \Omega_i^2 \epsilon_{rms}^2} \tag{2.10}$$

Equation (2.10) is the maximum SNR performance of an ideal ADC without jitter compensation, which is also the SNR limitation of an ADC theoretically. In this thesis, we introduce a novel jitter compensation technique. The proposed jitter compensation scheme can improve the ADC SNR and break the performance limitation predicted by Equation (2.10).

2.3 Proposed Jitter Compensation Configuration

The basic principle of the proposed jitter compensation is illustrated in Figure 2.2. An ADC samples and quantizes the analog signal $V_i(t)$ and generates the corresponding digital sequence $D_i[k]$. A clock dictates the instants at which $V_i(t)$ is sampled. The k-th sampling time is $(k + \epsilon[k])T_s$ where T_s is the nominal sampling interval, and $\epsilon[k]$ is the clock jitter normalized to T_s. The clock jitter at the k-th sample is $\Delta t[k] = \epsilon[k]T_s$. A jitter-to-digital converter (JDC) measures the $\epsilon[k]$ jitter and produces a jitter estimation $\hat{\epsilon}[k]$ in digital form. The relationship between $\epsilon[k]$ and $\hat{\epsilon}[k]$ is defined as

$$\hat{\epsilon}[k] = \epsilon[k] + \epsilon_e[k] \tag{2.11}$$

where $\epsilon_e[k]$ is the JDC measurement error. A jitter compensation filter (JCF) uses the $\hat{\epsilon}[k]$ data to correct the sampling error in $D_i[k]$. The corrected output from the JCF is $D_c[k]$. The jitter error is compensated in the digital domain, thus no high-precision analog circuit is required and all the problems in [27] can be eliminated.

The theory of jitter compensation is discussed as follows. A band-limited signal $V_i(t)$ can be expressed in inverse Fourier transform as

$$V_i(t) = \frac{1}{2\pi}\int_{-\omega_B}^{+\omega_B} V(j\omega)e^{j\omega t}d\omega \tag{2.12}$$

where $V(j\omega)$ is the Fourier transform of $V_i(t)$, and ω_B is its bandwidth. Neglecting the quantization noise, the $D_i[k]$ signal in Figure 2.2 is simply the $V_i(t)$ input sampled at $t = (k + \epsilon[k])T_s$. It can be expressed as

$$D_i[k] = \frac{1}{2\pi}\int_{-\omega_B}^{+\omega_B} V(j\omega)e^{j\omega(kT_s+\epsilon[k]T_s)}d\omega \tag{2.13}$$

Assume that only the k-th sample $D_i[k]$ contains a sampling error. Then, the sampling error is caused by a jitter $\epsilon[k]$ at the k-th sampling. To correct this sampling error, the required JCF is a filter with a frequency-domain transfer function of $e^{-j\omega\epsilon[k]T_s}$. Therefore, the corrected output from JCF is

$$\begin{aligned} D_c[k] &= \frac{1}{2\pi}\int_{-\omega_B}^{+\omega_B} \left[V(j\omega) \times e^{-j\omega\epsilon[k]T_s}\right] e^{j\omega(kT_s+\epsilon[k]T_s)}d\omega \\ &= \frac{1}{2\pi}\int_{-\omega_B}^{+\omega_B} V(j\omega)e^{j\omega kT_s}d\omega \\ &= V_i(kT_s) \end{aligned} \tag{2.14}$$

which is a correct sample at $t = kT_s$.

Figure 2.3 gives an intuitive interpretation of this jitter compensation scheme. Assume that a sampling error occurs at the k-th sampling instant. The $V_i(t)$ is sampled at $t = (k + \epsilon[k])T_s$ instead of $t = kT_s$. The magnitude at point A' is quantized as $D_i[k]$. Thus, the ADC perceives a different $\hat{V}_i(t)$ input instead of $V_i(t)$. The signal $\hat{V}_i(t)$ has a value of $D_i[k]$ at $t = kT_s$, denoted as point B. The filter of Equation (2.15) interpolates the value of $\hat{V}_i(t)$ at $t = (k - \epsilon[k])T_s$, denoted as point B', which is a correct estimation of $V_i(t)$ at $t = kT_s$, denoted as point A.

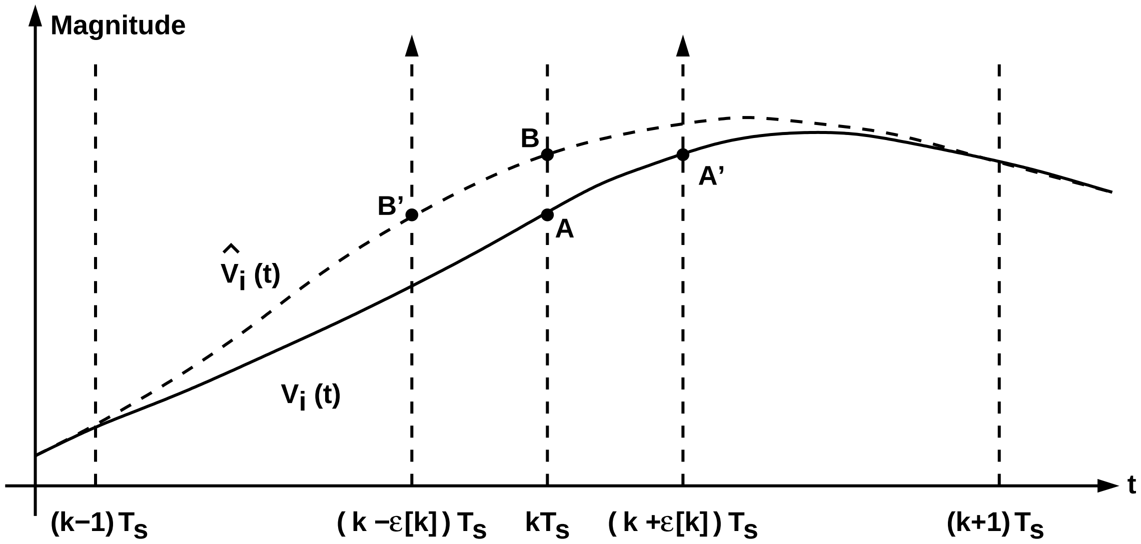


Figure 2.3: A graphic illustration of jitter compensation principle.

2.4 Jitter Compensation Filter

Since the JCF is a filter with a frequency-domain transfer function of $e^{-j\omega\epsilon[k]T_s}$, the discrete-time impulse response of the linear-phase filter can be obtained by using inverse Fourier transform, i.e.,

$$\begin{aligned} h_c\left[n, \epsilon[k]\right] &= \frac{1}{2\pi}\int_{-\pi}^{+\pi} e^{-j\Omega\epsilon[k]}e^{j\Omega n}d\Omega \\ &= \frac{\sin\left(\pi(n-\epsilon[k])\right)}{\pi(n-\epsilon[k])} = \mathrm{sinc}(n-\epsilon[k]) \end{aligned} \tag{2.15}$$

where $\Omega = \omega T_s$ is the normalized frequency.

Figure 2.4 shows a finite-impulse-response (FIR) filter with $2M + 1$ taps that approximates the JCF of Equation (2.15). Applying the measured $\hat{\epsilon}[k]$ data from the JDC and assuming $\hat{\epsilon}[k] \ll 1$, the filter's output can be expressed as

$$\begin{aligned} D_c[k] &= \sum_{n=k-M}^{k+M} D_i[n] \times h_c\left[k-n, \hat{\epsilon}[k]\right] \\ &\approx D_i[k] + \sum_{n=1}^{M}\left[\frac{1}{2\pi}\int_{-\omega_B}^{+\omega_B} V(j\omega)e^{j\omega kT_s}\frac{2j\sin(n\omega T_s)}{(-1)^n \times n}\hat{\epsilon}[k]d\omega\right] \end{aligned} \tag{2.16}$$

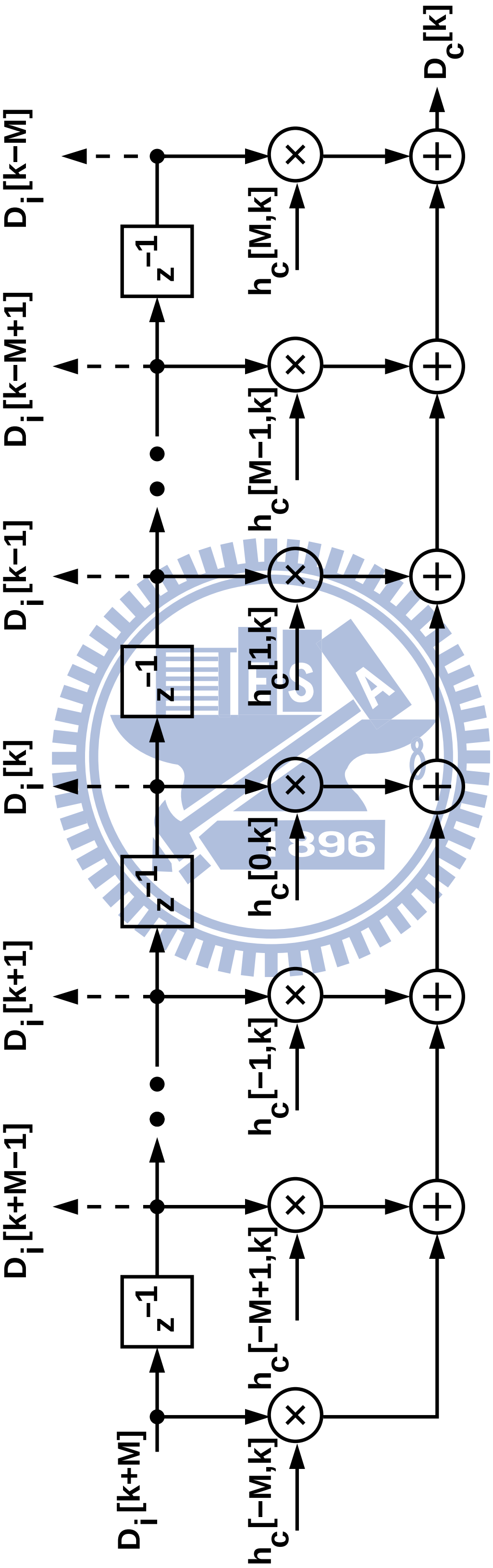


Figure 2.4: A jitter compensation filter with $2M + 1$ taps.

Exchanging the order of integration and summation, Equation (2.16) becomes

$$D_c[k] \approx D_i[k] + \frac{j\hat{\epsilon}[k]}{\pi} \int_{-\omega_B}^{+\omega_B} V(j\omega)e^{j\omega kT_s} F_c(M, \omega T_s) d\omega \tag{2.17}$$

where

$$F_c(M, \omega T_s) = \sum_{n=1}^{M} \frac{\sin(n\omega T_s)}{(-1)^n \times n} \tag{2.18}$$

Let $\Omega = \omega T_s$, if M approaches infinity, Equation (2.18) becomes (Appendix A details the derivation)

$$\lim_{M\to\infty} F_c(M, \Omega) = -\frac{\Omega}{2} \tag{2.19}$$

Furthermore, if there is no JDC measurement error so that $\hat{\epsilon}[k] = \epsilon[k]$, Equation (2.17) becomes $D_c[k] = V_i(kT_s)$. This proves that the JCF can compensate the sampling errors and recover the original $V_i(kT_s)$.

As an example, consider a sine wave input $V_i(t)$ expressed as

$$V_i(t) = A_i \times \sin(\omega_i t + \phi_i) \tag{2.20}$$

where A_i is the amplitude, ω_i is the input frequency, and ϕ_i is the phase. The corresponding output from the ADC is

$$D_i[k] = A_i \sin(\Omega_i(k + \epsilon[k]) + \phi_i) = V_i(kT_s) + q[k] + V_e[k] \tag{2.21}$$

From Equation (2.16), the resulting $D_c[k]$ from the JCF can be expressed as

$$D_c[k] \approx D_i[k] + \hat{\epsilon}[k] F_c(M, \Omega_i) \times 2A_i \cos(k\Omega_i + \phi_i) \tag{2.22}$$

The residual sampling error after the jitter compensation, defined as $V_e[k] \equiv D_c[k] - V_i(kT_s)$, can be approximated by

$$\begin{aligned} V_e[k] \approx{} & [\Omega_i + 2F_c(M, \Omega_i)]\, \epsilon[k] \times A_i \cos(k\Omega_i + \phi_i) \\ & + 2F_c(M, \Omega_i)\epsilon_e[k] \times A_i \cos(k\Omega_i + \phi_i) \end{aligned} \tag{2.23}$$

The averaged power of $V_e[k]$ is

$$P_e = \frac{A_i^2}{2}\left[\Omega_i + 2F_c(M,\Omega_i)\right]^2 \epsilon_{rms}^2 + 2A_i^2\left[F_c(M,\Omega_i)\right]^2 \epsilon_{e,rms}^2 \tag{2.24}$$

where ϵ_{rms} is the rms of the clock jitter, $\epsilon[k]$, and $\epsilon_{e,rms}$ is the rms of the JDC measurement error, $\epsilon_e[k]$. If $A_i = 1$ and the quantization noises are included, the SNR of the signal $D_c[k]$ can be calculated using Equation (2.5), Equation (2.7), and Equation (2.24). From Equation (2.19), if M is so large that $F_c(M,\Omega_i) \approx -\Omega_i/2$, the SNR becomes

$$\mathrm{SNR}_{c,\infty} = \frac{1}{(2/3)2^{-2B} + \Omega_i^2 \epsilon_{e,rms}^2} \tag{2.25}$$

Comparing Equation (2.25) with Equation (2.10), clock jitter rms ϵ_{rms} in Equation (2.10) is replaced by the jitter measurement error rms $\epsilon_{e,rms}$ in Equation (2.25). Thus, the jitter compensation can break the SNR limitation caused by clock jitter if the measured jitter error is smaller than the clock jitter.

2.5 Non-Ideal Effects in Jitter Compensation

The SNR after jitter compensation can be predicted using Equation (2.5), Equation (2.7), and Equation (2.24). There are several assumptions when deriving Equation (2.24). These assumptions include:

1. The quantization errors are ignored when sampling $D_i[k-M]$ to $D_i[k-1]$ and $D_i[k+1]$ to $D_i[k+M]$. However, as the number of the filter taps increases, the quantization noise power also increases.

2. When calculating $D_c[k]$, the sampling errors caused by clock jitter when sampling the neighboring samples are also assumed ignorable.

3. The filter coefficients of Equation (2.15) are irrational numbers. The effect of the finite precision is inevitable when implementing the jitter compensation technique in a system-on-chip (SOC).

The effects of these assumptions are discussed in the following subsections.

2.5.1 Quantization Error and Sampling Error

The quantization errors and the sampling errors of $D_i[k-M]$ to $D_i[k-1]$ samples and $D_i[k+1]$ to $D_i[k+M]$ samples are ignored when Equation (2.16) is used to calculate the compensated ADC output $D_c[k]$. If these errors are taking into consideration, the result of the jitter compensation can be expressed as

$$\begin{aligned} D_c[k] &= \sum_{n=k-M}^{k+M} D_i[n] \times h_c\left[k-n, \hat{\epsilon}[k]\right] \\ &= \sum_{n=k-M}^{k+M} \left\{V_i(nT_s) + q[n] + V_e[n]\right\} \times h_c\left[k-n, \hat{\epsilon}[k]\right] \end{aligned} \tag{2.26}$$

where $q[n]$ is the quantization error of $D_i[n]$ and $V_e[n]$ is the sampling error of $D_i[n]$ caused by the clock jitter.

As discussed in Section 2.4, $D_c[k]$ is a correct sample of $V_i(t)$ at kT_s if there is no JDC measurement error and M is large enough. However, the summation in Equation (2.26) not only compensates the sampling error $V_e[k]$ but also accumulates the quantization errors and the sampling errors of $D_i[k-M]$ to $D_i[k-1]$ and $D_i[k+1]$ to $D_i[k+M]$. The accumulation of these errors may degrade the performance of the jitter compensation.

The error in $D_c[k]$ contributed by sampling $D_i[k-M]$ to $D_i[k-1]$ and $D_i[k+1]$ to $D_i[k+M]$ is defined as

$$\begin{aligned} V_n[k] = & \sum_{n=k-M}^{k-1} (q[n] + V_e[n]) \times h_c\left[k-n, \hat{\epsilon}[k]\right] \\ & + \sum_{n=k+1}^{k+M} (q[n] + V_e[n]) \times h_c\left[k-n, \hat{\epsilon}[k]\right] \\ \approx & \sum_{n=k-M}^{k-1} (q[n] + V_e[n]) \times \frac{(-1)^{n+1}\hat{\epsilon}[k]}{n} \\ & + \sum_{n=k+1}^{k+M} (q[n] + V_e[n]) \times \frac{(-1)^{n+1}\hat{\epsilon}[k]}{n} \end{aligned} \tag{2.27}$$

The worst case of the signal power of the $V_n[k]$ sequence occurs when $M \to \infty$. Thus,

$$P_n \equiv \overline{(V_n[k])^2} \approx 2\zeta(2) \times \hat{\epsilon}^2_{rms} \times (P_q + P_e) \tag{2.28}$$

where P_q and P_e is defined as Equation (2.7) and Equation (2.9) respectively, and

$$\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \tag{2.29}$$

is a well-known Riemann zeta function [29]. Therefore, Equation (2.28) becomes

$$P_n = \frac{1}{3}\pi^2 \times \hat{\epsilon}^2_{rms} \times (P_q + P_e) \tag{2.30}$$

Comparing Equation (2.30) with Equation (2.7), P_n is usually much smaller than P_q or P_e in practical because that the error power is multiplied by $\hat{\epsilon}^2_{rms}$. Thus, the effect of the quantization errors and the sampling errors of $D_i[k-M]$ to $D_i[k-1]$ and $D_i[k+1]$ to $D_i[k+M]$ are indeed negligible even when the number of filter taps approach to infinity when calculating $D_c[k]$.

2.5.2 Finite Precision of the Filter Coefficients h_c

In order to represent the irrational coefficients into digital form, quantization processes are required to convert irrational numbers into digital codes. Similar to an ADC, the conversion also introduces quantization error. Take this effect into account when evaluating $D_c[k]$, the result of JCF can be expressed as

$$D_c[k] = \sum_{n=k-M}^{k+M} D_i[n] \times \left\{ h_c\left[k-n, \epsilon[k]\right] + q_h[n, \epsilon[k]] \right\} \tag{2.31}$$

where $q_h[n, \epsilon[k]]$ is the error induced by quantizing the filter coefficients $h_c\left[k-n, \epsilon[k]\right]$. The error of $D_c[k]$ induced by the finite precision of the filter coefficients is defined as $V_{h,q}$ and

$$V_{h,q}[k] = \sum_{n=k-M}^{k+M} D_i[n] \times q_h[n, \epsilon[k]] \tag{2.32}$$

As long as the $D_i[n]$ sequence are independent of $\epsilon[k]$ sequence, the power of the $V_{h,q}[k]$ sequence can be expressed as

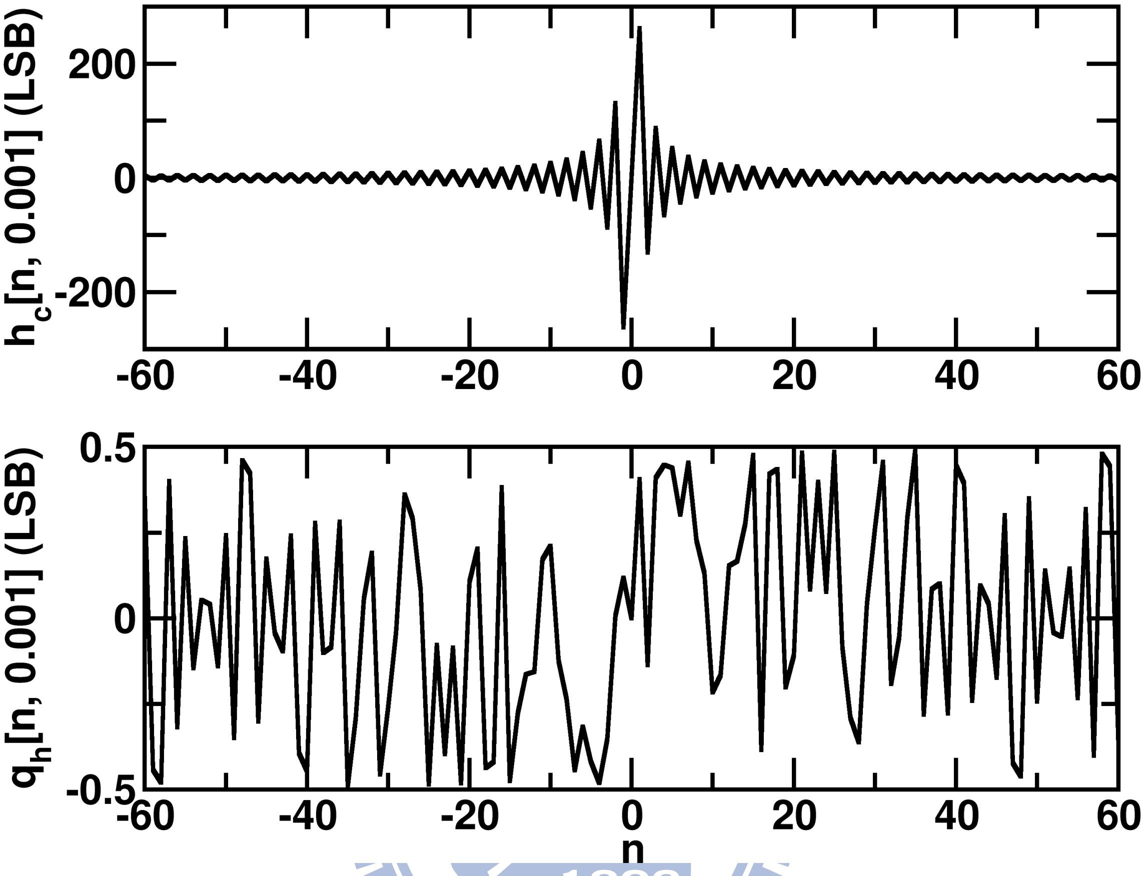


Figure 2.5: $h_c[n, \hat{\epsilon}[k]]$ and $q_h[n, \hat{\epsilon}]$ plots example when $\hat{\epsilon}$=0.001 and the filter coefficients are represented by binary fraction of 18-bit wide.

$$P_{h,q} = \overline{(D_i[n])^2} \times \overline{\sum_{n=k-M}^{k+M} q_h^2[n, \epsilon[k]]} \tag{2.33}$$

Similar to the quantization error power of an ADC expressed as Equation (2.7), the binary fraction expression for $h_c[k-n, \epsilon[k]]$ of B_h-bit wide results in a quantization error which can be expressed as

$$\sum_{n=k-M}^{k+M} q_h^2[n, \epsilon[k]] \approx 2M \times \frac{1}{12} 2^{-2B_h} \tag{2.34}$$

Figure 2.5 shows an example of $h_c[n, \epsilon[k]]$ and $q_h[n, \epsilon[k]]$ plots with B_h=18 and $\hat{\epsilon}[k]$=0.001.

For an ideal B-bit ADC with full-swing sine wave input, $P_{h,q}$ must smaller than P_q to preserve the resolution, i.e.,

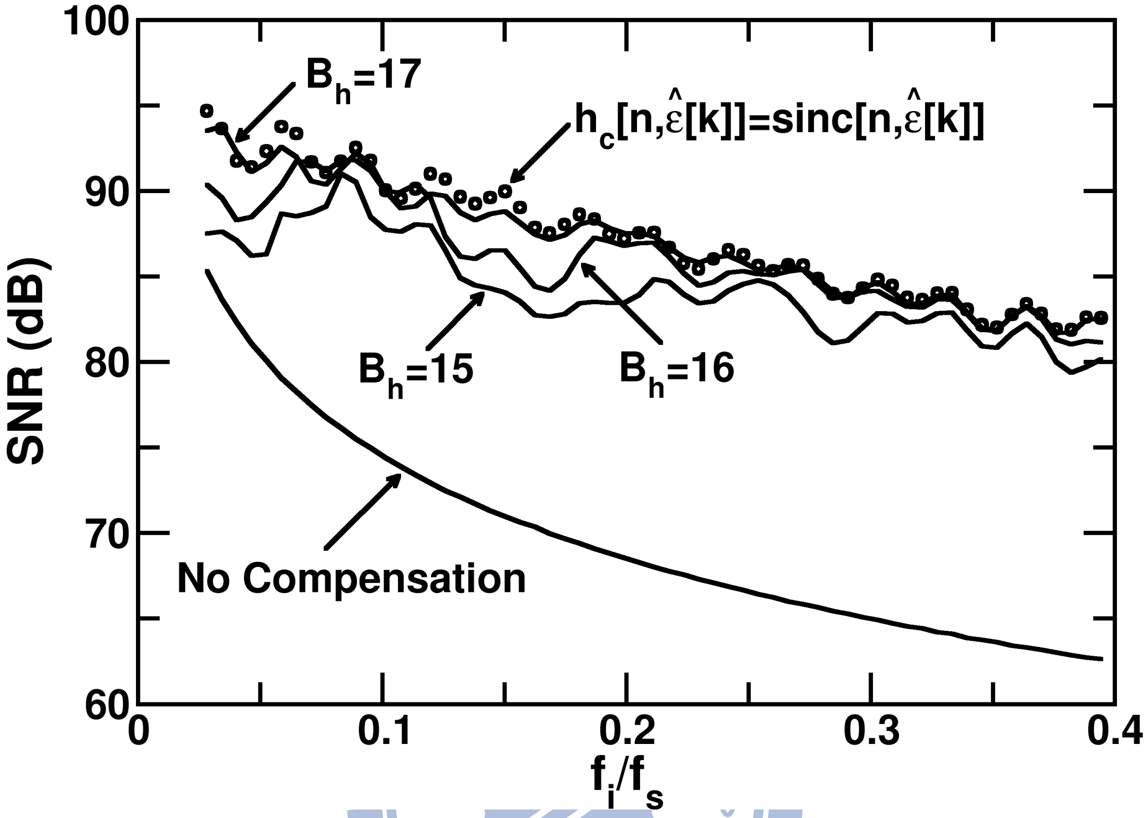


Figure 2.6: SNR results of jitter compensation of different B_h. Circles are simulations with irrational $h_c[n, \hat{\epsilon}[k]] = \mathrm{sinc}(n - \hat{\epsilon}[k])$.

$$\frac{A^2}{2} \times 2M \times \frac{1}{12} 2^{-2B_h} \le \frac{1}{12} \left(\frac{2A}{2^B} \right)^2 \tag{2.35}$$

Thus,

$$B_h \ge B - 1 + \frac{1}{2} \log_2(M) \tag{2.36}$$

For example, if B=16 and M=16, B_h=17 can be chosen according to Equation (2.36). Figure 2.6 shows the simulated SNR performance of a 16-bit ADC when using $B_h = 15$, $B_h = 16$ and $B_h = 17$ to approximate the irrational tap coefficients $h_c\,[n, \epsilon[k]]$. Assume that $\epsilon_{rms} = 0.0003$ and the resolution of the JDC is 0.0001. The circles are simulations without quantizing the tap coefficients, i.e., the irrational $h_c[n, \hat{\epsilon}[k]] = \mathrm{sinc}(n - \hat{\epsilon}[k])$ are used in the simulations. As predicts by Equation (2.36), the SNR performance after jitter compensation when $B_h = 17$ are nearly the same with the SNR performance when

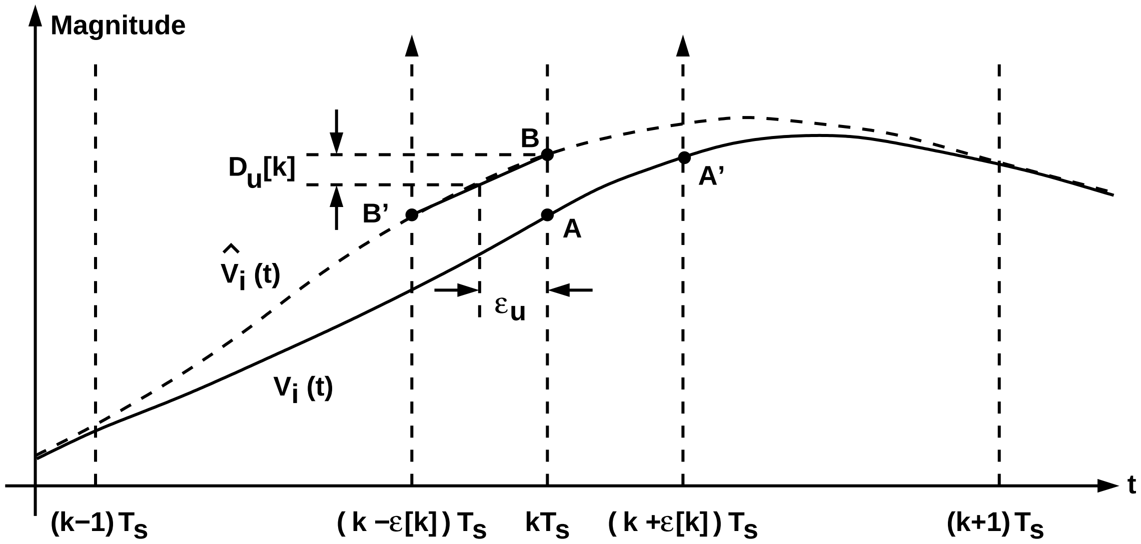


Figure 2.7: A graphic illustration of simplified jitter compensation principle.

$B_h > 17$.

Note that Equation (2.36) is overestimated for large M. As M increases, $h_c[M, \epsilon[k]] = \mathrm{sinc}(M - \epsilon[k])$ will convergence to zero eventually, so does $q_h[M, \epsilon[k]]$.

2.6 Simplified Jitter Compensation Filter

In Figure 2.4, the tap coefficients, $h_c\,[n, \epsilon[k]]$, are recalculated every clock cycle due to the different $\hat{\epsilon}[k]$ at different k. The hardware cost of this time-variant JCF is very high. It is also difficult for this JCF to achieve high-speed operation. A simplified JCF is proposed to reduce the difficulty when implementing the JCF.

As shown in Figure 2.7, if a sampling error occurs at the k-th sampling instant, the $V_i(t)$ is sampled at $t = (k + \epsilon[k])T_s$ instead of $t = kT_s$. The magnitude at point A' is quantized as $D_i[k]$. Thus, the ADC perceives a different $\hat{V}_i(t)$ input instead of $V_i(t)$. The signal $\hat{V}_i(t)$ has a value of $D_i[k]$ at $t = kT_s$, denoted as point B. The filter of Equation (2.15) interpolates the value of $\hat{V}_i(t)$ at $t = (k - \epsilon[k])T_s$, denoted as point B', which is a correct estimation of $V_i(t)$ at $t = kT_s$. If $\epsilon[k] \ll 1$, the curve between point B to point B' is nearly a straight line. Thus, if the clock jitter is ϵ_u and using Equation (2.16) to compensate the sampling error, the magnitude required to be compensated is defined as $D_u[k]$ and

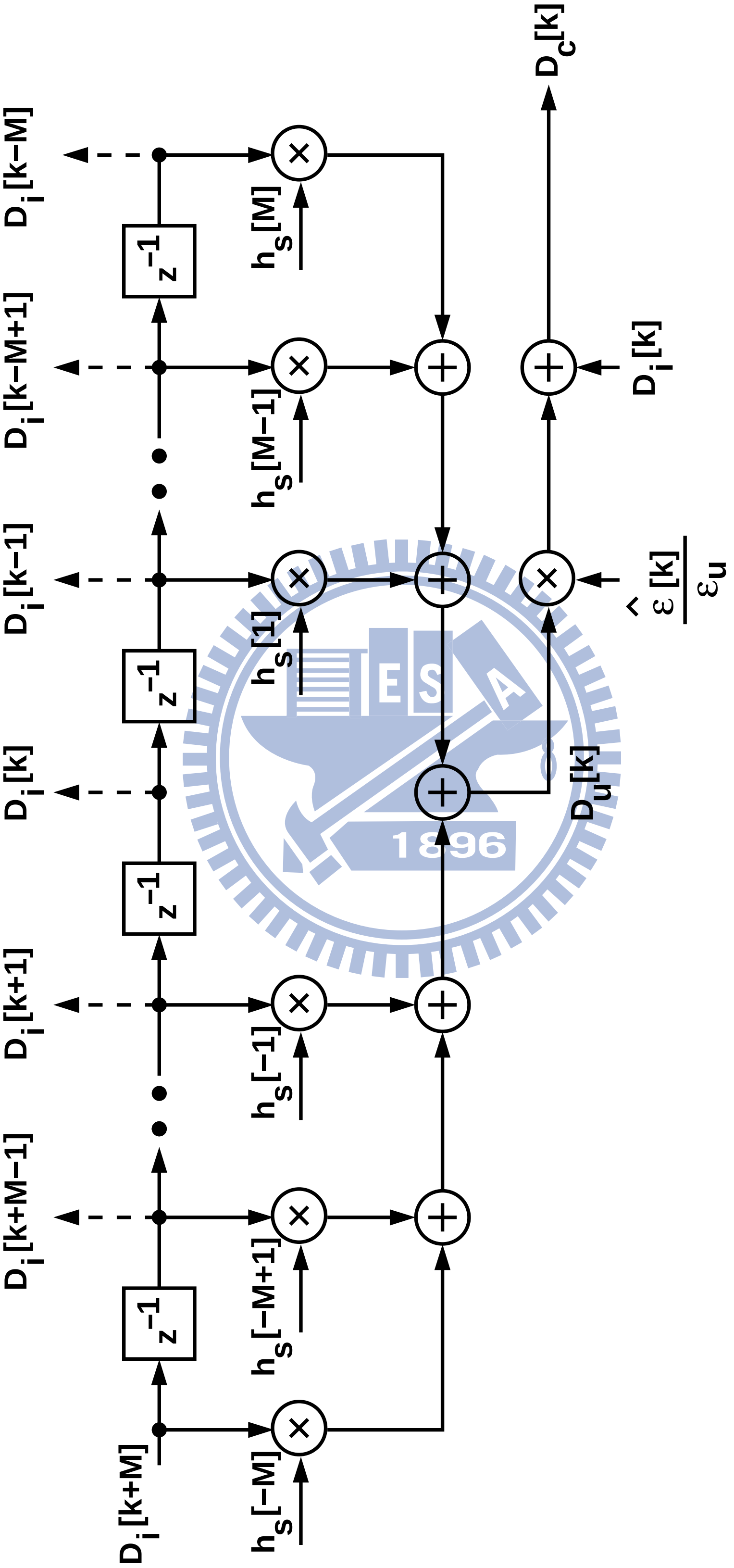


Figure 2.8: A simplified jitter compensation filter with $2M + 1$ taps.

$$\begin{aligned} D_u[k] &= D_c[k] - D_i[k] \\ &= \sum_{n=k-M}^{k-1} D_i[n] \times h_c\left[k-n, \hat{\epsilon}[k]\right] + \sum_{n=k+1}^{k+M} D_i[n] \times h_c\left[k-n, \hat{\epsilon}[k]\right] \end{aligned} \tag{2.37}$$

and the slope of the straight line between point B to point B' is

$$D_s[k] = \frac{D_u[k]}{\epsilon_u} \tag{2.38}$$

Therefore, the magnitude required to be compensated is simply this slope at the k-th sample, multiplied by the k-th sampling jitter, $\hat{\epsilon}[k]$. As a consequence, Equation (2.16) can be approximated by

$$\begin{aligned} D_c[k] &\approx D_i[k] + \hat{\epsilon}[k] \times \frac{D_u[k]}{\epsilon_u} \\ &= D_i[k] + \frac{\hat{\epsilon}[k]}{\epsilon_u} \times \left\{ \sum_{n=k-M}^{k-1} \left\{ D_i[n] \times h_c[k-n, \epsilon_u] \right\} \right. \\ &\quad \left. + \sum_{n=k+1}^{k+M} \left\{ D_i[n] \times h_c[k-n, \epsilon_u] \right\} \right\} \end{aligned} \tag{2.39}$$

We can define a simplified JCF with the filter coefficients as

$$h_s[n] = h_c[n, \epsilon_u] = \mathrm{sinc}(n - \epsilon_u) \tag{2.40}$$

Thus, $D_u[k]$ becomes

$$D_u[k] = \sum_{n=k-M}^{k-1} \left\{ D_i[n] \times h_s[k-n] \right\} + \sum_{n=k+1}^{k+M} \left\{ D_i[n] \times h_s[k-n] \right\} \tag{2.41}$$

and the output of the simplified JCF is

$$D_c[k] \approx D_i[k] + \frac{\hat{\epsilon}[k]}{\epsilon_u} \times D_u[k] \tag{2.42}$$

In the above equations, ϵ_u is a predefined jitter constant. The value of ϵ_u is not crucial. It can be chosen so that ϵ_u approximates to the standard deviation of $\epsilon[k]$.

Figure 2.8 shows the resulting JCF. In this implementation, once a specific ϵ_u is chosen, the $h_s[n]$ tap coefficients are fixed. The $D_u[k]$ signal of Equation (2.41) is calculated using a time-invariant FIR filter with the fixed $h_s[n]$ tap coefficients. The sampling error is estimated by multiplying $D_u[k]$ with the $\hat{\epsilon}[k]/\epsilon_u$ ratio. In Figure 2.8, $\hat{\epsilon}[k]$ is only used at one place. Note that the division is not needed here because the results of the TDC calibration are in the form of divided by ϵ_u already. The TDC background calibration will be introduced in Chapter 4 and Chapter 5.

As derived in Section 2.4, for an ideal JCF without JDC measurement error, the output of the JCF becomes $D_c[k] = V_i(kT_s)$ which is a correct sample without sampling error. Therefore, the sampling error can be derived from Equation (2.16)

$$\begin{aligned} V_e[k] = D_c[k] - D_i[k] = &\sum_{n=k-M}^{k-1} D_i[n] \times h_c\,[k-n, \epsilon[k]] \\ &+ \sum_{n=k+1}^{k+M} D_i[n] \times h_c\,[k-n, \epsilon[k]] \end{aligned} \tag{2.43}$$

From Equation (2.41) and Equation (2.42), the sampling error calculated by the simplified JCF can be expressed as

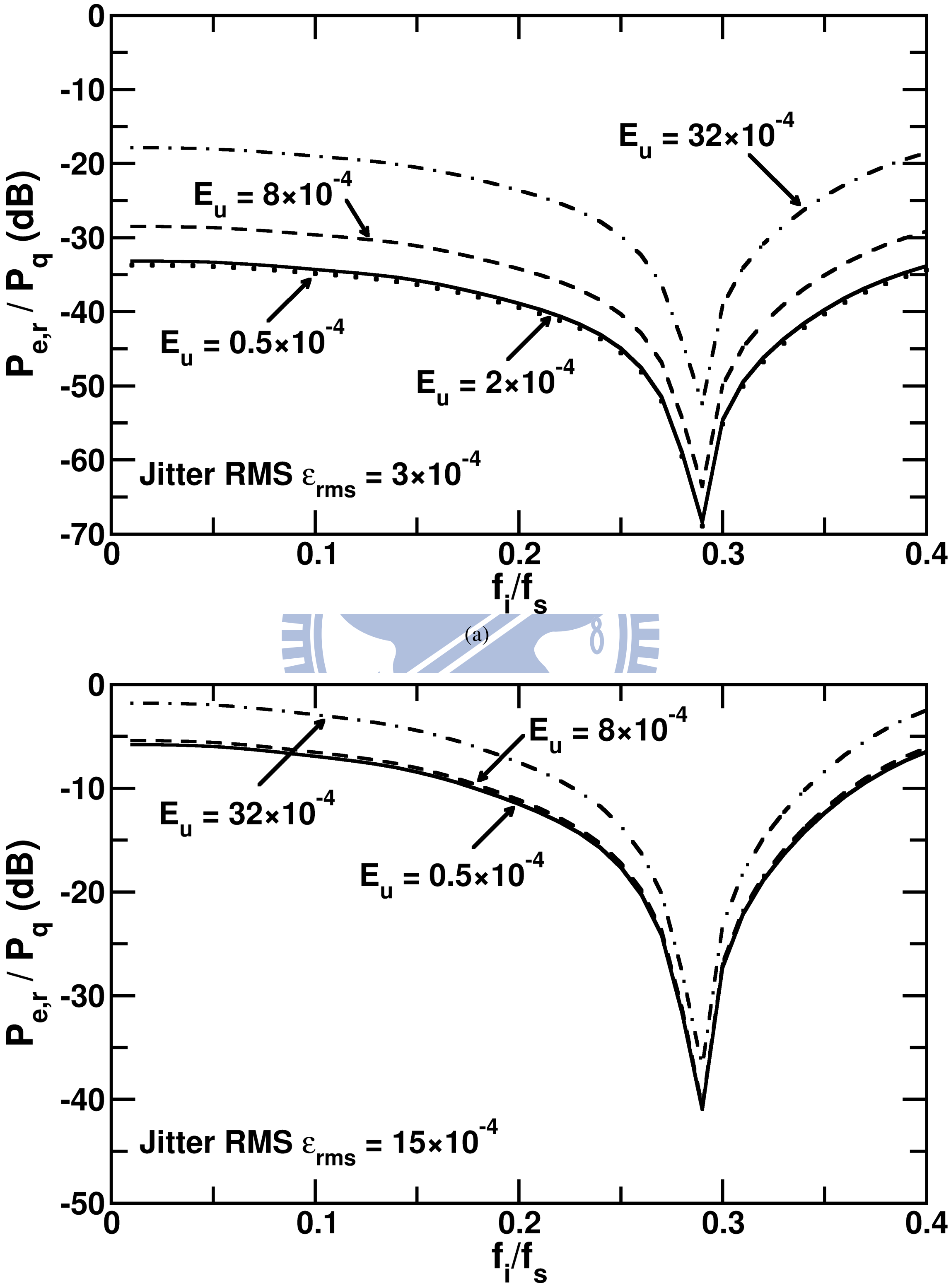
$$\begin{aligned} V_{e,s}[k] = &\sum_{n=k-M}^{k-1} D_i[n] \times h_s[k-n] \times \frac{\epsilon[k]}{\epsilon_u[k]} \\ &+ \sum_{n=k+1}^{k+M} D_i[n] \times h_s[k-n] \times \frac{\epsilon[k]}{\epsilon_u[k]} \end{aligned} \tag{2.44}$$

In order to decide the value of ϵ_u, we defined E_s which is the difference between $h_s[n]$ and $h_c[n]$,

$$E_h[n, \epsilon[k]] \equiv h_c[k-n, \epsilon[k]] - h_s[k-n] \times \frac{\epsilon[k]}{\epsilon_u[k]} \tag{2.45}$$

The residual sampling error caused by the simplified JCF, defined as $V_{e,r}[k] \equiv V_e[k] - V_{e,s}[k]$, can be derived from Equation (2.43) and Equation (2.44),

$$V_{e,r}[k] = \sum_{n=k-M}^{k-1} D_i[n] \times E_h[n, \epsilon[k]] + \sum_{n=k+1}^{k+M} D_i[n] \times E_h[n, \epsilon[k]] \tag{2.46}$$



(a)

(b)

Figure 2.9: $P_{e,r}$ normalized to the quantization noise P_q Plots. (a) $\epsilon_{rms} = 3 \times 10^{-4}$ and (b) $\epsilon_{rms} = 15 \times 10^{-4}$.

As an example, consider a sine wave input $V_i(t)$ expressed as

$$V_i(t) = A_i \times \sin(\omega_i t + \phi_i) \tag{2.47}$$

Using Equation (2.46) and $E_h[n, \epsilon[k]] \approx E_h[-n, \epsilon[k]]$, we have

$$V_{e,r}[k] = A_i \sin(\omega_i k T_s + \phi_i) \times \left\{ \sum_{n=1}^{M} 2\cos(n\omega_i T_s) \times E_h[n, \epsilon[k]] \right\} \tag{2.48}$$

Thus, the error power of the $V_{e,r}[k]$ sequence is

$$P_{e,r} \equiv \overline{(V_{e,r}[k])^2} = 2A_i^2 \times \overline{\left\{ \sum_{n=1}^{M} \cos(n\omega_i T_s) \times E_h[n, \epsilon[k]] \right\}^2} \tag{2.49}$$

In order to calculate the power of the $V_{e,r}[k]$ sequence, the distribution of the clock jitter $\epsilon[k]$ must be considered. Assume that the clock jitter $\epsilon[k]$ is normal-distributed with zero mean and the standard deviation is ϵ_{rms}. Figure 2.9 shows the $P_{e,r}$ normalized to the quantization noise P_q of a 16-bit ADC. The value of E_u is not crucial. For $E_u < 32 \times 10^{-14}$, the $P_{e,r}$ is smaller than the quantization noise even when the clock jitter $\epsilon_{rms} = 15 \times 10^{-4}$. $\epsilon_{rms} = 15 \times 10^{-4}$ is equivalent to $\Delta_{tms} = 18.75$ ps at 80 MHz sampling rate, which is an extremely large value of jitter.

The simplified JCF can be interpreted in a more intuitively way. The Taylor series of the input signal expanded at $t = x$ can be expressed as

$$V_{in}(t) = V(x) + \frac{V'(x)}{1!}(t - x) + \frac{V''(x)}{2!}(t - x)^2 + \frac{V'''(x)}{3!}(t - x)^3 + \ldots \tag{2.50}$$

At the k-th sample, the nominal smaling time is $t = kT_s$. The jitter at the k-th sample is $\Delta t[k] = \epsilon[k] \times T_s$. Therefore, the Taylor series expansion of the input signal at $t = kT_s + \epsilon[k] \times T_s$ can be obtained by substituting $x = (kT_s + \Delta t[k])$ in Equation (2.50), i.e.,

$$\begin{aligned} V_{in}(t) &= V(kT_s + \varDelta t[k]) \\ &+ \frac{V'(kT_s + \varDelta t[k])}{1!}\left\{t - \left[kT_s + \varDelta t[k]\right]\right\} \\ &+ \frac{V''(kT_s + \varDelta t[k])}{2!}\left\{t - \left[kT_s + \varDelta t[k]\right]\right\}^2 \\ &+ \frac{V'''(kT_s + \varDelta t[k])}{3!}\left\{t - \left[kT_s + \varDelta t[k]\right]\right\}^3 + \ldots \end{aligned} \tag{2.51}$$

The correct sample ,$V_{in}(kT_s)$, can be obtained by substituting $t = kT_s$ into Equation (2.51),

$$\begin{aligned} V_{in}(kT_s) &= V(kT_s + \varDelta t[k]) \\ &+ \frac{V'(kT_s + \varDelta t[k])}{1!}\left[-\varDelta t[k]\right] \\ &+ \frac{V''(kT_s + \varDelta t[k])}{2!}\left[-\varDelta t[k]\right]^2 \\ &+ \frac{V'''(kT_s + \varDelta t[k])}{3!}\left[-\varDelta t[k]\right]^3 + \ldots \\ &= V_c[k] \end{aligned} \tag{2.52}$$

With the information of clock jitter$\varDelta t[k]$, the input signal $V_{in}(t)$ and its n-th derivative at $t = kT_s + \varDelta t[k]$, the correct sample can be obtained at incorrect sampling time by Equation (2.51). Comparing Equation (2.42) with Equation (2.52), the simplified JCF is the first order Taylor series approximation of jitter correction. and $-D_u[k]/\epsilon_u$ is the first order derivative of $V_{in}(t)$ at $t = kT_s + \varDelta t[k]$.

As derived from Section 2.5.2, a 17-bit wide binary fraction expression for $h_c[k - n, \epsilon[k]]$ is required for a 16-bit ADC system. Therefore, $2M$ multipliers with 17-bit × 16-bit is required in a JCF, which results in an area-hunger digital circuit. However, the multipliers can be replaced with a few adders for the simplified JCF since the coefficients are time-invariant. Table 2.1 shows the binary sign-magnitude expression of the filter coefficients, $h_s[n]$, for a JCF with 15 filter taps. $\epsilon_u = 0.0002$ is chosen and 20 bit is used to approximate the irrational number. The number of the adders is proportional to the number of bits with a digital output 1 in the binary expression of the filter coefficients. For example, $h_s[1] = 0.00000000000011010001$, only 4 adders are required to calculate $D_i[k-1] \times h_s[1]$. Thus, The area of the simplified JCF is decreased dramatically.

Table 2.1: Sign-magnitude expression of $h_s[n]$

	$h_s[n]$		
	Decimal Fraction	Binary Fraction	
n		Sign	Magnitude
-7	-0.000028570610	1	0.00000000000000011101
-6	0.000033332220	0	0.00000000000000100010
-5	-0.000039998397	1	0.00000000000000101001
-4	0.000049997497	0	0.00000000000000110100
-3	-0.000066662218	1	0.00000000000001000101
-2	0.000099989994	0	0.00000000000001101000
-1	-0.000199959995	1	0.00000000000011010001
1	0.000200039995	0	0.00000000000011010001
2	-0.000100009994	1	0.00000000000001101000
3	0.0000666711073	0	0.00000000000001000101
4	-0.000050002497	1	0.00000000000000110100
5	0.000040001597	0	0.00000000000000101001
6	-0.000033334442	1	0.00000000000000100010
7	0.000028572243	0	0.00000000000000011101

2.7 A 16-bit 80 MS/s ADC Design Example

A 16-bit ADC system operating at a sampling rate of 80 MS/s was simulated by using a C program to verify the proposed jitter compensation techniques. Its sampling period is $T_s = 12.5$ ns. Assume that the rms of the clock jitter is $\Delta t_{rms} = \epsilon_{rms} T_s = 3.75$ ps, i.e., $\epsilon_{rms} = 3 \times 10^{-4}$. From Equation (2.5), Equation (2.7), and Equation (2.24), to ensure better than 80 dB SNR for an input frequency up to 32 MHz, i.e., $\Omega_i < (4/5)\pi$, one can choose $\epsilon_{e,rms} T_s < 1.25/\sqrt{12}$ ps and $M = 7$. The $\epsilon_{e,rms}$ indicates the required resolution and accuracy for the JDC. To simplify the simulations, the JDC is an ideal one with a uniform quantization step size of 1.25 ps, so that $\epsilon_{e,rms} T_s = 1.25/\sqrt{12}$ ps. The number of taps for the JCF is $2M + 1 = 15$. The bandwidth limitation is due to the proposed JDC calibration, which will be discussed in Chapter 4 and Chapter 5.

Figure 2.10 shows the simulated ADC output spectrum before and after the jitter compensation for the system in Figure 2.2 and $M = 7$ is chosen. The JCF is the one shown in Figure 2.8, i.e., a simplified JCF is adopted in simulations. The input is a sine wave signal with frequency of 26 MHz. As the figure reveals, the noise floor is decreased by about

20 dB after the jitter compensation. In other words, the jitter compensation can improve the SNR by 20 dB.

Figure 2.11 shows the simulated ADC output spectrum before and after the jitter compensation when the input is a wide-band signal. In this simulation, the input is composed of four sine wave signals with different frequencies and phases. The jitter compensation for this multi-tone test can improve the ADC SNR from 65.46 dB to 83.55 dB.

Figure 2.12 shows the SNR performance from simulations of the system in Figure 2.2 for different input frequencies. The circle symbols in Figure 2.12 are simulation results. Also shown as the solid line in the figure are the calculated SNR of a JCF with $M = 7$ using Equation (2.5), Equation (2.7), and Equation (2.24). Perfect match between the simulated results and the calculated results can be observed in the figure. The dashed line shown in Figure 2.12 is the SNR of an ideal JCF calculated using Equation (2.25). This dashed line is the limitation of the jitter compensation for a given JDC's resolution, which is assumed to be 1.25 ps in simulation. At $f_i/f_s = 0.4$, i.e., $\Omega_i = (4/5)\pi$, the JCF can improve the SNR by 20 dB at input frequency close to $0.4f_s$. As a reference, the uncompensated SNR in Figure 2.12 is calculated using Equation (2.10).

As the number of taps of the JCF increased, the SNR after jitter compensation will approach to the ideal performance gradually as predict by Equation (2.25). Figure 2.13 shows the SNR performance when $M = 31$ is chosen. The circle symbols in Figure 2.13 are simulation results. The solid line is the calculated SNR of a JCF with $M = 31$ using Equation (2.5), Equation (2.7), and Equation (2.24). Perfect match between the simulated results and the calculated results still can be observed in the figure. The compensated SNR for a JCF with $M = 31$ is almost the same with the SNR calculated using Equation (2.25).

2.8 Summary

A digital jitter compensation for ADCs is presented in this chapter. If the jitter of the sampling clock is measured, the jitter compensation technique can correct the sampling error caused by the clock jitter, thus allowing the use of a cheaper clock source. Most of the compensation overhead is the digital circuitry; therefore both the circuit area and the power consumption are scaled along with the technology scaling. Since the jitter compensation is performed in the digital domain and no feedback signal is required for the ADC, the jitter compensation circuit and the ADC can be connected in parallel. No modification is required for the ADC.

The SNR improvement of the proposed jitter compensation technique is determined by the resolution of the JDC, the number of taps for the JCF and the rms of input clock jitter. Theoretical analyses and System simulations are both provided to verify the proposed jitter compensation technique in this chapter.

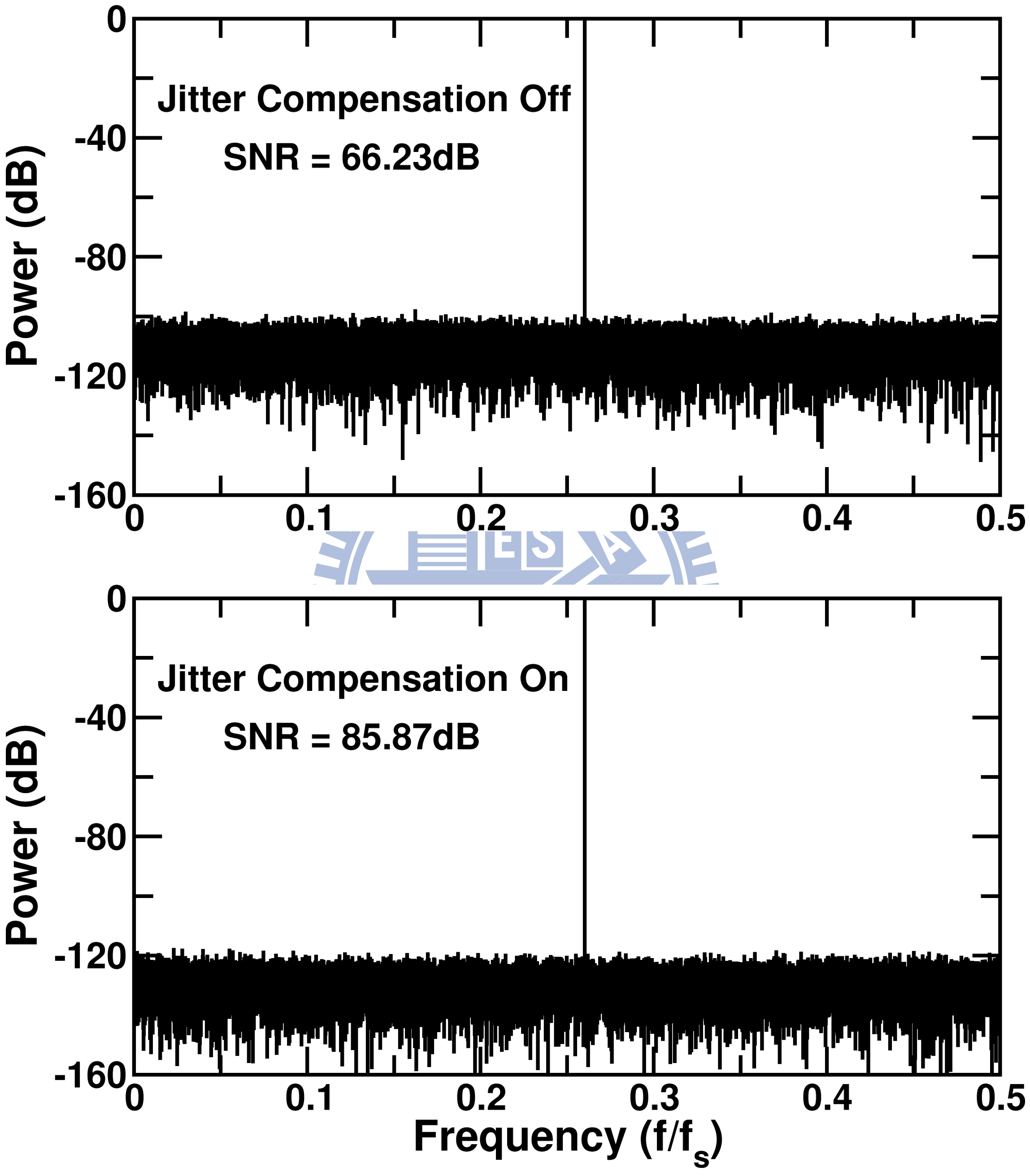


Figure 2.10: The ADC output power spectrum before and after the jitter compensation for $M = 7$.

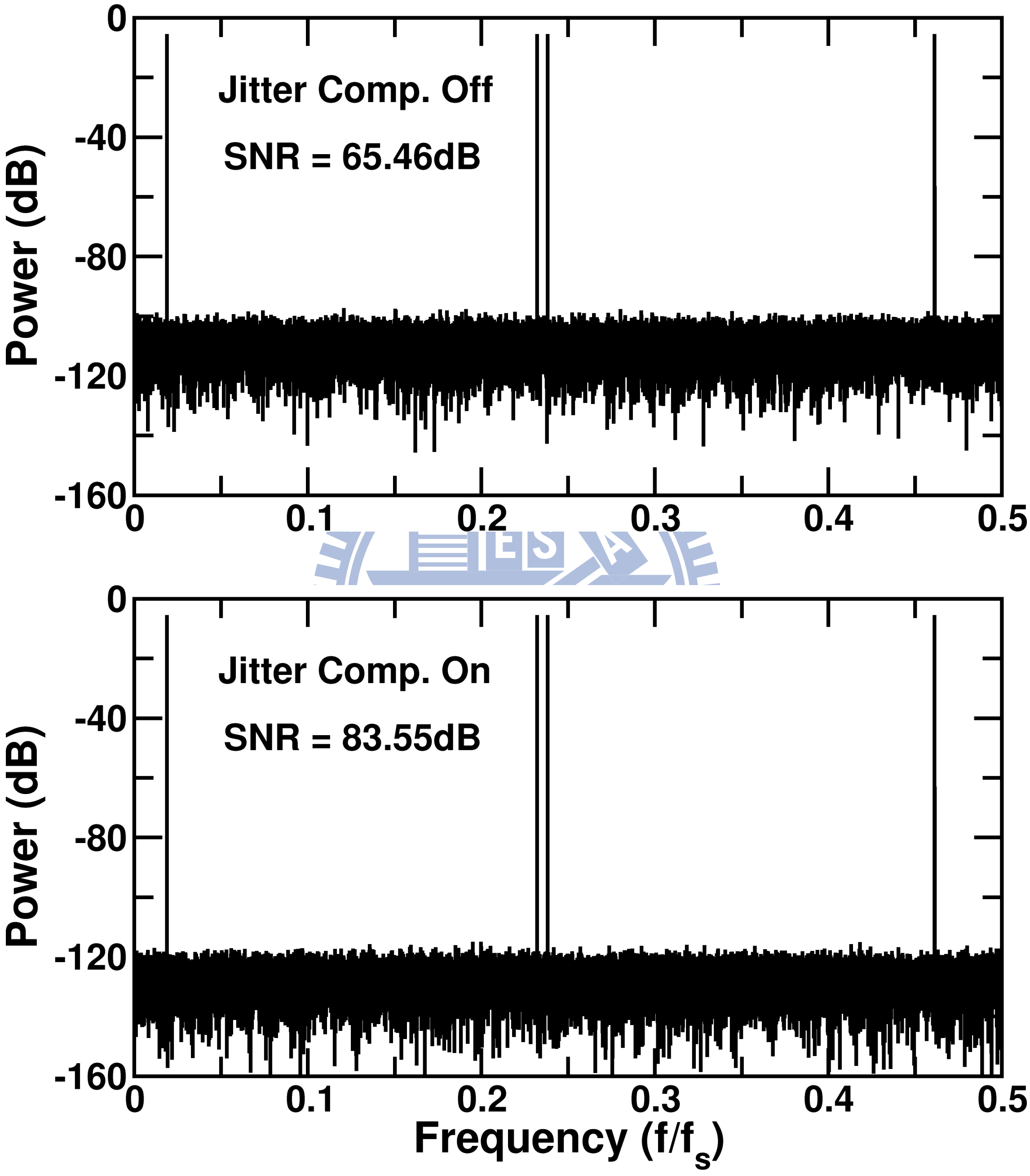


Figure 2.11: The ADC output power spectrum before and after the jitter compensation for $M = 7$.

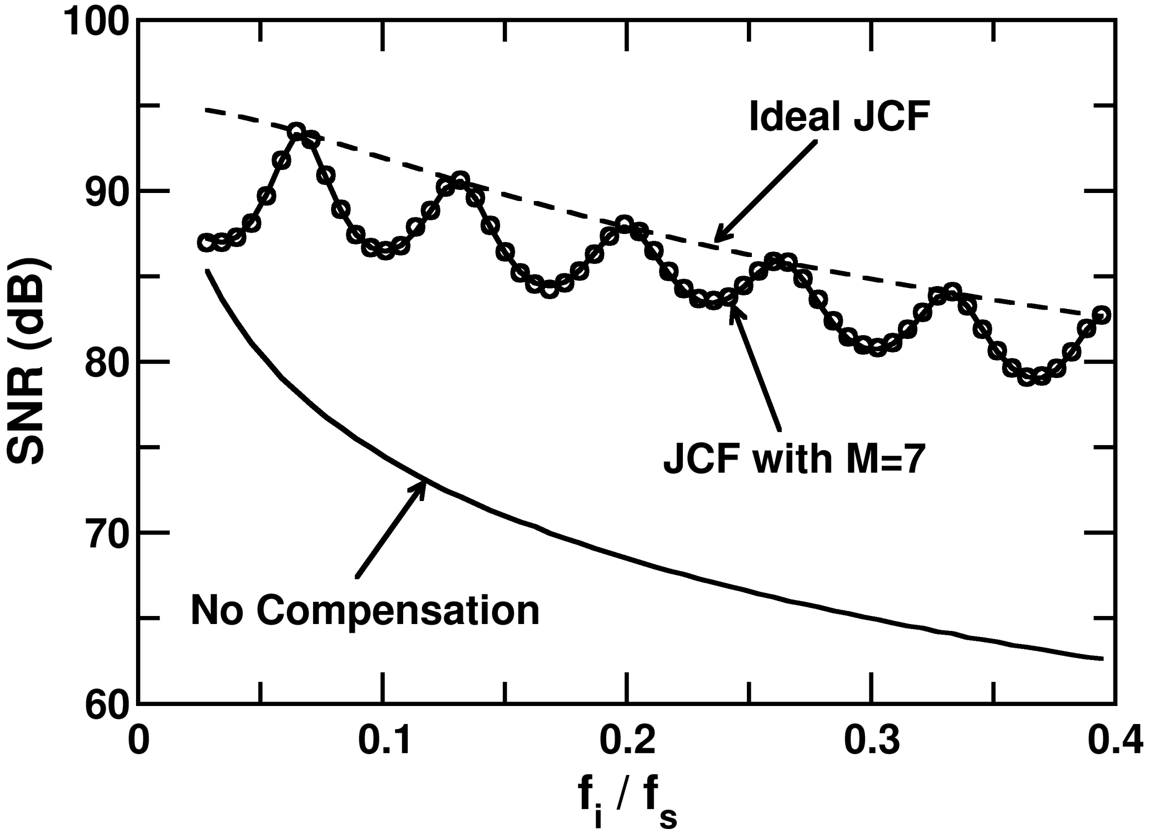


Figure 2.12: SNR results of jitter compensation for $M = 7$. Circles are from simulations. Lines are from calculations. $f_i = \omega_i/(2\pi)$ is the input frequency, and $f_s = 1/T_s$ is the sampling rate.

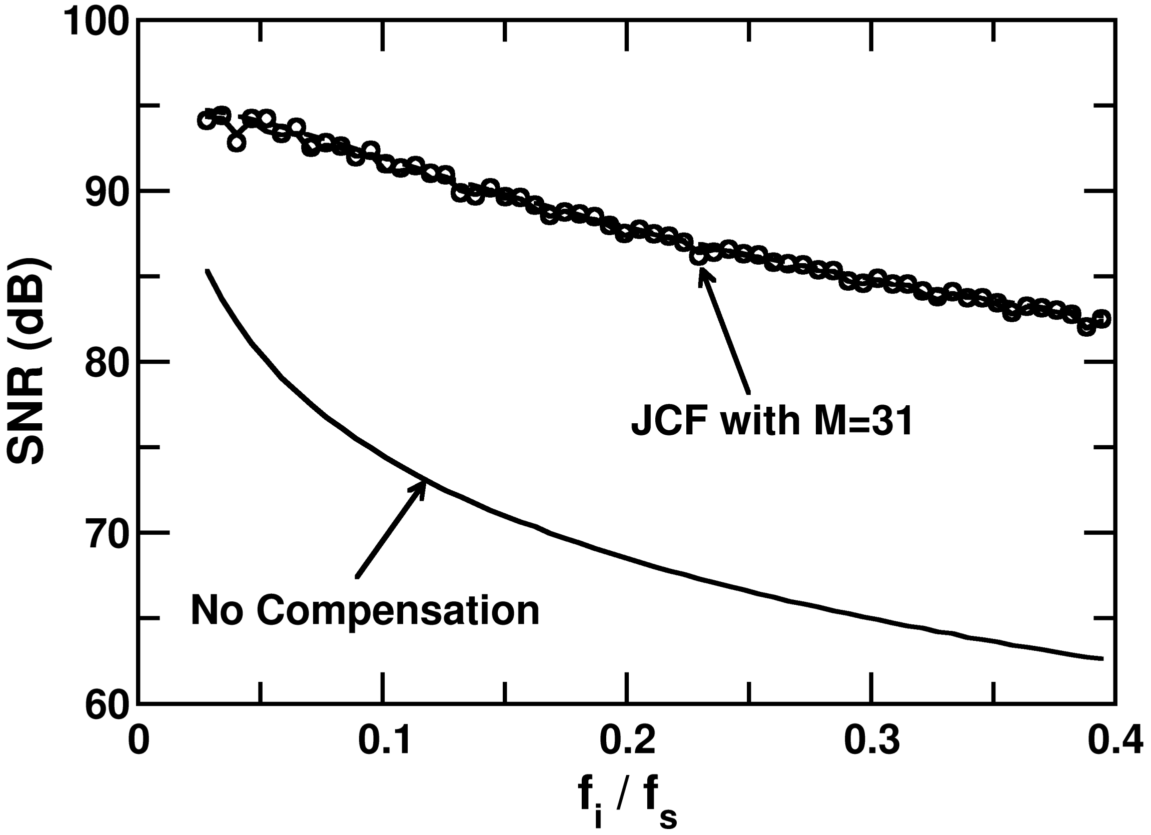


Figure 2.13: SNR results of jitter compensation for $M = 31$. Circles are from simulations. Lines are from calculations

Chapter 3

Time-to-Digital Converter

3.1 Introduction

The proposed jitter compensation scheme requires a timing measurement circuit to perform the jitter-to-digital conversion so that the sampling error introduced by the clock jitter can be compensated in the digital domain. The JDC in Figure 2.2 contains a time-to-digital converter (TDC) that digitizes the timing (or phase) difference between two clocks of identical frequency. As shown in Figure 3.1, a TDC is used to compare the rising edges of two clocks V_1 and V_2. For a 16-bit 80 MS/s ADC system, the TDC must perform the conversion every clock cycle and have a resolution better than 1 ps.

In order to quantized the timing information into the digital code, several time-to-digital conversion circuits had been demonstrated previously in literature. An up-to-date survey can be found in [30]. Most of them require certain types of precision circuits, such as delay elements with precise delay [31, 32], oscillators with precise frequency [33], or time-to-voltage converters with precise conversion function [34, 35]. In this chapter, we will discuss various types of TDC and their limitations.

3.2 Counter-Based TDC

The most direct method for measuring a time interval is to use a counter. As shown in Figure 3.2, the counter is triggered at the raising edge of V_1 signal and stopped at the

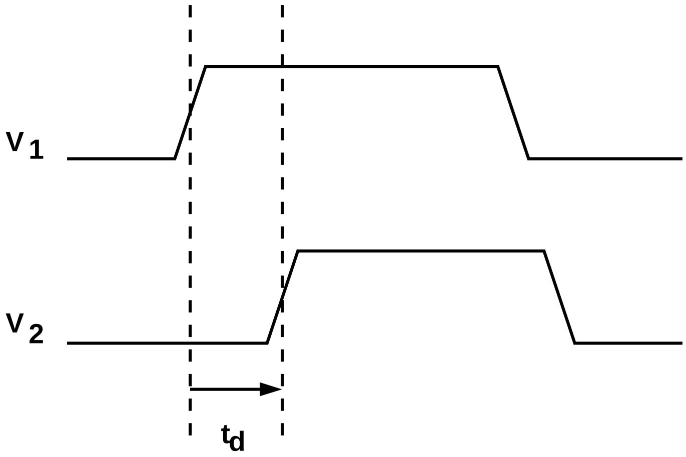


Figure 3.1: Timing difference measured by a TDC.

raising edge of V_2 signal. Therefore, the timing interval to be measured is proportional to the resulting count,

$$t_d \approx N \times T_c \tag{3.1}$$

where T_c is the clock period of the counter and N is the resulting count which is 6 in Figure 3.2.

The resolution of such TDC is limited by the period of the counter clock. Like the quantization noise in an ADC, the root-mean-squared value (rms) of the quantization step for the counter based TDC can be derived as [36]

$$\Delta t_{q,rms} = \frac{\pi T_c}{8} \tag{3.2}$$

The accuracy of counter based measurements can be improved by taking a series of measurements of the same interval t_d, and averaging the results [37]. Therefore, this technique is often used for frequency measurement since several periods can be measured and a better resolution can be obtained by average. For example, if M successive periods are measured for a periodic signal with frequency of f_s, i.e. $t_d = M \times (1/f_s) \approx N \times T_c$. Thus, the frequency under measured is

$$f_s = \frac{M}{N \times T_c} \tag{3.3}$$

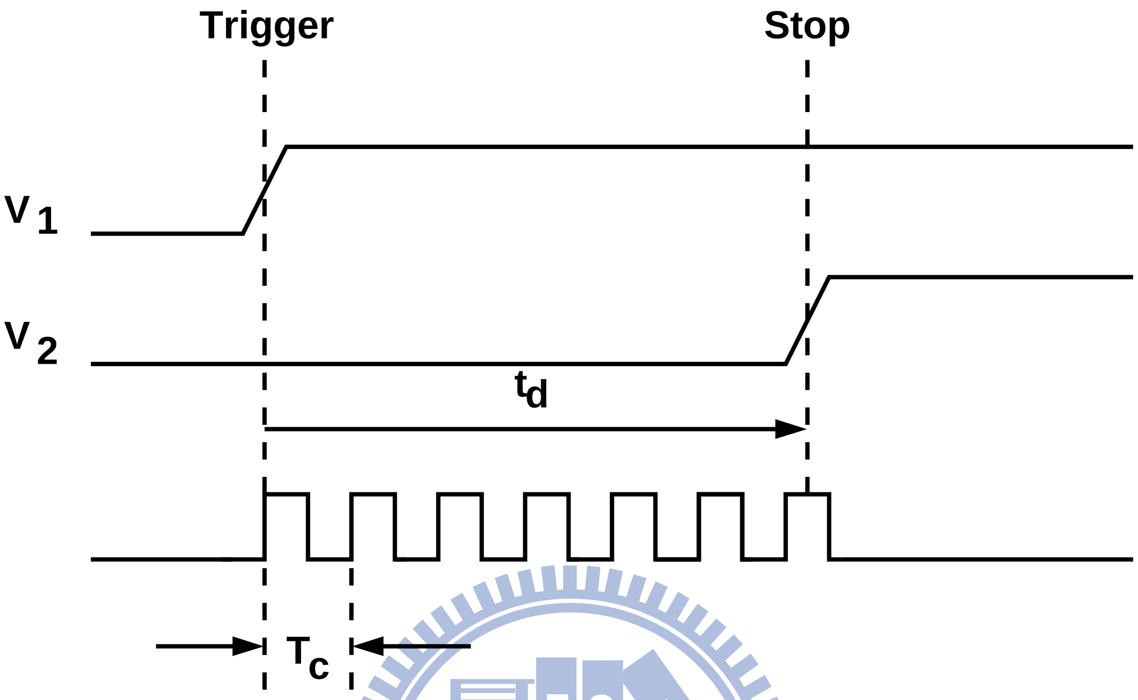


Figure 3.2: A counter-based TDC timing diagram of conversion.

This technique is unsuitable for measuring aperiodic signals with short time intervals in the order of picoseconds, such as the clock jitter.

3.3 Time-to-Amplitude Method

Another commonly used method is based on time-to-amplitude conversion [38, 39, 40]. As shown in Figure 3.3, a capacitor is first charged or discharged by a fixed current for the time interval to be measured, then an ADC is used to digitize the voltage on the capacitor. The capacitor voltage is reset to zero between measurements. The voltage on the capacitor after it stops charging is

$$V_c = \frac{I_c}{C} \times t_d \tag{3.4}$$

For a short t_d, a high resolution ADC is required to digitize the voltage on the capacitor and is difficult to implement. To simplify the design, a dual-slope ADC is often used together with the time-to-amplitude method. The dual-slope technique first charged a capacitor with fixed current, I_c, for the time interval to be measured then a smaller current,

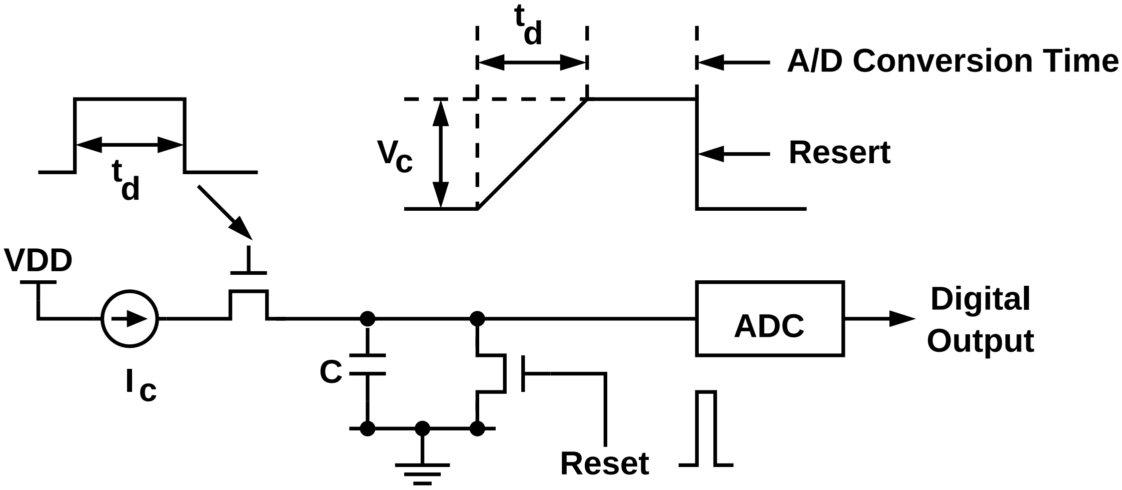


Figure 3.3: A time-to-amplitude TDC timing diagram of conversion.

I_d, is used to discharge the capacitor to zero. As shown in Figure 3.4, the relation between the time interval to be measured, t_d, and the time interval for discharging the capacitor to zero, t_m, is

$$t_d = \frac{I_d}{I_c} \times t_m \tag{3.5}$$

If $I_c = 100I_d$, t_m is equal to $100 \times t_d$. Thus, we can measure t_m instead of t_d to obtain a better resolution. However, if t_d is small, the switching noise such as clock feed through and charge injection will dominant the resolution.

3.4 Tapped Delay Line TDC

In a tapped delay line TDC as shown in Figure 3.5a, V_1 signal is passing through a delay line and each delay buffer produces a delay equal to τ_1. The output of each delay buffer is connected to the data input of a flip-flop. All the flip-flops are triggered at the raising edge of V_2 signal and the TDC's output m is generated by summing the digital outputs from all flip-flops. Like a flash ADC, the adder can be replaced by an edge detector to perform the thermometer code to binary code conversion. As shown in Figure 3.5b, the timing interval to be measures is

$$t_d \approx m \times \tau_1 \tag{3.6}$$

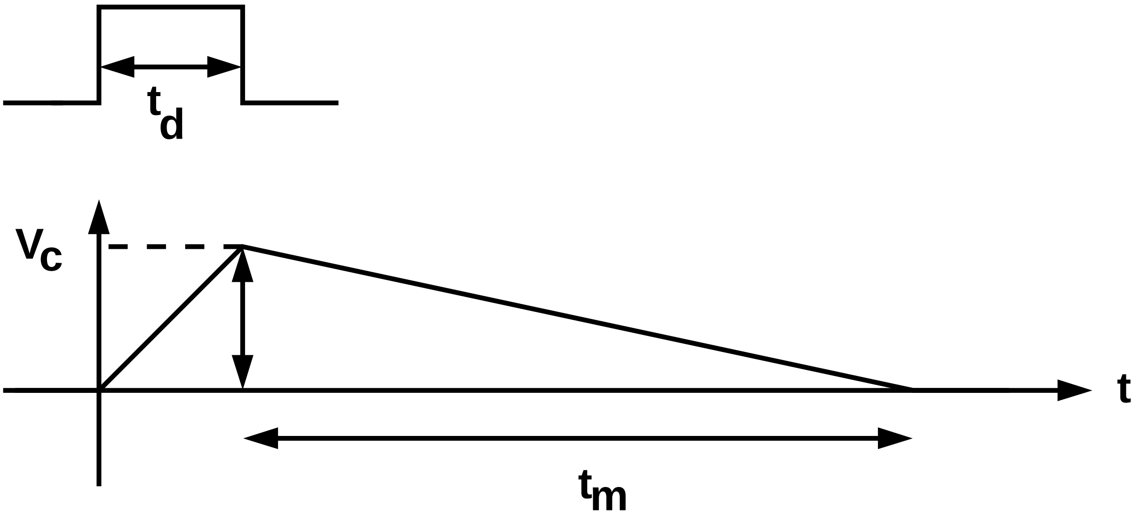


Figure 3.4: A dual-slope TDC timing diagram of conversion.

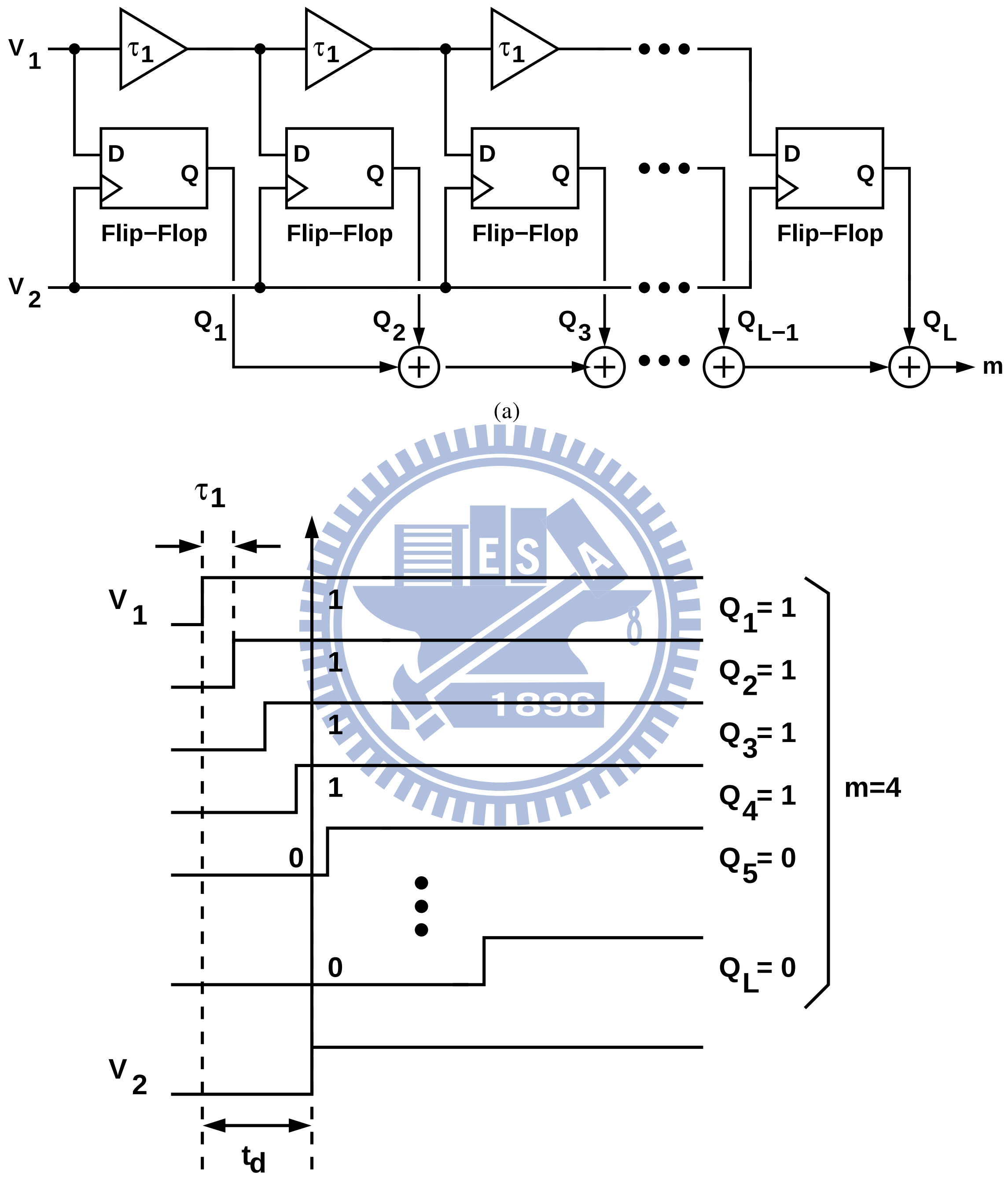
and m is equal to 4 in Figure 3.5b. To ensure that τ_1 is known accurately, the delay chain is often controlled by a delay-locked loop (DLL) [31, 41, 42] or a phase-locked loop (PLL) [33, 43].

The resolution of the tapped delay line TDC is determined by the delay of the delay element τ_1, which is limited to a gate delay. To provide a finer resolution, a vernier tapped delay line technique is used [31, 43, 44]. In the vernier tapped delay line technique, one tapped delay line drives the flip-flop clock inputs, while the other tapped delay line drives the flip-flop data inputs as shown in Figure 3.6a. The clock tap delay is slightly longer (or shorter) than the data tap delay. The vernier-based TDC is equivalent to a delay line TDC shown in Figure 3.6b, the effective tap delay is then the difference between the clock and data tap delays, i.e.

$$t_d \approx m \times (\tau_1 - \tau_2) \tag{3.7}$$

Therefore, resolutions better than a gate delay can be achieved.

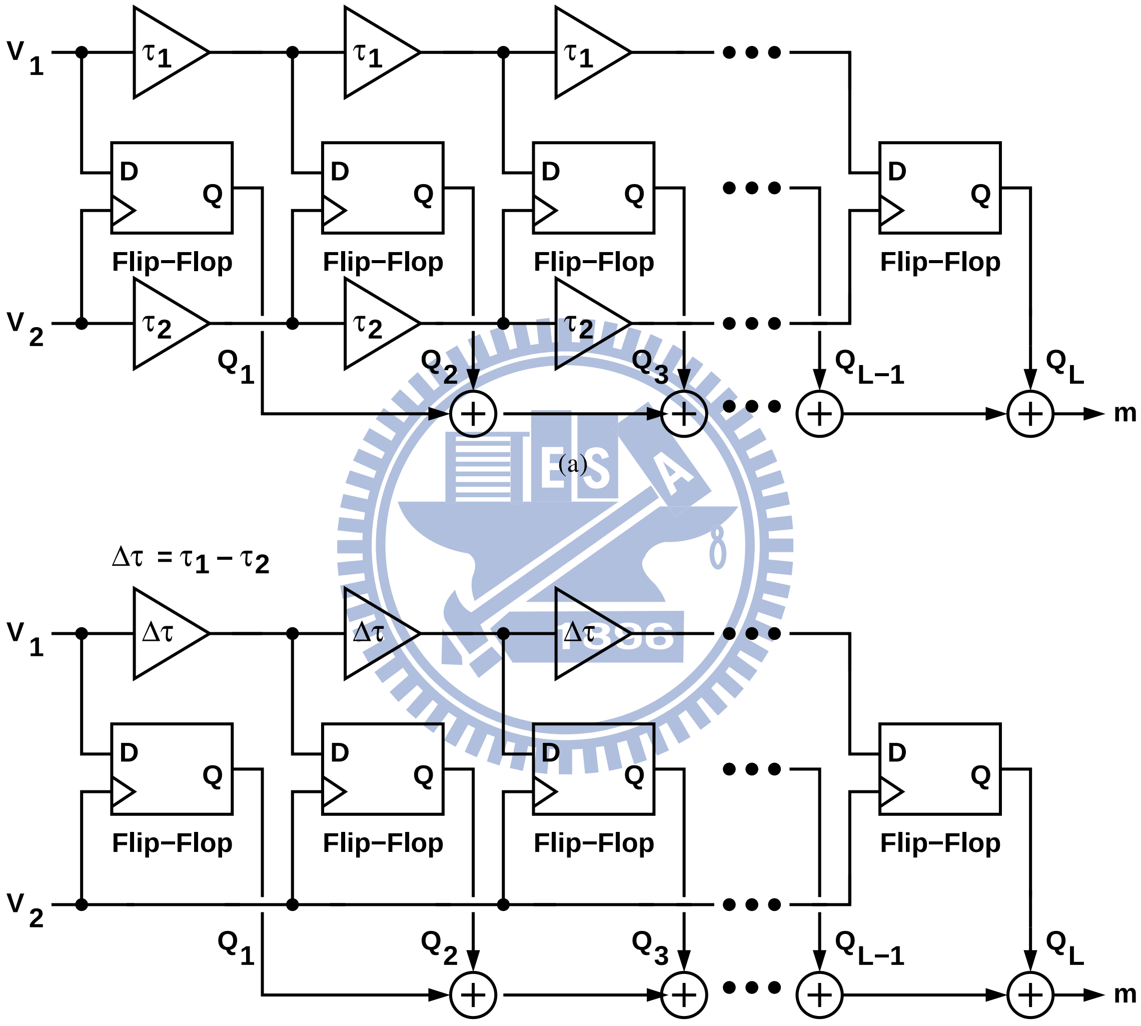
The resolution is sensitive to the gate delays, thus timing calibration of the delay chain is necessary [44, 45, 46, 47]. Even with appropriate calibration, this method still suffered from the noise induced by the delay line itself, and the error is accumulated along the delay line.



(a)

(b)

Figure 3.5: (a) TDC utilizing a tapped delay-line. (b) Timing diagram of a tapped delay-line TDC.



(a)

(b)

Figure 3.6: (a) A vernier delay-line TDC. (b) An equivalent circuit model of a vernier delay-line TDC containing a single delay line, and $\Delta\tau = \tau_1 - \tau_2$.

3.5 Stochastic TDC

The TDC used in our design is based on the stochastic TDC architecture [28, 48]. It does not use any precision circuit, and can be easily realized in a standard CMOS VLSI technology.

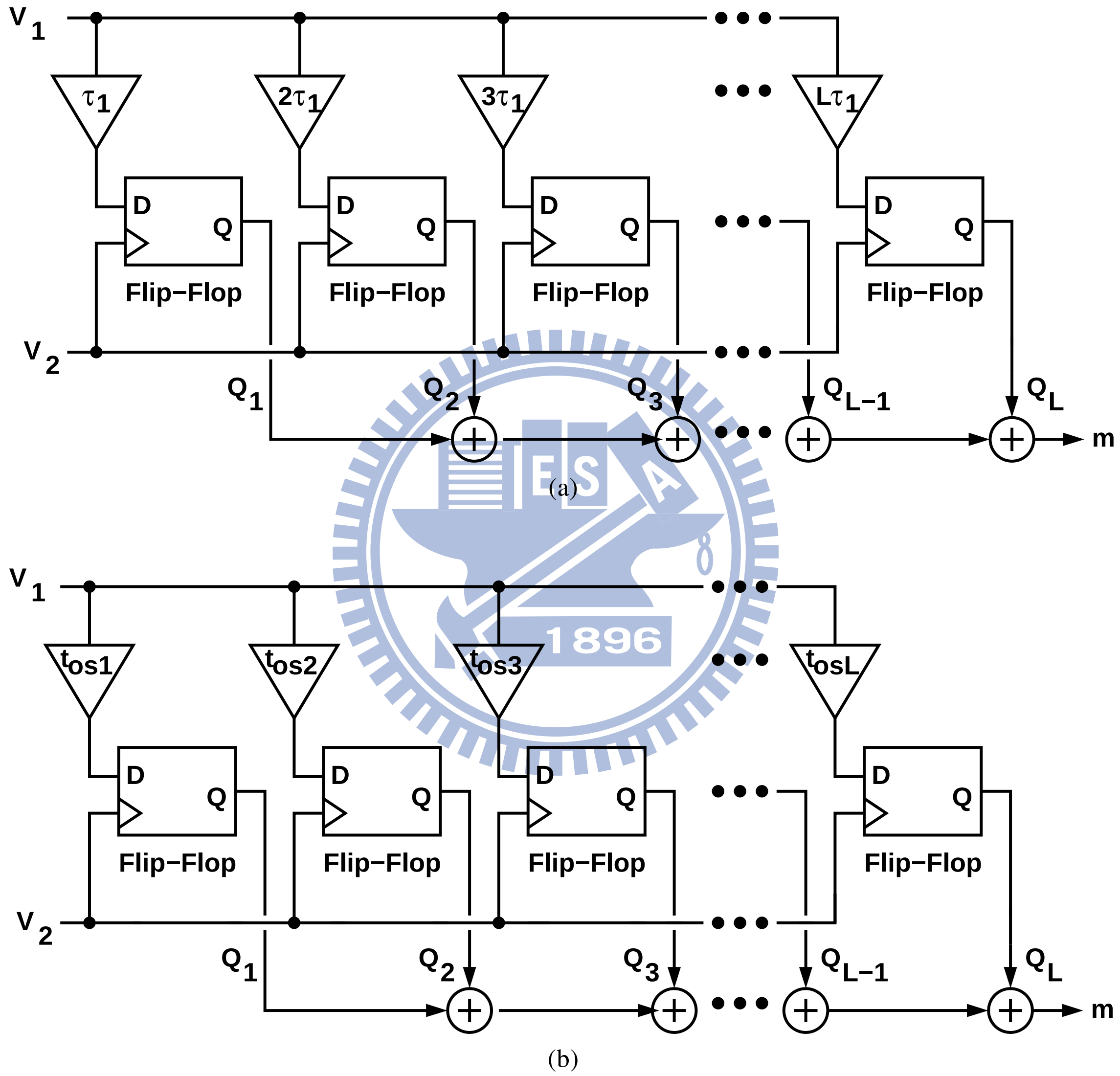
To understand the stochastic TDC in an easy way, Figure 3.7 shows the relationship between a stochastic TDC and a delay-line TDC. Each delay cell in the delay-line is replaced by a timing offset t_{osn}. The timing offset originates from the mismatch of the flip-flop itself. Thus, a stochastic TDC conducts the time-to-digital conversion by exploring the statistics of a group of flip-flops or timing comparators (TCMPs). Like a flash ADC, it can easily complete the conversion in one clock cycle. It can provide the conversion for every clock cycle continuously. Furthermore, it can improve the conversion resolution simply by adding more TCMPs. Figure 3.8 shows a TCMP example [28]. It compares the rising edges of two clocks, V_1 and V_2. Ideally, its output is a digital 1 if the timing difference $t_d > 0$. If $t_d < 0$, the output is a digital 0. However, a practical TCMP exhibits an offset, t_{os}. The offset is mainly caused by devices mismatches and interconnect mismatches. The TCMP now yields an output of digital 1 only if $t_d > t_{os}$; otherwise, the output is a digital 0.

Figure 3.9 shows the architecture of a stochastic TDC. It contains L TCMPs. Each TCMP detects the polarity of $(t_d - t_{os})$ and has its own t_{os} offset. For every clock cycle, the TDC's output, m, is generated by summing the digital outputs from all TCMPs. Thus, m is the number of TCMPs with a digital 1 output. Figure 3.10 illustrates the probability density function (pdf) of t_{os} of a TCMP and the TDC transfer function. From the central limit theorem, the pdf of t_{os} is approximately a normal distribution, $G(t_{os})$, which is

$$G(t_{os}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-t_{os}^2/(2\sigma^2)} \tag{3.8}$$

where σ is the standard deviation of t_{os}. The averaged t_d-to-m TDC transfer function can be obtained by integrating over this pdf, i.e.,

$$m = L \times \int_{-\infty}^{t_d} G(t_{os})dt_{os} \tag{3.9}$$



(a)

(b)

Figure 3.7: Relationship between a delay line TDC and a stochastic TDC. (a) Alternative representation of delay line TDC. (b) A stochastic TDC by utilizing the offset.

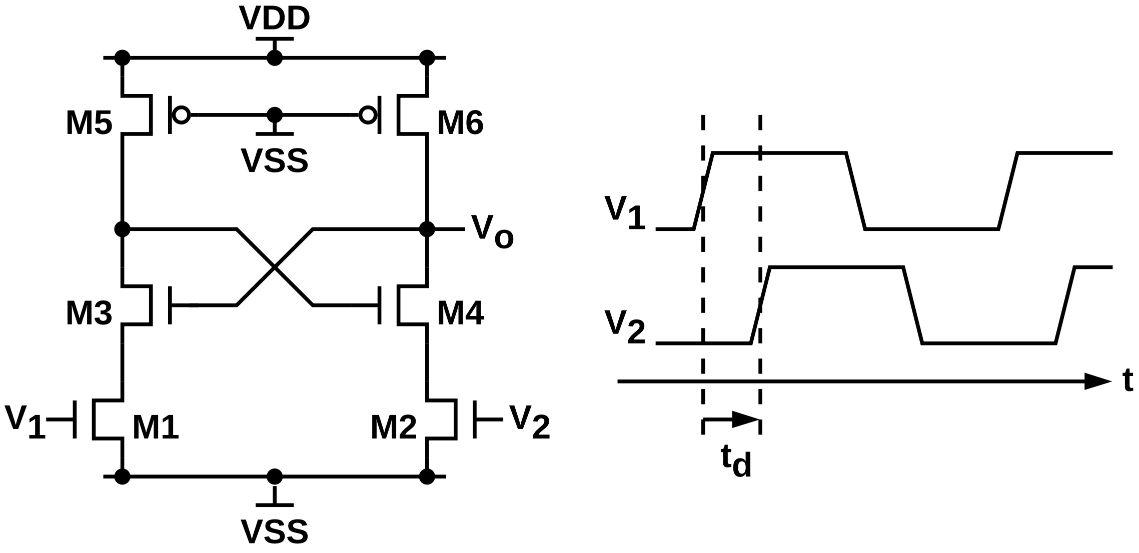


Figure 3.8: Timing comparator (TCMP).

In practice, m can only be an integer between 0 and L. This TDC has its input range proportional to σ. Its resolution is a function of L.

A TDC converts an input t_d into a digital code m. The input range of a TDC is divided into different segments. Each segment is mapped to a different m code. Figure 3.11 shows an example of time-to-digital conversion for $L = 6$. The t_{os1} to t_{os6} are the timing offset of the six TCMPs respectively. Note that t_{osn} is not necessary the timing offset of the n-th TCMP in Figure 3.9. If $m = 2$ at the k-th sampling, i.e., $D_t[k] = 2$, there are two TCMPs with a digital 1 output. The clock jitter at the k-th sample is $\Delta t[k]$ and $t_{os2} < \Delta t[k] < t_{os3}$. Thus, as long as the timing offset of all the TCMPs are known, the TDC can be used to measure the timing difference between two signals. However, similar to an ADC, a TDC also introduces quantization noise, $\Delta t_q[k]$ as shown in Figure 3.12. For example, as long as $t_{os2} < t_d < t_{os3}$, the output of the TDC is always $m = 2$. The power of the quantization error when $m = 2$ is

$$\Delta t_{q,2}^2 = \frac{\Delta t_{s,2}^2}{12} \tag{3.10}$$

and $\Delta t_{s,2}$ is the step size for $m = 2$. Thus, the quantization error for a giving TDC can be derived as

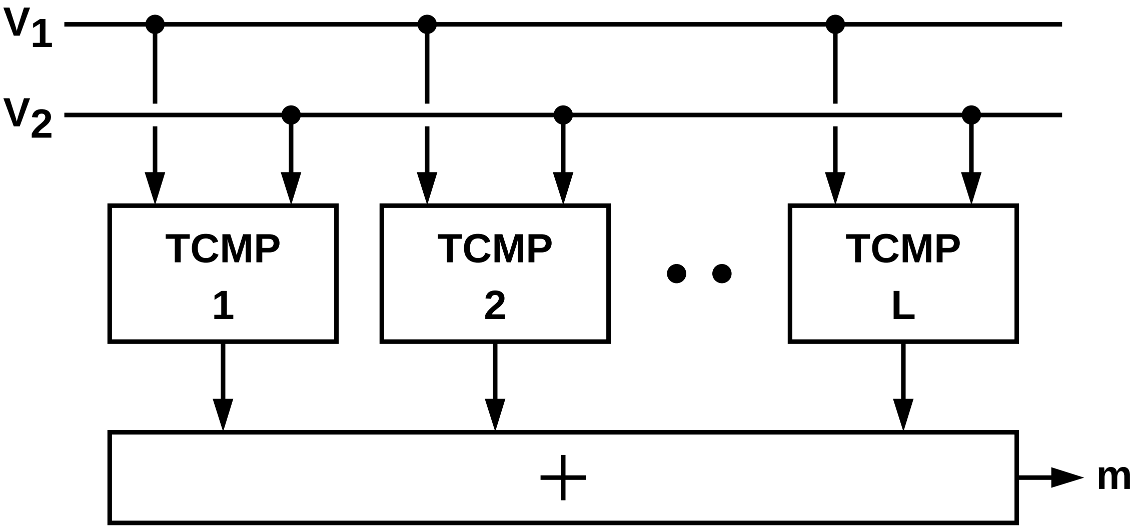


Figure 3.9: Stochastic time-to-digital converter (TDC).

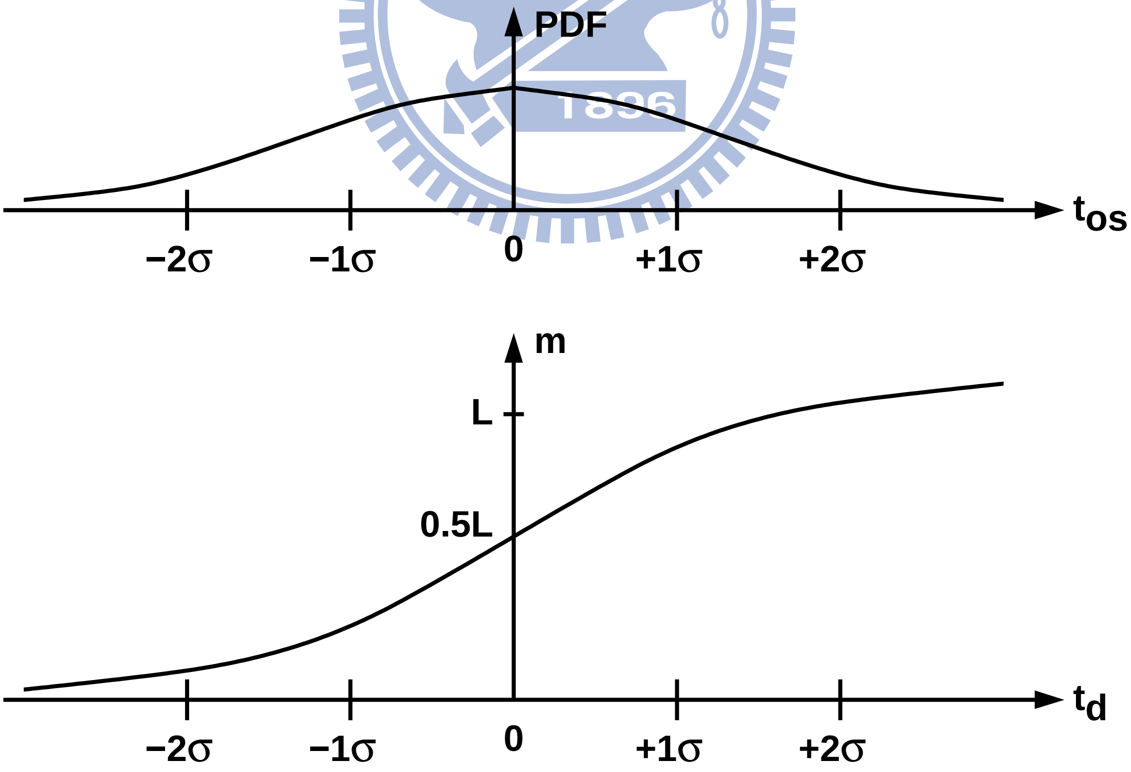


Figure 3.10: TDC Transfer function and t_{os} probability density function (pdf).

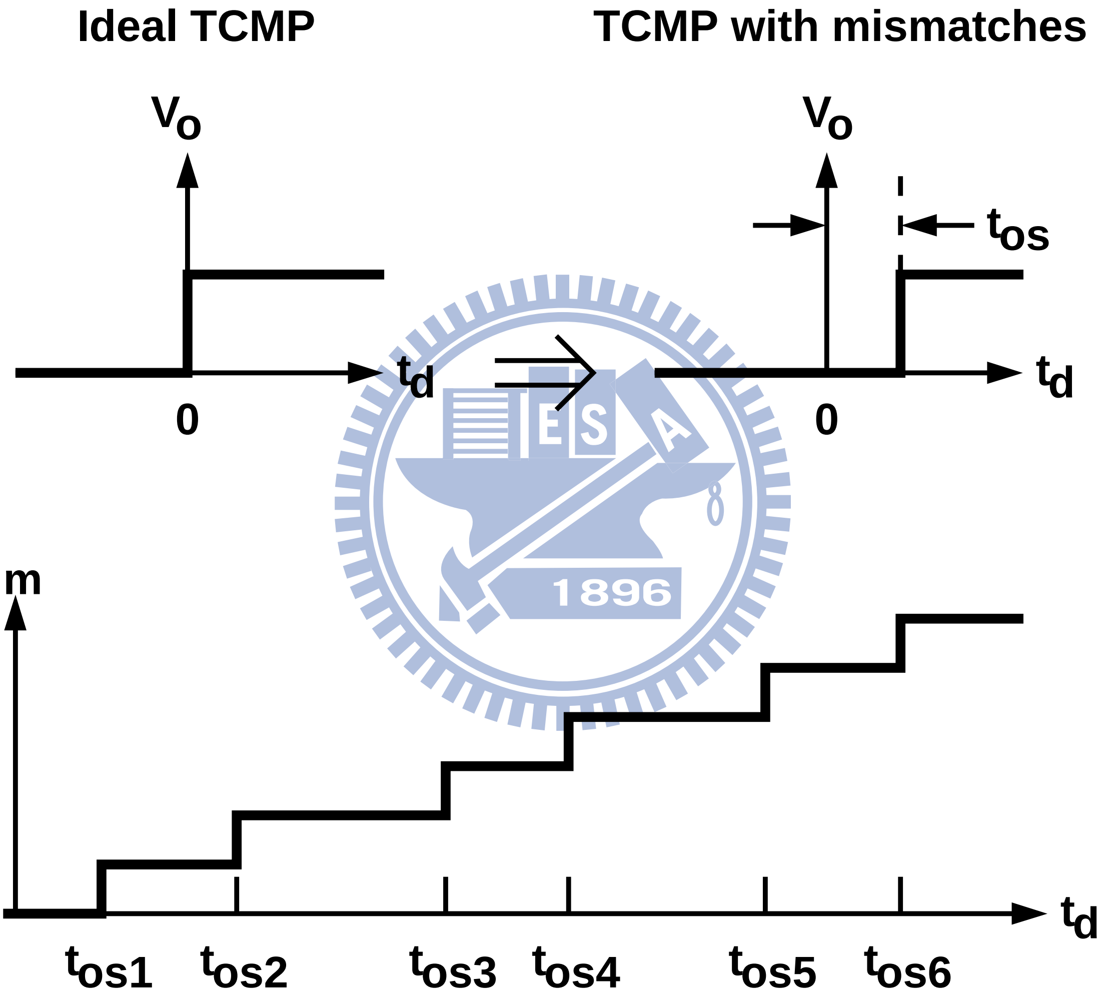


Figure 3.11: An example of time-to-digital conversion for $L = 6$.

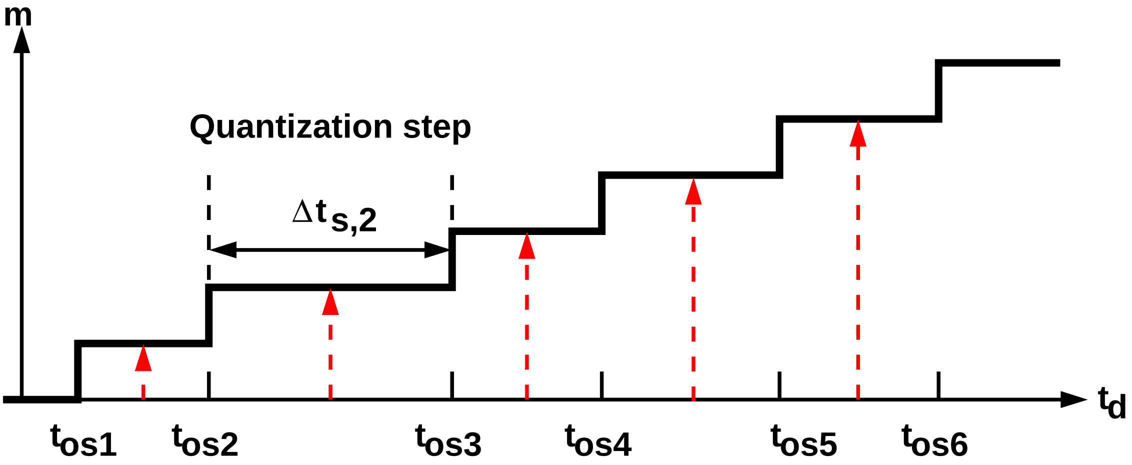


Figure 3.12: Quantization error of a TDC.

$$\Delta t_{q,rms} = \sqrt{\sum_{m=1}^{L-1} \frac{\Delta t_{s,m}^2}{12} \times P(m)} \tag{3.11}$$

where $P(m)$ is the probability for the TDC's output is a digital code of m.

The quantization step size, $\Delta t_{s,m}$, is the range of the t_d segment mapped to the digital code m. A larger $\Delta t_{s,m}$ results in a larger $\Delta t_{q,m}$. The quantization error limits the resolution of the TDC.

A Monte Carlo simulation with a timing offset searching loops similar to [49] is applied to estimate the standard deviation of t_{os}. The search loop for timing offset is composed of the TCMP netlist and a binary search block written by Verilog-A behavioral circuit element. The algorithm of the binary search is illustrated in Figure 3.13. The TCMP is the one shown in Figure 3.8. Initially, t_d is set to 0, which means V_1 is phase aligned with V_2. For a giving searching range $\pm T_r$, if V_o is a digital 1 after the transient analysis, t_d is set to $t_d = t_d - T_r/2$, and repeat the same transient analysis. Otherwise, if V_o is a digital 0 after the transient analysis, t_d is set to $t_d = t_d + T_r/2$.

Figure 3.14 shows the t_{os} statistics of a TCMP realized in a 90 nm CMOS technology by using the binary search methodology. The data were collected from 1000 Monte Carlo circuit simulations. The standard deviation of t_{os} is $\sigma = 6.36$ ps.

Assume an ideal TDC which has identical $\Delta t_{s,m}$ for all m. Similar to Equation (2.7), an ideal TDC that has a uniform quantization step size of $\Delta t_s = 1$ ps gives $\Delta t_{q,rms} =$

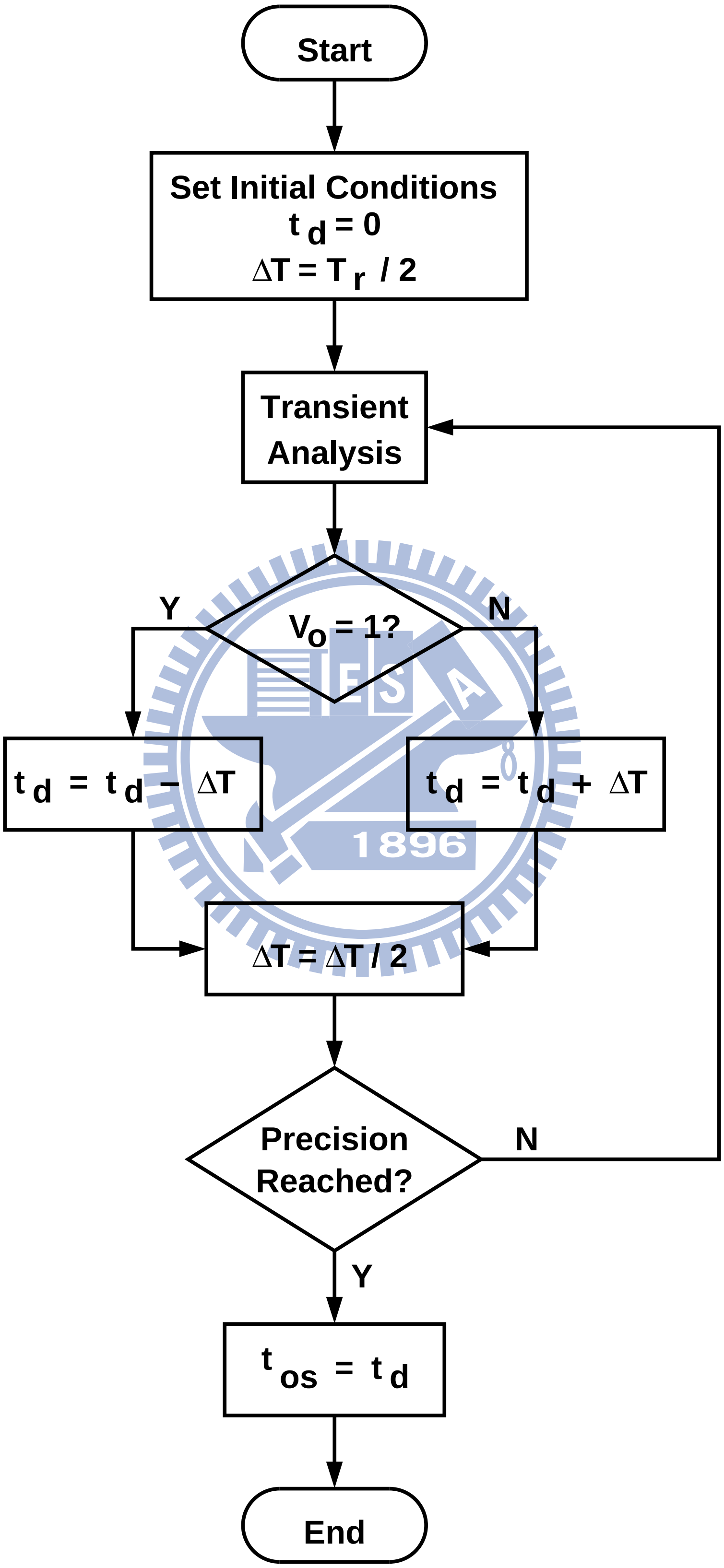


Figure 3.13: Binary search process for timing offset extraction.

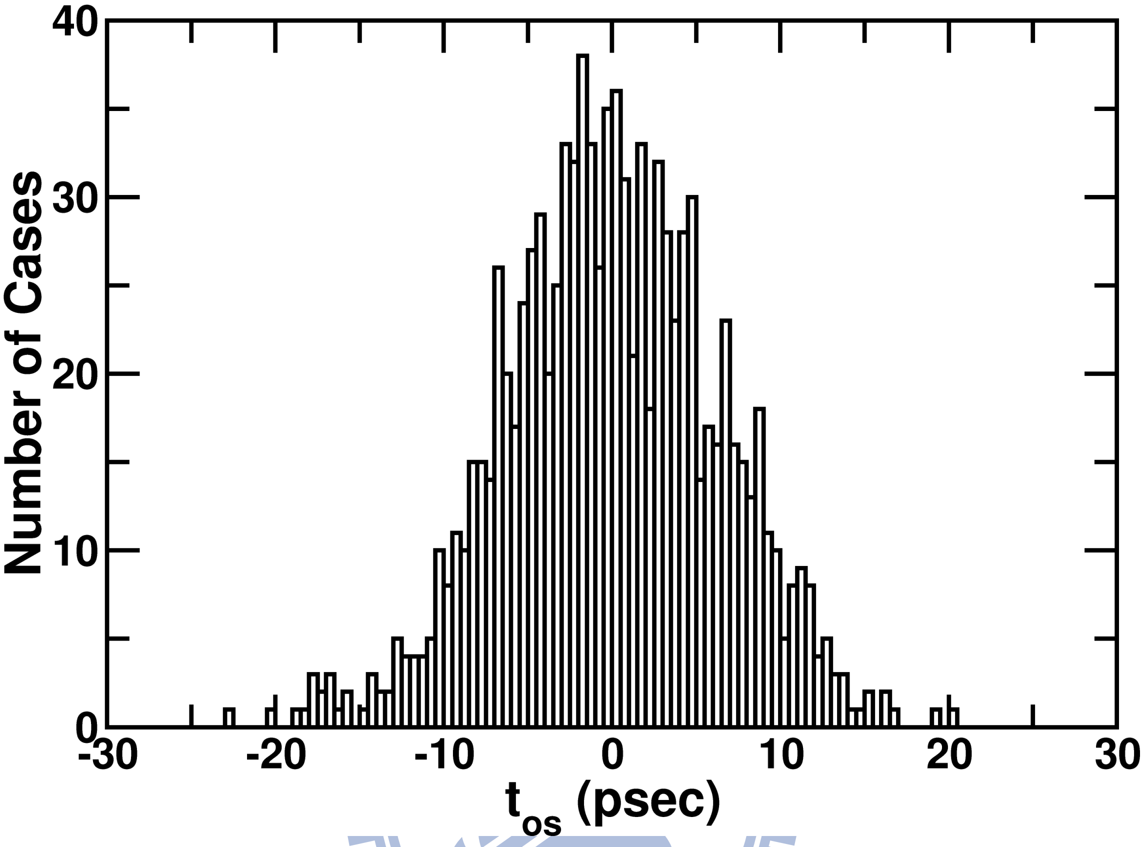


Figure 3.14: Histogram of timing comparator offset, t_{os}, from 1000 Monte Carlo simulations.

$1/\sqrt{12}$ ps. For a stochastic TDC, the $\Delta t_{s,m}$ may vary for different m and $\Delta t_{q,rms}$ can be derived using Equation (3.11). However, an explicit expression for its $\Delta t_{q,rms}$ is difficult to obtain. Brute-force simulations were used instead. Figure 3.15 shows the $\Delta t_{q,rms}$ achieved by 99% of the stochastic TDCs with a given L. The t_{os} of each TCMP in the TDC is randomly chosen. The statistics of t_{os} is the normal distribution of Equation (3.8). The data are obtained from simulations of 100,000 different TDC cases for each L. The input t_d is assumed to have a normal distribution with a mean of zero and a rms of Δt_{rms}. Note that, depending on applications, the input t_d may have different statistical distributions. Consider a stochastic TDC whose internal TCMPs exhibit a t_{os} with standard deviation $\sigma = 6.36$ ps. In order to achieve the same $\Delta t_{q,rms} = 1/\sqrt{12}$ ps, i.e. $\Delta t_{q,rms}/\sigma = 0.045$, this TDC needs the number of TCMPs $L > 70$ if the RMS of the input t_d is $\Delta t_{rms} = (1/3)\sigma$. The TDC needs $L > 94$ if $\Delta t_{rms} = (2/3)\sigma$.

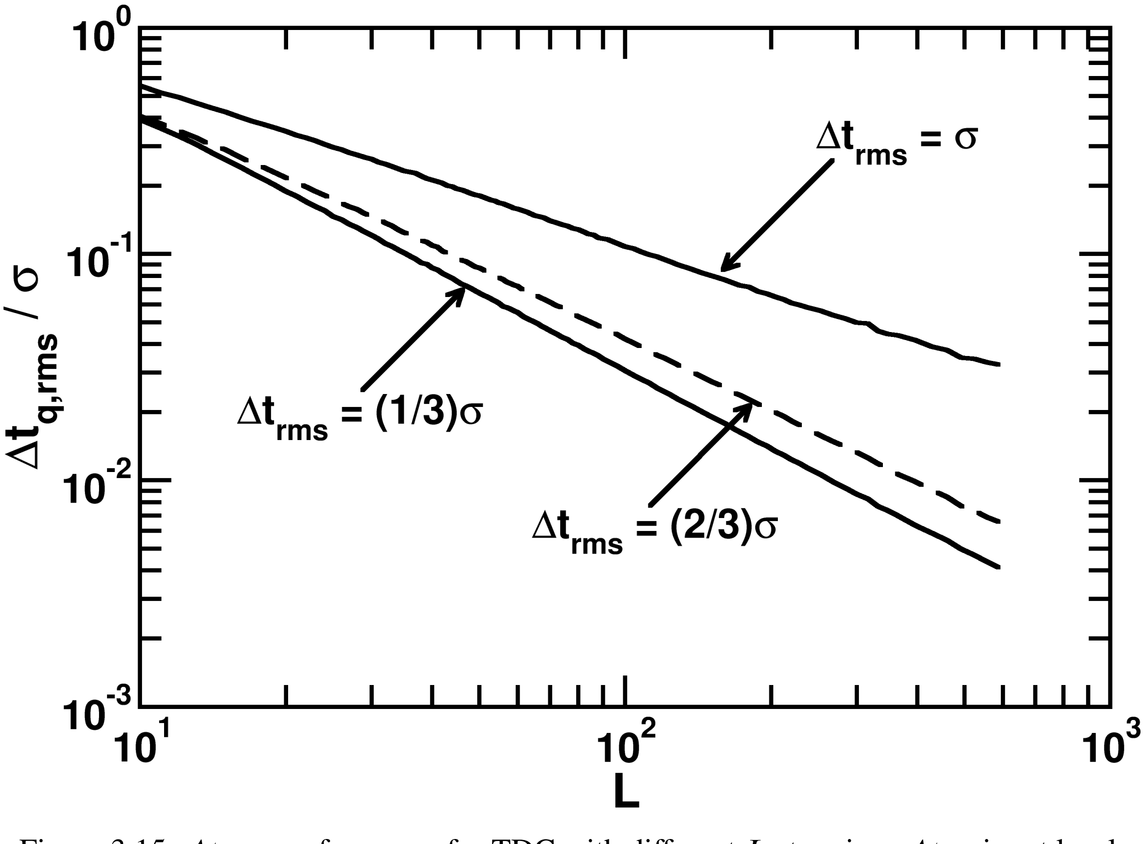


Figure 3.15: $\Delta t_{q,rms}$ performance for TDC with different L at various Δt_{rms} input level.

The t_{os} of a TCMP is very sensitive to process, supply voltage, and temperature (PVT) variations. It is also sensitive to the waveforms of the two input clocks, V_1 and V_2. Therefore, it is necessary to calibrate a stochastic TDC so that an accurate estimation of t_d can be extracted from its digital output. Techniques to calibrate the stochastic TDC in the background are proposed in the following two chapters.

In the following discussion, the rms of the clock jitter, the TDC quantization noise and its rms are normalized as

$$\epsilon_{rms} = \frac{\Delta t_{rms}}{T_s} \quad \epsilon_q[k] = \frac{\Delta t_q[k]}{T_s} \quad \epsilon_{q,rms} = \frac{\Delta t_{q,rms}}{T_s} \tag{3.12}$$

where T_s is the nominal clock period. If $\Delta t_{q,rms} = 1.25/\sqrt{12}$ ps and $T_s = 12.5$ ns, $\epsilon_{q,rms} = (1/\sqrt{12}) \times 10^{-4}$.

3.6 Summary

The proposed jitter compensation scheme requires a TDC to quantize the clock jitter so that the sampling error can be compensated in the digital domain. Various types of TDC are described in this chapter. To measure the clock jitter accurate at every clock cycle, a stochastic TDC is adopted. The stochastic TDC performs the time-to-digital conversion by exploring the statistics of a group of TCMPs. The conversion resolution can be improved simply by adding more TCMPs.

Although a resolution better than 1 ps can be achieved easily by a stochastic TDC, the TDC is sensitive to the PVT variations. It requires calibration for accurate jitter measurement. A novel TDC background calibration will be introduced in the following chapters.

Chapter 4

Jitter Compensation with Clean External Clock

4.1 Introduction

A Nyquist-rate ADC first samples a continuous-time analog signal, and then quantizes the sampled data into a discrete-time digital sequence. A periodic clock is required to provide a reference for the sampling time. If the sampling clock exhibits jitter, sampling error occurs during the sampling process [14, 15]. Excess clock jitter can degrade the SNR performance of an ADC. A perfect ADC will yield poor SNR performance just because the clock edges are moving around.

To relax the clock jitter requirement, a jitter compensation scheme had been proposed in Chapter 2. The clock jitter is measured and digitized by a jitter-to-digital converter (JDC). The JDC is composed of a stochastic TDC and a calibration circuit. The measured jitter information is used to compensate the sampling error of an ADC by a jitter compensation filter (JCF). Two different system scenarios will be covered in this thesis: 1) an ADC with a clean external clock and 2) an ADC with an external clock as the main jitter source. The first scenario will be investigated in this chapter and the second scenario will be discussed in Chapter 5. The TDC background calibration techniques for both jitter compensation scenarios are also proposed.

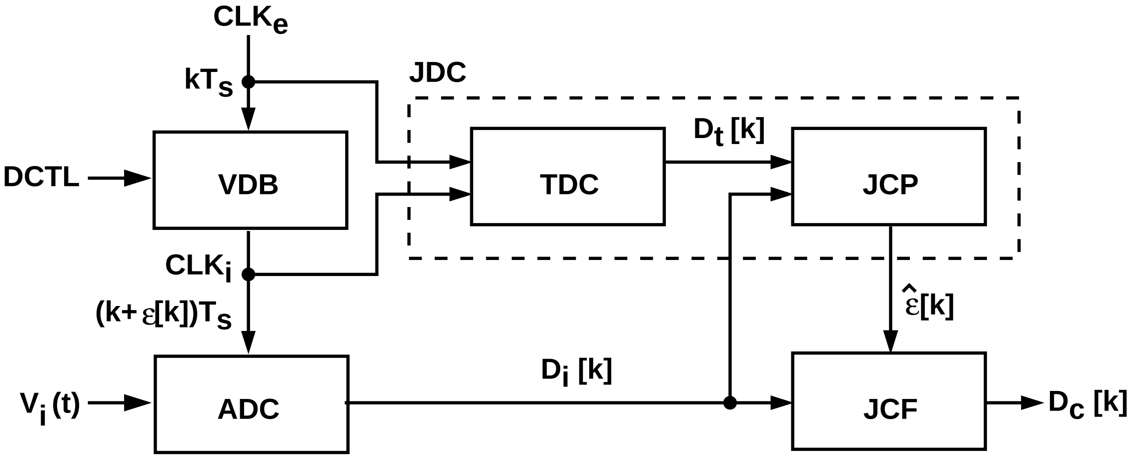


Figure 4.1: A jitter compensation scheme with clean external clock.

4.2 Jitter Compensation with Clean External Clock

Figure 4.1 shows an ADC system with jitter compensation when a ideal clock source is available. In this scenario, the external clock, CLK$_e$, is assumed to be clean and without jitter. It defines the sampling instants, denoted as kT_s. A variable delay buffer (VDB) receives the CLK$_e$ clock and generates the internal CLK$_i$ clock that drives the ADC. The delay of the VDB is controlled by a delay-control (DCTL) signal such that CLK$_i$ is phase-aligned with CLK$_e$. The DCTL signal can be generated from a delay-locked loop (DLL) [31, 41, 42] or a phase-locked loop (PLL) [33, 43]. It is assumed that the VDB adds jitter to the CLK$_i$ clock, changing the sampling instants to $(k + \epsilon[k])T_s$. The normalized absolute clock jitter $\epsilon[k]$ is measured and digitized by a JDC. The JDC is composed of a stochastic TDC and a jitter calibration processor (JCP1). A JCF as shown in Figure 4.2 uses the measured jitter data $\hat{\epsilon}[k]$ to convert the $D_i[k]$ signal from the ADC into the corrected signal $D_c[k]$. The detail of the jitter compensation had been demonstrated in Chapter 2. The JCF eliminates the sampling error in $D_i[k]$ caused by the clock jitter $\epsilon[k]$.

The output of the TDC, $D_t[k]$, is an integer between 0 and L where L is the number of TCMPs in the TDC as shown in Figure 3.9. Although the output of the TDC, $D_t[k]$, represents the CLK$_i$ absolute jitter at the k-th clock cycle, the exact value of the jitter it represents cannot be determined by the TDC alone. Therefore, calibrations are required to ensure an accurate measurement result. The most commonly used method to calibrate

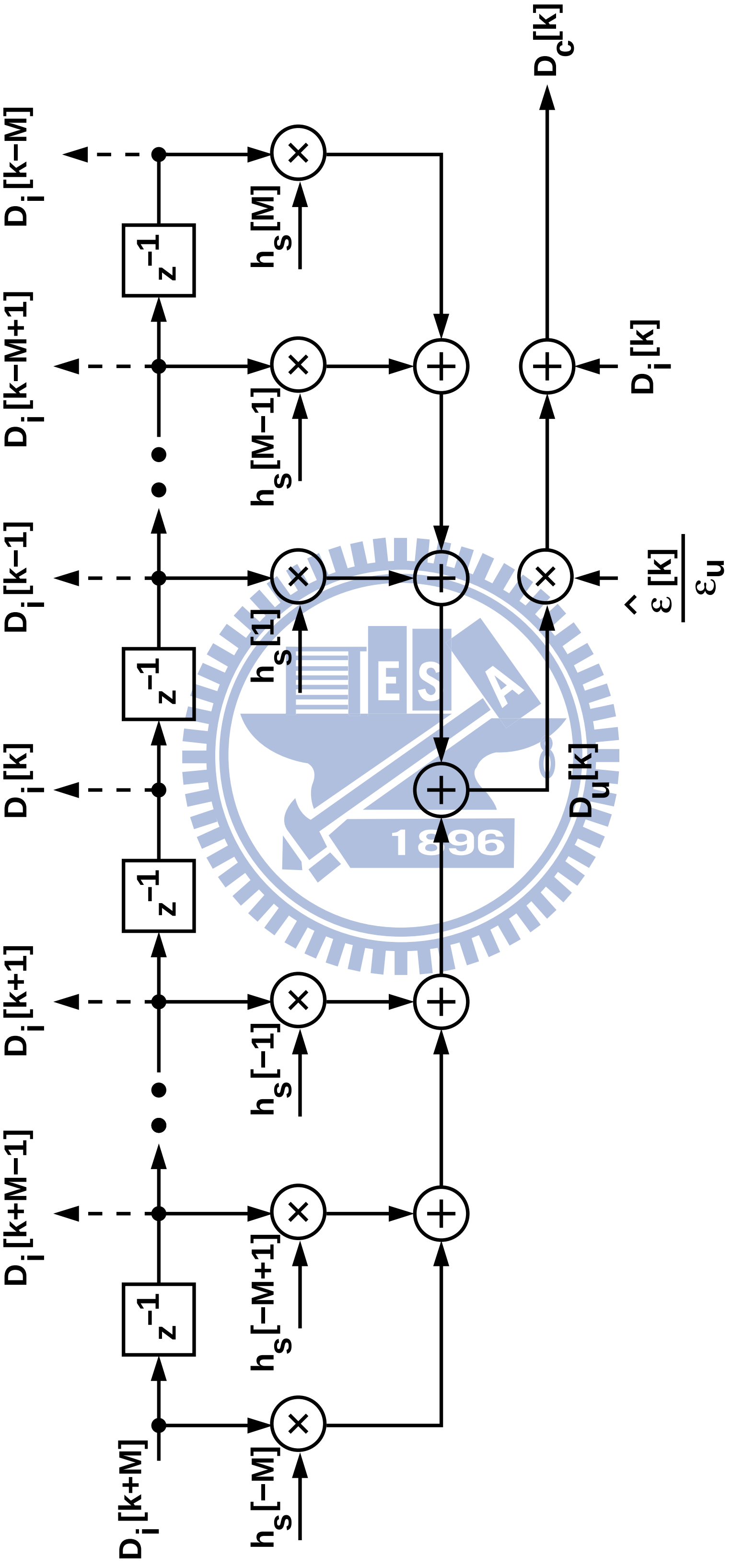


Figure 4.2: A simplified jitter compensation filter with $2M + 1$ taps.

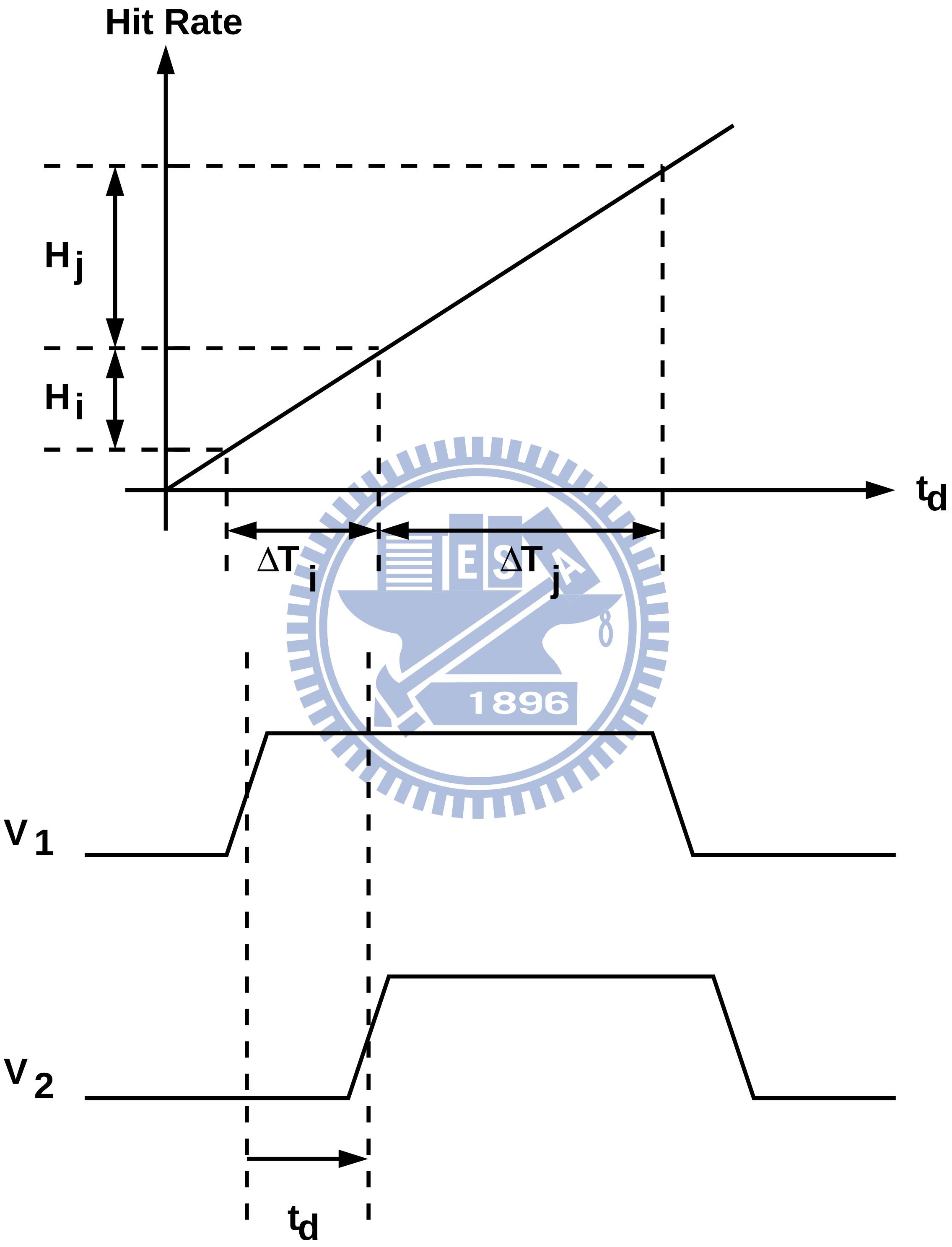


Figure 4.3: Code-density test.

the TDC is the foreground code-density test [28, 44, 50, 51, 52]. It is well-known that the probability distribution of the timing difference, t_d, of two periodic signals with uncorrelated frequencies is a uniform distribution. As shown in Figure 4.3, the probability that t_d falls in the interval Δt_j is proportional to the width of Δt_j itself. Therefore, to calibrate the TDC, two periodic signals with uncorrelated frequencies are input to the TDC. For a stochastic TDC whose transfer curve is the one shown in Figure 3.11, the probability that the TDC's output code, $D_t[k] = m$, is proportional to the its quantization step. For example $D_t[k] = 5$ will occur with a frequency of $(t_{os5} - t_{os4}) \times f_1 f_2$ where f_1 and f_2 are the frequency of the calibration signal V_1 and V_2 respectively. Thus, the TDC can be calibrated by monitoring the hit rate of each code.

However, the transfer function of a stochastic TDC is sensitive to the waveforms of both V_1 and V_2. The calibration results will be incorrect if the waveforms of V_1 and V_2 is different from CLK$_e$ and CLK$_i$. The TDC transfer function is also sensitive to process, voltage, and temperature (PVT) variations. Background calibration is required to convert the digital code $D_t[k]$ into the corresponding jitter information accurately. We propose using the ADC to calibrate the TDC in the background. The jitter calibration processor performs self-calibration which will be discussed in the next section.

4.3 TDC Background Calibration Principle

The principle of the TDC background calibration is described as follows. As shown in Figure 4.4, assuming that a sampling error occurs at the k-th sampling instant and the TDC output code is $D_t[k] = m$. We define that $\epsilon(m)$ is the jitter represented by $D_t[k] = m$. Therefore, the $V_i(t)$ is sampled at $t = (k + \epsilon(m))T_s$ instead of $t = kT_s$. The magnitude at point A' is quantized as $D_i[k]$. Thus, the ADC perceives a different $\hat{V}_i(t)$ input instead of $V_i(t)$. The signal $\hat{V}_i(t)$ has a value of $D_i[k]$ at $t = kT_s$, denoted as point B. If the clock jitter $\epsilon[k]$ is known, where $\epsilon[k] = \epsilon(m)$ when $D_t[k] = m$, the correct sample can be obtained by interpolating the $\hat{V}_i(t)$ signal at $t = (k - \epsilon(m))T_s$. As derived from Chapter 2, the corrected ADC output is

$$D_c[k] = D_i[k] + \epsilon(m) \times \frac{D_u[k]}{\epsilon_u} \tag{4.1}$$

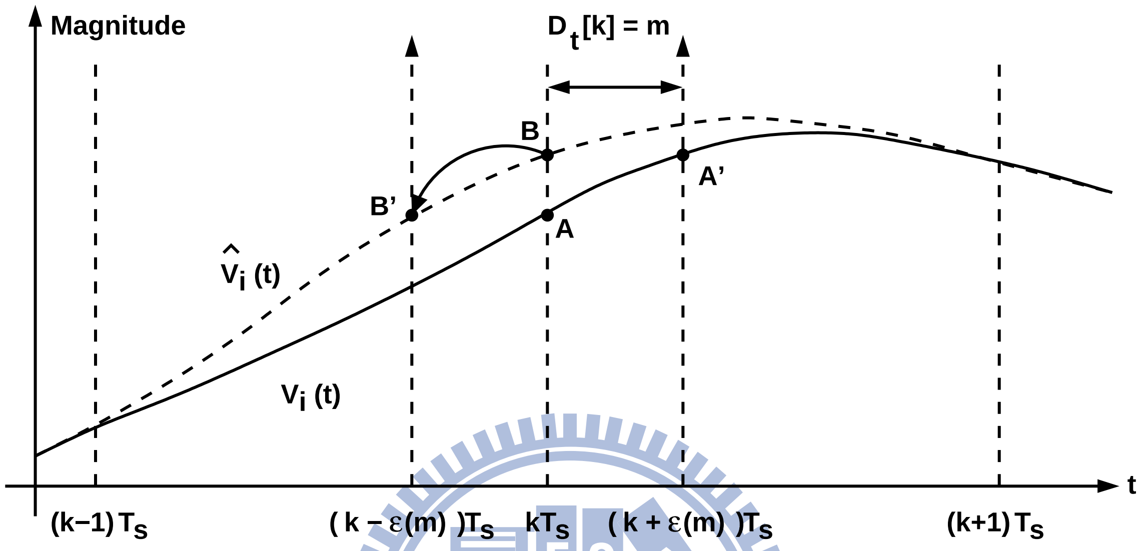


Figure 4.4: Jitter compensation at the k-th sampling and $D_t[k] = m$.

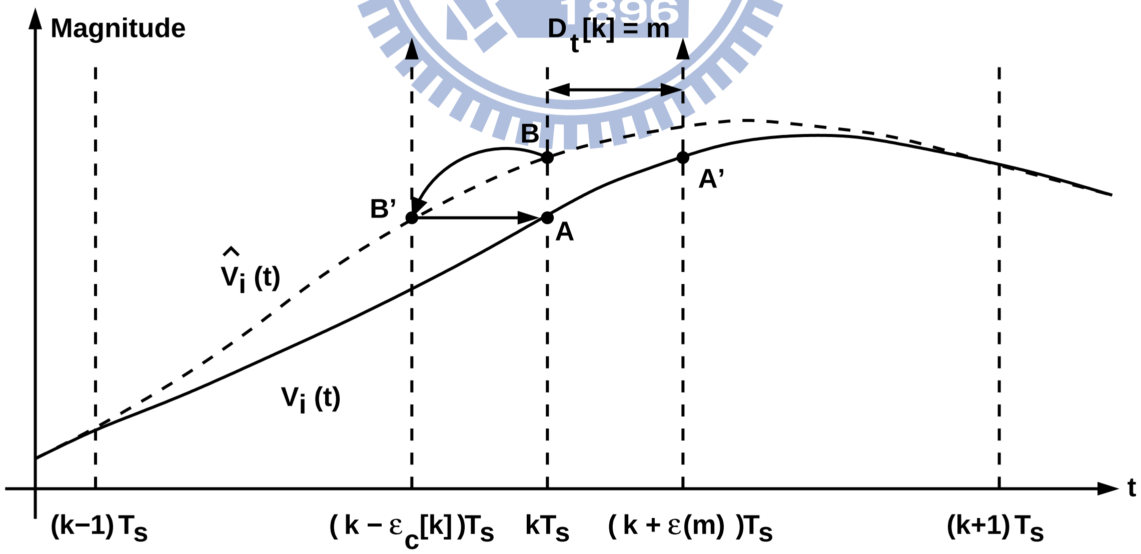


Figure 4.5: TDC background calibration principle.

where

$$D_u[k] = \sum_{n=k-M}^{k-1} \left\{ D_i[n] \times h_s[k-n] \right\} + \sum_{n=k+1}^{k+M} \left\{ D_i[n] \times h_s[k-n] \right\} \tag{4.2}$$

with

$$h_s[n] = \mathrm{sinc}(n - \epsilon_u) \tag{4.3}$$

and ϵ_u is a predefined jitter constant. Since $D_u[k]/\epsilon_u$ represents the $V_i(t)$ signal slope at the k-th sampling instant, the value of ϵ_u is not crucial. The magnitude at point B', which is equal to $D_c[k]$, is a correct estimation of $V_i(t)$ at $t = kT_s$, denoted as point A.

However, the only jitter information we have at the k-th sampling so far is the output of the TDC, $D_t[k]$. If $D_t[k] = m$ and the value of the jitter it represents is known, the correct sample can be calculated using Equation (4.1).

Equation (4.1) can be interpreted as the correct sample $D_c[k]$ is a function of the clock jitter,

$$D_c[k] = f(\epsilon[k]) \tag{4.4}$$

If it exists an inverse function of f, defined as f^{-1}, Equation (4.4) can be rewritten as

$$\epsilon(m) = f^{-1}(D_c[k]) \tag{4.5}$$

If the jitter represents by $D_t[k] = m$ is not known but the magnitude at point A, i.e. $V_i(kT_s)$, is known instead, we can use Equation (4.5) to calibrate the TDC. The calibrated result, $\epsilon_c[k]$, which is the jitter corresponded to $D_t[k]$, can be obtained as

$$\epsilon_c[k] = f^{-1}(V_i(kT_s)) \tag{4.6}$$

In other words, $\epsilon_c[k]$ is the jitter used in the JCF of Equation (4.1) to make the interpolated magnitude of $\hat{V}_i(t)$ at $t = (k - \epsilon_c[k])T_s$, denoted as point B', equal to the magnitude at point A. Therefore, if the magnitude of the ideal sample is available, the TDC can be calibrated as illustrated in Figure 4.5. The ideal sample can be obtained by a signal reconstruction filter (SRF) described in Section 4.4.

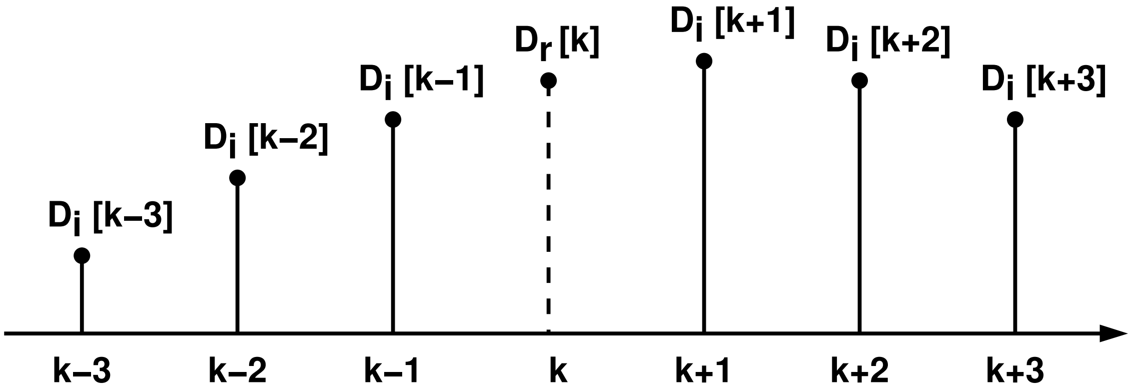


Figure 4.6: Signal reconstruction.

4.4 Signal Reconstruction Filter

Consider a band-limited analog input $V_i(t)$ and the corresponding digital ADC output $D_i[k]$. As shown in Figure 4.6, we want to reconstruct the k-th sample, $D_i[k]$, from its neighboring samples, which can be expressed as $D_i[k-n]$ and $D_i[k+n]$ for integer $n \geq 1$. The output of the reconstruction filter can be expressed as

$$D_r[k] = \sum_{n=-N}^{-1} h_r[n] \times D_i[k-n] + \sum_{n=1}^{N} h_r[n] \times D_i[k-n] \tag{4.7}$$

Since the number of the $D_i[k-n]$ samples is equal to the number of the $D_i[k+n]$ samples, from signal symmetric property, the $h_r[-n]$ filter coefficient must equal to the $h_r[n]$ filter coefficient. Thus, Equation (4.7) can be rearranged as

$$D_r[k] = \sum_{n=1}^{N} \left\{ h_r[n] \times \left(D_i[k-n] + D_i[k+n] \right) \right\} \tag{4.8}$$

Figure 4.7 shows the block diagram of the resulting SRF with $2N+1$ taps.

Neglecting the clock jitter $\epsilon[k]$, the output of the ADC can be expressed as

$$D_i[k] = \frac{1}{2\pi} \int_{-\omega_B}^{+\omega_B} V(j\omega) e^{j\omega(kT_s)} d\omega \tag{4.9}$$

where ω_B is the bandwidth of the input signal $V_i(t)$ and $V(j\omega)$ is the Fourier transform of $V_i(t)$. Combining Equation (4.8) and Equation (4.9) yields

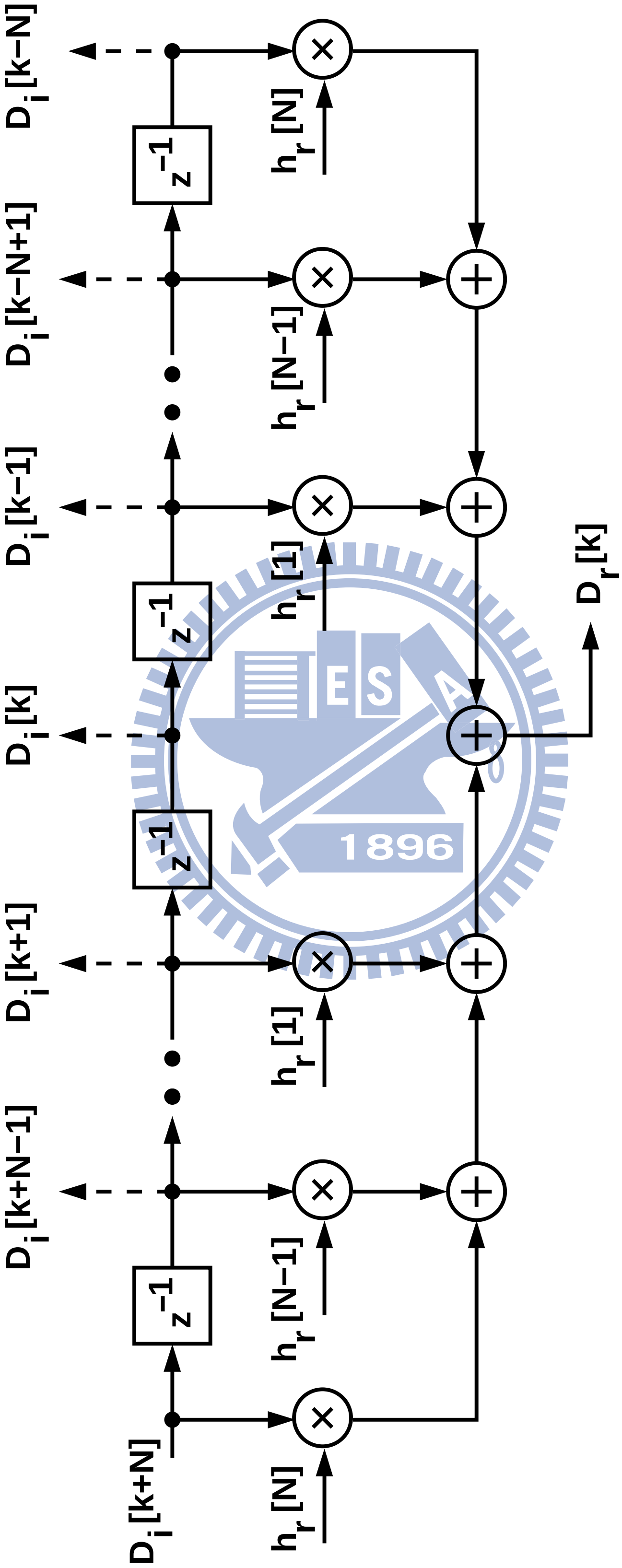


Figure 4.7: A signal reconstruction filter with $2N + 1$ taps.

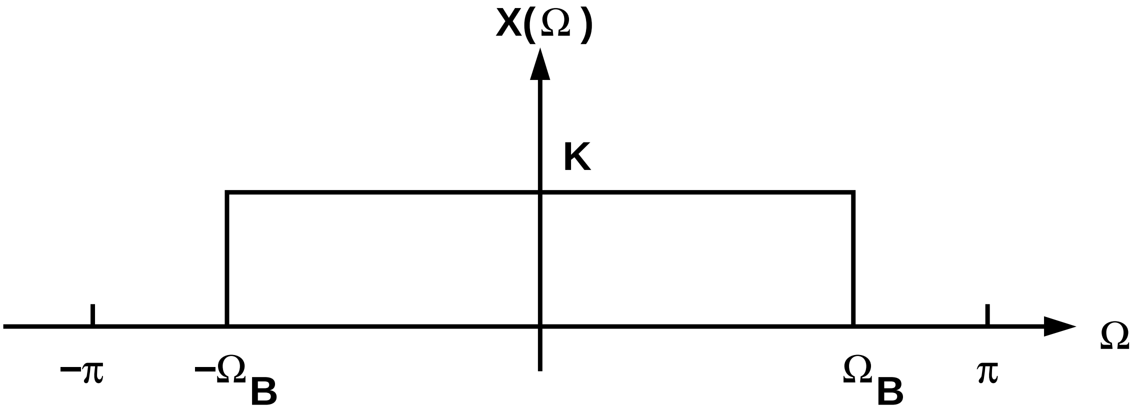


Figure 4.8: An ideal low-pass filter.

$$\begin{aligned} D_r[k] &= \sum_{n=1}^{N} \left\{ \frac{1}{2\pi} \int_{-\omega_B}^{+\omega_B} V(j\omega) e^{j\omega k T_s} \left[2h_r[n]\cos(n\omega T_s)\right] d\omega \right\} \\ &= \frac{1}{2\pi} \int_{-\omega_B}^{+\omega_B} V(j\omega) e^{j\omega k T_s} \sum_{n=1}^{N} \left[2h_r[n]\cos(n\omega T_s)\right] d\omega \end{aligned} \qquad (4.10)$$

For a valid reconstruction filter, the $D_r[k]$ of Equation (4.10) must be equal to the $D_i[k]$ of Equation (4.9). Thus, within the frequency band of interest, the $h_r[n]$ coefficients must satisfy

$$\sum_{n=1}^{N} \left[h_r[n]\cos(n\Omega)\right] = \frac{1}{2} \qquad (4.11)$$

where $\Omega = \omega T_s$.

To find a solution for the $h_r[n]$ coefficients, consider an ideal low-pass filter $X(\Omega)$ as shown in Figure 4.8. For $|\Omega| \leq \Omega_B$, $X(\Omega) = K$; otherwise, $X(\Omega) = 0$. The Fourier series representation of this filter is

$$X(\Omega) = \frac{K \times \Omega_B}{\pi} + \sum_{n=1}^{\infty} \left[\frac{2K \times \sin(n\Omega_B)}{n\pi} \cos(n\Omega) \right] \qquad (4.12)$$

Thus, for $|\Omega| \leq \Omega_B$,

$$\sum_{n=1}^{\infty} \left[\frac{2K \times \sin(n\Omega_B)}{n\pi} \cos(n\Omega) \right] = K - \frac{K \times \Omega_B}{\pi} = \frac{K(\pi - \Omega_B)}{\pi} \qquad (4.13)$$

Comparing Equation (4.11) with Equation (4.13) for N approaching infinity, K and $h_r[n]$can be found to be

$$K = \frac{\pi}{2(\pi - \Omega_B)} \tag{4.14}$$

$$h_r[n] = \frac{2K \times \sin(n\Omega_B)}{n\pi} \tag{4.15}$$

Substitute Equation (4.14) into Equation (4.15) yields

$$h_r[n] = \frac{\sin(n\Omega_B)}{n(\pi - \Omega_B)} \tag{4.16}$$

For example, if the bandwidth of $V_i(t)$ is ω_B, and $\Omega_B = \omega_B T_s = (4/5)\pi$, we can choose

$$h_r[n] = \frac{5 \sin \frac{4n\pi}{5}}{n\pi} \tag{4.17}$$

As an example, consider a sine wave input

$$V_i(t) = A_i \times \sin(\omega_i t + \phi_i) \tag{4.18}$$

and the corresponding digital output

$$D_i[k] = A_i \sin[\Omega_i(k + \epsilon[k]) + \phi_i] \tag{4.19}$$

where $\Omega_i = \omega_i T_s$ and $\epsilon[k]$ is the k-th sampling jitter normalized to T_s. For the reconstruction filter of Equation (4.8) with a finite value of N and the coefficients of Equation (4.17), the output of the SRF is

$$\begin{aligned} D_r[k] \approx & \left[\sum_{n=1}^{N} \frac{10}{\pi} \frac{\cos(n\Omega_i) \sin \frac{4n\pi}{5}}{n} \right] \times A_i \sin(k\Omega_i + \phi_i) \\ & + \frac{5A_i\Omega_i}{\pi} \left\{ \sum_{n=1}^{N} \frac{\cos[\Omega_i(k+n) + \phi_i] \sin \frac{4n\pi}{5}}{n} \epsilon[k+n] \right\} \\ & + \frac{5A_i\Omega_i}{\pi} \left\{ \sum_{n=1}^{N} \frac{\cos[\Omega_i(k-n) + \phi_i] \sin \frac{4n\pi}{5}}{n} \epsilon[k-n] \right\} \end{aligned} \tag{4.20}$$

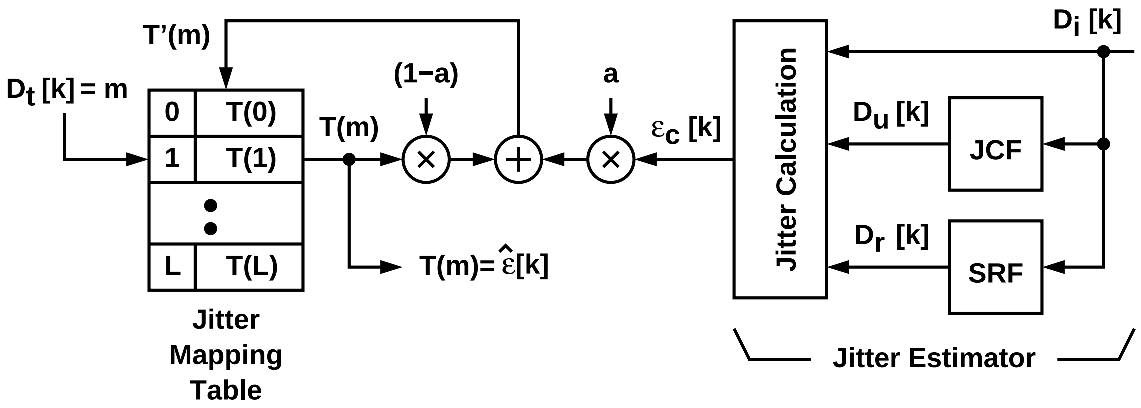


Figure 4.9: Block diagram of jitter calibration processor, JCP1.

As N increases, it can be shown that the first term on the right-hand side approximates $A_i \sin(k\Omega_i + \phi_i)$, which is the exact value of $V_i(t)$ sampled at $t = kT_s$. The last two terms in Equation (4.20) are errors caused by the clock jitters when sampling $D_i[k-n]$ and $D_i[k+n]$. Note that $D_r[k]$ is independent to $\epsilon[k]$ and will be used in the TDC background calibrations described in Section 4.5 and in Chapter 5.

The quantization error and the thermal noise when sampling $D_i[k-n]$ and $D_i[k+n]$ are not considered in Equation (4.20). These errors can be added to the equation easily with tedious calculation. These errors affect the accuracy of the reconstructed result $D_r[k]$. However, as will be discussed in Section 4.5, these errors have no effect on the accuracy of the TDC calibration.

4.5 Jitter Calibration Processor

The JDC in Figure 4.1 consists of a stochastic TDC and a jitter calibration processor JCP1. The TDC measures the timing difference between CLK$_e$ and CLK$_i$, and outputs a digital code that corresponds to this timing difference. The TDC can also be used as the phase detector of a digital delay-locked loop that generates the DCTL signal [48]. The JCP1 translates the output of the TDC, $D_t[k]$, into the corresponding jitter estimation, $\hat{\epsilon}[k]$. The JCP1 also performs self-calibration to ensure an accurate $\hat{\epsilon}[k]$.

Figure 4.9 shows the JCP1 block diagram. It contains a jitter mapping table (JMT). The JMT receives $D_t[k] = m$ as an address, and outputs the content stored in that address.

The content $T(m)$ is the corresponding jitter estimation, $\hat{\epsilon}[k]$. Every address m has its own content $T(m)$. Note that $0 \leq m \leq L$, where L is the number of TCMPs in the TDC. The content $T(m)$ is acquired from a jitter estimator (JE). The JE extracts the jitter information $\epsilon[k]$ from $D_i[k]$ and its neighboring samples, which are digital outputs from the ADC. The JE includes a JCF employing the structure of Figure 4.2.

This JCF generates the $D_u[k]$ signal as defined in Equation (4.23). Also included in the JE is a signal reconstruction filter (SRF) that takes $2N$ samples, $D_i[k-n]$ and $D_i[k+n]$ where $1 \leq n \leq N$, and applies Equation (4.8) to reconstruct the $V_i(kT_s)$ sample, denoted as $D_r[k]$. The JE makes a new estimation of the jitter $\epsilon[k]$ based on the outputs from both JCF and SRF. Its output, $\epsilon_c[k]$, updates the $T(m)$ by

$$T'(m) = (1-a) \times T(m) + a \times \epsilon_c[k] \tag{4.21}$$

where $a < 1$ is a constant. The above equation is a low-pass filter with a single pole at $z_p = 1 - a$. Its function is to remove the high-frequency components in $\epsilon_c[k]$ so that $T(m)$ can approximate the dc value of $\epsilon_c[k]$.

The JE calculates its output $\epsilon_c[k]$ based on the criterion described as follows. If a JCF employs this $\epsilon_c[k]$ for jitter compensation, its output $D_c[k]$ should be equal to $D_r[k]$ which is the output of the SRF. Recall that the output of the JCF can be expressed as

$$D_c[k] \approx D_i[k] + \hat{\epsilon}[k] \times \frac{D_u[k]}{\epsilon_u} \tag{4.22}$$

where

$$D_u[k] = \sum_{n=k-M}^{k-1} \left\{ D_i[n] \times h_s[k-n] \right\} + \sum_{n=k+1}^{k+M} \left\{ D_i[n] \times h_s[k-n] \right\} \tag{4.23}$$

with

$$h_s[n] = \mathrm{sinc}(n - \epsilon_u) \tag{4.24}$$

Since $\epsilon_c[k]$ is the jitter used in the JCF to make the interpolated signal of $\hat{V}_i(t)$ at $t = (k - \epsilon_c[k])T_s$ equal to the reconstructed signal $D_r[k]$, from Equation (4.22) and letting $D_c[k] = D_r[k]$, $\epsilon_c[k]$ is calculated as

$$\epsilon_c[k] = \frac{D_r[k] - D_i[k]}{D_u[k]} \times \epsilon_u \tag{4.25}$$

or

$$\frac{\epsilon_c[k]}{\epsilon_u} = \frac{D_r[k] - D_i[k]}{D_u[k]} \tag{4.26}$$

As mentioned in Chapter 2.4, if the JMT collects the result of Equation (4.26) rather than Equation (4.25), the results of calibration can be stored in the form of divided by ϵ_u so that the division in Figure 4.2 is not needed.

Consider a sine wave input where $V_i(t)$ and $D_i[k]$ are expressed as Equation (4.18) and Equation (4.19) respectively. The filter coefficients for the SRF are the $h_r[n]$ of Equation (4.17). By equating Equation (2.22) and Equation (4.20), and letting $\hat{\epsilon}[k] = \epsilon_c[k]$, we have

$$\begin{aligned}\epsilon_c[k] \approx & -\frac{\Omega_i}{2F_c(M, \Omega_i)} \times \epsilon[k] \\ & + \left[\sum_{n=1}^{N} \left(\frac{10}{\pi} \frac{\cos(n\Omega_i) \sin \frac{4n\pi}{5}}{n}\right) - 1\right] \times \frac{\tan(k\Omega_i + \phi_0)}{2F_c(M, \Omega_i)}\end{aligned} \tag{4.27}$$

Contributions by the last two terms in Equation (4.20) are not shown in the above equation. They are removed by the low-pass filter of Equation (4.21). If he quantization error and the thermal noise when sampling $D_i[k-n]$ and $D_i[k+n]$ are also considered in the SRF, these error can also be filtered out by the low-pass filter.

For each m, the $T(m)$ is updated only when the TDC has its output $D_t[k] = m$. Every time when $D_t[k] = m$, the TDC detects a similar $\epsilon[k]$ jitter, denoted as $\epsilon(m)$. From Equation (4.27), we have

$$T(m) \approx -\frac{\Omega_i}{2F_c(M, \Omega_i)} \times \epsilon(m) \tag{4.28}$$

The second term on the right-hand side of Equation (4.27) does not appear in Equation (4.28), since it is removed by the low-pass filter of Equation (4.21).

Figure 4.10 shows the ratio of $T(m)/\epsilon(m)$ calculated from Equation (4.28). Note that $\Omega_i = \omega_i T_s = 2\pi f_i / f_s$. The ratio represents the JDC conversion gain. Different M yields

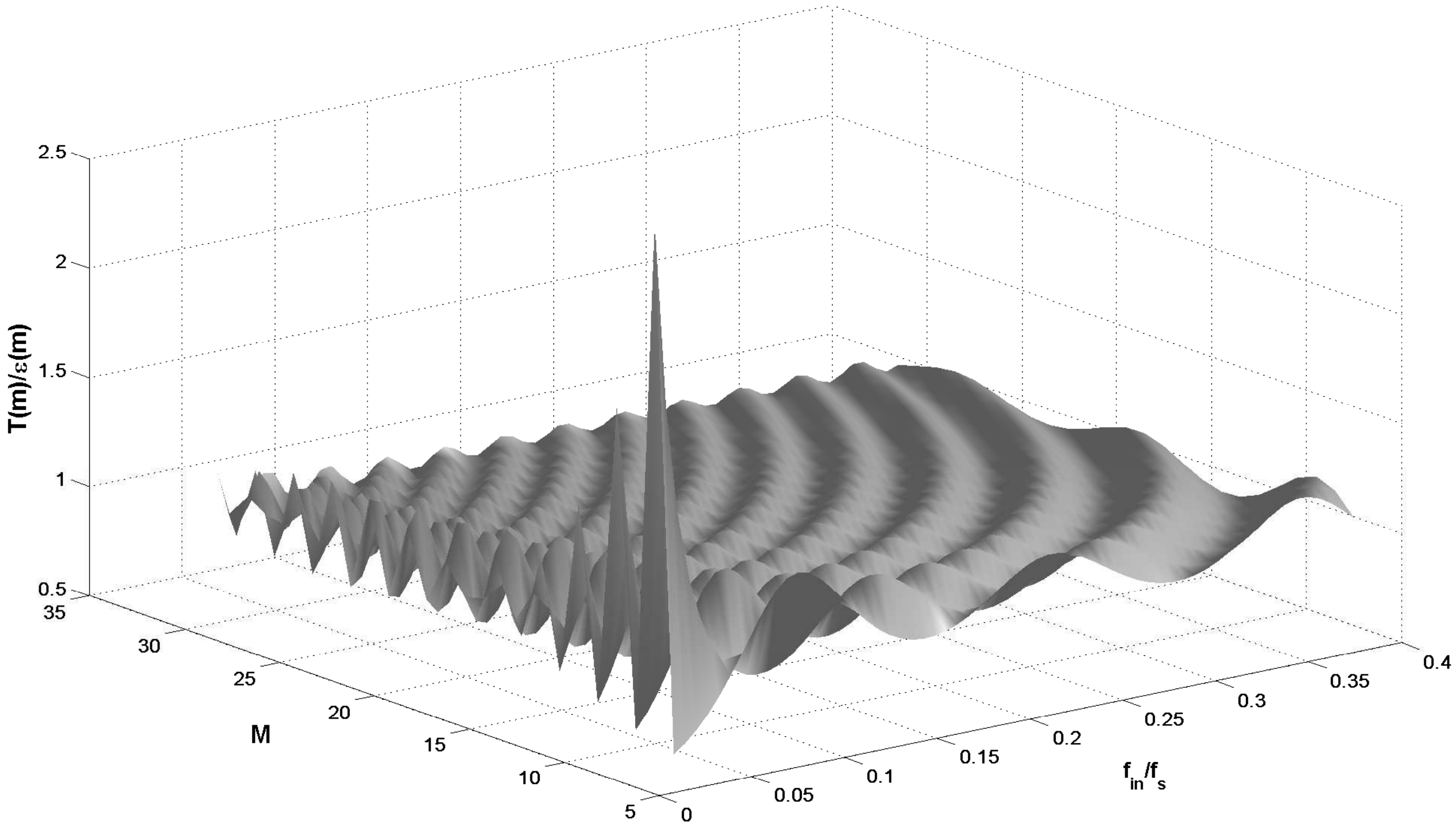


Figure 4.10: Conversion gain of the JDC using JCP1 for different M and different input frequencies.

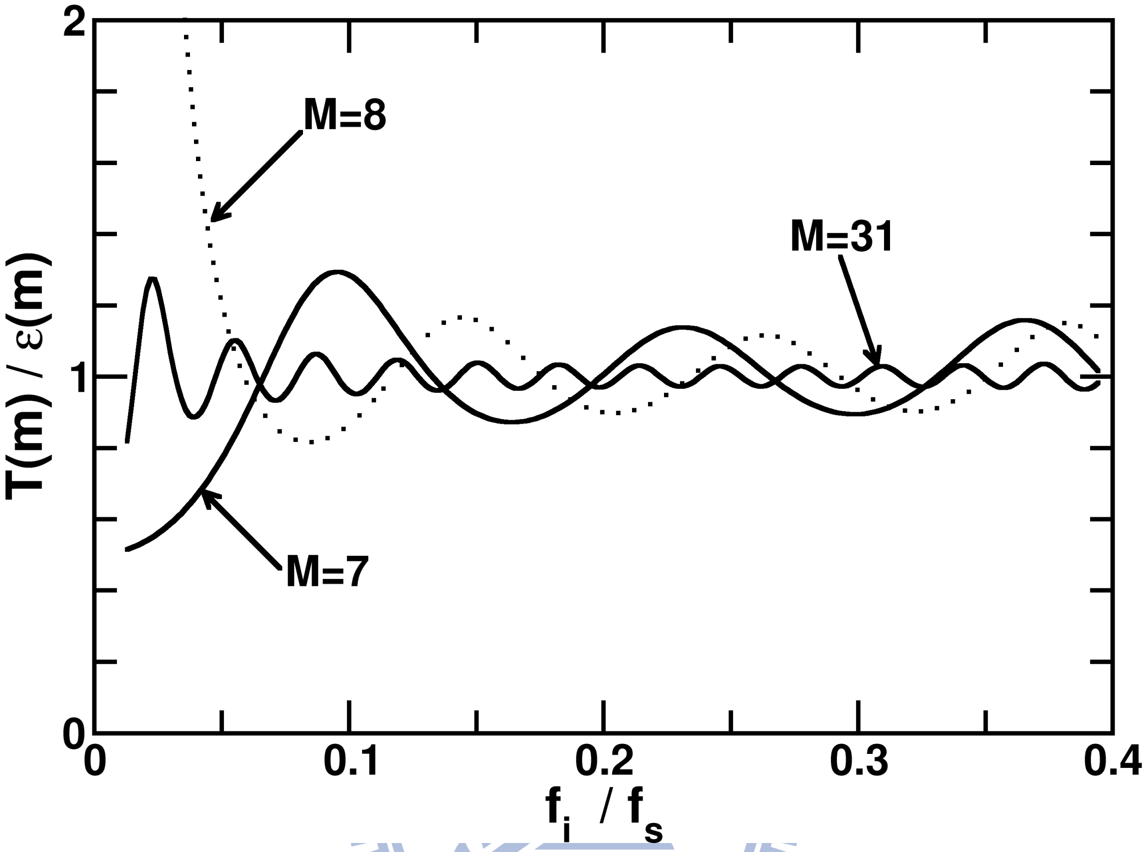


Figure 4.11: Conversion gain of the JDC using JCP1 for M=7, M=8 and M=31.

different conversion gain characteristic, where $2M + 1$ is the number of taps for the JCF shown in Figure 4.9. Figure 4.11 shows the ratio of $T(m)/\epsilon(m)$ for M=7, $M = 8$ and $M = 31$. If $M \to \infty$, the conversion gain approaches 1 for all input frequencies up to $f_i = 0.4f_s$, which is determined by the $h_r[n]$ coefficients of the SRF. The conversion gain is less accurate at low f_i frequencies. An odd M is preferred if $V_i(t)$ is a narrow-band low-f_i signal. If an even M value is selected, the resulting $|T(m)|$ can be excessively large under the same input condition.

Although a larger M for the JCF results in a better JDC conversion gain characteristic, a large M is not necessary if the JCF in Figure 4.9 is identical to the JCF in Figure 4.1. If $V_i(t)$ is wide-band and has many different frequency components, the averaged JDC conversion gain approximates 1 even for small M. To illustrate a narrow-band input condition, let $V_i(t) = A_i \sin(\omega_i t + \phi_i)$ as in Equation (4.18). For jitter compensation, the $D_c[k]$ signal is calculated as Equation (2.22) and is rewritten here

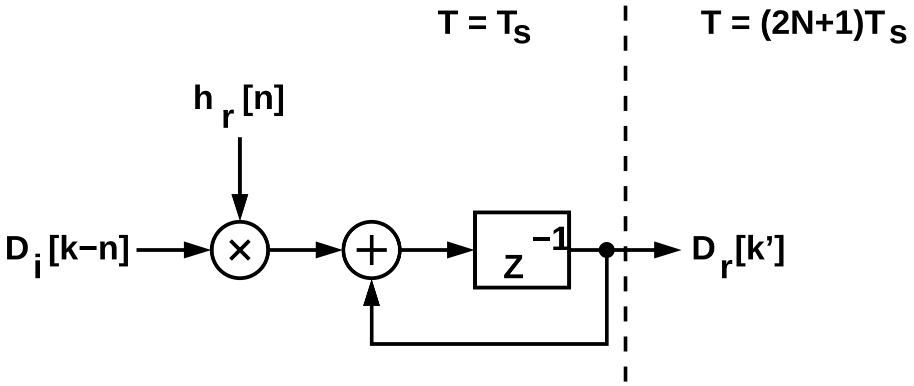


Figure 4.12: Cyclic multiply-and-accumulate architecture to realize the SRF.

$$D_c[k] \approx D_i[k] + \hat{\epsilon}[k] \times F_c(M, \Omega_i) \times 2A_i \cos(k\Omega_i + \phi_i) \tag{4.29}$$

and

$$D_i[k] = V_i\Big(\big(k + \epsilon[k]\big) \times T_s\Big) \approx V_i(kT_s) + \epsilon[k] \times \Omega_i \times A_i \cos(k\Omega_i + \phi_i) \tag{4.30}$$

where $\epsilon[k]$ is the k-th jitter normalized to clock period T_s and $\hat{\epsilon}[k]$ is the jitter measured by the JDC. Substitute Equation (4.30) into Equation (4.29) yields

$$D_c[k] \approx V_i(kT_s) + \Big\{\epsilon[k] \times \Omega_i + \hat{\epsilon}[k] \times 2F_c(M, \Omega_i)\Big\} \times A_i \cos(k\Omega_i + \phi_i) \tag{4.31}$$

From Equation (4.28), if an ideal low-pass filter is used to filter out the noise in Equation (4.27), the measured jitter $\hat{\epsilon}[k]$ after the TDC background calibration is

$$\hat{\epsilon}[k] = -\frac{\Omega_i}{2F_c(M, \Omega_i)} \times \epsilon[k] \tag{4.32}$$

Substitute Equation (4.32) into Equation (4.31) yields $D_c[k] \approx V_i(kT_s)$. Thus, the JDC conversion gain error is automatically compensated. It has little effect on the SNR of the $D_c[k]$ signal.

For an accurate signal reconstruction, the SRF requires a very large N. It is not practical to implement the SRF hardware with $2N$ multipliers. We propose using the cyclic multiply-and-accumulate (MAC) architecture to realize the SRF as shown in Figure 4.12.

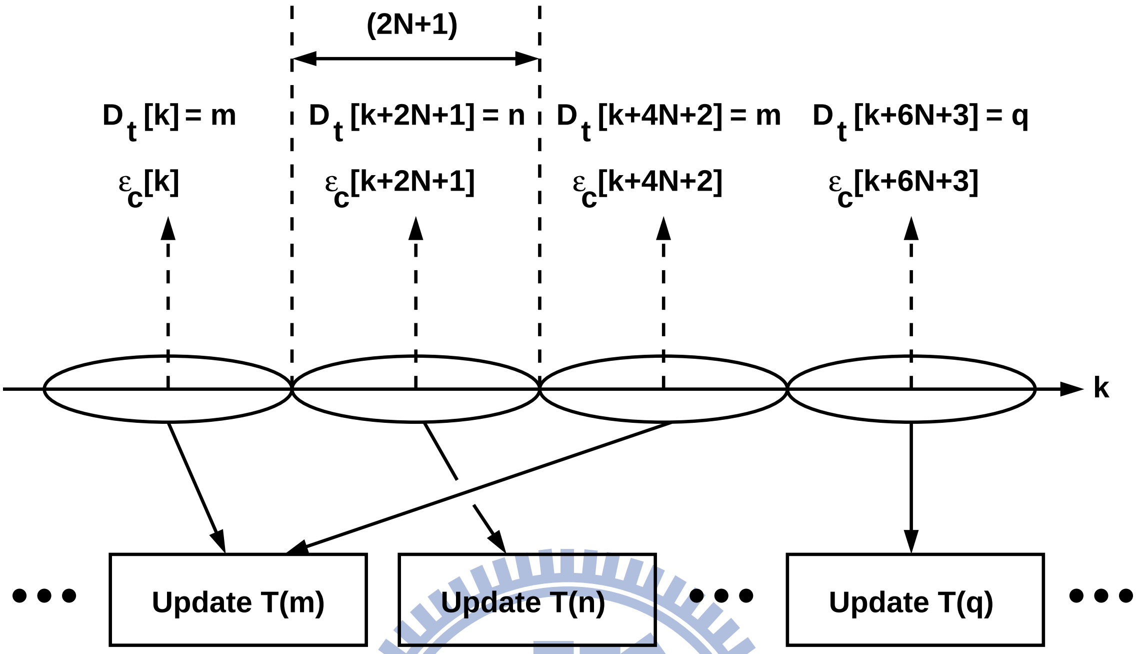


Figure 4.13: Calibration for different TDC output codes, $D_t[k]$.

The MAC architecture requires only one multiplier and one accumulator. As a result, the SRF can generate only one $D_r[k]$ signal every $2N + 1$ clock cycles. Therefore, the proposed jitter estimator generates one valid $\epsilon_c[k]$ output every $2N + 1$ clock cycles as shown in Figure 4.13. Note that for each m, the $T(m)$ is updated only when the TDC has its output $D_t[k] = m$.

For the jitter compensation configuration of Figure 4.1, the jitter measurement error of the JDC, defined as $\epsilon_e[k]$ in Equation (2.11), is mainly caused by the quantization noise $\epsilon_q[k]$ of its TDC. We can assume $\epsilon_e[k] = \epsilon_q[k]$ and $\epsilon_{e,rms} = \epsilon_{q,rms}$. The JDC's $\epsilon_{e,rms}$ requirement can be estimated by using Equation (2.5) and Equation (2.24). If M is large, it can be found by using Equation (2.25). Since $\epsilon_{e,rms} = \epsilon_{q,rms}$, we also obtain the $\epsilon_{q,rms}$ requirement for the TDC, which dictates the TDC's resolution.

4.6 A 16-bit 80MS/s ADC Design Example

The ADC system of Figure 4.1 is simulated using a C program. As in Section 2.7, the ADC has 16-bit resolution and operates at 80 MS/s sampling rate. The rms of the clock

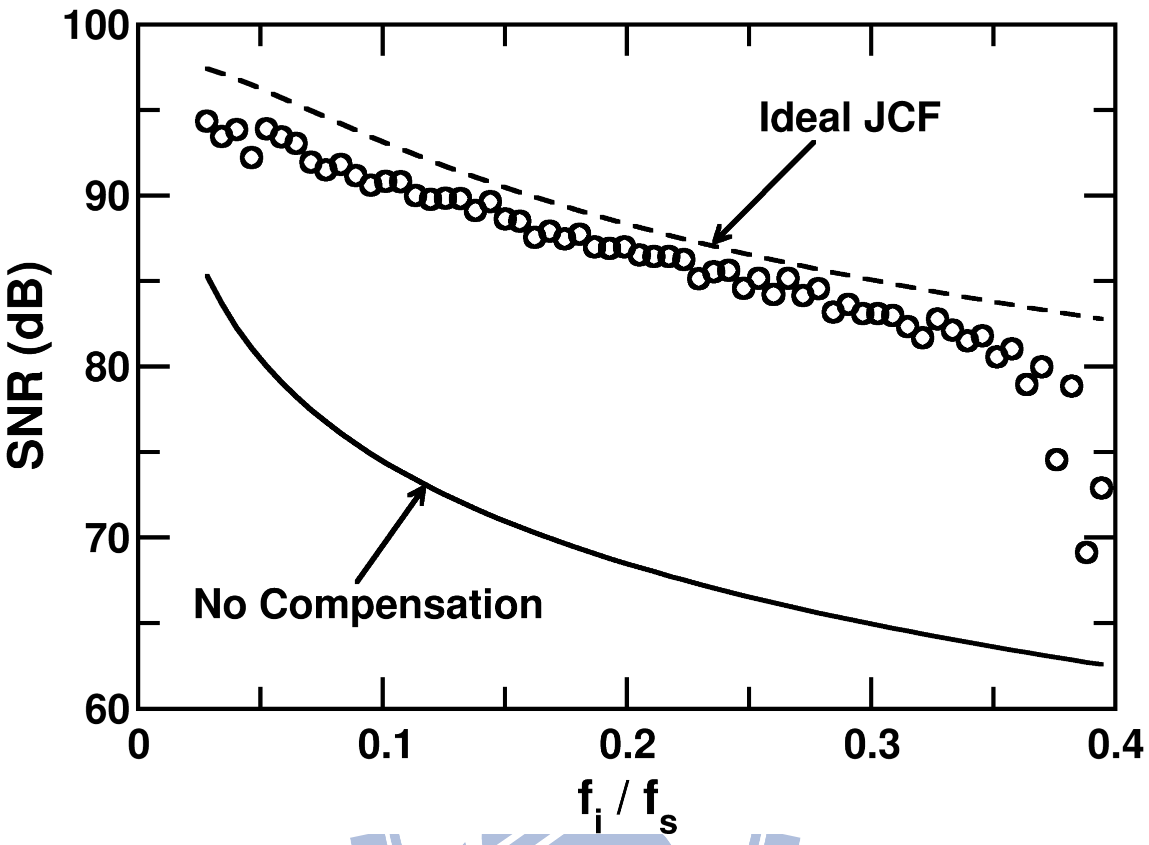


Figure 4.14: SNR of the ADC system of Figure 4.1 at various input frequencies.

jitter $\epsilon[k]$ is $\epsilon_{rms} = 3 \times 10^{-4}$, i.e., $\Delta t_{rms} = \epsilon_{rm}T_s = 3.75$ ps. To make simulation results agree with the theoretical analyses more closely, the TCMPs in the TDC are assumed to contain uniformly-distributed t_{os} so that the TDC has a uniform quantization step of size $\Delta t_s = 1.25$ ps. Similar to a ideal ADC, the resulting rms of quantization noise is $\epsilon_{q,rms} = (1/\sqrt{12}) \times 10^{-4}$. In practice, 72 TCMPs are required in the TDC to achieve the same $\epsilon_{q,rms}$ if the TCMP's t_{os} has the normal distribution $G(t_{os})$ of Equation (3.8) with a standard deviation $\sigma = 6.36$ ps. For the JCF, $M = 7$. For the SRF, $N = 2^{10}$ and $\Omega_B = (4/5)\pi$. For the low-pass filter of Equation (4.21), $a = 2^{-13}$. In Figure 4.14, the circles are the simulation results. They are the SNRs of the $D_c[k]$ signal under various f_i frequencies. From Equation (2.25) with $\epsilon_{e,rms}$ replaced by $\epsilon_{q,rms}$, the theoretical SNR with ideal JCF compensation is shown as the dash line. From Equation (2.10), the theoretical SNR for the ADC without jitter compensation is shown as the solid line. The calibration scheme of Figure 4.1 improves the SNR by about 20 dB.

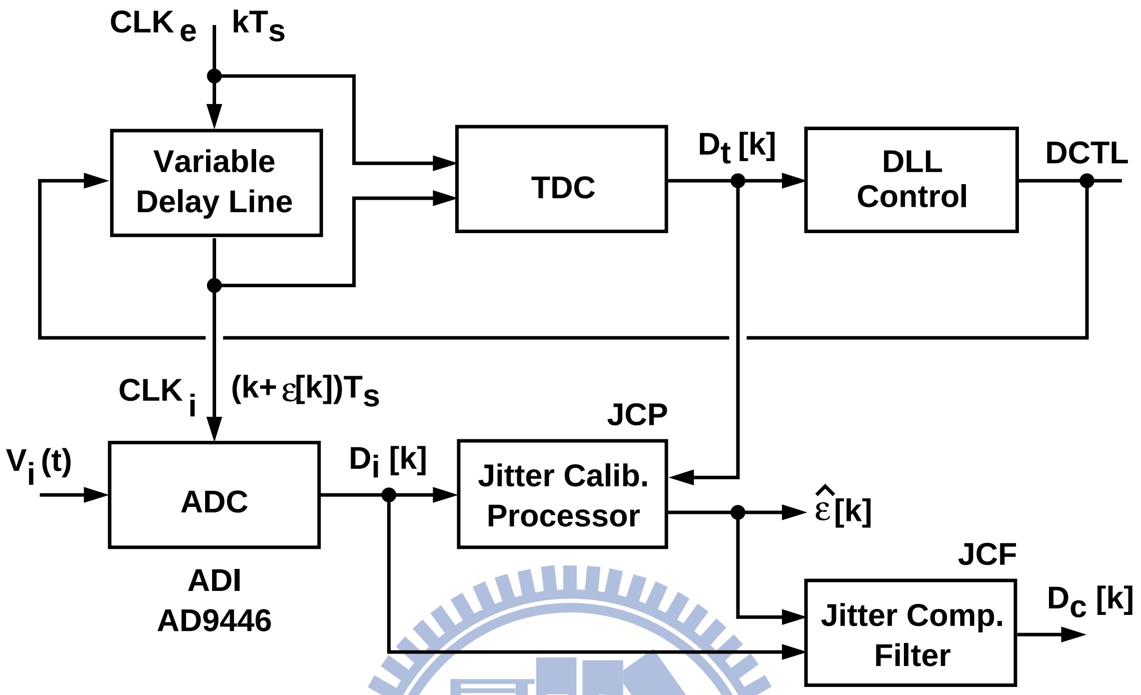


Figure 4.15: Jitter compensation experiment setup.

4.7 Circuit Implementations

Figure 4.15 shows the setup for the jitter compensation experiment. Assume that the external clock CLK$_e$ is clean. Its rising edges are at kT_s, where k is an integer and T_s is the period of CLK$_e$. A variable-delay line (VDL) receives the clock CLK$_e$ and delivers the internal clock CLK$_i$ to an ADC. The VDL introduces jitter to the clock CLK$_i$. Thus, the rising edges of the CLK$_i$ are at $(k + \epsilon[k])T_s$, where $\epsilon[k]$ is a real-valued random variable, representing the jitter in CLK$_i$. Employing the clock CLK$_i$ as a sampling time reference, the ADC samples and digitizes the analog input $V_i(t)$ and generates the digital stream $D_i[k]$. The ADC output $D_i[k]$ contains a sampling error due to the jitter $\epsilon[k]$. We use a TDC and a jitter calibration processor to generate a jitter estimate $\hat{\epsilon}[k]$ for every k. A jitter compensation filter then employs $\hat{\epsilon}[k]$ to correct the sampling error in $D_i[k]$. The resulting $D_c[k]$ is a corrected ADC digital output with improved SNR. The TDC also functions as the phase detector of a DLL. The DLL adjusts the VDL delay to align the CLK$_i$ phase with the CLK$_e$ phase. The DLL is used to reduce the required input range for the TDC. We will detail the circuit architectures in this section.

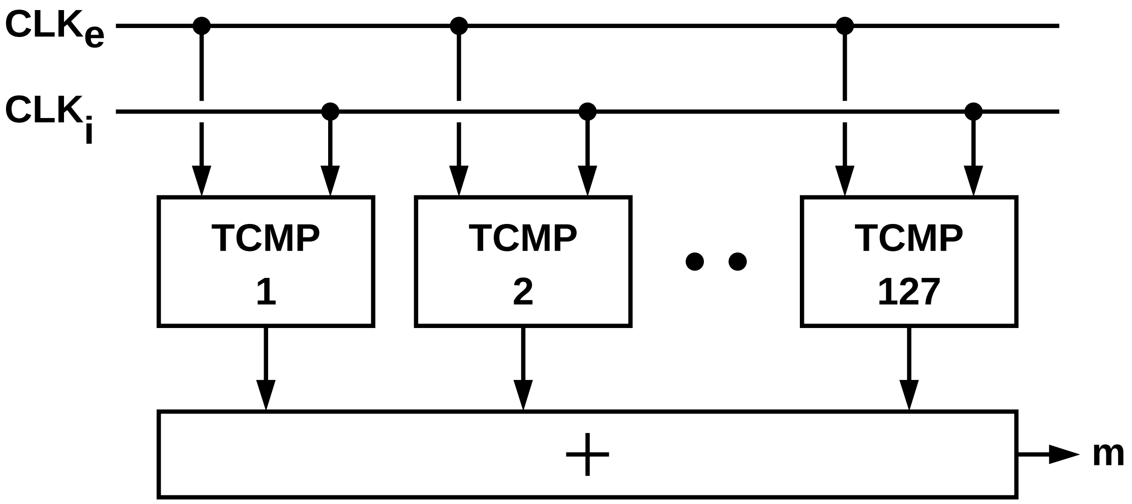


Figure 4.16: Stochastic TDC block diagram.

4.7.1 Stochastic Time-to-Digital Converter

Figure 4.16 shows the stochastic TDC block diagram [28, 48]. The TDC is a collection of 127 timing comparators (TCMP). Figure 4.17 shows the TCMP schematic. The TCMP is essentially an asynchronous latch. The fully differential configuration makes the circuit less sensitive to common-mode noise. When both CLK$_e$ and CLK$_i$ are low, the TCMP is reset, and both of its outputs, $V_{o,p}$ and $V_{o,n}$, are charged to V_{DD}. If the rising edge of the CLK$_e$ arrives before the rising edge of the CLK$_i$, the TCMP output, $V_{o,p} - V_{o,n}$, is a digital 1. If the rising edge of the CLK$_e$ arrives after the rising edge of the CLK$_i$, the TCMP output is a digital 0. Thus, the TCMP compares the timing difference between the input clocks CLK$_e$ and CLK$_i$.

The TCMP exhibits an input offset t_{os} due to device mismatches, i.e., its output becomes 1 only if CLK$_e$ leads CLK$_i$ by an amount greater than t_{os}. The offset t_{os} is a random variable with zero mean and standard deviation of σ. The TDC output $m = D_t[k]$ is a summation of all TCMP outputs. Thus, the TDC transfer function is an integration of the probability density function (pdf) of the random variable t_{os}.

As shown in Figure 3.15, in order to achieve a given resolution with minimum TCMPs in a TDC, $\sigma = 3 \times \Delta t_{rms}$ can be chosen. Thus, if the rms jitter of the clock is 3.75 ps, the optimal σ of the TCMP is 11.25 ps. From Monte Carlo simulations, the t_{os} standard deviation is 11 ps for the TCMP realized in 65 nm CMOS technology.

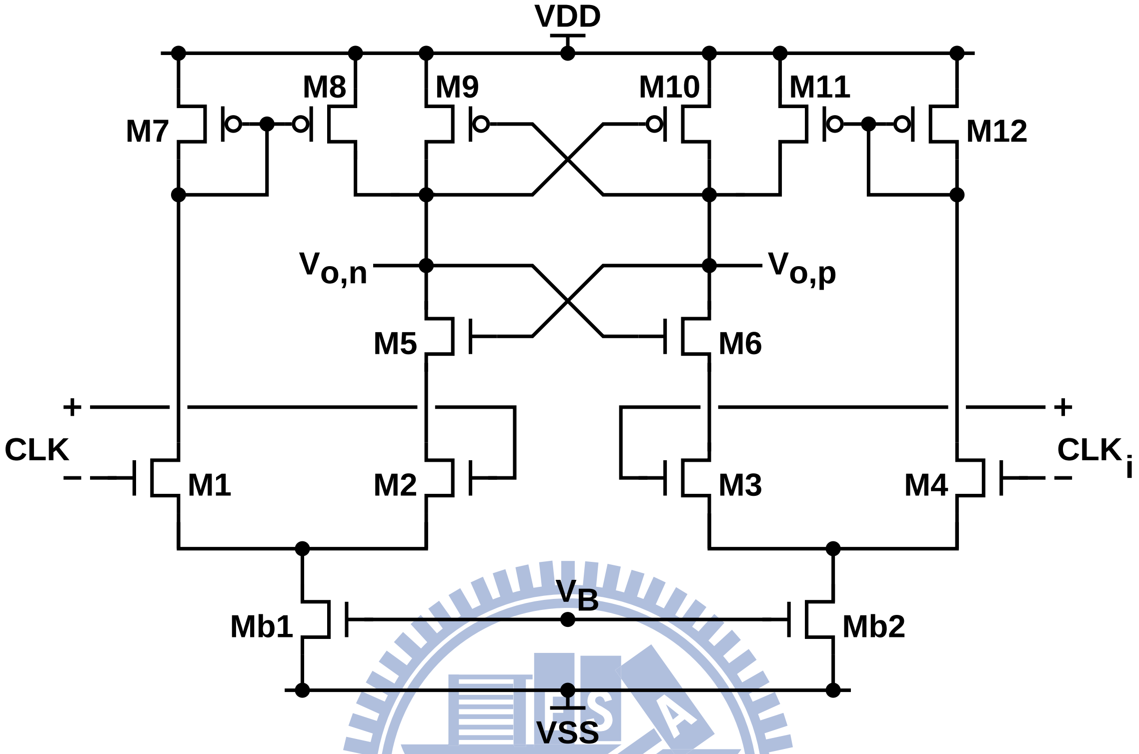


Figure 4.17: Timing comparator (TCMP) schematic.

In our design, the TCMPs are divided into 3 groups. Each group has a different t_{os} mean so that the overall TDC input range is increased to reserve additional margin. There are 63 normal TCMPs in group 1 with a t_{os} mean of 0 ps. Both group 2 and group 3 have 32 TCMPs, their t_{os} mean are +14.5 ps and −14.5 ps respectively. The t_{os} mean is introduced by changing the W/L ratios of M2 and M3. From Monte Carlo simulations, the t_{os} standard deviation for each group is 11 ps. Figure 4.18 shows the pdf of t_{os} for each group. The overall t_{os} pdf is shown as the solid line. The dots are the t_{os} data collected from Monte Carlo circuit simulations.

4.7.2 Variable Delay Line

The VDL is a cascade of 9 differential delay cells. There are 8 identical delay cells with coarse control and one delay cell with fine control. Figure 4.19 shows the delay cell schematic. The shunt-capacitor scheme is used to digitally control the delay [53]. The shunt capacitors are MOS transistors with binary-weighted area. The capacitors are

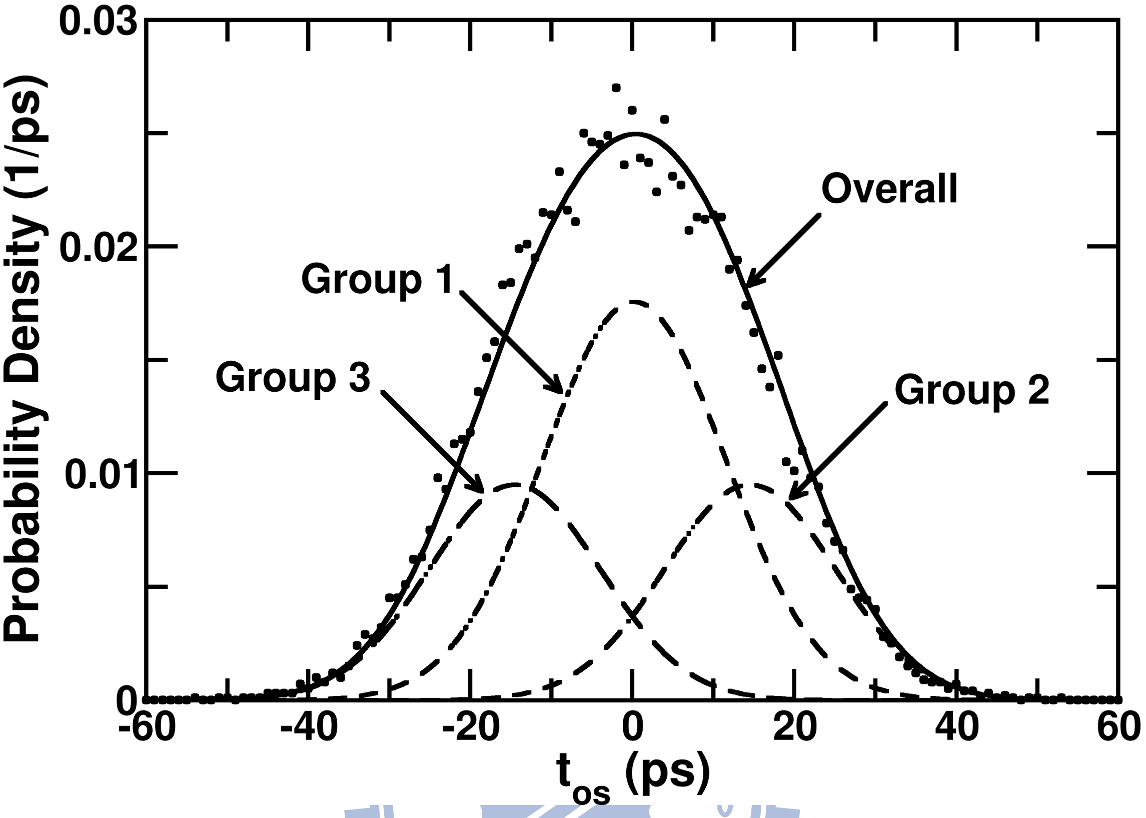


Figure 4.18: The probability density function (pdf) of TCMP t_{os}.

connected to the 8-bit digital control signal (DCTL). The coarse and fine delay cells are identical except the capacitors in the coarse delay cell are 4 times larger than the capacitors in the fine delay cell. From simulations, the delay resolution for the coarse control is 34 ps, and the delay resolution for the fine control is 1 ps. The maximum delay of the fine-controlled delay cell is 256 ps, which is large enough to cover the PVT variations.

The DLL control first adjusts the VDL coarse control, freezes its value, then constantly adjusts the VDL fine control to minimize the timing difference between the clocks CLK$_e$ and CLK$_i$. Thus, the DLL can achieve wide locking range and low jitter performance at the same time.

4.7.3 Delay-Locked Loop

In Figure 2.2, the TDC, VDL, and the DLL control form a DLL. The block diagram of the DLL is shown in Figure 4.20. The DLL control is a simple digital integrator. Figure 4.21

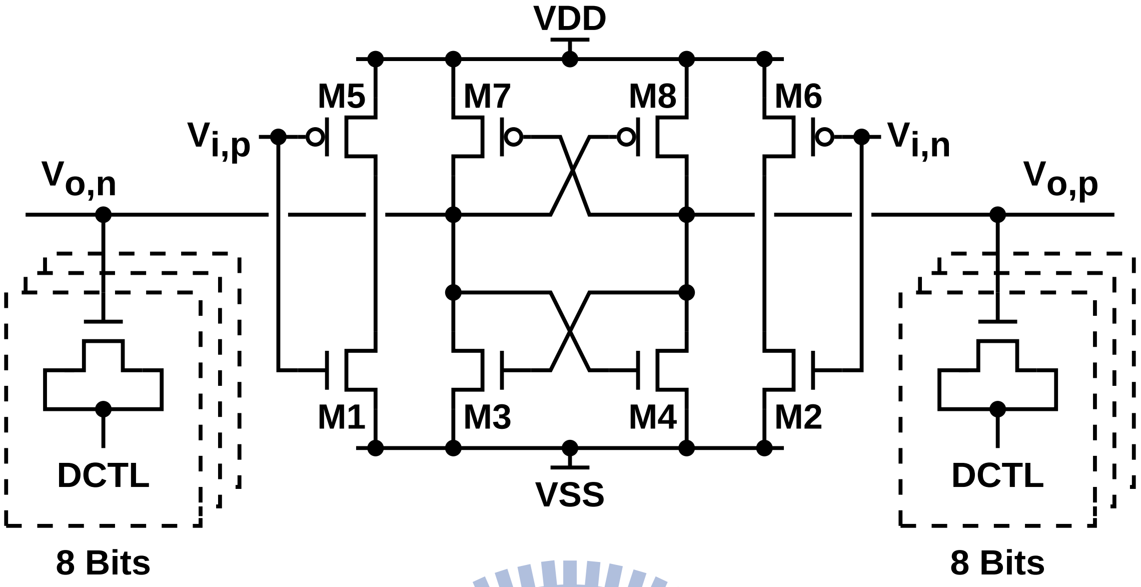


Figure 4.19: An 8-bit digitally-controlled delay cell.

shows the discrete-time jitter transfer model of this digital DLL [54], where K_{TDC} is the transfer gain of the TDC, K_{VDL} is the transfer gain of the VDL, and K_{INT} is the gain of the digital integrator. The jitter of the CLK$_e$ clock is $\epsilon_i[k]$ and the jitter introduced by the VDL is $\epsilon_d[k]$. Then the jitter of the CLK$_i$ clock can be expressed as

$$\epsilon(z) = \frac{1 - z^{-1} + K_{\mathrm{DLL}} z^{-2}}{1 - z^{-1} + K_{\mathrm{DLL}} z^{-3}} \epsilon_i(z) + \frac{1 - z^{-1}}{1 - z^{-1} + K_{\mathrm{DLL}} z^{-3}} \epsilon_d(z) \tag{4.33}$$

where $K_{\mathrm{DLL}} = K_{\mathrm{TDC}} K_{\mathrm{INT}} K_{\mathrm{VDL}}$ is the DLL loop gain. The DLL loop is stable if $K_{\mathrm{DLL}} \leq 0.618$. The loop gain K_{DLL} can be adjusted by changing K_{INT}. In our experimental setup, $\epsilon(z)$ is dominated by the VDL jitter $\epsilon_d(z)$.

For nonzero K_{DLL}, the jitter amplification can never be eliminated [54]. The jitter rms of the CLK$_i$ clock can be expressed as

$$\sigma_{clki} = A_{clk} \times \sigma_{clk} + A_{dl} \times \sigma_{dl} \tag{4.34}$$

where σ_{clk} and σ_{dl} are the jitter rms of the clock CLK$_e$ and the delay line respectively. The A_{cke} and A_{dl} can be derived from the impulse response of (4.33). The A_{cke} and A_{dl} for different K_t are plot in Figure 4.22. In this design we chose $K_t = 0.14$, the resulting A_{cke} and A_{dl} are 1.01 and 1.05.

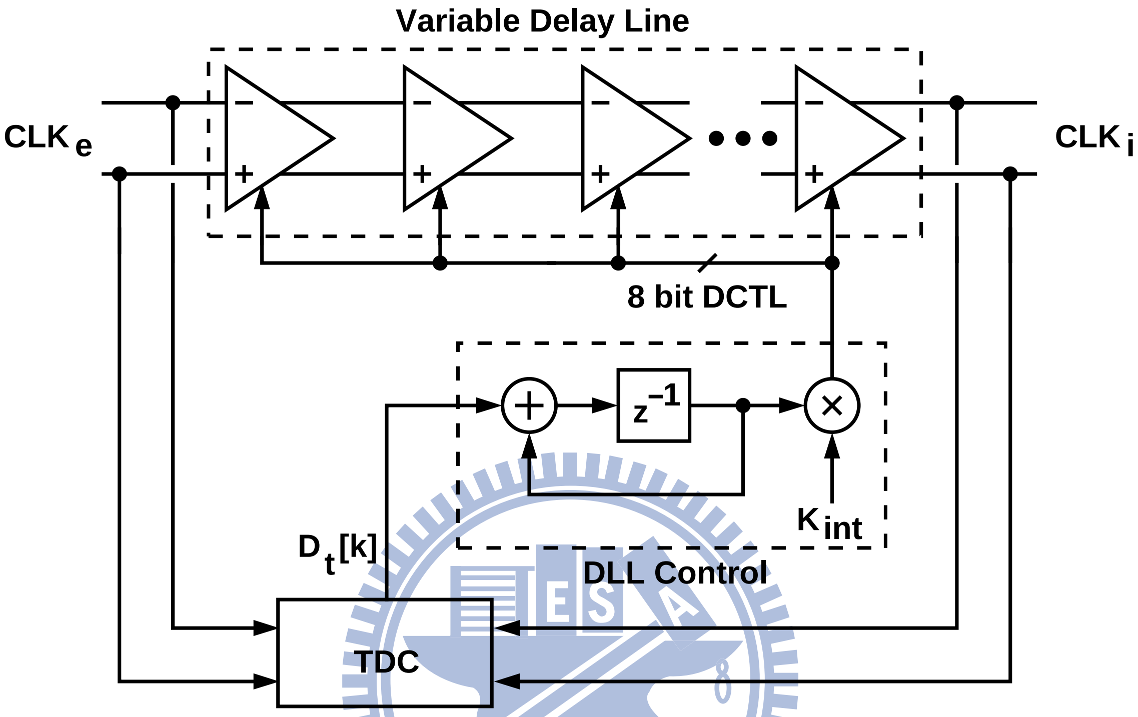


Figure 4.20: Block diagram of the digital delay-locked loop.

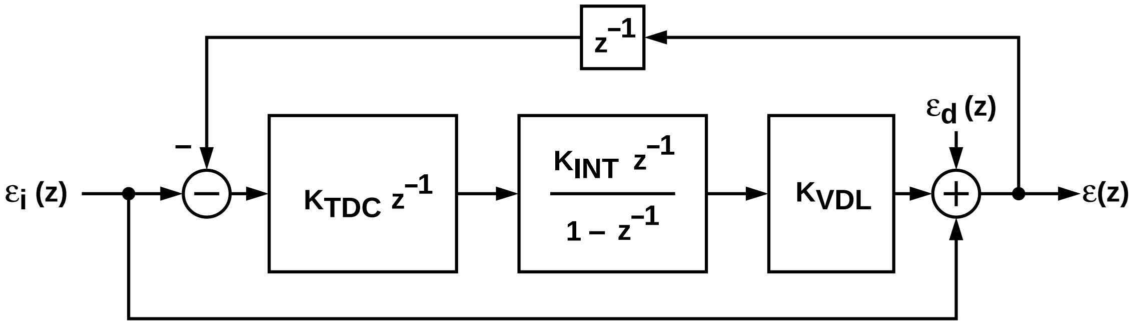


Figure 4.21: Discrete-time model of the digital delay-locked loop.

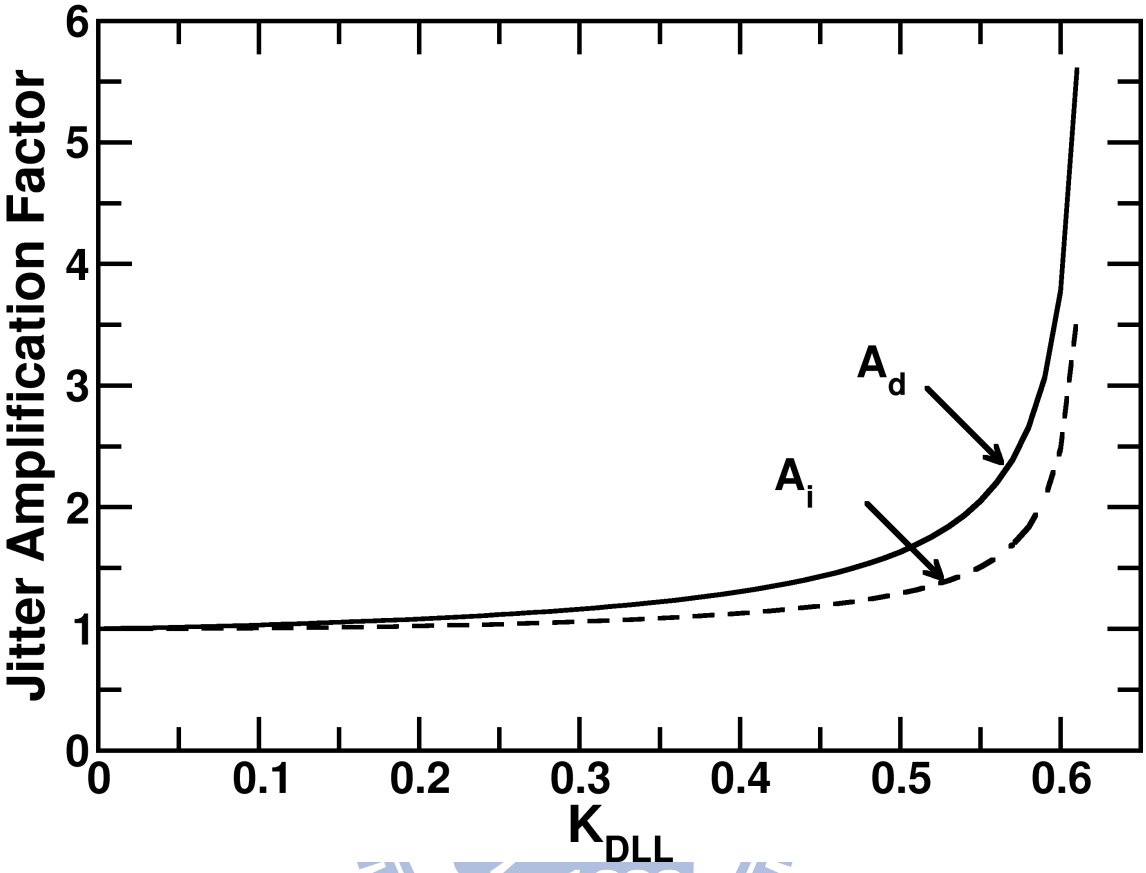


Figure 4.22: Jitter amplification of the DLL.

4.8 Experimental Results

Figure 4.23 shows the chip micrograph, which contains a TDC, a VDL, and a DLL control. The chip was fabricated using a 65 nm CMOS technology. The circuits occupied an area smaller than 480 × 100 μm^2. The chip area including pads is 1500 × 1000 μm^2.

Supply voltage is 1.2 V. Operating at 80 MHz, each TCMP consumes 86 μW of power. Total power consumption is 20 mW for the TDC and the VDL, and 60 mW for the output drivers.

Figure 4.24 shows the measurement environment. The 80 MHz clock signal is generated by Agilent ESG-D3000A vector signal generator. The analog input signal is generated by Agilent E4438C ESG vector signal generator. The clock and the input signal are both passing through a band-pass filter to ensure the clock and the input signal are both clean. The clock signal is then delivered to the DLL. The output of the DLL is used as the

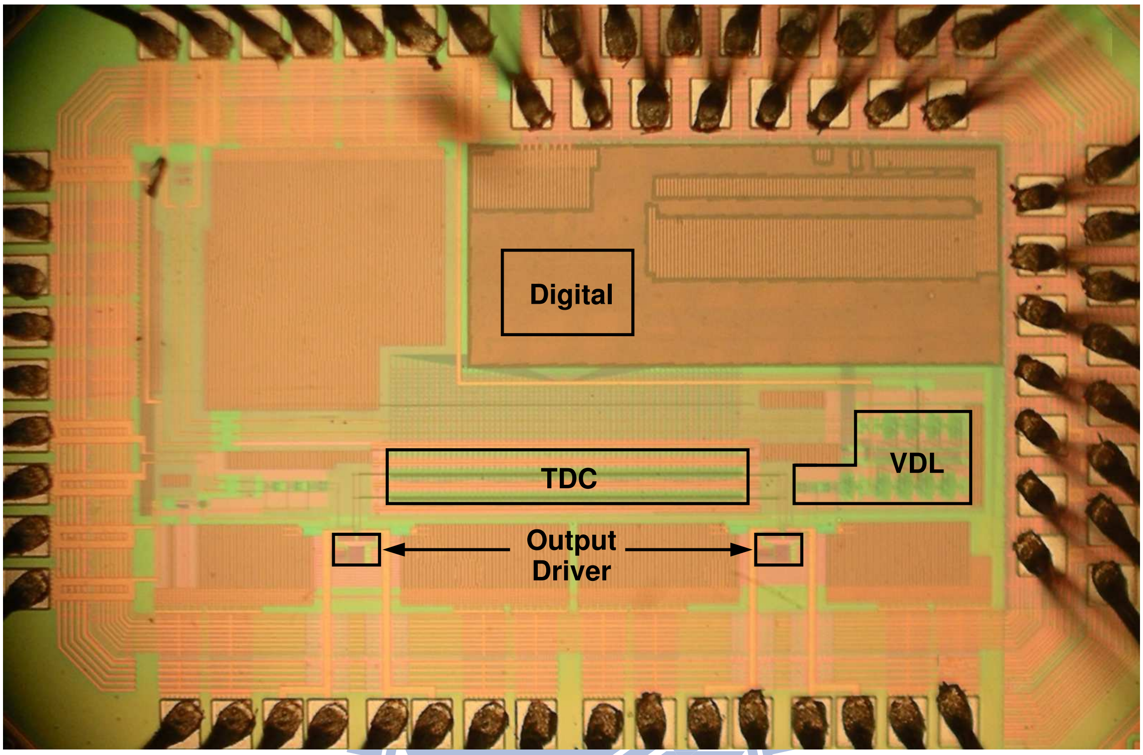


Figure 4.23: Chip micrograph.

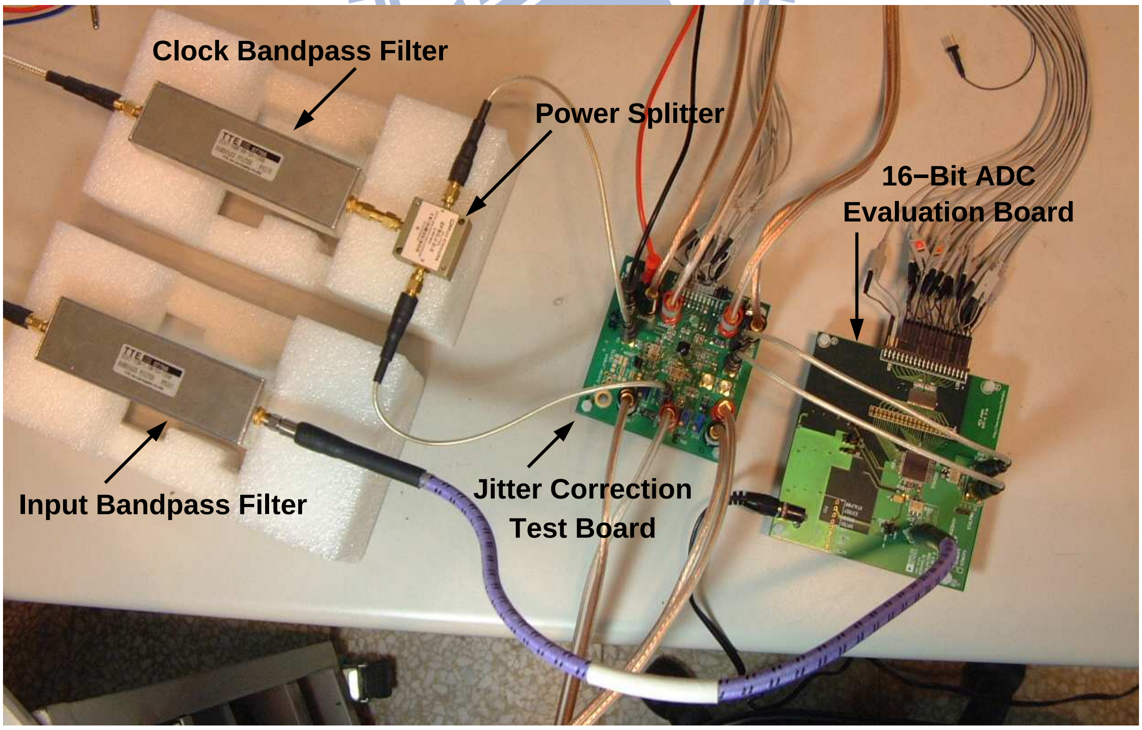


Figure 4.24: Photo of the jitter correction setup with the ADC evaluation board.

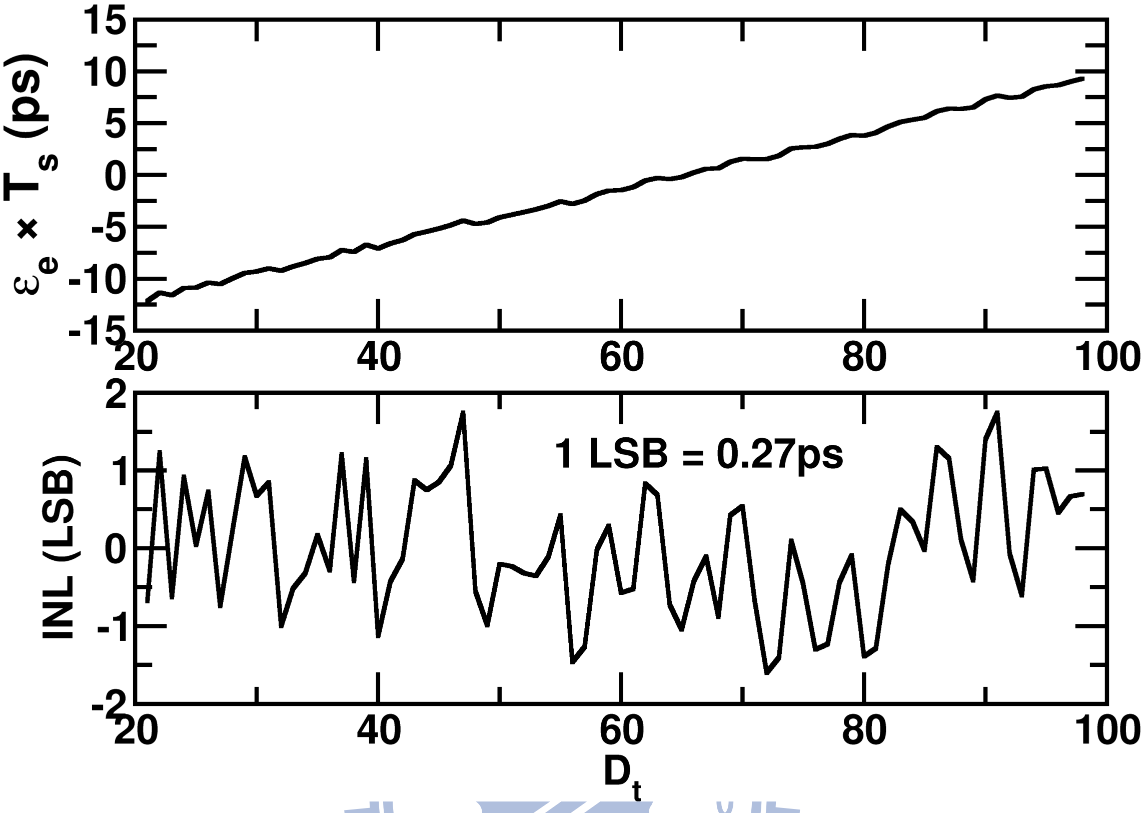


Figure 4.25: Measured TDC transfer function and INL.

sampling clock of the ADC. The TDC measures the timing difference between the input clock and the output of the DLL. A logic analyser is used to collect the output of the TDC and the output of the ADC. By using these data, the JCP and the JCF can be realized in digital post processing.

Figure 4.25 shows the contents of the JMT after calibration. It also represents the measured TDC transfer function. The data does not extend to the full range of D_t, i.e., from 0 to 127. During actual measurement, the jitter input to the TDC did not cover the entire TDC input range. Thus, those D_t codes outside the active range never appeared and did not have the chance to get calibrated. Also shown in the figure is the TDC integral nonlinearity (INL) plot. It displays the difference between the measured transfer function against a best-fitted linear function. The LSB step size is 0.27 ps, which also represents the TDC resolution. The INL of the TDC is $+1.7/-1.6$ LSB.

Figure 4.26 shows the measured ADC output power spectrum before and after the

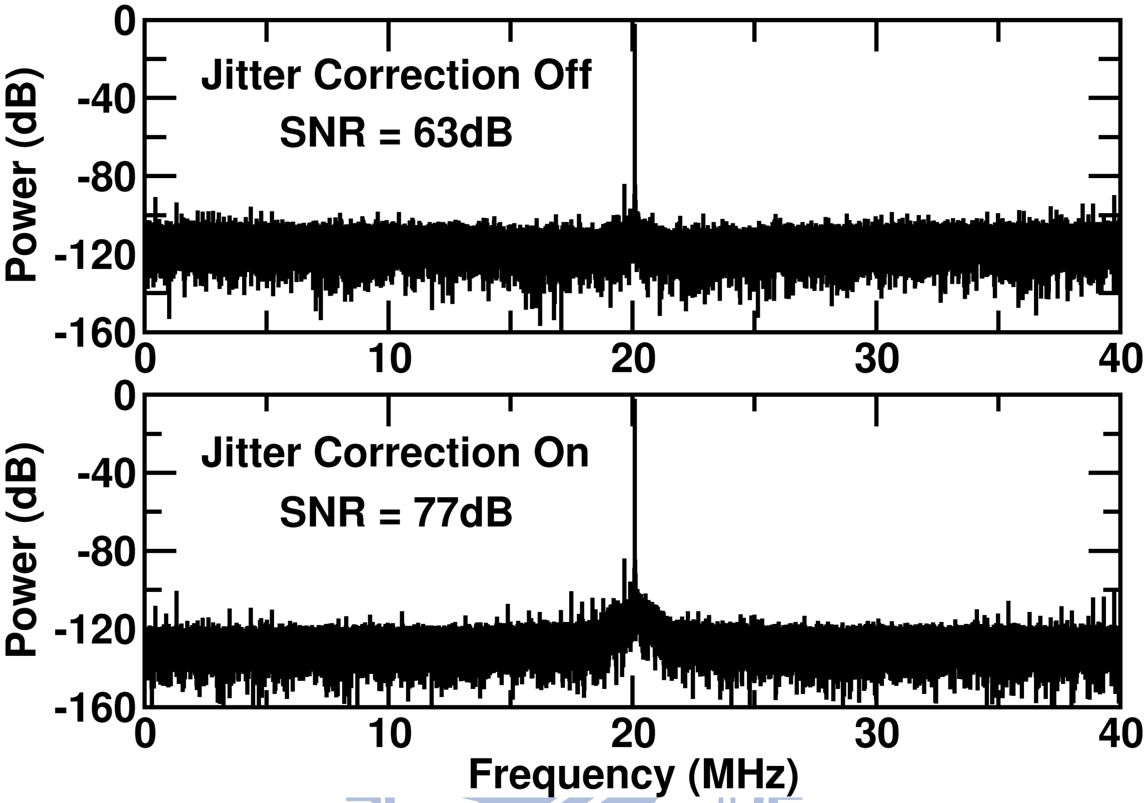


Figure 4.26: Measured ADC output power spectrum before and after the jitter correction.

jitter correction for a sine wave input of 20 MHz. The noise floor is decreased after jitter correction. The skirt around the fundamental signal is contributed by the signal source.

Figure 4.27 shows the measured SNR performance of the ADC before and after jitter correction. Two different cases are illustrated. In case 1, the power lines are well-conditioned by adding bypass capacitors and $K_{\mathrm{DLL}} = 0.14$ is chosen for the DLL. The jitter rms of the CLK$_i$ clock is $\sigma = 1.2$ ps as measured by the TDC. The jitter correction can improve the ADC SNR from 71.2 dB to 77.3 dB for a 29 MHz sine wave input. In case 2, the power lines are ill-conditioned by removing some bypass capacitors and $K_{\mathrm{DLL}} = 0.56$ is chosen for the DLL. The resulting jitter rms of the CLK$_i$ clock is $\sigma = 4.5$ ps as measured by the TDC. The jitter correction can improve the ADC SNR from 60.8 dB to 74.4 dB for a 29 MHz sine wave input.

Figure 4.28 shows the measured jitter of the CLK$_i$ clock when the V_{DD} supply of the VDL is modulated by a 1 MHz 20 mV$_{\mathrm{pp}}$ sine wave. During this measurement, the DLL control is disabled. The delay of the VDL is manually adjusted to align the CLK$_e$ and

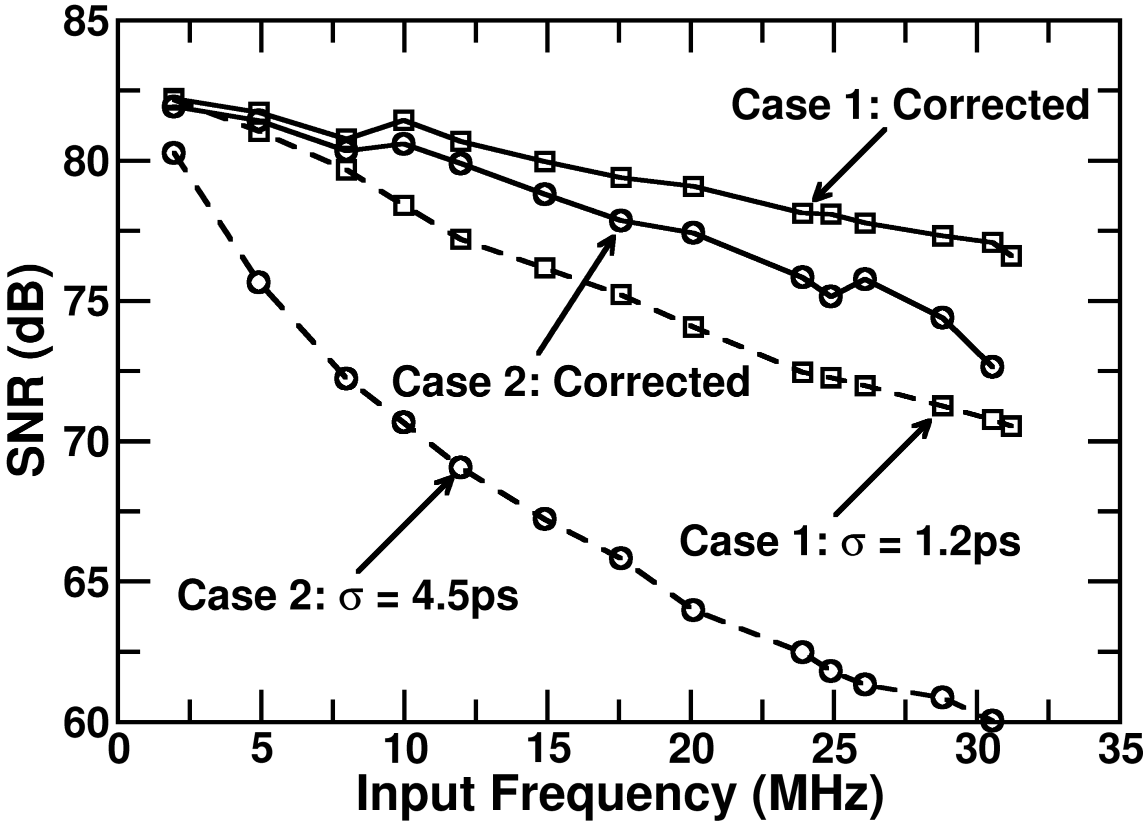


Figure 4.27: Measured ADC SNR versus input frequency.

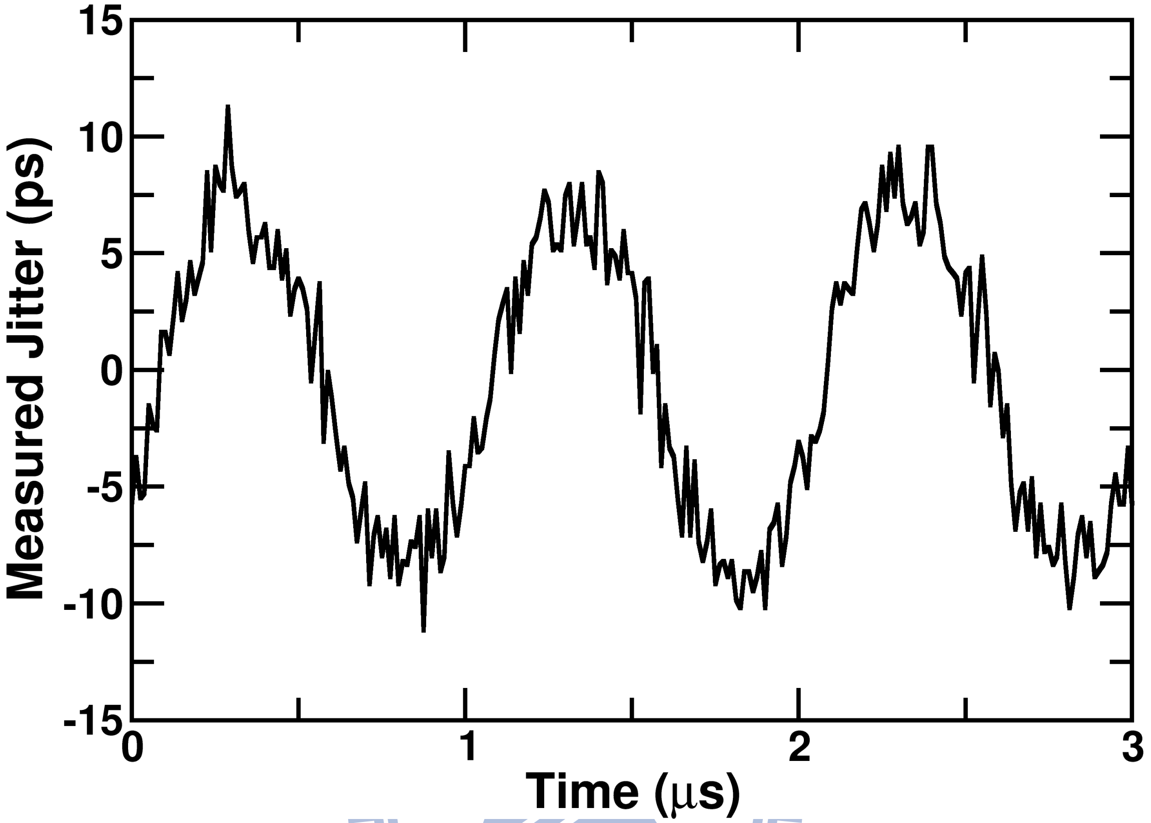


Figure 4.28: Measured sine wave jitter of the CLK$_i$ clock.

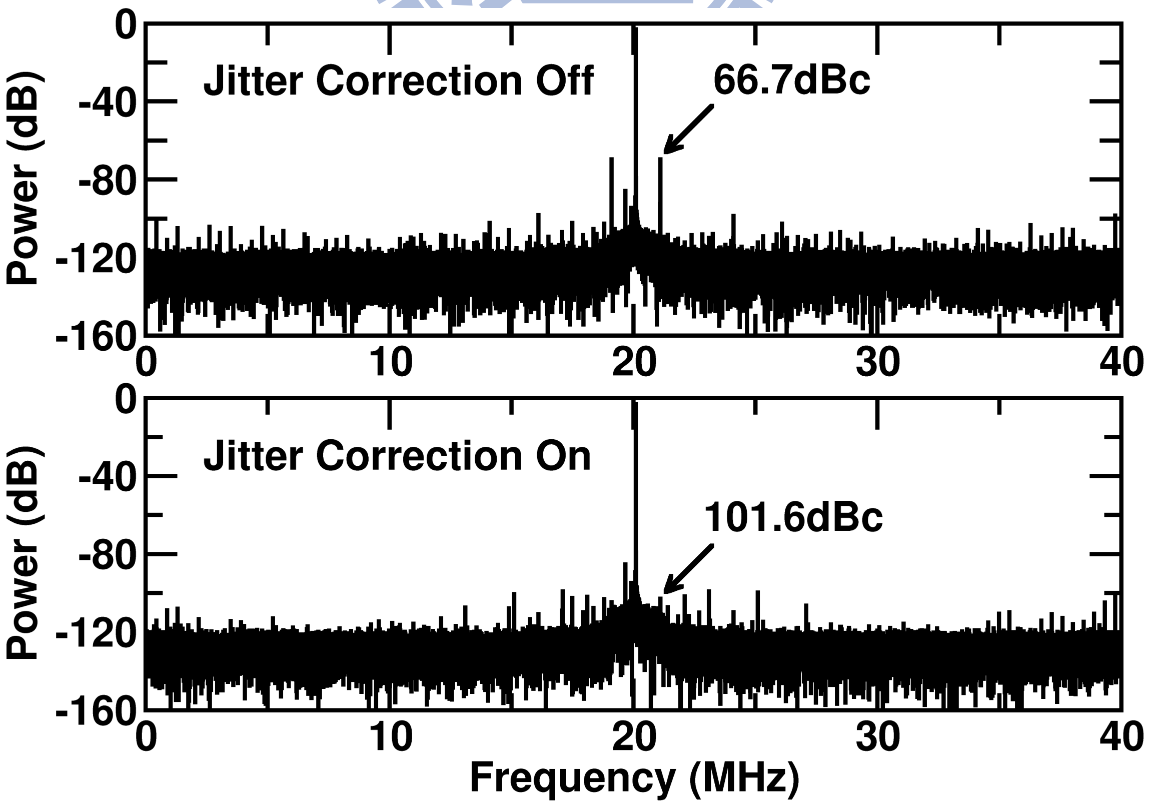


Figure 4.29: Measured ADC output power spectrum before and after the jitter correction when the CLK$_i$ jitter is a sine wave.

CLK_i clocks. Figure 4.29 shows the resulting output power spectrum before and after the jitter correction. The spurious tones caused by the sine wave jitter are 66.7 dBc. After jitter correction, the spurious tones are suppressed below 101.6 dBc.

4.9 Summary

We have demonstrated that clock jitter can be accurately measured and the measured data can be used to correct the ADC sampling errors in the digital domain. The accurate jitter measurement is enabled by using a stochastic TDC. Realized in a 65 nm CMOS technology, the TDC achieves a resolution of 0.27 ps. For a stochastic TDC, its resolution can be improved by adding more timing comparators. We have also demonstrated a digital background calibration technique that can calibrate the TDC without interrupting the operations of both the TDC and the ADC.

The jitter measurement and correction system described in the thesis require a clean external clock CLK_e. However, it is possible to improve the ADC SNR even when a clean external clock is not available. By adding a delay line of T_s delay, the cycle-to-cycle jitter can be measured by the TDC [35]. The absolute jitter can be obtained by integrating the cycle-to-cycle jitter. The TDC can also be calibrated in the cycle-to-cycle jitter measurement setup. We will discuss the jitter compensation and TDC calibration techniques when the external clock is the main jitter source in Chapter 5.

Chapter 5

Jitter Compensation with Jittering External Clock

5.1 Introduction

A Nyquist-rate analog-to-digital converter (ADC) requires a clock to sample its continuous-time analog input periodically. If the sampling clock exhibits jitter, sampling uncertainty occurred and the ADC suffers from sampling error. The sampling error degrades the signal-to-noise ratio (SNR) performance of the ADC. Low-jitter clocks, which are mostly based on crystal oscillators, are inflexible and expensive.

Chapter 4 describes a jitter compensation configuration where the main source of the jitter is the VCDL in the DLL. The application of this configuration is restricted since we can simply use the clean external clock as the sampling clock. In this chapter, we introduce another jitter compensation scheme in which the external clock is the main jitter source. The system can correct the sampling errors caused by the external clock jitter, thus allowing the use of cheaper clock source. In other words, the proposed jitter compensation technique mitigates the input clock requirement.

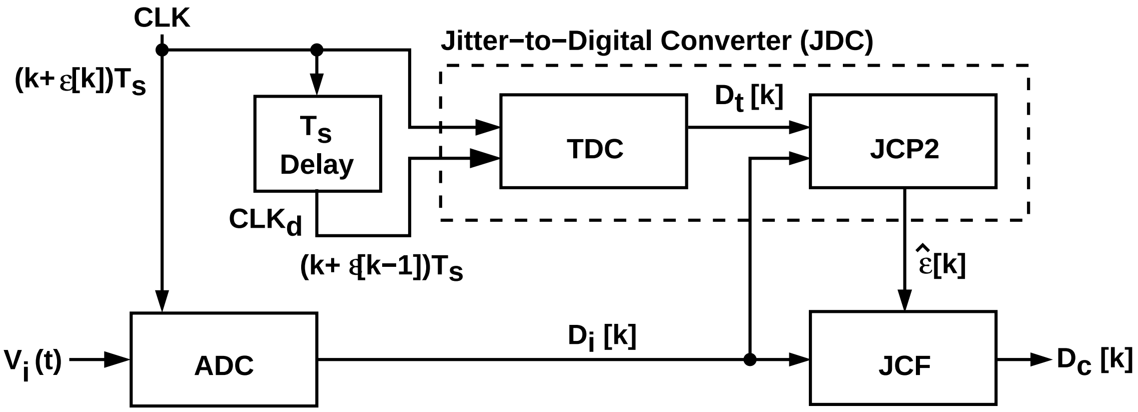


Figure 5.1: A jitter compensation scheme with jittering external clock.

5.2 Jitter Compensation with Jittering External Clock

The ADC system of Figure 4.1 assumes a clean external clock, which is used as a reference to measure the jitter caused by the internal clock buffer. Figure 5.1 shows an alternative ADC system with the external clock exhibiting $\epsilon[k]$ jitter. The system can correct the sampling errors caused by the clock jitter $\epsilon[k]$, thus allowing the use of cheaper clock source. However, jitter compensation for this scenario is more complex. Unlike the system of Figure 4.1, there is no clean clock that the TDC can refer to for measuring the absolute jitter. What can be measured is the relative jitter between two consecutive sampling instants [35]. In Figure 5.1, the CLK clock dictates the time of k-th sampling at $(k + \epsilon[k])T_s$, where T_s is the nominal sampling interval and $\epsilon[k]$ is the absolute jitter normalized to T_s. The CLK$_d$ clock is the CLK delayed by one T_s. The sampling time provided by CLK$_d$ is denoted as $(k + \epsilon[k - 1])T_s$. Figure 5.2 shows the relationship between the two clocks. Sampling instants are at the rising edges of the clocks. The TDC in Figure 5.1 measures the time difference between the k-th edge of the CLK clock and the $(k - 1)$-th edge of the CLK$_d$ clock. The measurement result is the cycle jitter defined as

$$\tau[k] = \epsilon[k] - \epsilon[k - 1] \tag{5.1}$$

The delay between the clock CLK and the clock CLK$_d$ should not introduce additional jitter. It can be realized using a passive delay line. For the calibration scheme described below, the accuracy of the realized delay is not crucial as long as it is a constant. If the

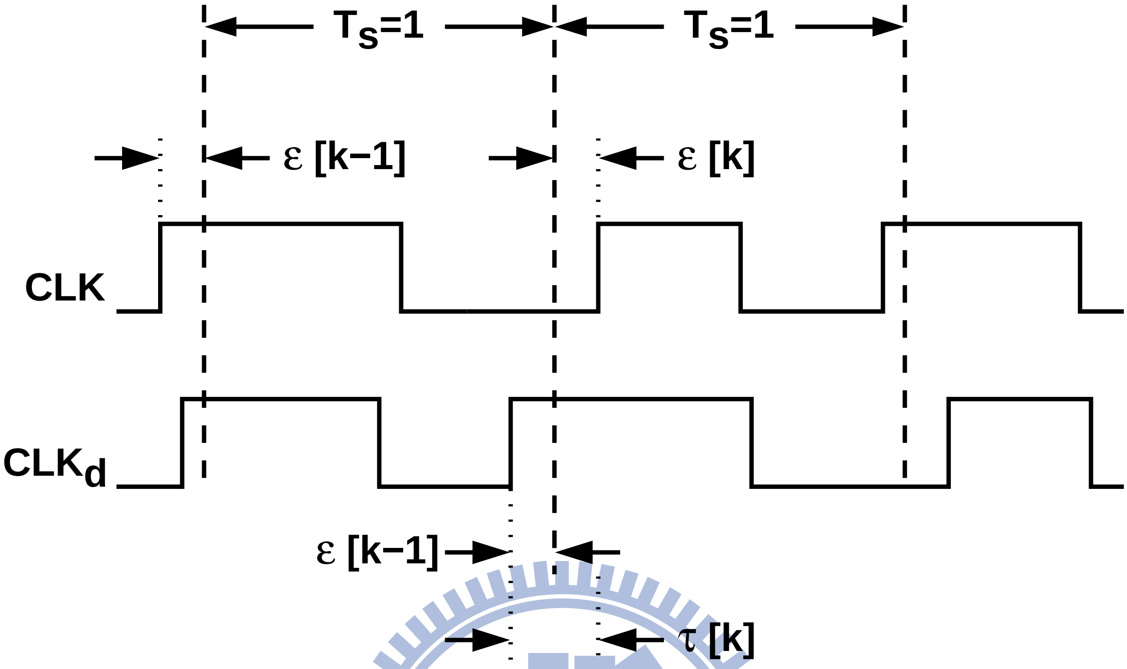


Figure 5.2: Relations between the CLK and CLK$_d$ clocks.

realized delay is different from T_s, the difference becomes a constant timing shift between the two input clocks of the TDC. This effect can be treated as a dc offset of the TDC. In the proposed scheme, the realized delay is only required to be close to T_s so that the offset is small enough to avoid overloading the TDC.

In Figure 5.1, a TDC converts the cycle jitter $\tau[k]$ into a $D_t[k]$ sequence. An ADC digitizes the analog input $V_i(t)$ and generates the corresponding digital output $D_i[k]$. A jitter calibration processor, JCP2, generates the absolute jitter estimation $\hat{\epsilon}[k]$. A JCF then uses this $\hat{\epsilon}[k]$ to correct $D_i[k]$. The corrected output is $D_c[k]$. Figure 5.3 shows the block diagram of the proposed JCP2. A JMT receives $D_t[k] = m$ from the TDC as an address, and outputs the content stored in that address. The content $T(m)$ is an estimation of the cycle jitter defined in Equation (5.1), denoted as $\hat{\tau}[k]$. The absolute jitter, $\hat{\epsilon}[k]$, is obtained by lossy accumulation of the $\hat{\tau}[k]$ sequence. The operation is expressed as

$$\hat{\epsilon}[k] = \hat{\tau}[k] + b \times \hat{\epsilon}[k-1] \tag{5.2}$$

where $b < 1$ is a constant. The reason to use a lossy accumulator is that a dc component may appear in the quantization noise of the TDC, which can overflow a lossless accumu-

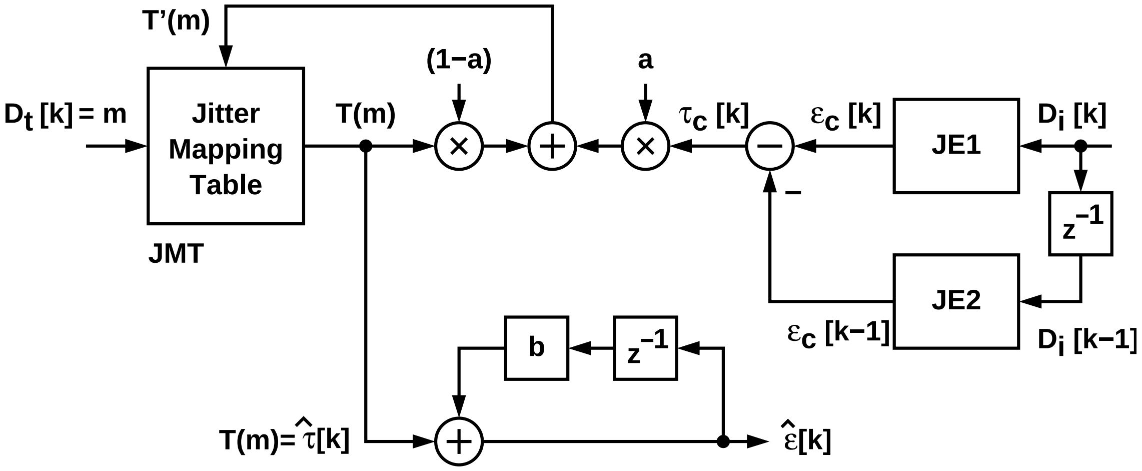


Figure 5.3: Block diagram of jitter calibration processor, JCP2.

lator.

To calibrate the content $T(m)$ when $D_t[k] = m$, JCP2 includes two jitter estimators, JE1 and JE2. Both are identical to the one shown in Figure 4.9. JE1 collects $D_i[k]$ and its neighboring $2N$ samples to extract $\epsilon_c[k]$, which is an estimation of the absolute jitter $\epsilon[k]$. In addition, JE2 collects $D_i[k-1]$ and its neighboring $2N$ samples to extract $\epsilon_c[k-1]$. Two jitter estimators are required, since neither estimator can calculate both $\epsilon_c[k]$ and $\epsilon_c[k-1]$ consecutively. Each estimator can generate only one valid output for every $2N+1$ samples. Similar to Equation (5.1), we define $\tau_c[k]$ as

$$\tau_c[k] = \epsilon_c[k] - \epsilon_c[k-1] \tag{5.3}$$

In Figure 5.3, the content $T(m)$ is updated by applying the low-pass-filter function of Equation (4.21) on $\tau_c[k]$ so that $T(m)$ can approximate the mean value of $\tau_c[k]$. Note that $T(m)$ only records the estimation of the cycle jitter $\tau[k]$ when $D_t[k] = m$. This cycle jitter is denoted as $\tau(m)$. The TDC's dc offset may affect the value of m, but has no effect on the accuracy of $T(m)$.

Combining Equation (5.3) and Equation (4.27), neglecting the high-frequency components that can be removed by the low-pass filter of Equation (4.21), $\tau_c[k]$ can be ap-

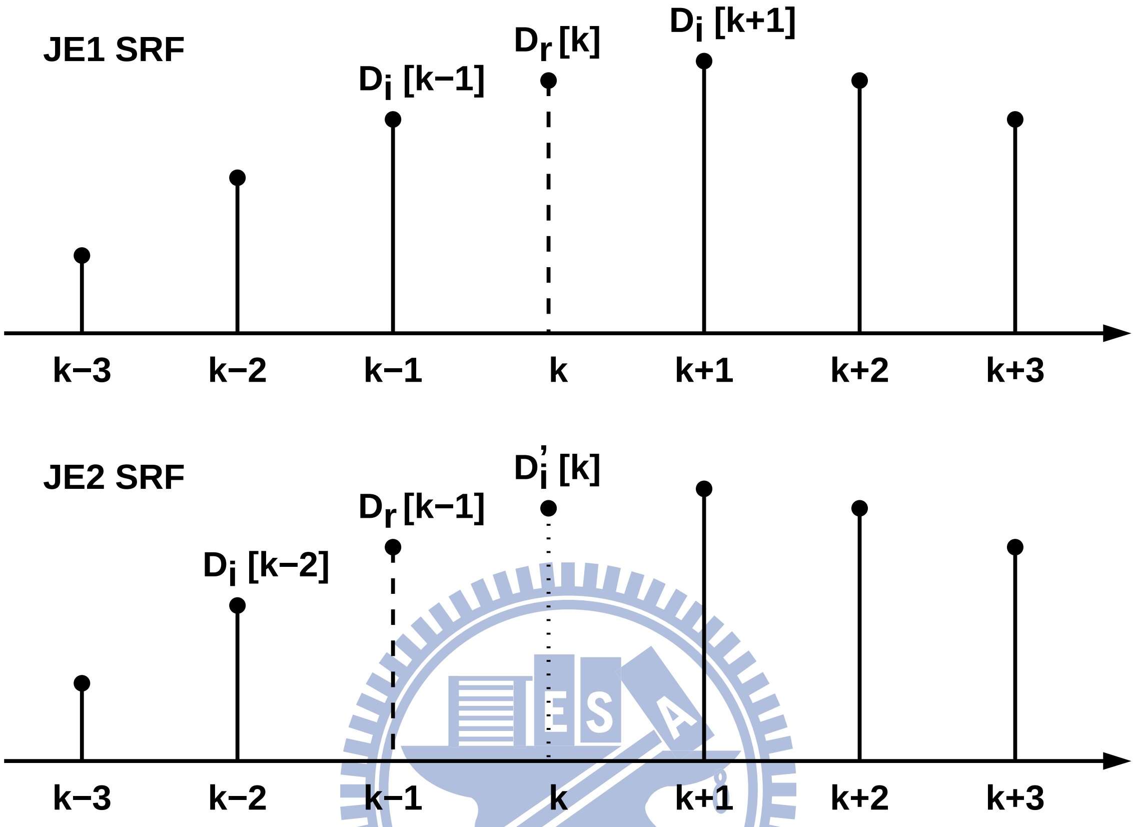


Figure 5.4: Signal reconstruction in JE1 and JE2.

proximated by

$$\begin{aligned}\tau_c[k] \approx & -\frac{\Omega_i}{2F_c(M,\Omega_i)} \times \tau[k] \\ & -\frac{5\Omega_i \cos\Omega_i}{2\pi F_c(M,\Omega_i)} \sin\left(\frac{4\pi}{5}\right) \times \tau[k] \\ & +\left\{\left[\sum_{n=1}^{N} \frac{10}{\pi} \frac{\cos(n\Omega_i) \times \sin\frac{4n\pi}{5}}{n}\right] - 1\right\} \\ & \times \frac{\tan(k\Omega_i + \phi_i) - \tan\left[(k-1)\Omega_i + \phi_i\right]}{2F_c(M,\Omega_i)}\end{aligned} \tag{5.4}$$

The first term on the right-hand side of Equation (5.4) is the desired jitter estimation, which approaches $\tau[k]$ for large M. The third term on the right-hand side has a non-zero mean. However, it is significant only when Ω_i is close to $\Omega_B = (4/5)\pi$ or $\Omega_B = 0$.

The second term on the right-hand side of Equation (5.4) demands special attention. It originates from the fact that, for any specific m, the sampling interval between the

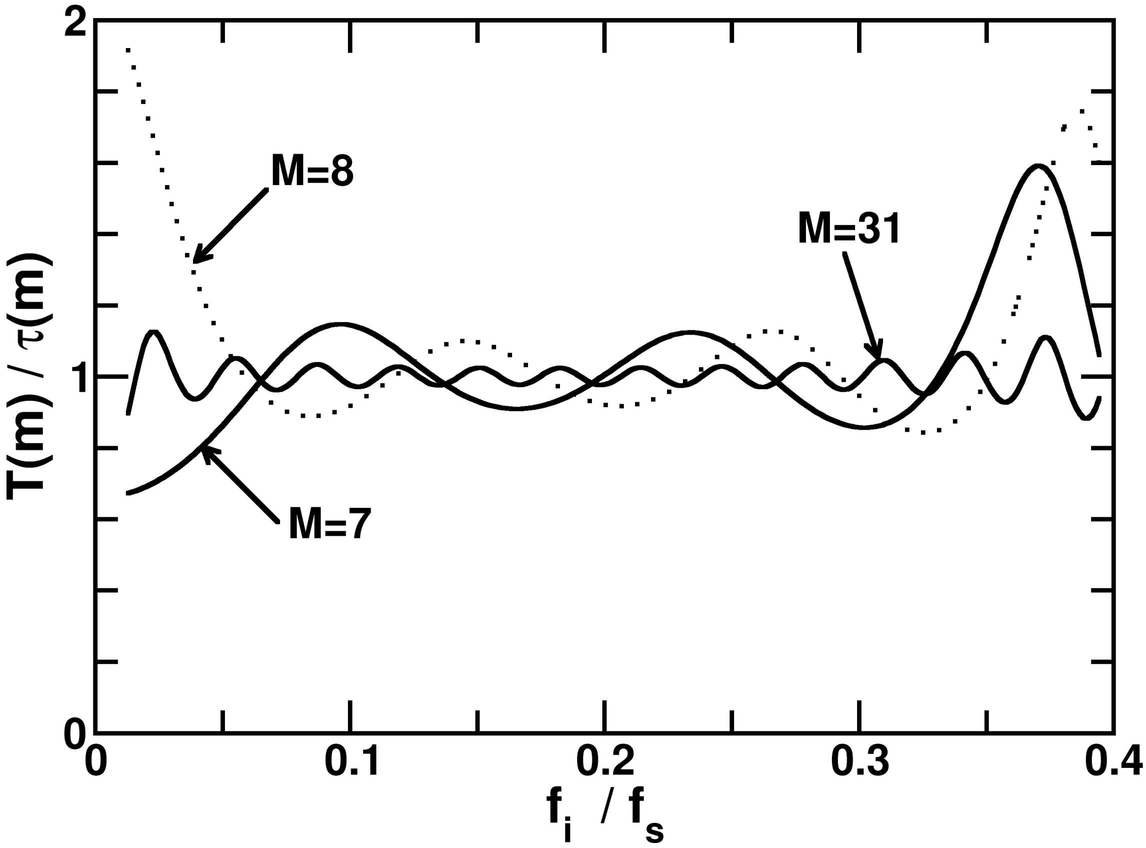


Figure 5.5: Conversion gain of the JDC using JCP2.

reconstructed $D_r[k]$ in JE1 and the reconstructed $D_r[k-1]$ in JE2 is always $T_s + \tau(m)$, as illustrated in Figure 5.4. To solve this constant $\tau(m)$ issue, the $D_i[k]$ sample used in the SRF of JE2 is replaced by an interpolated $D_i'[k]$, which has its sampling time moved backward by $\tau(m)$. The $D_i'[k]$ can be calculated by using

$$D_i'[k] = D_i[k] + \frac{T(m)}{\epsilon_u} \times D_u[k] \tag{5.5}$$

where $D_u[k]$ and ϵ_u come from the JCF shown in Figure 5.1. When the calibration process converges, the resulting $T(m)$ can be expressed as

$$T(m) = \frac{-\pi\Omega_i - 5\Omega_i \cos\Omega_i \times \sin(\frac{4\pi}{5})}{2\pi F_c(M,\Omega_i) - 5\Omega_i \cos\Omega_i \times \sin(\frac{4\pi}{5})} \times \tau(m) \tag{5.6}$$

Figure 5.5 shows the ratio of $T(m)/\tau(m)$ of Equation (5.6). For $M \to \infty$, the conversion gain approaches 1 for all input frequencies up to $f_i = 0.4 f_s$.

In Figure 5.3, the final absolute jitter estimate, $\hat{\epsilon}[k]$, is obtained by the lossy accumulation of $\hat{\tau}[k]$, as expressed in Equation (5.2). Assuming $T(m) = \tau(m)$, from (2.11), (5.1), and (5.2), the RMS of the jitter estimation error can be found as

$$\epsilon_{e,rms}^2 = \frac{1-b}{1+b} \times \epsilon_{rms}^2 + \frac{1}{1-b^2} \times \epsilon_{q,rms}^2 \tag{5.7}$$

For an accurate estimation of the jitter $\epsilon[k]$, it is necessary to choose the b coefficient close to 1. However, as b approaching 1, the accumulation of the low-frequency components in ϵ_q can become the major source of $\epsilon_e[k]$. For a given ϵ_{rms} and $\epsilon_{q,rms}$, the optimal value of b for a minimum $\epsilon_{e,rms}$ is

$$b = \frac{2\epsilon_{rms}^2 + \epsilon_{q,rms}^2 - \sqrt{\epsilon_{q,rms}^4 + 4\epsilon_{rms}^2 \epsilon_{q,rms}^2}}{2\epsilon_{rms}^2} \tag{5.8}$$

Figure 5.6 shows the simulated ADC output spectrum with jitter compensation for different value of b. The frequency of the input signal is $f_i/f_s = 0.249$. The normalized input jitter rms is $\epsilon_{e,rms} = 3 \times 10^{-4}$. The quantization error of the JDC is $\epsilon_{q,rms} = (0.1/\sqrt{12}) \times 10^{-4}$. Before the jitter compensation, the SNR of the ADC is 66.86 dB. For an optimized design of jitter compensation, $b = 0.99$ can be chosen according to Equation (5.8), and the SNR improves to 83.62 dB after jitter compensation.

As described above, the lossy accumulator is adopted to prevent the low-frequency components in ϵ_q overflowing the accumulator. However, the lossy accumulator also filtered out the low-frequency components in the clock jitter $dt[k]$, which results in the skirt around the signal. For $b < 0.99$, the lossy accumulator ignored more low-frequency components in ϵ_q when calculating the absolute jitter from cycle-to-cycle jitter using Equation (5.2). This phenomenon leads to a flatter but wider skirt around the signal on the ADC output power spectrum. As shown in Figure 5.6, the SNR becomes 81.38 dB for $b = 0.95$.

For $b > 0.99$, there are more low-frequency components in ϵ_q been integrated in the accumulator. Thus, for $b = 0.9999$, the skirt around the signal is appeared above the original noise floor, which degrades the SNR to 71.23 dB. Note that although different value of b results in different SNR improvement, the noise floor far from the signal is the same for each case as shown in Figure 5.6. It is results form that the lossy accumulator acts like a low-pass filter, thus it has little effect on the high-frequency components in ϵ_q.

5.3 A 16-bit 80MS/s ADC Design Example

The ADC system of Figure 5.1 is simulated using a C program. System and design parameters are identical to the ADC described in Section 4.6. Since the RMS of jitter $\epsilon[k]$ is $\epsilon_{rms} = 3 \times 10^{-4}$, and the RMS of the TDC quantization noise is $\epsilon_{q,rms} = (1/\sqrt{12}) \times 10^{-4}$, an optimal $b = 0.9$ is chosen for the lossy accumulation of $\hat{\tau}[k]$. The circle symbols in Figure 5.7 are the simulation results. They are the SNRs of the $D_c[k]$ signal under various f_i frequencies. The proposed jitter compensation can improve the SNR by about 10 dB. The theoretical SNR by assuming an ideal JCF for jitter compensation is shown as the dash line. It is calculated by Equation (2.25) with $\epsilon_{e,rms}$ of Equation (5.7).

The SNR can be improved by increasing the resolution of the TDC. Consider a TDC with a uniform quantization step of size reduced to $\Delta t_s = 0.25$ ps. Its RMS of quantization noise then becomes $\epsilon_{q,rms} = (0.25/\sqrt{12}) \times 10^{-4}$. The corresponding optimal value for b is 0.97. The simulation results are shown as the cross symbols in Figure 5.7. The jitter compensation can improve the SNR by about 16 dB. However, to achieve an equivalent RMS of quantization noise $\epsilon_{q,rms} = (0.25/\sqrt{12}) \times 10^{-4}$, the TDC must contain $L = 236$

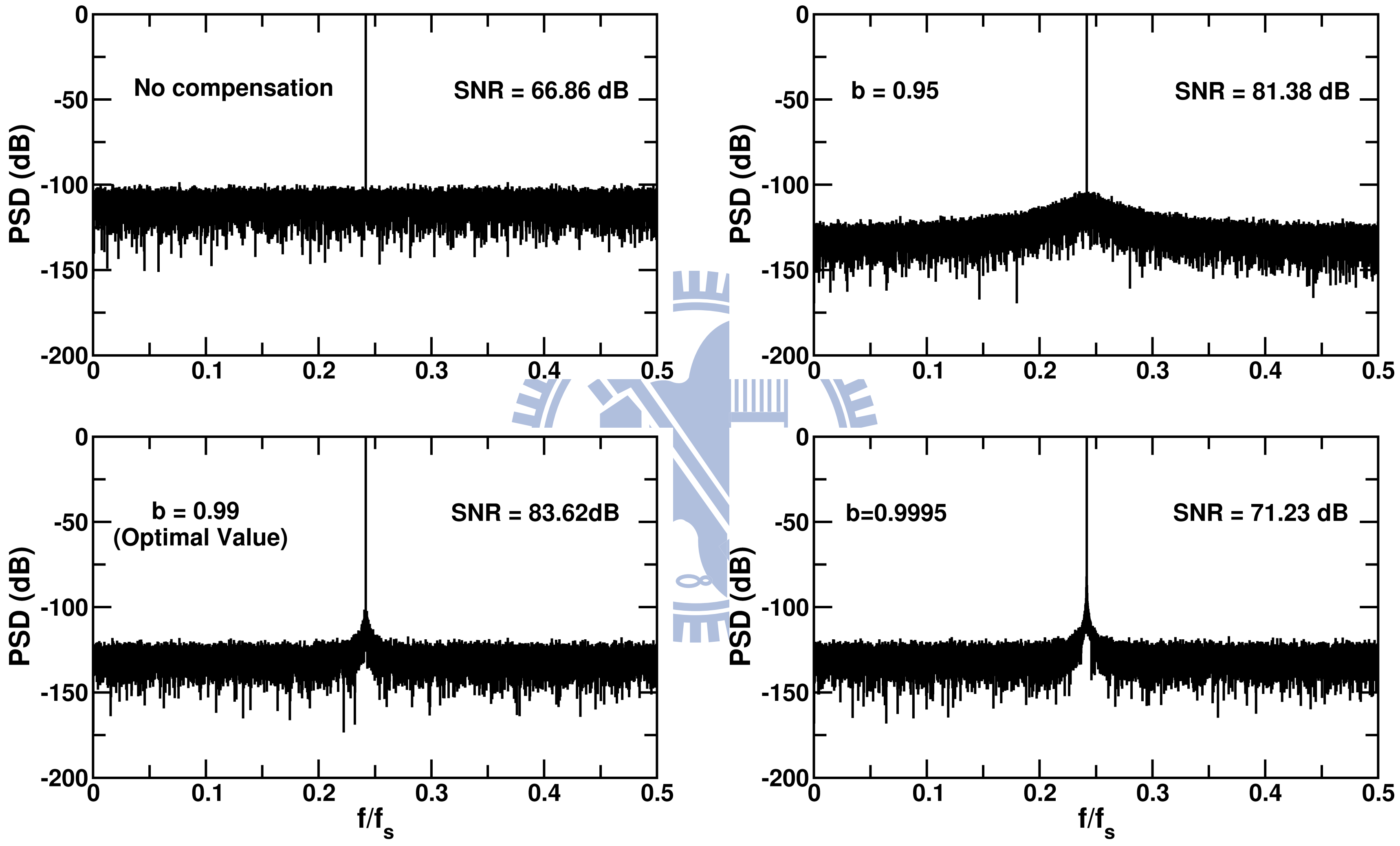


Figure 5.6: Simulated ADC output spectrum with jitter compensation for different value of *b*.

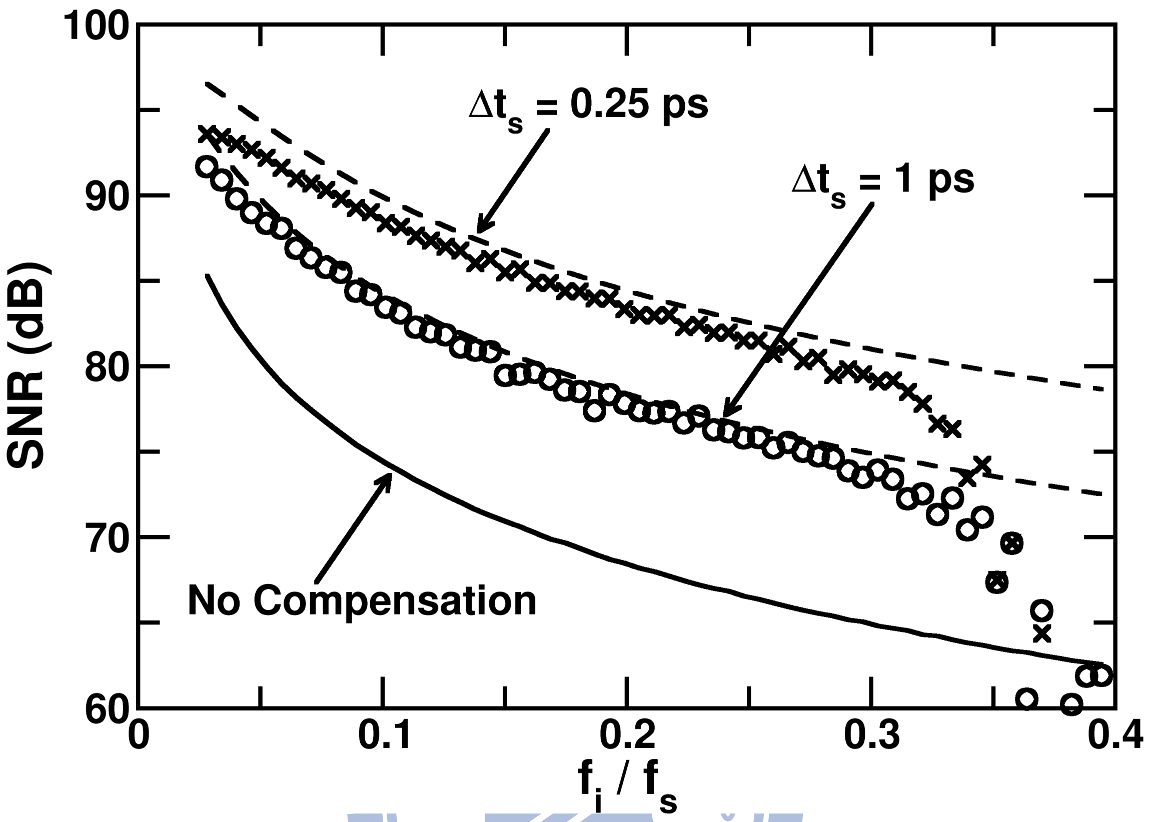


Figure 5.7: Simulated SNRs of the ADC system of Figure 5.1.

TCMPs, if the TCMPs have a t_{os} with standard deviation σ = 6.36 ps.

5.4 Circuit Implementations

Figure 5.8 shows the experimental setup for the jitter compensation. The rising edges of the sampling clock CLK are at $(k + \epsilon[k])T_s$, where k is the sampling index, T_s is the period of CLK and $\epsilon[k]$ is the k-th sampling jitter normalized to T_s. A variable-delay line (VDL) in a delay-locked loop (DLL) provides a delay equal to T_s. The rising edges of the CLK$_\mathrm{d}$ signal are at $(k + \epsilon[k-1] + \epsilon_d[k])T_s$, where $\epsilon_d[k]$ is the k-th jitter induced by the VDL. The TDC measures the relative jitter $\tau[k]$ between CLK and CLK$_\mathrm{d}$ signal, and $\tau[k] = \epsilon[k] - \epsilon[k-1] - \epsilon_d[k]$. Assume that $\epsilon_d[k]$ is small enough, the cycle-to-cycle jitter is measured [35]. The absolute jitter can be obtained by integrating the cycle-to-cycle jitter. Note that $\epsilon_d[k]$ can be considered as part of the JDC measurement error.

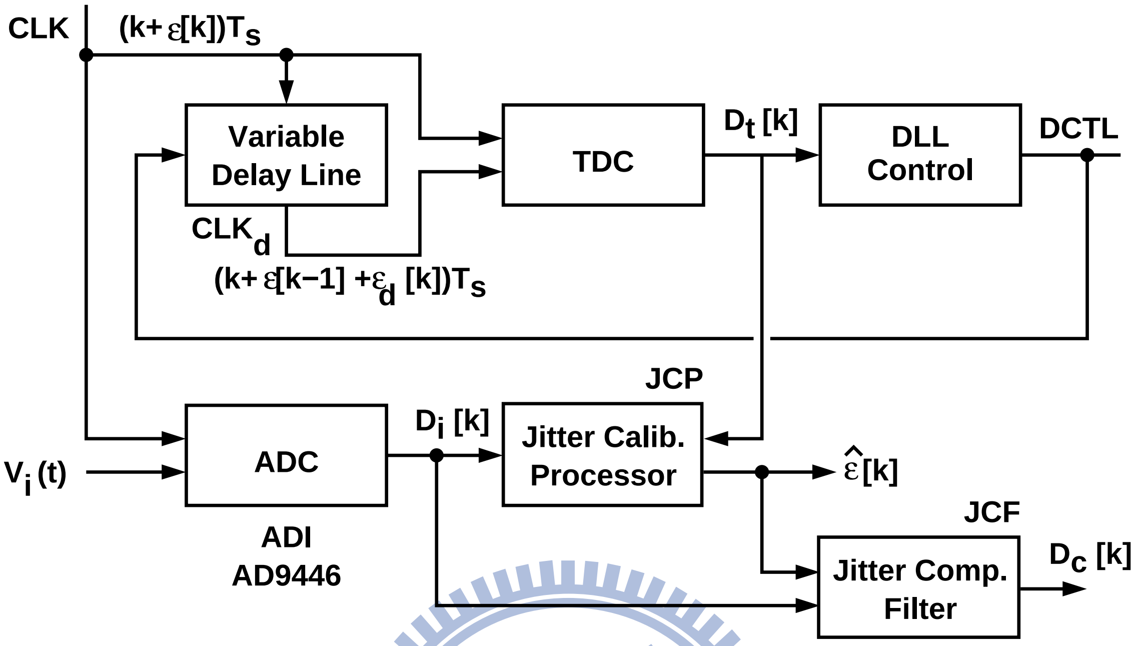


Figure 5.8: Jitter compensation experiment setup.

Using the clock CLK as a sampling time reference, the ADC samples and digitizes the analog input $V_i(t)$ and generates the digital data $D_i[k]$. The ADC output $D_i[k]$ contains a sampling error due to the jitter $\epsilon[k]$. We use a TDC and a jitter calibration processor (JCP) to generate a jitter estimate $\hat{\epsilon}[k]$ for every k. A jitter compensation filter (JCF) then employs $\hat{\epsilon}[k]$ to correct the sampling error in $D_i[k]$. The resulting $D_c[k]$ is a corrected ADC digital output with improved SNR. The TDC also functions as the phase detector of a DLL. The DLL adjusts the VDL delay so that the total delay is equal to T_s.

5.5 Experiment Results

Figure 5.9 shows the chip testing circuit board together with the ADC evaluation board [55]. To decrease the jitter induced by the DLL, a passive cable is used to provide a 6.8 ns delay. Figure 5.10 shows the contents of the JMT after calibration. It also represents the measured TDC transfer function. Figure 5.11 shows the measured ADC output power spectrum before and after the jitter correction for a sine wave input of 25 MHz. The 80 MHz clock signal is generated by Agilent 33250A arbitrary waveform generator. The

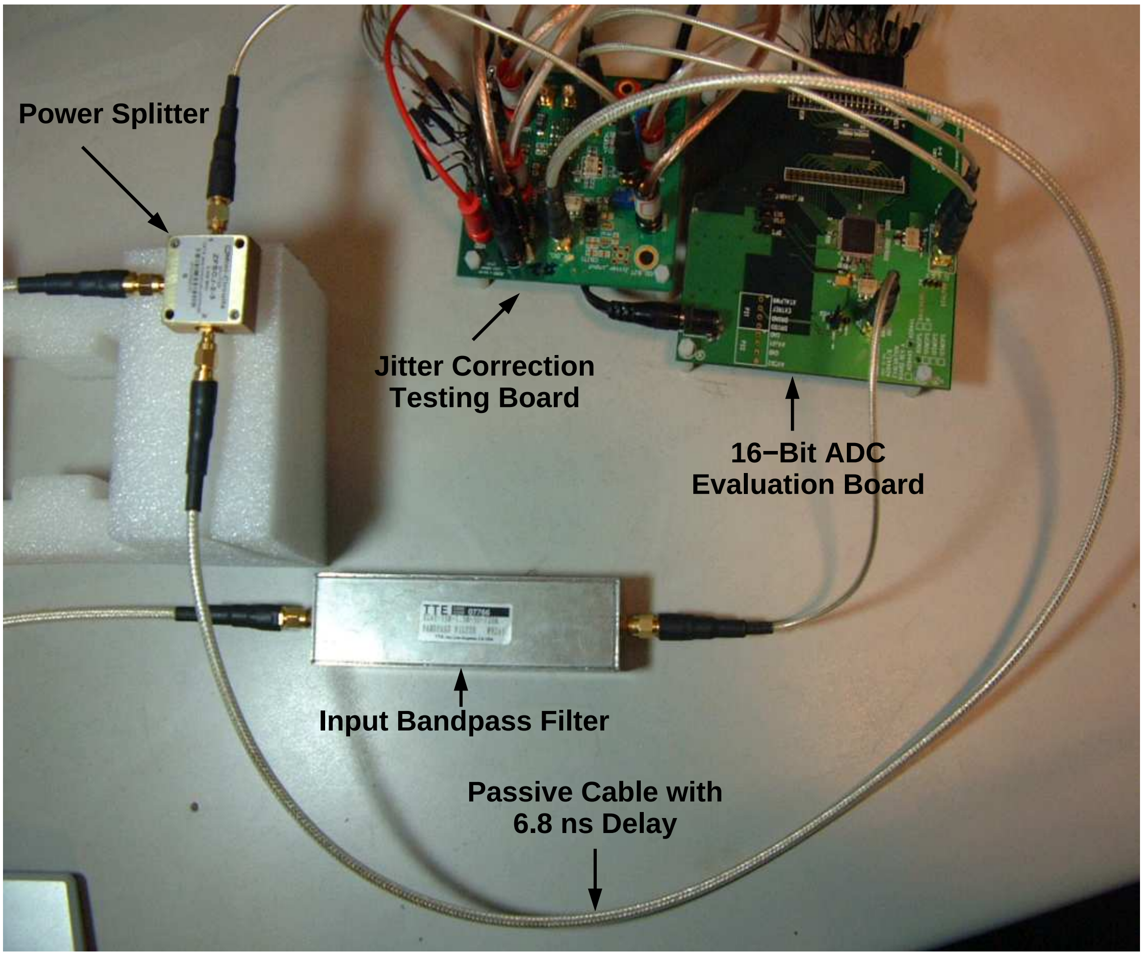

Power Splitter
Jitter Correction Testing Board
16−Bit ADC Evaluation Board
Input Bandpass Filter
Passive Cable with 6.8 ns Delay

Figure 5.9: Photo of the jitter correction setup with the ADC evaluation board.

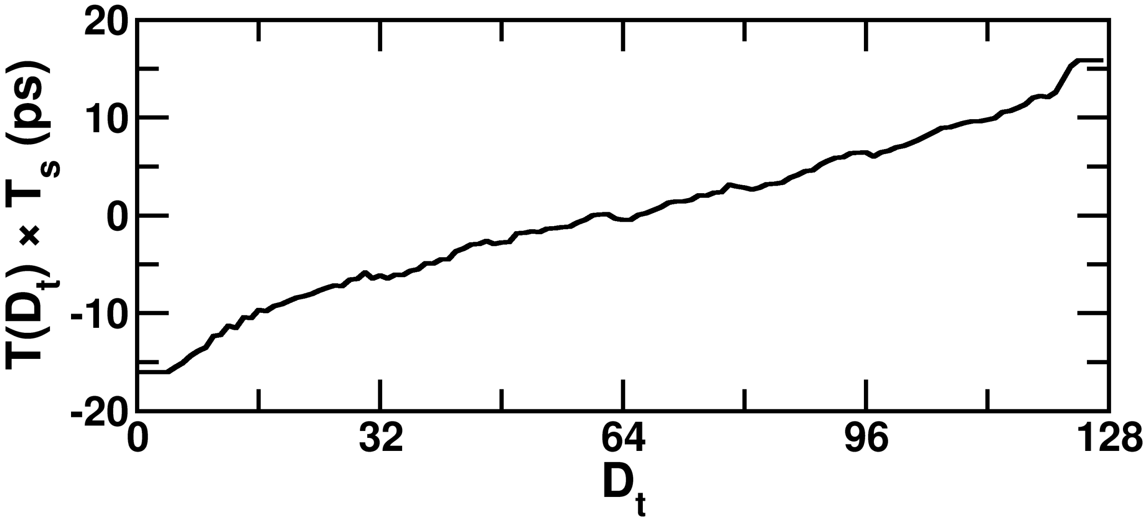


Figure 5.10: Measured TDC transfer function.

jitter rms of the clock is 8.2 ps. The noise floor is decreased after jitter correction. The skirt around the fundamental signal stems from the lossy accumulation when calculating the absolute jitter from cycle-to-cycle jitter. Low frequency components of the jitter will be filtered out by the lossy accumulation, which degrades the jitter correction performance.

Figure 5.12 shows the SNR performance of the ADC before and after jitter correction. The circular symbols are measurement results. The jitter correction improves the ADC SNR from 56.45 dB to 62.55 dB for a 31 MHz sine wave input. After the jitter correction, the equivalent sampling jitter rms decreased to about 4 ps. The VDL introduces additional $\sigma_d = \epsilon_d \times T_s$=1.2 ps rms jitter.

The theoretical ADC SNR after jitter compensation can be expressed as

$$\text{SNR} = \frac{1}{P_T/2 + \omega_i^2 \times (\sigma_q^2 \ + \sigma_d^2)} \tag{5.9}$$

where ω_i is the ADC input frequency, σ_q is the quantization noise power of the TDC, and P_T is the total noise power in the ADC except the sampling error power. $P_T = 5 \times 10^{-9}$ is chosen so that the ideal SNR is equal to 80 dB, The results of the calculation are the solid lines shown in Figure 5.12. In this experiment, the VDL jitter σ_d is much larger than the TDC quantization errors. If σ_d is reduced to a smaller value of 0.3 ps, it is possible to achieve 12 dB SNR improvement. If $\sigma_d = 0$, it is possible to achieve 17 dB SNR improvement.

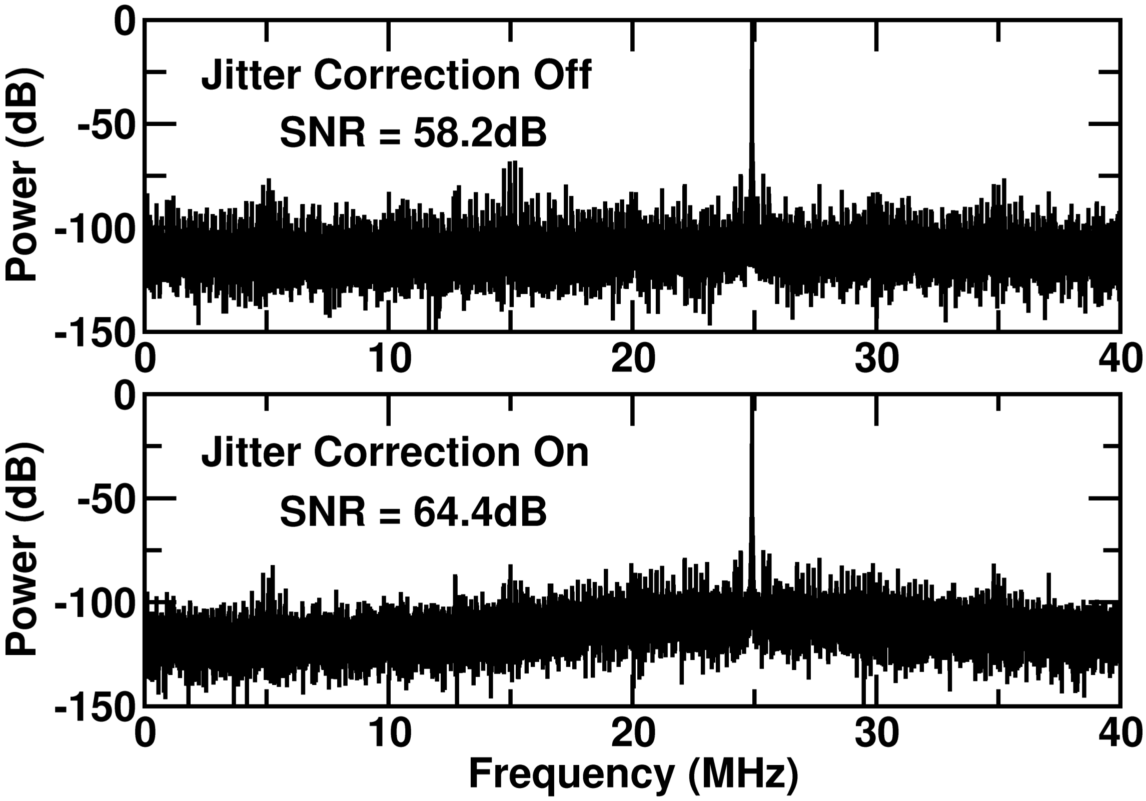


Figure 5.11: Measured ADC output power spectrum before and after the jitter correction.

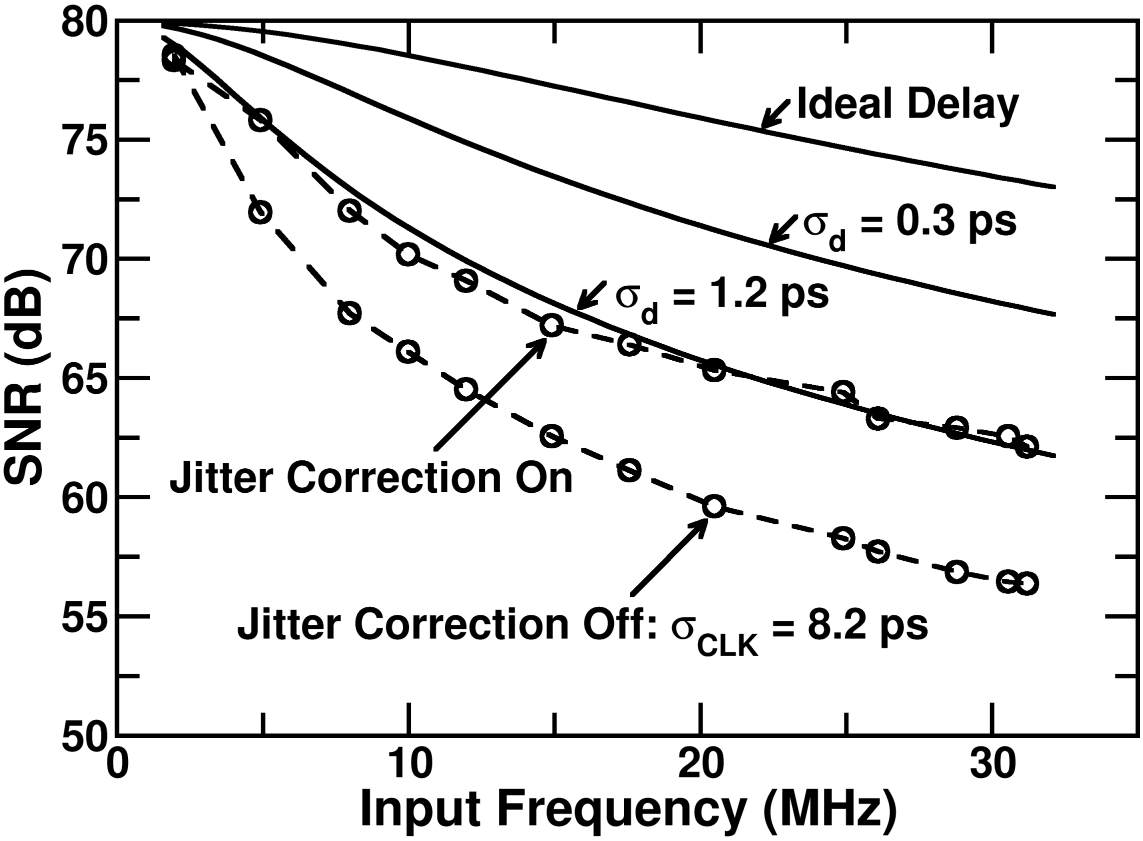


Figure 5.12: Measured ADC SNR versus input frequency.

5.6 Implementation Issues

Consider a 16-bit 80 MS/s ADC. To relax the clock jitter requirement, it employs the jitter compensation system shown in Figure 5.1. For this system, a passive delay line can be used to provide the $T_s = 12.5$ ns delay to generate the CLK$_d$ clock. Functional blocks, such as TDC, JCP2, and JCF, can be realized in a standard CMOS technology. They can also be integrated with the ADC on the same chip.

In order to estimate the overhead of the jitter compensation circuits, assume that the jitter compensation system is realized in a 90 nm CMOS technology. The TDC consists of $L = 127$ TCMPs. The TCMP shown in Figure 3.8 consumes 26.7 μW of power at 80 MHz clock rate. Thus, the TDC consumes a total power of about 3.39 mW. Its chip area is estimated to be $1,000$ μm^2. Note that the resolution and input range of the TDC depend on the PVT variation of the TCMPs. Comprehensive characterization of the fabrication technology is required for optimized design of the TDC.

The JCF is a digital FIR filter, as shown in Figure 2.8. If M=7, it includes 200 flip-flops, 15 multipliers, and 14 adders. Although the $D_i[k]$ signal is 16-bit wide, 8-bit width is sufficient when calculating the $D_u[k]$ signal. The $h_s[n]$ coefficients are 6-bit wide. Synthesized by computer-aided-design (CAD) software, this JCF consumes 2.35 mW of power and occupies an area of $17,000$ μm^2.

The JCP2 is also a digital functional block. It contains a JMT with 128 entries. Each entry is 23-bit wide. The JCP2 shown in Figure 5.3 includes two jitter estimators, JE1 and JE2, for generating $\epsilon_c[k]$ and $\epsilon_c[k-1]$. The two JCFs in JE1 and JE2 and the JCF in Figure 5.1 can share the same hardware. Each jitter estimator contains a SRF. Although the SRF is a FIR filter, to save power and area, it is realized using the MAC architecture. Each SRF consists of a multiplier and an accumulator. The two SRFs share a ROM that stores the filter coefficients $h_r[n]$. The ROM has $N = 2^{10}$ entries, and each entry is 18-bit wide. Due to the MAC operation, each SRF produces only one valid output every $2N + 1$ clock cycles. Synthesized by CAD software, this JCP2 consumes 4.42 mW of power and occupies an area of $185,000$ μm^2. Note that the JMT occupies about 58% of the JCP2's total area. The area and power can be reduced by customized design of the JMT.

Realized in a 90 nm CMOS technology, this jitter compensation system consumes a total power of about 10 mW.

5.7 Summary

A 7-bit stochastic TDC is realized in a 65 nm CMOS technology. The cycle-to-cycle clock jitter is measured by the TDC and the absolute jitter is obtained by integrating the cycle-to-cycle jitter. The measured absolute jitter can be used to correct the ADC sampling errors in the digital domain, thus mitigates the clock jitter requirement.

The proposed jitter compensation achieves an effective sampling jitter rms of 4 ps when the rms jitter of the original input clock is 8.2 ps. The performance of the jitter correction is limited by the jitter of the delay line. It can achieve an effective sampling jitter rms smaller than 1 ps if no additional jitter is induced by the delay line.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

High-performance Nyquist-rate ADCs require low-jitter clocks for input sampling. In this thesis, we have proposed digital signal processing techniques to relax the clock jitter requirement. Enabling techniques are the stochastic TDC and the JCF. The TDC comprises a large group of TCMPs and relies on the statistic variation of the TCMPs for time-to-digital conversion. The TDC is sensitive to the PVT variations. It requires calibration for accurate measurement. Our proposed calibration schemes are based on signal reconstruction. They can be performed in the background without interrupting the normal ADC operation.

All the signal processing described in this paper can be realized using digital circuitry. The timing comparators in the TDC are simple latches and can be realized in any standard CMOS technology. Thus, both the chip area and power consumption of the proposed techniques are expected to be reduced as CMOS technologies advance.

Finally, a 7-bit 80-MS/s TDC is fabricated to demonstrate the feasibility of the proposed technique. A 16-bit 80-MS/s ADC adopts the proposed jitter compensation technique can achieve 14 dB SNR improvement when a clean reference clock is available, or 7 dB SNR improvement with a jittering reference clock.

6.2 Recommendations for Future Investigation

This section presents several suggestions for future investigations in jitter compensation for an ADC.

- The background TDC calibration techniques proposed in this thesis need long calibration time to converge. TDC calibration with fewer samples is necessary to make this technique more feasible in industry application.

- The measured SNR performance of the implemented jitter compensation technique in Chapter 4 is better than the performance of the jitter compensation technique in Chapter 5. It stems from the jitter accumulation when calculating the absolute jitter from cycle-to-cycle jitter, thus the jitter induced by the delay line is also accumulated. Minimize the delay provided by the internal delay line can mitigate this jitter accumulation effect, thus further improving the SNR performance.

- As mention above that the jitter compensation scheme with clean external clock can achieve better SNR performance since there is no jitter accumulation phenomenon. Therefore, this technique is especially suitable for time-interleaved ADCs with multiphase clock generator for several reasons.

  - Time-interleaved ADC is used to multiply the sampling rate, increasing the analog input bandwidth. As the input bandwidth increased, the jitter requirement is also increased.

  - The multiphase clocks are usually generated by a PLL or a DLL. The use of PLL or DLL will induce additional clock jitter inevitable even when the reference clock is clean.

  - If the reference clock is clean, the absolute jitter can be measured directly. Jitter compensation technique in Chapter 4 can be used to obtain a better SNR improvement. There is no jitter accumulation effect mentioned in Chapter 5 if absolute jitter is measured.

Appendix A

In this appendix, we derive the equation

$$\lim_{M\to\infty} F_c(M,\Omega) = -\frac{\Omega}{2} \tag{A.1}$$

where

$$F_c(M,\Omega) = \sum_{n=1}^{M} \frac{\sin(n\Omega)}{(-1)^n \times n} \tag{A.2}$$

Considering the Maclaurin Series expansion for $-\ln(l+x)$

$$\begin{aligned} -\ln(1+x) &= -x + \frac{x^2}{2} - \frac{x^3}{3} + \frac{x^4}{4} + \ldots \\ &= \lim_{M\to\infty} \sum_{n=1}^{M} \frac{x^n}{(-1)^n \times n} \end{aligned} \tag{A.3}$$

Comparing (A.2) and (A.3), we can find

$$\lim_{M\to\infty} \sum_{n=1}^{M} F_c(M,\Omega) = \Im\left\{ \sum_{n=1}^{M} \frac{e^{j\Omega}}{(-1)^n \times n}) \right\} = -\Im\left\{ \ln(1+e^{j\Omega}) \right\} \tag{A.4}$$

Since

$$\ln(1+e^{j\Omega}) = \ln(re^{j\phi}) = \ln(r) + j\phi \tag{A.5}$$

with

$$\begin{aligned} r &= 2 + 2\cos(\Omega) \\ \phi &= \tan^{-1}\frac{\sin\Omega}{1+\cos\Omega} = \frac{\Omega}{2} \end{aligned} \tag{A.6}$$

Therefore, (A.4) can be rewriten as

$$\lim_{M\to\infty}\sum_{n=1}^{M} F_c(M,\Omega) = -\phi = -\frac{\Omega}{2} \tag{A.7}$$

Bibliography

[1] J. Li and U.-K. Moon, “Background calibration techniques for multistage pipelined ADCs with digital redundancey,” *IEEE Trans. Circuits Syst. II*, vol. 50, no. 9, pp. 531–538, September 2003.

[2] B. Murmann and B. E. Boser, “A 12-bit 75-MS/s pipelined ADC using open-loop residue amplification,” *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2040–2050, December 2003.

[3] E. Siragusa and I. Galton, “A digitally enhanced 1.8-V 15-bit 40-MSample/s CMOS pipelined ADC,” *IEEE J. Solid-State Circuits*, vol. 39, no. 12, pp. 2126–2138, December 2004.

[4] H.-C. Liu, Z.-M. Lee, and J.-T. Wu, “A 15-b 40-MS/s CMOS pipelined analog-to-digital converter with digital background calibration,” *IEEE J. Solid-State Circuits*, vol. 40, no. 5, pp. 1047–1056, May 2005.

[5] J. P. Keane, P. J. Hurst, and S. H. Lewis, “Background interstage gain calibration technique for pipelined ADCs,” *IEEE Trans. Circuits Syst. I*, vol. 52, no. 1, pp. 32–43, January 2005.

[6] A. Panigada and I. Galton, “Digital background correction of harmonic distortion in pipelined ADCs,” *IEEE Trans. Circuits Syst. I*, vol. 53, no. 9, pp. 1885–1895, September 2006.

[7] J.-L. Fan, C.-Y. Wang, and J.-T. Wu, “A robust and fast digital background calibration technique for pipelined ADCs,” *IEEE Trans. Circuits Syst. I*, vol. 54, no. 6, pp. 1213–1223, June 2007.

[8] H. V. de Vel, B. A. J. Buter, H. van der Ploeg, M. Vertregt, G. J. G. M. Geelen, and E. J. F. Paulus, "A 1.2-V 250-mW 14-b 100-MS/s digitally calibrated pipeline ADC in 90-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1047–1056, 2009.

[9] C.-C. Huang and J.-T. Wu, "A background comparator calibration technique for flash analog-to-digital converters," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 9, pp. 1732–1740, September 2005.

[10] G. V. der Plas, S. Decoutere, and S. Donnay, "A 0.16pJ/conversion-step 2.5mW 1.25GS/s 4b ADC in a 90nm digital CMOS process," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, February 2006, pp. 566–567.

[11] S. Park, Y. Palaskas, and M. P. Flynn, "A 4GS/s 4b flash ADC in 0.18μm CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, February 2006, pp. 570–571.

[12] Y.-Z. Lin, C.-W. Lin, and S.-J. Chang, "A 2-GS/s 6-bit flash ADC with offset calibration," in *IEEE Asian Solid-State Circuits Conference*, November 2008, pp. 385–388.

[13] Y.-H. Chung and J.-T. Wu, "A CMOS 6-mW 10-bit 100-MS/s two-step ADC," in *IEEE Asian Solid-State Circuits Conference*, November 2009, pp. 137–140.

[14] M. Shinagawa, Y. Akazawa, and T. Wakimoto, "Jitter analysis of high-speed sampling systems," *IEEE J. Solid-State Circuits*, vol. 25, no. 1, pp. 220–224, February 1990.

[15] H. Kobayashi, M. Morimura, K. Kobayashi, and Y. Onaya, "Aperture jitter effects in wideband sampling systems," in *Proc. IEEE Instrumentation and Measurement Tech. Conf.*, vol. 2, May 1999, pp. 880–885.

[16] M. Gustavsson, J. J. Wikner, and N. N. Tan, *CMOS Data Converters for Communications*. Boston, MA: Kluwer, 2000.

[17] B. Brannon and R. Reeder, *Understanding High Speed ADC Testing and Evaluation*. Norwood, MA: Analog Devices, Inc., Application Note AN-835, April 2006.

[18] K. Lim, C.-H. Park, D.-S. Kim, and B. Kim, “A low-noise phase-locked loop design by loop bandwidth optimization,” *IEEE J. Solid-State Circuits*, vol. 35, no. 6, pp. 807–815, June 2000.

[19] C.-H. Lee, K. McClellan, and J. John Choma, “A supply-noise-insensitive CMOS PLL with a voltage regulator using DC–DC capacitive converter,” *IEEE J. Solid-State Circuits*, vol. 36, no. 10, pp. 1453–1463, October 2001.

[20] V. von Kaenel, D. Aebischer, C. Piguet, and E. Dijkstra, “A 320 MHz, 1.5 mW@1.35V CMOS PLL for microprocessor clock generation,” *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1715–1722, November 1996.

[21] S. Sidiropoulos, D. Liu, J. Kim, G. Wei, and M. Horowitz, “Adaptive bandwidth DLLs and PLLs using regulated supply CMOS buffers,” in *Symposium on VLSI Circuits Digest of Technical Papers*, June 2000, pp. 124–127.

[22] M. Brownlee, P. K. Hanumolu, K. Mayaram, and U.-K. Moon, “A 0.5-GHz to 2.5-GHz PLL with fully differential supply regulated tuning,” *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2720–2728, December 2006.

[23] E. Alon, J. Kim, S. Pamarti, K. Chang, and M. Horowitz, “Replica compensated linear regulators for supply-regulated phase-locked loops,” *IEEE J. Solid-State Circuits*, vol. 41, no. 2, pp. 413–424, Feburary 2006.

[24] T. H. Lee and A. Hajimiri, “Oscillator phase noise: A tutorial,” *IEEE J. Solid-State Circuits*, vol. 35, no. 3, pp. 326–336, March 2000.

[25] R. Aparicio and A. Hajimiri, “A noise-shifting differential Colpitts VCO,” *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1728–1736, December 2002.

[26] L. Dai and R. Harjani, “Design of low-phase-noise CMOS ring oscillators,” *IEEE Trans. Circuits Syst. II*, vol. 49, no. 5, pp. 328–338, May 2002.

[27] J. Tourabaly and A. Osseiran, “A jittered-sampling correction technique for ADCs,” in *IEEE International Symposium on Electronic Design, Test and Applications*, January 2008, pp. 249–252.

[28] V. Gutnik and A. Chandrakasan, "On-chip picosecond time measurement," in *Symposium on VLSI Circuits Digest of Technical Papers*, June 2000, pp. 52–53.

[29] T. M. Apostol, "A proof that Euler missed: evaluating $\zeta(2)$ the easy way," *The Mathematical Intelligencer*, vol. 5, no. 3, pp. 59–60, September 1983.

[30] P. Napolitano, A. Moschitta, and P. Carbone, "A survey on time interval measurement techniques and testing methods," in *IEEE Instrumentation and Measurement Technology Conference (I2MTC)*, May 2010, pp. 181–186.

[31] P. Dudek, S. Szczepański, and J. V. Hatfield, "A high-resolution CMOS time-to-digital converter utilizing a vernier delay line," *IEEE J. Solid-State Circuits*, vol. 35, no. 2, pp. 240–247, February 2000.

[32] J.-P. Jansson, A. Mäntyniemi, and J. Kostamovaara, "A CMOS time-to-digital converter with better than 10 ps single-shot precision," *IEEE J. Solid-State Circuits*, vol. 41, no. 6, pp. 1286–1296, June 2006.

[33] A. K. M. K. Mollah, R. Rosales, S. Tabatabaei, J. Cicalo, and A. Ivanov, "Design of a tunable differential ring oscillator with short start-up and switching transients," *IEEE J. Solid-State Circuits*, vol. 54, no. 12, pp. 2669–2682, December 2007.

[34] T. Xia and J.-C. Lo, "Time-to-voltage converter for on-chip jitter measurement," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 6, pp. 1738–1748, December 2003.

[35] M. Ishida, K. Ichiyama, T. J. Yamaguchi, M. Soma, M. Suda, T. Okayasu, D. Watanabe, and K. Yamamoto, "A programmable on-chip picosecond jitter measurement circuit without a reference-clock input," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, February 2005, pp. 512–513.

[36] J. Kalisz, "Review of methods for time interval measurements with picosecond resolution," *Metrologia*, vol. 41, pp. 17–32, 2004.

[37] *Fundamentals of Time Interval Measurements*. Santa Clara, CA: Agilent Technologies, Inc, Application Note 200-3, June 1997.

[38] J. Kostamovaara and R. Myllylä, "Time-to-digital converter with an analog interpolation circuit," *Review of Scientific Instruments*, vol. 57, no. 11, pp. 2880–2885, November 1986.

[39] E. Räisänen-Ruotsalainen, T. Rahkonen, and J. Kostamovaara, "A time digitizer with interpolation based on time-to-voltage conversion," in *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, vol. 1, August 1997, pp. 197–200.

[40] K. Määttä and J. Kostamovaara, "A high-precision time-to-digital converter for pulsed time-of-flight laser radar applications," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 2, pp. 521–536, April 1998.

[41] J.-P. Jansson, A. Mäntyniemi, and J. Kostamovaara, "A CMOS time-to-digital converter with better than 10 ps single-shot precision," *IEEE J. Solid-State Circuits*, vol. 41, no. 6, pp. 1286–1296, June 2006.

[42] M. Lee and A. A. Abidi, "A 9 b, 1.25 ps resolution coarse-fine time-to-digital converter in 90 nm CMOS that amplifies a time residue," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 769–777, April 2008.

[43] J. Yu, F. F. Dai, and R. C. Jaeger, "A 12-bit vernier ring time-to-digital converter in 0.13μm CMOS technology," in *Symposium on VLSI Circuits Digest of Technical Papers*, June 2009, pp. 232–233.

[44] R. Rashidzadeh, M. Ahmadi, and W. C. Miller, "An all-digital self-calibration method for a vernier-based time-to-digital converter," *IEEE Trans. Instrum. Meas.*, vol. 2, no. 2, pp. 463–469, February 2010.

[45] P. M. Levine and G. W. Roberts, "A high-resolution flash time-to-digital converter and calibration scheme," in *Proceedings of International Test Conference*, October 2004, pp. 1148–1157.

[46] R. B. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P. T. Balsara, "1.3 V 20 ps time-to-digital converter for frequency synthesis in 90-nm CMOS," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 3, pp. 220–224, March 2006.

[47] T. Hashimoto, H. Yamazaki, A. Muramatsu, T. Sato, and A. Inoue, “Time-to-digital converter with vernier delay mismatch compensation for high resolution on-die clock jitter measurement,” in *Symposium on VLSI Circuits Digest of Technical Papers*, June 2008, pp. 166–167.

[48] V. Kratyuk, P. K. Hanumolu, K. Ok, U.-K. Moon, and K. Mayaram, “A digital PLL with a stochastic time-to-digital converter,” *IEEE Trans. Circuits Syst. I*, vol. 56, no. 8, pp. 1612–1621, August 2009.

[49] C.-C. Huang, “Digitally-calibrated comparator and its application in flash analog-to-digital converters,” Ph.D. dissertation, Department of Electronics Engineering and Institute of Electronics, National Chiao-Tung University, Hsin-Chu, Taiwan, January 2010.

[50] J. Rivoir, “Fully-digital time-to-digital converter for ATE with autonomous calibration,” in *EEE International Test Conference*, October 2006, pp. 1–10.

[51] ——, “Statistical linearity calibration of time-to-digital converters using a free-running ring oscillator,” in *15th Asian Test Symposium*, November 2006, pp. 45–50.

[52] I. Nissinen and J. Kostamovaara, “On-chip voltage reference-based time-to-digital converter for pulsed time-of-flight laser radar measurements,” *IEEE Trans. Instrum. Meas.*, vol. 58, no. 6, pp. 1938–1948, June 2009.

[53] F. Baronti, D. Lunardini, R. Roncella, and R. Saletti, “A non-linearity self-calibration technique for delay-locked loop delay-lines,” in *IEEE Instrumentation and Measurement Technology Conference*, May 2002, pp. 1007–1010.

[54] M.-J. E. Lee, W. J. Dally, T. Greer, H.-T. Ng, R. Farjad-Rad, J. Poulton, and R. Senthinathan, “Jitter transfer characteristics of delay-locked loops - theories and design techniques,” *IEEE J. Solid-State Circuits*, vol. 38, no. 4, pp. 614–621, April 2003.

[55] Analog Devices, “AD9446 Datasheet,” [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/AD9446.pdf.

自傳

Chi-Wei Fan was born in Hsin-Chu, Taiwan, R.O.C.. He received the B.S degree in electrical engineering from the National Central University, Chung Li, Taiwan, in 2000. From 2001 to 2010, he worked toward the Ph.D. degree in electronics engineering with the National Chiao-Tung University, Hsinchu. His research interests in mixed-signal, high speed and high resolution integrated circuits design in data communication. In 2010, he joined the MediaTek where he was engaged in the design of analog and mixed-signal ICs.

住址: 新竹縣竹北市國強街 88 號

本論文使用 LaTeX[1] 系統排版.

[1] LaTeX 是 TeX 之下的 macros 集. TeX 是 American Mathematical Society 的註冊商標. 本論文 macros 的原始作者是 Dinesh Das, Department of Computer Sciences, The University of Texas at Austin. 交大中文版的作者是吳介琮, 交通大學電子工程學系, 新竹, 台灣.

Publication List

- Journal Paper:
  - **C.-W. Fan** and J.-T. Wu, "Jitter Measurement and Compensation for Analog-to-Digital Converters," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 11, pp. 3874–3884, November 2009.

- Conference Paper:
  - **C.-W. Fan** and J.-T. Wu, "ADC Clock Jitter Measurement and Correction Using a Stochastic TDC," accepted by *2010 IEEE Asia Pacific Conference on Circuits and Systems Program Committee*.
  - W.-H. Tseng, **C.-W. Fan** and J.-T. Wu, "A 12-Bit 1.25-GS/s DAC in 90nm CMOS with $>$ 70dB SFDR up to 500MHz," submitted to *2011 IEEE International Solid-State Circuits Conference*.