

2001

Design techniques and implementations of high-speed analog communication circuits: two analog-to-digital converters and a 3.125Gb/s receiver

Ahmed Abdell-Ra'ouf Younis
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Electrical and Electronics Commons](#)

Recommended Citation

Younis, Ahmed Abdell-Ra'ouf, "Design techniques and implementations of high-speed analog communication circuits: two analog-to-digital converters and a 3.125Gb/s receiver " (2001). *Retrospective Theses and Dissertations*. 1419.
<https://lib.dr.iastate.edu/rtd/1419>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

Design Techniques and Implementations of High-speed Analog Communication Circuits:
Two Analog-to-Digital Converters and a 3.125Gb/s Receiver

by

Ahmed Abdell-Ra'ouf Younis

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee:
Marwan Hassoun, Major Professor
William Black
Chris Chu
Gurpur Prabhu
Robert Weber

Iowa State University

Ames, Iowa

2001

Copyright © Ahmed Abdell-Ra'ouf Younis, 2001. All rights reserved.

UMI Number: 3098488

UMI[®]

UMI Microform 3098488

Copyright 2003 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Ahmed Abdell-Ra'oof Younis

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

TABLE OF CONTENTS

Abstract	viii
CHAPTER 1. Introduction	1
CHAPTER 2. Terminology And Metrics.....	3
2.1. Introduction	3
2.2. ADC definition	3
2.3. ADC Characteristics	5
2.3.1. Resolution and Accuracy	6
2.3.2. Bins and Trip Points	7
2.3.3. Gain and Offset.....	8
2.3.4. INL and DNL.....	11
2.3.5. SNR, SNDR and ENOB	13
2.3.6. Dynamic Range and SFDR.....	15
2.3.7. Latency	15
2.3.8. Aperture Jitter	15
2.3.9. PSRR (Power Supply Rejection Ratio).....	15
2.3.10. THD	15
2.4. Conclusions	16
References	17
CHAPTER 3. ADC Architectures	18
3.1. Introduction	18
3.2. Flash ADCs	18
3.2.1. Advantages	20
3.2.2. Limitations.....	20
3.3. Two-step Flash ADCs.....	21
3.4. Folding ADCs.....	22
3.5. Multistep ADCs.....	25
3.6. Successive Approximation and Algorithmic converters ADCs	26
3.6.1. Successive Approximation ADCs	26
3.6.2. Algorithmic Converters ADCs	27
3.7. Pipeline ADCs	27
3.8. Parallel Pipeline ADCs.....	28

3.9. Oversampling ADCs.....	29
3.9.1. Sigma Delta Modulation.....	30
3.10. Conclusions	31
References	33
CHAPTER 4. Pipeline ADCs.....	35
4.1. Introduction	35
4.2. Pipeline Building Blocks	42
4.2.1. Operational Amplifier.....	42
4.2.2. Common Mode feedback circuit.....	44
4.2.3. CMOS Comparator design	44
4.3. Error Sources in Pipeline ADCs	46
4.3.1. Capacitor Mismatch.....	49
4.3.2. Comparator Offsets.....	51
4.3.3. Thermal noise	53
4.3.4. Charge injection and Clock feedthrough	54
4.4. Channel-related errors	56
4.4.1. Channel Gain mismatches	56
4.4.2. Channel offset mismatches	58
4.4.3. Timing mismatch and Jitter	59
4.5. Conclusions	61
References	62
CHAPTER 5. Implementation Of 10-Bit And 100 Ms/S Pipeline ADC.....	63
5.1. Introduction	63
5.2. 10-bit Pipeline ADC	63
5.2.1. Operation of one stage of the ADC	65
5.2.2. Operation of the subADC	68
5.2.3. Operation of the comparators	69
5.2.4. Digital error correction	71
5.2.5. Implementation of one stage.....	74
5.2.6. Design of the operational amplifier	76
5.2.7. Comparator implementation	82
5.2.8. Capacitor design in TSMC 0.25u Process	84
5.3. Layout.....	85
5.3.1. Operational amplifier layout.....	85
5.3.2. Stage layout	85

5.3.3. Overall Layout	88
5.4. Testing Results	91
5.5. Conclusions	95
References	96
CHAPTER 6. ADC Error Correction And Calibration.....	97
6.1. Introduction	97
6.2. Analog versus digital calibration	98
6.2.1. Over-range and under-range stages	98
6.2.2. 1.5-bit per stage	98
6.2.3. Stage gain < 2	99
6.2.4. Pros and cons for the above designs	99
6.3. Capacitor Error-Averaging	100
6.4. Continuous calibration.....	101
6.5. Proposed Single Path calibration algorithm.....	104
6.5.1. Overview	104
6.5.2. Comparator offsets.....	107
6.5.3. Correction Algorithm.....	111
6.5.4. Gain and DAC measurement algorithm.....	119
6.6. Multipath Calibration.....	123
6.6.1. Gain error randomization.....	123
6.6.2. Channel normalization.....	123
6.6.3. Hardware sharing	123
6.6.4. Digital Calibration	124
References	127
CHAPTER 7. VCO-Based ADCs	129
7.1. Introduction	129
7.2. Architecture	130
7.3. Frequency Detector Circuit.....	132
7.3.1. Introduction and overview	132
7.3.2. Proposed solution	132
7.3.3. Implementation	139
7.3.4. Derivation of Maximum Error in the FD Measurement	142
7.3.5. Example	143
7.4. The mapping Circuit.....	148
7.5. ADC Overall Picture.....	149

7.6. Summary.....	149
References	151
CHAPTER 8. High Speed Receiver Design	152
8.1. Introduction	152
8.2. Architecture	153
8.2.1. The Deserializer.....	154
8.2.2. The Coarse Loop	155
8.2.3. The Fine Loop	156
8.2.4. The Gm Circuit	158
8.3. Performance Enhancement	161
8.3.1. VCO Jitter Minimization	161
8.3.2. Buffer Separation.....	162
8.3.3. Power Supply Noise Reduction	163
8.4. Top Level Receiver Analog Simulations	165
8.5. Receiver Development.....	167
8.5.1. VCO layout.....	169
8.5.2. Parasitic insensitive clocking scheme	172
8.6. Test Setup	176
8.7. Measured Results.....	177
8.7.1. Jitter Tolerance	177
8.8. Summary And Conclusions	179
References	180
CHAPTER 9. Design Techniques and Engineering Practice For High-Speed Analog ICs.....	181
9.1. Introduction	181
9.2. Design of the operational amplifier	181
9.2.1. Operational Amplifier: Theoretical Analysis.....	182
9.2.2. Operational Amplifier: Practical Design	185
9.2.3. CMFB circuit design of the main amplifier	187
9.2.4. CMFB circuit design of the boosting amplifiers.....	187
9.3. Comparator implementation	188
9.4. Metal capacitor design.....	190
9.5. Thermal noise	192
9.6. Charge injection.....	193
9.7. Clock feedthrough	195
9.8. Gm cells.....	195

9.8.1. Designing for Figures of Merit and Simulation Results	196
9.9. Parasitic insensitive clocking Scheme	198
9.10. Layout.....	198
9.10.1. Operational amplifier layout.....	198
9.10.2. ADC Stage layout.....	198
9.10.3. Overall Layout.....	200
9.11. Conclusions	202
References	203
CHAPTER 10. Conclusions	204

Abstract

Low-cost and high performance analog building blocks are essentials to the realization of today's high-speed networking and communications systems. Two such building blocks are analog-to-digital converters (ADCs) and multi-gigabit per second transceivers. The ADCs are paramount to translating the real world analog signals into the digital processing world. The multi-gigabit transceivers are becoming a necessity for high-speed systems and chips to transfer the enormous amount of digital data between each other. This thesis addresses two different ADC architectures and a 3.125Gb/s receiver architecture.

The first ADC architecture is a 10-bit, 100MS/s pipeline ADC. Techniques that enhance the gain-bandwidth of the operational amplifier, a key building block in analog-to-digital converters, as well as to increase its dc gain are presented. Layout techniques to reduce the effect of parasitics on the performance of the ADC are also discussed. Since any ADC will have inherent errors in it, two calibration techniques that reduce the effect of these errors on the performance of the ADC are also presented.

In this thesis, the design of the ADC as well as the implementation of those techniques will be presented and discussed.

For the second ADC, a new architecture is proposed that is capable of achieving higher performance than many current ADC architectures. The new architecture is based on a voltage controlled oscillator and a frequency detector. One reason for the high performance of the new ADC is the novel design of the frequency detector. This thesis includes detailed analysis as well as examples to illustrate the operation of the frequency detector.

Designing high-speed CMOS transceivers is a challenging process, especially, when using digital CMOS process that exhibits poor analog performance. Circuit implementation and design techniques that are used to design and enhance the performance of the receiver block of a 3.125Gb/s transceiver in a 0.18 μ m digital CMOS process will be presented and fully explained in this thesis. Silicon results have shown that these techniques have resulted in outstanding and very robust receiver performance under different operating conditions.

The thesis also includes a chapter on design techniques and engineering practices for high speed analog ICs. These techniques were used extensively in the design of the ADC as well as the receiver.

CHAPTER 1. Introduction

With the great advances in digital circuits, the demand on analog circuits increases as well. There are digital systems that require analog front-end (AFE) subsystem to make them alive. One example is communication systems. This is mainly due to the nature of the signals being transmitted. Signals are analog by nature.

Even if a signal is transmitted as a digital one over a cable or in the air, after a while, it will no longer be digital. Noise, interference, attenuation and many other impairments will distort the signal and make it look like an analog one.

Although analog circuits can understand digital signals, digital circuits cannot understand analog signals. This requires analog signals to be handled by analog circuits.

In this thesis, analog design techniques and implementations of high-speed circuits that are used to work on analog signals are presented.

It is important to familiarize the reader of this thesis with some terminology that will be used throughout the thesis. CHAPTER 2 presents the terminology and metrics that are used in data converters, as they are one of the most important and challenging analog circuits in many systems.

Analog-to-digital converters (ADCs) are used in digital systems whenever the input is an analog signal. The ADC converts the analog signal into an equivalent digital value that can be used by the digital system. CHAPTER 3 presents some ADC architectures that are commonly used to do the job. In particular, CHAPTER 4, talks about the pipeline ADC architecture in more details as it is being chosen to design an ADC that is capable of achieving 10 bits of resolution when running at 100MHz. The details of the implementation of this ADC are presented in CHAPTER 5. CHAPTER 6 presents the techniques that are commonly used to enhance the performance of data converter circuits. In addition, it presents two new algorithms that can be used to enhance what is called the *DNL* of an ADC. In CHAPTER 7, a new architecture of ADCs is presented. This architecture is not only based on a new concept, but it also has the potential for achieving higher performance with lower power and smaller silicon area than current architectures.

When it comes to the design of any system, time-to-market plays a great role in deciding on a specific architecture. The new architecture uses a very common analog block called *voltage-controlled oscillator (VCO)* as its main conversion engine. VCOs are used mainly in communication systems as well as in clock synthesizers. This block is well researched and many of its design issues are well known to the analog

designers. This, in turn, will have a great impact on the time-to-market factor. The new architecture also includes a novel *frequency detector (FD)* circuit that enables the ADC to run at very fast speeds.

CHAPTER 8, presents analog design techniques as well as circuit implementation of a CMOS 3.125Gb/s receiver. This receiver achieved high performance with minimum power consumption as will be shown in the measurement section of the same chapter.

All the design techniques and engineering practices used in this dissertation are collected in CHAPTER 9. These techniques can be applied in any analog system.

The conclusions of this dissertation are presented in CHAPTER 10, which provides a brief summary of each chapter of this dissertation as well as the contributions of this research.

CHAPTER 2. Terminology And Metrics

2.1. Introduction

This chapter contains the necessary material that is required to understand data converters, their function and how to differentiate between them. The first section in this chapter will define the analog-to-digital converter as a system and identify its function. There are many data converter architectures in the market and each one has its own advantages over the others. In order to characterize them and quantify their performance, many parameters have to be evaluated. Some of the most common parameters and their definitions will be covered in the section 2.3.

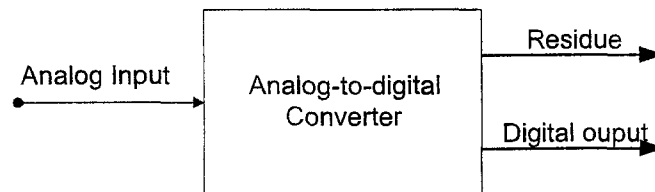


Figure 1 Analog-to-Digital Converter system.

2.2. ADC definition

An Analog-to-Digital Converter (ADC) is a device that has an analog input and produces a digital output that is equivalent to the analog input. The analog input is an electrical signal that might be a current or a voltage. An ADC can be modeled as shown in Figure 1, which shows that in addition to the digital output, another analog output, called the *residue*, might also be generated by the ADC, although in almost all practical ADCs this output is ignored. The ADC performs what is called *quantization* on the input signal. Quantization is the process of transforming a continuous analog signal by a set of digital values that closely approximates the original signal. A good way of illustrating the quantization process is by an example.

Example 1.

Consider the grading system at school, where the instructor has to submit the grades using the system A, A-, B+, B, B-, ..., F. The instructor gathers the grades during the semester out of 200, and at the end of the semester, he/she *quantizes* those grades to the equivalent letter system. For instance, a

student whose total is 172 will be given a **B+**, while another student whose total is 165 will be given a **B**.

Figure 2 depicts this process. If another student has a total of 180, he will get a **B+**, too. The **B+** grade can be assigned to any total in the range of 167 to 183. According to Figure 2, the quantization level of **B+** is 167, so, any total that lies in the range 167–183 will be assigned a **B+**. The difference between any total and its quantization level is equivalent to the residue, while the grades: **A**, **A-**, **B+**, ..., **F** are equivalent to the digital output in the ADC.

The digital output will not exactly reflect the input signal, rather, it will be equivalent to the closest quantization level smaller than the input signal. Consider the following example:

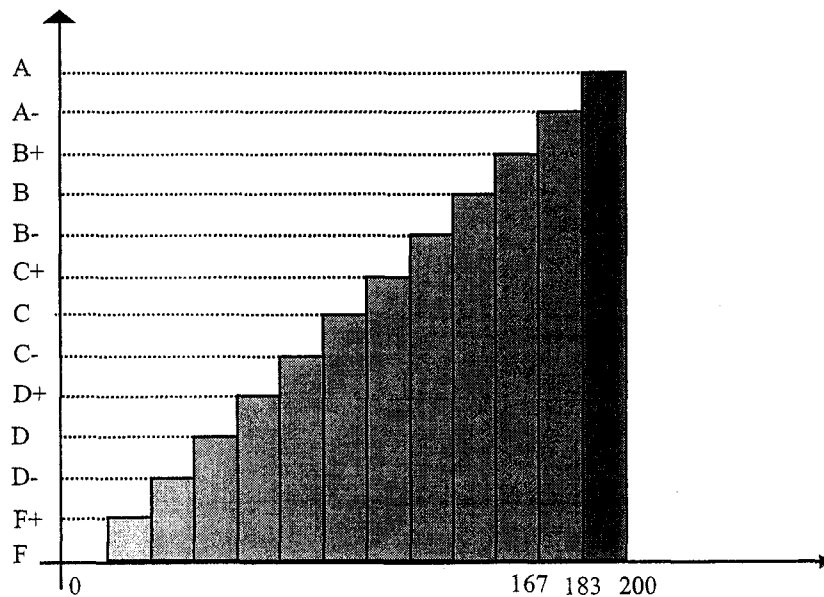


Figure 2 The grading system as a quantization process.

Example 2.

Suppose that we have an ADC that quantizes an input signal into integer values. The input signal is a continuous time voltage signal that ranges from 0 to 5V. If the input signal value is, say, 4.3V, the digital output will be 4 and the residue will be $(4.3V - 4.0V) = 0.3V$. Note that when the residue was calculated, the exact analog equivalent value was subtracted from the input signal value.

If we represent the analog signals by the real numbers line, an ADC can be viewed as dividing the real line into subranges and the input signal is mapped to one of those subranges. Those subranges are given codes and the ADC generates the code of the subrange to which the input signal belongs, in addition to the location of the input signal in that subrange. This is illustrated in Figure 3.

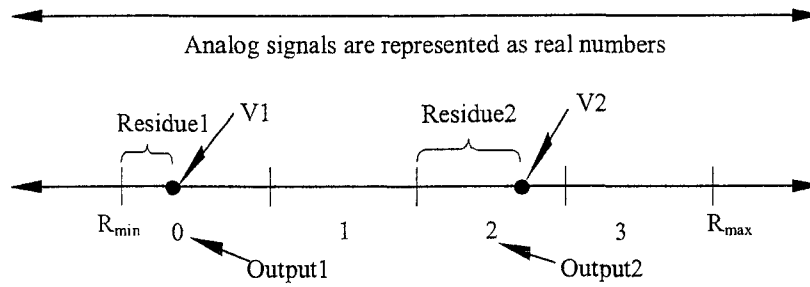


Figure 3 ADC system as a real line.

Figure 3 illustrates the function of the ADC. The input signals are represented as dots on the real line. Given the first value, $V1$, is an input to the ADC, the ADC will generate a digital output, 0, which says that the input value occurred in the subrange that is marked by the code 0. For the second value, $V2$, the ADC will generate a digital value, 2, that correspond to the subrange to which $V2$ belongs. Note that the residue values are different for the two input values.

2.3. ADC Characteristics

ADCs are categorized according to their ability to digitize the input signal range into distinct subranges or levels. The more number of levels the ADC is able to generate, the closer the equivalent value of the digital code generated by the ADC is to the actual input signal, which, in turn, means the smaller the residue is. Each level is assigned a unique code or number. Although it is not necessary, those numbers are always represented in binary form. For example, if we have 4 levels in the input signal range, the first level can be given the code 00, the second one can be given 01, the third 10 and the last one can be given the code 11. The assignment of those codes to the levels, called *code assignment*, can also be useful as will be shown later since, if they are chosen carefully, they might relax the design of some parts of the ADC. An easier and more practical way to categorize an ADC can be achieved by taking $\log_2(\text{number of levels})$, which is generally referred to as *the number of bits*. In practice, we might encounter a 13-bit ADC, which means that the ADC is able to digitize the input signal into one level out of ($2^{13} = 8192$) levels that span over the input signal range.

Since the ADC is a system, it must have a transfer function that relates the digital output to the analog input and this is shown as the solid line in Figure 4 for an ideal one. An ideal ADC is one whose behavior agrees completely with theoretical calculations of its parameters, i.e., it has ideal parameters. Those parameters will be illustrated as we proceed through the following sections.

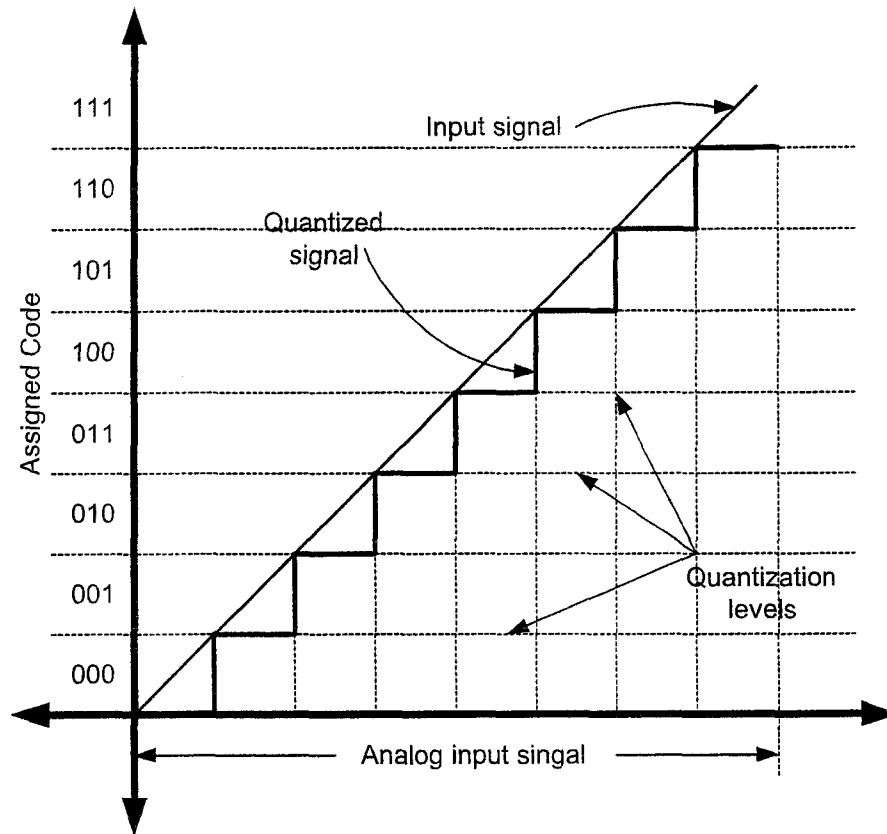


Figure 4 ADC transfer characteristic

2.3.1. Resolution and Accuracy

Resolution is the number of bits an ADC can have, and it is a measure of the ability of the ADC to digitize the input signal's range into larger number of subranges. So, an 8-bit ADC means that the resolution of the ADC is 8 bits or equivalently, the ADC can resolve 8 bits, and it can digitize the input signal's range into 2^8 subranges. If the overall range of the ADC is normalized to 1, i.e., the range becomes from 0.0 to 1.0, then the size of a subrange is called the *Least Significant Bit*, LSB. Mathematically,

$$LSB = \frac{\text{overall signal range}}{2^n} \quad (1)$$

where n is the number of bits the ADC can resolve. For example, a 1 LSB of a voltage signal that ranges between 0 and 5V in a 6-bit ADC is $5.0/2^6 = 78.125\text{mV}$.

The accuracy of an ADC is defined as the precision with which the subrange is calculated. The accuracy of the ADC is usually related to the *DNL* of the ADC as will be described later. As an example to the accuracy, consider an 8-bit ADC with accuracy of 9 bits. This means that each subrange width is at most

1.5LSBs. If the width of any subrange is guaranteed to be less than 1.25LSB, then the accuracy of the same ADC is 10 bits.

2.3.2. Bins and Trip Points

As the ADC divides the range of the input signal into subranges, those subranges are called *bins*, (*Bs*). The value of the input signal at which the ADC changes the quantization from one bin to a next one is called a *trip point*, (*TP*). Those two definitions will be used frequently when we talk about the error sources in an ADC system later on. Bins and TPs will be illustrated in the following example.

Example 4.

Consider an ideal 3-bit ADC with a voltage input signal that ranges from 0V to 5V. If the ADC is ideal, it will have ideal bins and ideal TPs. The ideal bin size will be:

$$\frac{(5-0)}{2^3}V = \frac{5}{8}V = 0.625V \quad (2)$$

The ADC bins and TPs are shown in Figure 5.

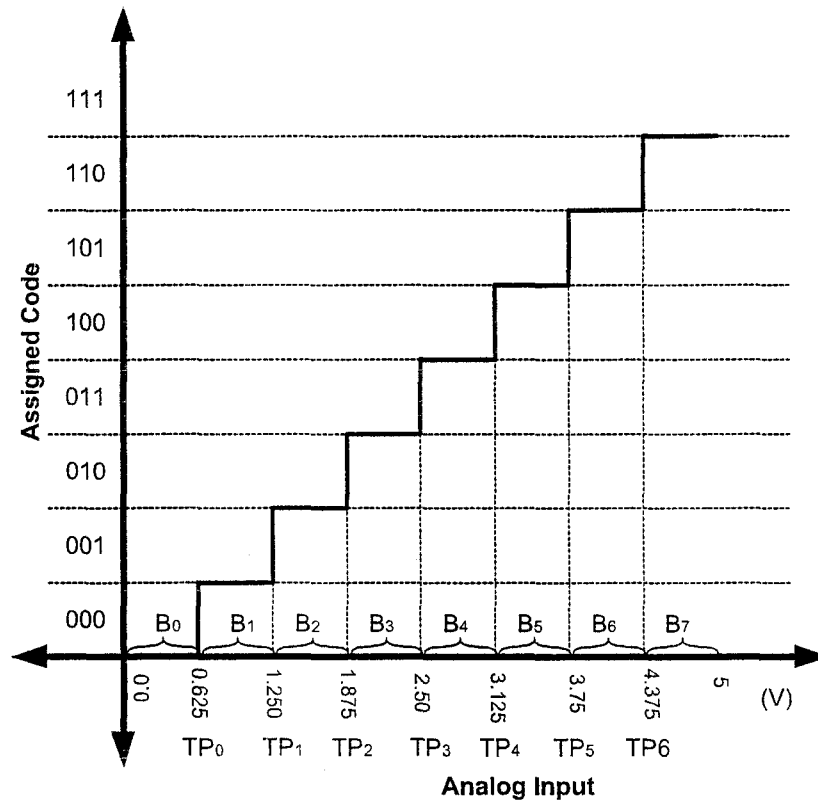


Figure 5 Bins and Trip Points.

Figure 5 shows the bins to which the ADC divides the input signal range. As an example, the first bin, B_0 , covers the range from 0V to 0.625V, while, B_3 , covers the range 1.875V to 2.5V. Table 1 shows the values of the trip points of the ADC. If the input signal value is less than the first trip point, the ADC will quantize that signal into 0V, which means that the ADC will generate a digital value that corresponds to a 0V input signal. If the input signal has a value in B_3 , i.e., between TP_4 and TP_5 , the ADC will quantize it to TP_4 . As will be shown later, this might not be true in general, but it holds for the above example.

Table 1 Trip point values for the 3-bit ideal ADC.

TP_0	TP_1	TP_2	TP_3	TP_4	TP_5	TP_6
0.625	1.25	1.875	2.5	3.125	3.75	4.375

2.3.3. Gain and Offset

In example3 above, an ideal ADC is assumed to quantize the signal exactly as described above, however, in practice, ADCs are not ideal. When an ADC processes the input signal, some kinds of impairments are introduced to the signal. Examples of those impairments are system noise, distortion, and change in ADC parameters due to environmental changes such as drifts in temperature, power supply and process variation. Those impairments will affect the transfer characteristic of the ADC and result in what is called *ADC Errors*. There are two kinds of ADC errors; *linearity* and *nonlinearity* errors.

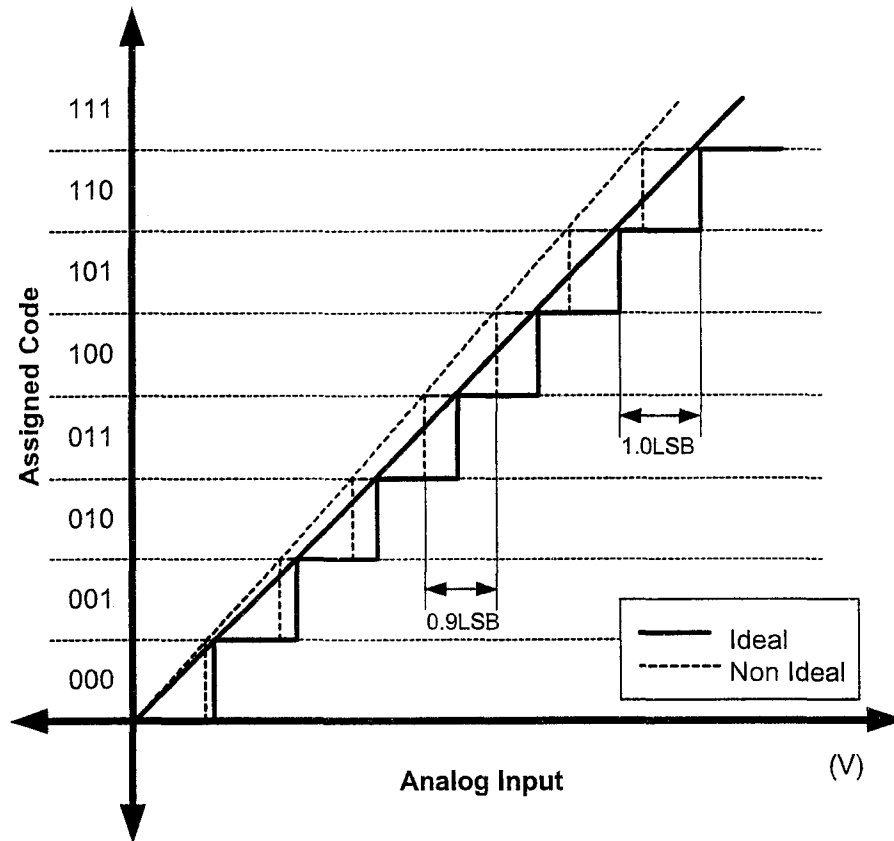


Figure 6 Effect of errors in bin size.

Linearity errors are those kinds of errors that affect all bins of the transfer characteristics by the same amount. One example of this effect might be the reduction of *all* bin sizes by the same value. Ideally, all bin sizes have to be 1 *LSB* wide, but, they all might have a size of 0.9*LSB*. Figure 6 shows two transfer characteristics;

Ideal, which is plotted as a solid line. The edges of the steps are connected by a solid straight line to distinguish them from the second plot.

Nonideal, which is plotted as a dotted line. The edges of the steps are connected by a dotted straight line to distinguish them from the first plot.

The slope of the straight line is called the *gain* of the ADC. The change in the sizes of all bins just described will result in a change in the ADC overall gain, as shown in Figure 6.

Although the straight lines plotted in Figure 6 connect the edges of the steps, in practice, however, there are many ways to plot the straight line, some of those will be discussed in the next subsection.

Another example of linearity errors is the drift of *all* trip point values by the same amount. A drift of 750mV will result in the following TPs shown in Table 2 instead of those in Table 1.

Table 2 A linearity error might cause trip points to drift by 0.75V.

TP ₀	TP ₁	TP ₂	TP ₃	TP ₄	TP ₅
1.375	2.0	2.625	3.25	3.875	4.5

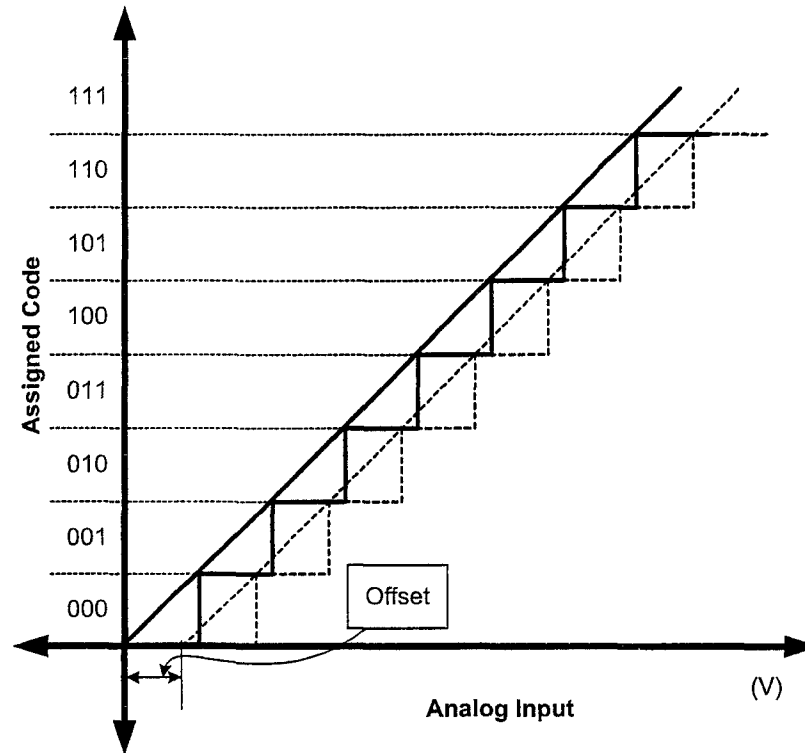


Figure 7 Effect of errors in the trip points.

The effect of this error is that a shift in the transfer characteristic occurs, and it is shown in Figure 7. The shift in the transfer characteristic from the origin point of the coordinate is called *ADC offset*.

Not all the ADCs are built with zero offset. For example, if the noise in the system has a zero average value, then it might be better to introduce an offset in the system so that the ADC will not keep jumping from the first bin to the second when there is no input signal [1]. Hence, the error in ADC offset is the difference between the actual offset and the offset set by design.

In summary, *ADC gain* is the overall gain of the ADC and it is the slope of the straight line that connects the steps of the transfer characteristic of the ADC. *Gain error*, on the other hand, is the difference in the gain of the nonideal ADC and the ideal one.

ADC offset is the offset of the straight line from the zero value of the analog input signal. *Offset error*, on the other hand, is the difference between the nonideal offset and the ideal one.

Nonlinearity errors will be discussed in the following subsection.

2.3.4. INL and DNL.

In most ADC's, the gain and offset specifications are not the most critical ones that determine an ADC's usefulness in specific applications. *Differential NonLinearity* (DNL) and *Integral NonLinearity* (INL) [2], which are considered as nonlinearity errors, are considered the most important specifications for the bulk of ADC applications, because they represent irreducible errors inherent to a practical ADC.

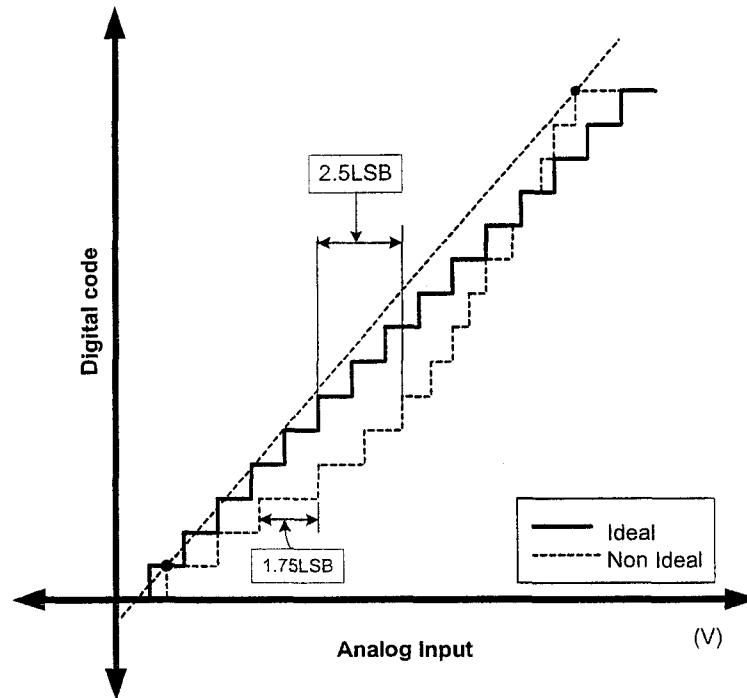


Figure 8 Ideal and nonideal transfer characteristic of an ADC.

Nonlinearity errors, in general, are those that affect the bins of the transfer characteristic unequally. Figure 8 shows a more practical transfer characteristic of an ADC.

INL is defined as the deviation of the transfer characteristic of a practical ADC from the ideal straight line. It is always measured at the quantization levels and expressed in terms of LSBs. There are many ways to draw the straight line that is shown in the figures above, some of those are:

1. End Points. A straight line is drawn between the first step edge and the last step edge. Figure 8 shows and INL of 2.5 LSB at B_6 , which happens to be the maximum INL of this ADC. When INL is specified in terms of the deviation from a straight line using this method it is called *end-point* INL.
2. Best-straight-line. The straight line is calculated such that the worst-case INL error, i.e., the maximum value of an INL error, is minimized. Usually, the straight line is calculated using least

square fitting curve procedure. INL specified in terms of the deviation from a straight line using this method is called *best-straight-line* INL.

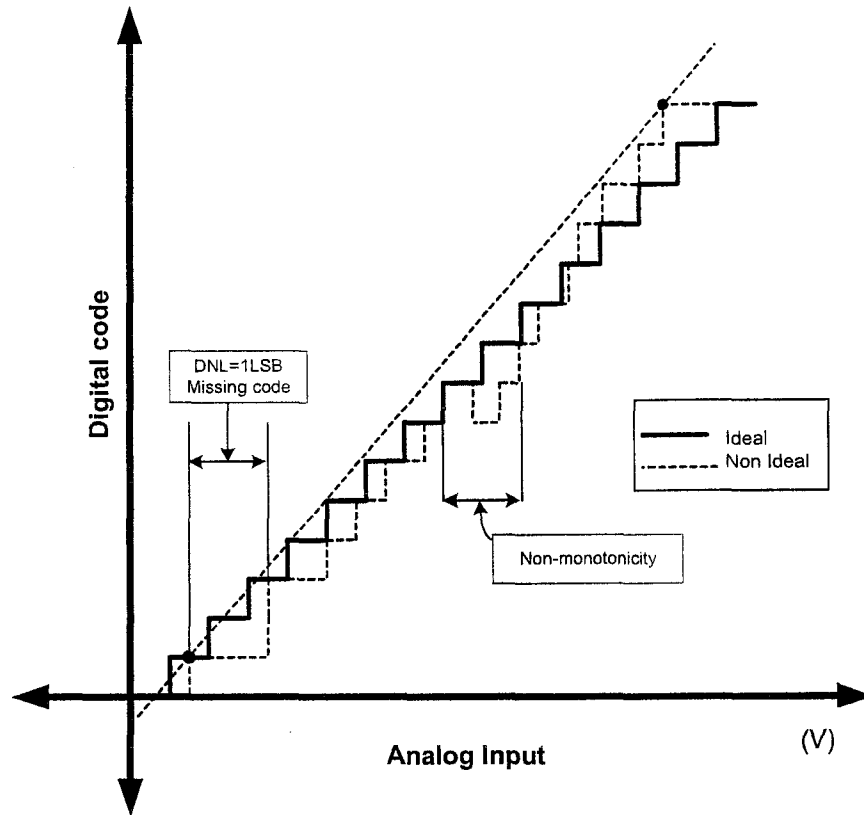


Figure 9 Missing codes and non-monotonicity due to large DNL errors.

DNL is the difference between the nonideal bin size and the ideal one, which is 1 LSB. As an example, B_3 of the ADC shown in Figure 8 has a DNL of 0.75 LSB. A DNL of 1 LSB results in what is called *missing code*, where one of the quantization levels will be *missing* in the transfer characteristic. If an ADC has a DNL greater than 1 LSB, then it will result in what is called *non-monotonicity*, where the quantization level of a certain bin is larger than its successor one. Figure 9 shows that B_2 is missing because the DNL at B_1 is 1 LSB. Figure 9 also shows the non-monotonicity in the transfer characteristic due to negative DNL for couple of the steps.

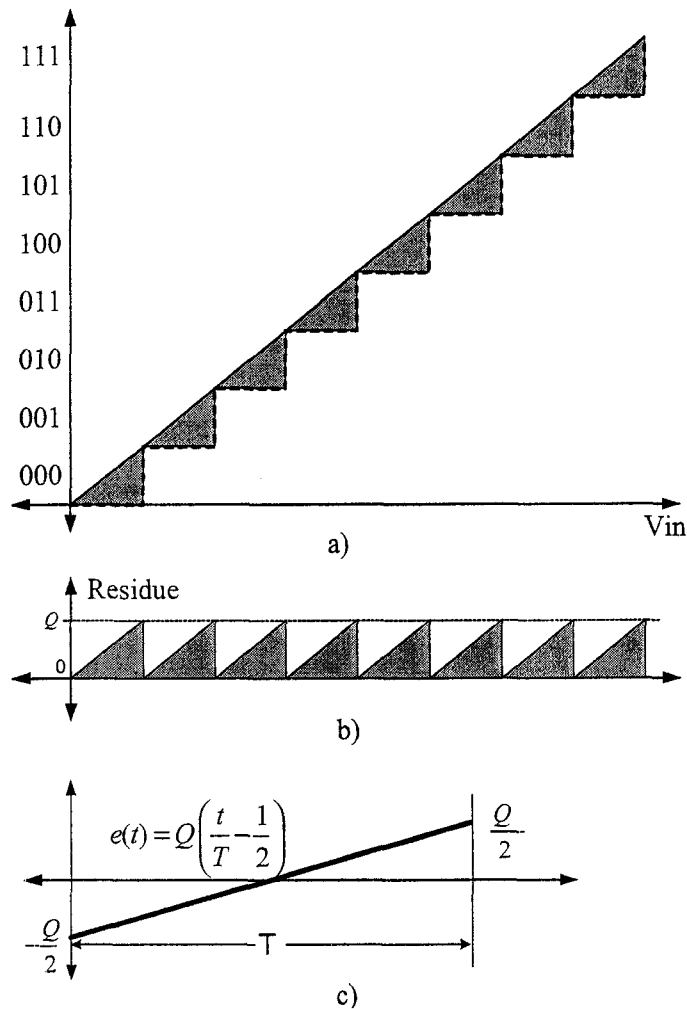


Figure 10 Residue and quantization noise.

2.3.5. SNR, SNDR and ENOB

SNR is the ratio of the *rms* signal amplitude (set at 1 dB below full scale) to the *rms* value of the sum of all other spectral components, excluding the first five harmonics and dc. Alternately, *SNR* can be calculated as the ratio of the signal power to the total noise power at the output. *SNR* is usually measured for a sinusoidal input signal [3].

Figure 10 shows the residue of the stages as a function of the input signal. Since the residue is the difference between the input signal and the corresponding quantization level, it is shown as the shaded area in Figure 10.a). Sometimes, the residue plot shown in Figure 10.b) is called the *quantization noise*. The term quantization noise is appropriate since the error that is produced manifests within a system much in the same way as other noise sources [4]. This is especially true when the quantization noise is not correlated with the

input signal. With this assumption in mind and ignoring all other sources of noise in the system, the SNR can be calculated as follows:

$$Q(rms) = \sqrt{\frac{1}{T} \int_0^T \left(Q \left(\frac{1}{T} - \frac{1}{2} \right) \right)^2 dt} = \frac{Q}{\sqrt{12}} = \frac{V_{FS}}{\sqrt{12}} \quad (3)$$

$$SNR(dB) = 20 \log \left(\frac{V_{FS}(rms)}{Q(rms)} \right) = 20 \log \left(\frac{V_{FS}}{2\sqrt{2} \frac{V_{FS}}{2^n \sqrt{12}}} \right) = 6.02n + 1.76 \quad (4)$$

Where V_{FS} is the full-scale value of the input signal and n is the nominal resolution of the ADC.

The above definition for both SNR and SNDR reflects the way they are measured in the lab, where the *spectral components* and *harmonics* are generated by using the FFT (*Fast Fourier Transform*). Mathematically, SNR is calculated as the difference in dB between the signal rms value and the noise rms value according to the following equation:

$$SNR = \text{signalrms}(dB) - \text{noiserms}(dB) \quad (5)$$

FFT takes a discrete number of time samples, M , and converts them into $M/2$ discrete spectral components. The spacing between the spectral lines is $\Delta f = F_s/M$, where F_s is the sampling frequency. Equation (3) is only valid if the noise is measured over the entire Nyquist bandwidth from DC to $F_s/2$. If the quantization noise is uncorrelated with the signal, it appears as Gaussian noise spread uniformly over the bandwidth from DC to $F_s/2$. The FFT acts as a narrowband filter with a bandwidth of Δf , and the FFT noise floor is therefore $10 \log_{10}(M/2)$ dB below the quantization noise level. This is referred to as *processing gain* of the FFT [6].

For example, a 4096 point FFT has a noise floor of 33dB below the theoretical rms quantization noise floor of 74dB for a 12-bit ADC, while the average noise floor is about $74+33 = 107$ dB below the full scale.

Also, if the signal bandwidth, BW , is less than $F_s/2$, then the SNR with the signal bandwidth is increased because the amount of quantization noise within the signal bandwidth is smaller [6]. The overall expression of the SNR will be:

$$SNR(dB) = 6.02n + 1.76 + 10 \log_{10} \left(\frac{f_s}{2 \cdot BW} \right) + 10 \log_{10} \left(\frac{M}{2} \right) \quad (6)$$

Another way of calculating the SNR can be achieved by measuring the powers of the signal and the quantization noise in the system.

Effective number of bits (ENOB) is defined by the following equation:

$$ENOB = \frac{SNDR_p - 1.76}{6.02} \quad (7)$$

Where $SNDR_p$ is the peak SNDR of the converter expressed in dB.

SNDR is defined to be the ratio, expressed in dB, of the RMS value of the input signal to the RMS value of all of the other spectral components below the Nyquist frequency including harmonics, but excluding DC.

2.3.6. Dynamic Range and SFDR.

Dynamic range is the ratio of the maximum allowable input swing to the minimum input level that can be sampled with specified accuracy [5].

Probably the most significant specification for an ADC used in a communications application is its *Spurious Free Dynamic Range* (SFDR), which is defined as the ratio of the rms signal amplitude to the rms value of the peak spurious spectral component (measured over the entire Nyquist bandwidth) and it may or may not be a harmonic. SFDR is generally plotted as a function of signal amplitude and may be expressed relative to the signal amplitude (dBc) or the ADC full scale (dBFS) [6]. This measurement indicates the amount of dynamic range that can be obtained from the ADC before distortion becomes dominant.

For a signal near full scale, the peak spectral component is generally determined by one of the first few harmonics of the fundamental. However, as the signal falls several dBs below full scale, other components generally occur which are not direct harmonics of the input signal. Therefore, SFDR considers all sources of distortion, regardless of their origin.

2.3.7. Latency

Latency is the time taken by an ADC to generate the digital equivalent of the analog input. It is measured by the number of clock cycles between conversion initiation and the associated output data being made available.

2.3.8. Aperture Jitter

It is the variation in the aperture delay from sample to sample. Aperture jitter shows up as input noise to the ADC.

2.3.9. PSRR (Power Supply Rejection Ratio)

It is the ratio of the change in DC power supply voltage to the resulting change in Full Scale Error, expressed in dB.

2.3.10. THD

Practically, THD is the ratio of the RMS value of the first six harmonic components to the RMS value of the measured input signal and is expressed as a percentage or decibels.

2.4. Conclusions

The terminology commonly used in ADCs was presented in this chapter. This terminology is key to the understanding of the specifications of ADCs as well as the measurement of their performance.

References

- [1] W. Black, EE501 Course at Iowa State University.
- [2] Engineering Staff at Analog Devices, Analog-Digital Conversion Handbook, Prentice-Hall, Inc. 1986.
- [3] Analog Devices data sheet AD9430.
- [4] M. Koen, "High performance analog to digital converter architectures," Proceedings of the 1989 Bipolar Circuits and Technology Meeting, pp. 35 - 43
- [5] B. Razavi, Principles of Data Conversion System Design, 1995 by AT&T.
- [6] W. Kester, High Speed Design Techniques, Analog Devices, 1996.

CHAPTER 3. ADC Architectures

3.1. Introduction

ADCs are becoming more and more important with the advancements in the digital processing design. Different systems require different specifications for the ADCs, and many architectures have been implemented to meet those different requirements. Among those requirements are high speed, high resolution, high SNR, low power, low DNL, small latency, small area or any combination of the above. None of the existing architectures meet all of the above requirements, which is expected from engineering sense, and so, the designer of the ADC should be able to pick the architecture that best meets the requirements. Among those architectures that will be discussed in this chapter are: Flash, Half Flash, or Two-step Flash, Multistep, Folding, Folding and Interpolating, Recycling, Successive Approximation, Pipeline and Parallel Pipelined.

3.2. Flash ADCs

The fastest of all types of high-speed analog to digital converters and perhaps the easiest to understand is the flash converter. The flash converter is considered to be the fastest because the conversion takes place in a single cycle, hence the name flash [2]. They have been implemented most commonly in Bipolar IC technology, where the excellent V_{BE} matching allows design of comparators accurate to 8 bits or better. In MOS technology, calibration cycles are typically required to eliminate comparator offset, which reduces the maximum available clock rate. Speeds up to 2GHz have been achieved [3], and conversion rates of up to 300MHz are readily available on the commercial market. The resolution of a flash converter tends to be limited to 8 bits due to the fact that the amount of circuitry doubles every time the resolution is increased by one bit.

In a flash architecture, $2^n - 1$ clocked comparators are used to simultaneously compare the input signal with a set of reference voltages generated with a resistor divider, where n is the nominal resolution of the ADC [4]. At the output, a so called linear code, or thermometer code, is generated. If a particular comparator's reference point is below the level of the input signal, the comparator's output is high, or ONE, while, if the reference point is above the input, the comparator's output is low, or ZERO. When every thing is ideal, the collection of comparators' outputs should resemble a thermometer; all ZEROs above the input signal level and all ONES below. The transition from ZERO-block to ONE-block is related to the value of the input signal. The thermometer is then converted to a 1-of-N code, which is subsequently encoded to n bits to produce the output

as shown in Figure 11, which shows a block diagram of an n -bit flash ADC. Usually, the encoder shown in the figure is implemented using a large but simple ROM.

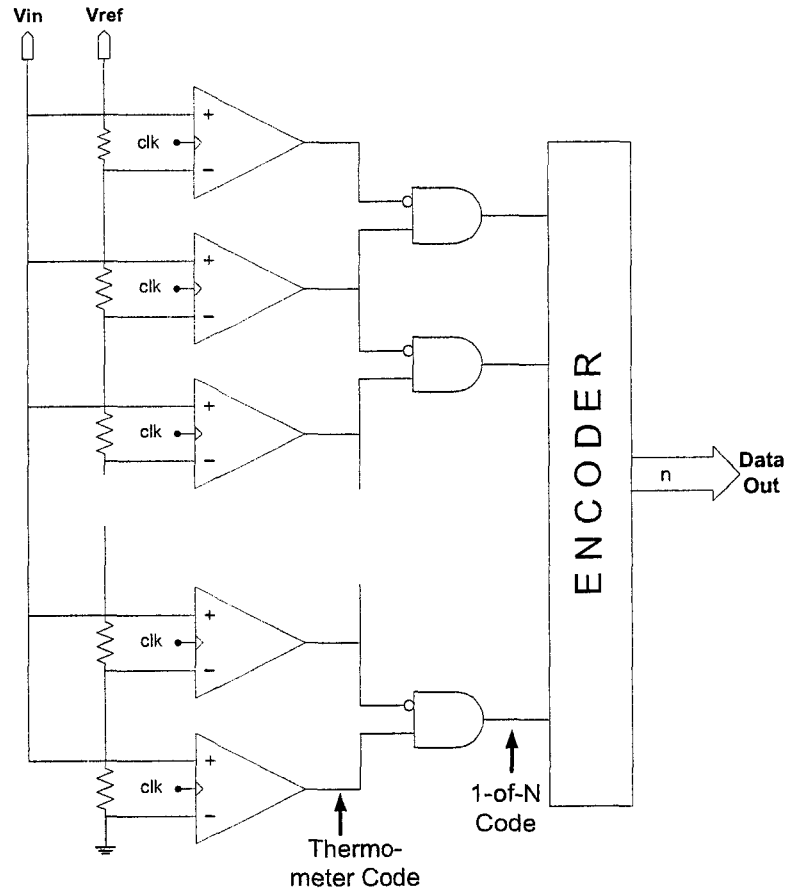


Figure 11 Block diagram of an n -bit flash ADC.

The flash structure is a simple one but uses a lot of chip area to implement the block of decision stages as well as the encoding ROM. The large chip area may result in a layout related problems such as skew in the clock signals, buffering of the sampling clock, ..., etc. The large number of comparators gives rise to problems such as dc deviation of the reference voltages generated by the ladder, large *nonlinear* input capacitance, and kickback noise at the analog input. The nonlinear input capacitance will introduce harmonic distortion in the sampled signal. That is mainly due to the fact that the input signal will encounter an amplitude-dependent delay. The kickback noise is the power of the transient noise observed at the comparator input due to switching of the amplifier and the latch. These two effects are explained well in [5]. Under extremely high input slew rate conditions, timing differences between signal paths or even slight differences in comparator response time can cause the effective trip point of one comparator to be different from another. Consequently, a ONE may be found above a ZERO in the thermometer code even though this cannot happen at dc.

Errors of this type are sometimes referred to as “bubbles” because they resemble a bubble in the “mercury” of the thermometer code [6]. Various circuit techniques have been devised to suppress the effect of bubbles. One approach is to use three-input gate, which will require two ZEROs and a ONE in order to indicate a transition. Other approaches to solve this problem include a voting process [6], Gray coding and “quasi-Gray” coding [7]. Another problem that appears in flash ADCs is what is called *metastability*, in which a small difference at the input of a comparator will cause the comparator to take a long time to produce a well-defined logic output.

This small difference occurs when the input signal level is very close to the reference value of a certain comparator and hence, the comparator output may not be a valid output, which will cause erroneous digital output for that particular conversion.

3.2.1. Advantages

The advantages of the flash converter can be summarized in the following points:

- The primary advantage of the flash conversion architecture is its high conversion rate. By pipelining the digital decoding operation, the input signal can be sampled and digitized at the same time the digital circuit is decoding a previous sample of the input signal; therefore, only 2 clock phases are required per conversion, corresponding to the latched and unlatched states of the comparators. The speed of this architecture is therefore only limited by the speed of the comparators and logic.
- If a resistor string divides the reference, the reference exhibits inherent monotonicity; that is, the reference voltage between any point on the string and the end with the lowest voltage is a non-decreasing function of increasing distance between the two points. The transfer curves of resistor string based flash converters can therefore be made monotonic.

3.2.2. Limitations

The main disadvantages of the flash type ADCs are:

- Large silicon area,
- large input capacitance,
- large power dissipation.

Unfortunately, those three issues grow exponentially as the number of bits increases.

3.3. Two-step Flash ADCs

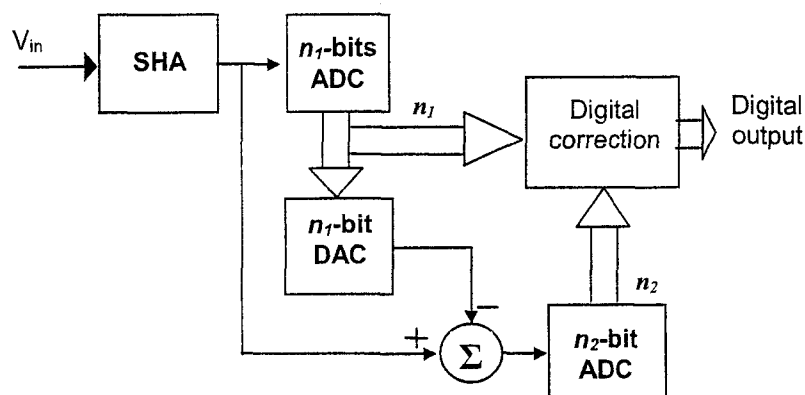


Figure 12 Two-Step flash ADC.

One approach to solve the exponential growth of power, area, and input capacitance of the flash ADC with its nominal resolution is to divide it into two less number of resolution flash ADCs. This architecture is called the *two-step flash ADC*, in which the first step performs a *coarse* conversion, while the second one does a *fine* conversion. For an n -bit two-step flash ADC, the first step will resolve n_1 bits, while in the second step, n_2 bits are resolved, where n_1 and n_2 are less than n and $n_1 + n_2 = n$.

The two-step flash architecture is an effective means of realizing high-speed, high resolution ADCs because it can be implemented without the need for operational amplifiers having either a high gain or a large output swing. Moreover, with conversion rates approaching half those of fully parallel designs, such half-flash architectures provide both a relatively small input capacitance and low power dissipation [8].

Problems associated with two-step flash ADCs are: poor linearity due to separated two-step comparison, slow-conversion rate, lack of precision and high speed internal sample-and-hold circuitry. Some solutions to the above problems include the implementation of pipelined and multiplexed two-step architecture to improve the conversion rate, while a development of an auto-zeroed differential sample-and-hold comparator will improve precision and speed [9].

The linearity of two-step A/D converters has been limited to a 10-bit level using passive component mismatches. To increase the resolution of the two-step ADC, error correction or calibration techniques have been used. One bit of redundancy, or overlap, can be used between the two stages to enable the second stage to correct for out-of-range errors in the first stage, thereby relaxing the precision required of the first-stage comparators. Furthermore, fully differential architecture increases the input dynamic range, eliminates even-order harmonic distortion, and suppresses common-mode noise due to supply transients and substrate coupling. Other techniques include a *direct code-error calibration* in the digital domain has been used by [10] to improve the linearity. This technique reduces feedthrough, offset and interstage gain errors simultaneously.

3.4. Folding ADCs

For resolution around 8 bit, flash ADC is the fastest possible architecture. The sampling speed of the flash converter is limited to the maximum speed of a comparator in that technology. On the other hand, the major disadvantage of the flash ADC is the exponential dependency of several of its parameters such as power consumption, area, and input capacitance [11].

A *folding* architecture can be considered as a continuous-time two-step architecture. In a two-step ADC converter, the signal conversion is split into two or more phases in time. The two stages work in tandem, where the second stage waits for the first stage to finish processing and pass the residue, and then it starts the quantization process.

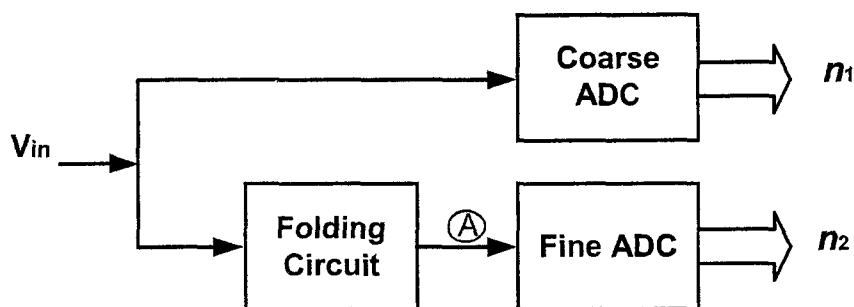


Figure 13 Block diagram of a folding ADC.

In folding ADC, signal conversion consists of a coarse and fine conversion stages, but those conversions are done in parallel. This gives the folding ADC the same maximum clock frequency that can be achieved with full-flash ADC with a power and area comparable to that of a two-step ADC. Figure 13 shows the block diagram of a folding ADC.

Figure 13 shows that the input signal is applied to two paths at the same time. The first path includes the coarse ADC that resolves the MSBs of the signal. In other words, if the coarse ADC is a 3-bit one, for example, it tells which octant, ($1/8$ of the whole range), of the input range the input signal lies. The second path does two operations on the signal. The first operation is what is known as *folding* the signal, while the second is a regular quantization process.

The folding operation is illustrated in Figure 14 and Figure 15. Figure 14 shows the folding operation where a piece of paper is folded into 4 folders. In Figure 15.a), another piece of sheet is folded into 11 smaller folders as numbered in the figure. The smaller folders are identical in their length, which is shown as height in Figure 15.a). The folding points, which are the points where the original sheet is going to be folded, are shown as dotted lines in the original sheet. There are three points on the original sheet; V_1 , V_2 and V_3 , shown as solid lines. Those three points are mapped to their location after folding as shown in Figure 15.b).

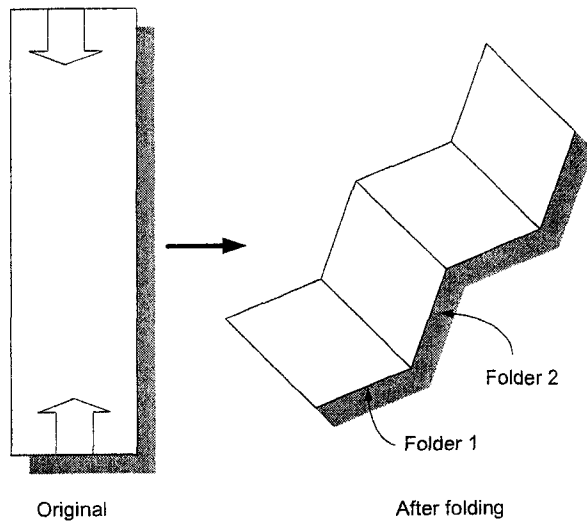


Figure 14 Folding operation.

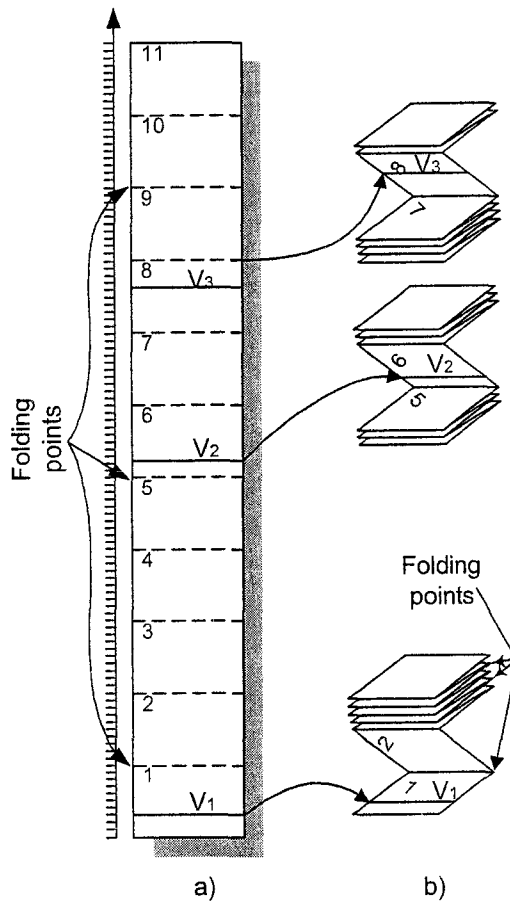


Figure 15 Folding operation; mapping of signal values.

The main function of the folding circuit is to map an input value, such as V_1 or V_2 to its place in one folder, while making the other folders transparent. This operation is similar to opening only one folder that contain the original value, while stacking the other folders over each others as shown in Figure 15.b).

The *folding factor* is the number of folds the input signal experiences. For example, a *folding factor* of 4 means that the signal is folded into 4 folds or folders.

A folder circuit can fold the input signal to a sawtooth waveform, as shown in Figure 16 or to a triangular waveform as shown in Figure 17. The folding factor in both figures is 4.

A mathematical model for the folding circuit is a useful tool in order to understand and imagine how the input waveform shown in Figure 15 will be at point A as a function of time, t , in the same figure after the folding circuit.

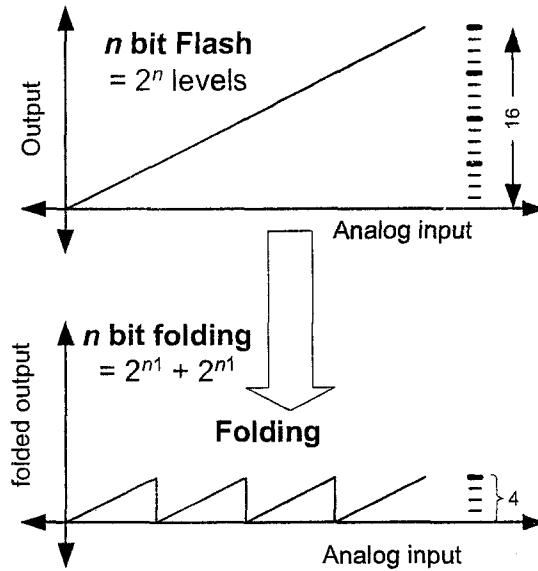


Figure 16 Output of the folder circuit as a sawtooth waveform.

The output of a folding circuit that generate a sawtooth waveform can be modeled mathematically as:

$$V_A(t) = \left(\frac{V_{in}(t)}{A} \times n - \left\lfloor \frac{V_{in}(t)}{A} \times n \right\rfloor \right) \times \frac{A}{n} \quad (1)$$

Where A is the amplitude of the input signal, $V_{in}(t)$, and n is the folding factor.

The output of a folding circuit that generate a triangular waveform can be modeled mathematically as:

$$i = V_{in}(t) \times \frac{n}{A} \quad (2)$$

$$V_A(t) = \frac{(1 - (-1)^i)}{2} \times \frac{A}{n} + (-1)^i \times \left(V_{in}(t) \times \frac{n}{A} - i \right) \times \frac{A}{n} \quad (3)$$

Where A is the amplitude of the input signal, $V_{in}(t)$, and n is the folding factor.

The input signal is applied to the folding circuit and the output of this circuit is then passed to the fine ADC. At the same time, the input signal is connected to the coarse ADC. The operation of the folding circuit is illustrated in Figure 17. The “zig-zag” shaped transfer curve covers the whole V_{in} range, and the output signal of the folding circuit needs to be converted to only 2^{n_2} levels corresponding to the n_2 least significant bits of the ADC converter output code [12].

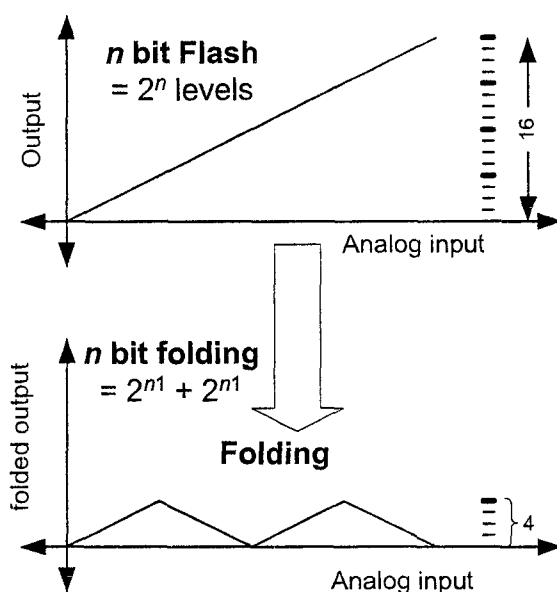


Figure 17 Output of the folder circuit as a triangular waveform.

A track-and-hold amplifier is not necessary in a folding ADC. However, the input signal frequency is multiplied in the folding circuit as a result of the *folding* operation. The maximum frequency multiplication in a folding system is determined by the *folding factor* of the ADC. A high folding factor results in a low number of comparators, but on the contrary, it lowers the maximum signal frequency of the ADC. A track-and-hold circuit might be used to overcome this bandwidth limitation [13].

3.5. Multistep ADCs

A multistep ADC architecture extends the concept of two step ADC to many stages. The total number of comparators in this ADC will be less than that of a two-step flash, but that will be on the expense of the conversion time, which will be reflected on the overall speed of the ADC.

3.6. Successive Approximation and Algorithmic converters ADCs

Both the successive approximation and algorithmic ADC topologies requires N clock cycles to perform an N -bit conversion. They both perform one bit of conversion per clock cycle. The successive approximation converter is a subclass of the subranging converter, in which, during each clock cycle only one bit of resolution is generated. The algorithmic converter is a variation of the pipelined converter, in which the pipeline is folded back into a loop. Both topologies essentially perform a binary search to generate the digital value, however, in the case of the successive approximation converter the binary search is performed on the reference voltage, while in the case of the algorithmic converter the search is performed on the input signal.

3.6.1. Successive Approximation ADCs

A block diagram of the successive approximation converter is shown in Figure 18. Because the conversion requires N clock cycles, a S/H version of the input signal is provided to the negative input of the comparator. The comparator controls the digital logic circuit that performs the binary search. This logic circuit is called the successive approximation register (SAR). The output of the SAR is used to drive the DAC that is connected to the positive input of the comparator.

The operation of the successive approximation ADC is described as follows: During the first clock period, the input is compared to the MSB, i.e., the MSB is temporarily raised high. If the output of the comparator remains high, then the input occurs somewhere between 0 and $V_{ref}/2$, and the MSB is reset to 0. However, if the comparator output is low, then the input signal is somewhere between $V_{ref}/2$ and V_{ref} , and the MSB is set to high. During the next clock, the MSB-1 bit is evaluated in the same manner. This procedure is repeated such that at the end of the N clock periods, all N bits have been resolved [16].

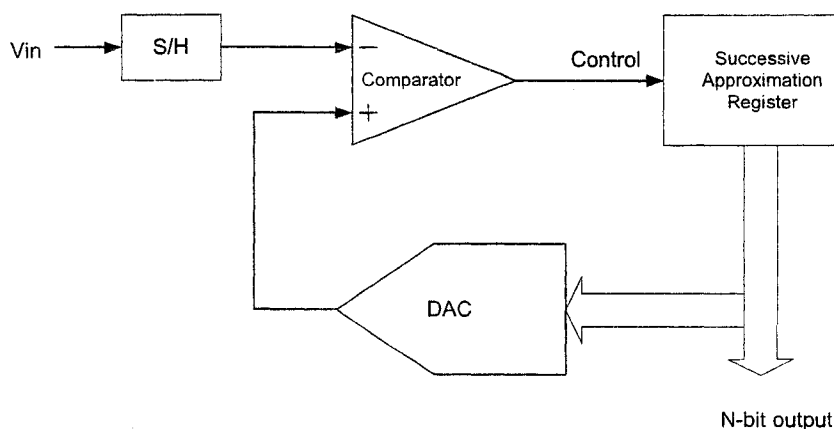


Figure 18 Successive approximation converter block diagram.

3.6.2. Algorithmic Converters ADCs

The algorithmic converter is formed by one stage that evaluates all of the N bits. This stage is configured as a loop that requires N clock cycles to finish the evaluation. A block diagram of this converter is shown in Figure 19 and consists of a S/H at the front, an amplifier that multiplies the input by 2, a comparator, and reference subtraction circuit. The operation of the circuit is as follows: The input is first sampled and held by setting S_1 to V_{in} , the signal is then multiplied by 2. The result of this multiplication, V_o , is compared to V_{ref} . If $V_{on} > V_{ref}$ then the most significant bit, b_N , is set to 1 or, otherwise, it is set to 0. In the next clock cycle, S_1 is switched to V_b while S_2 is connected to either V_{ref} or ground if b_N is equal to 1 or 0, respectively, such that:

$$V_{bN} = 2V_oN - b_N V_{ref} \quad b_N = \{0, 1\} \quad (4)$$

This voltage is then sampled-and-held and used to evaluate the MSB-1 bit. This procedure continues until all N -bits are resolved. The general expression for V_o is given by:

$$V_{O_i} = [2V_{O_{i-1}} - b_i V_{ref}] z^{-1} \quad (5)$$

where b_i is the comparator output for the i th evaluation and z^{-1} implies a delay of one clock period [16].

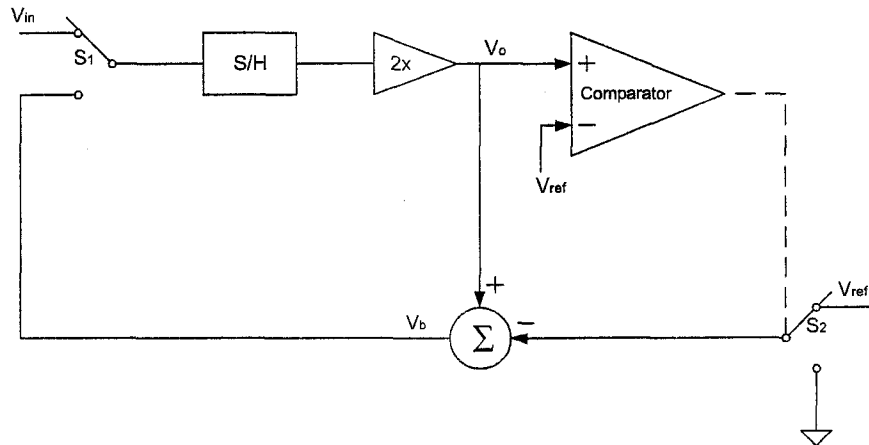


Figure 19 Block diagram of an algorithmic ADC.

3.7. Pipeline ADCs

Pipelining is an implementation technique whereby multiple operations are overlapped in execution. Today, fast CPUs in particular and digital systems in general are mainly attributed to pipelining.

A pipeline is like an assembly line. In an automobile assembly line, there are many steps, each is contributing something to the construction of the car. Each step operates in parallel with the other steps, though on a different car. A pipeline ADC consists of many *stages* that are usually, but not necessarily, identical. The stages are connected one to the next to form a pipeline. In most of the implementations, the pipeline is preceded by a circuit, called *Sample-and-Hold* (S/H), used to quantize the input of the pipeline, which is an analog signal.

Each stage does some kind of processing on the input signal and then passes a new signal to the next stage. The main function of each stage is to give some information about the input signal to that stage. Each stage quantizes the input signal to a certain value, or *bin*. The Pipelined ADC (PADC) is very similar to the multistage ADC, while the main difference between them is that PADC has S/H circuits between the stages. Each stage consumes one clock cycle to do the operation and all the stages are working at the same time, but each operates on a different sample of the input signal. More detailed description about the pipeline architecture will be presented in chapter 4.

3.8. Parallel Pipeline ADCs

Parallel pipeline, or sometimes called as *time-interleaved* pipeline ADCs, are used to increase the speed of the ADC beyond the technological limit. This parallelism can be achieved by connecting multiple ADCs in parallel, and work in a time-interleaved fashion. Time-interleaving means that a sampled version of the input signal is sent first to a first ADC which will start processing it, and then another sample is sent to a second ADC, which will start processing it, and then a third sample is sent to a third ADC, ..., etc, and then a sampled version of the input is sent to the first ADC and so on. Theoretically speaking, the speed of a time-interleaved ADC increases linearly with the increase of the number of the ADCs connected in parallel.

There are different configurations to implement the parallel ADC[17][18][19]. Each one of the ADCs in the parallel ADC can be of any type, however, the most common in the market are successive approximation ADCs, pipeline ADCs or sigma-delta ADCs. Figure 20 shows a parallel pipeline ADC that consists of $M=4$ pipeline channels or paths. Assuming that the overall ADC operates at a sampling frequency f_s , each ADC operates at a sampling rate of f_s/M . A major advantage of this approach is that a considerable saving in the silicon area can be achieved compared to other architectures with the same specifications. Although this architecture will, in some cases, result in increased noise or distortion, however, these effects are both predictable and consistent, and may be minimized in the design of an array of parallel ADCs[17].

Because of the time uncertainty (jitter) of the sample-and-hold circuit preceding each ADC in the array when switching from the sampling to the holding mode, a very accurate analog demultiplexer is needed at its input in order to convert a single high-speed signal into M lower speed analog sampled-and-held signals. Another problem with this architecture arises if mismatches occur among the channels creating aliasing and distortion in the resulting digital output [19]. Such kind of errors in the parallel architecture will be analyzed in chapter 4 in more details.

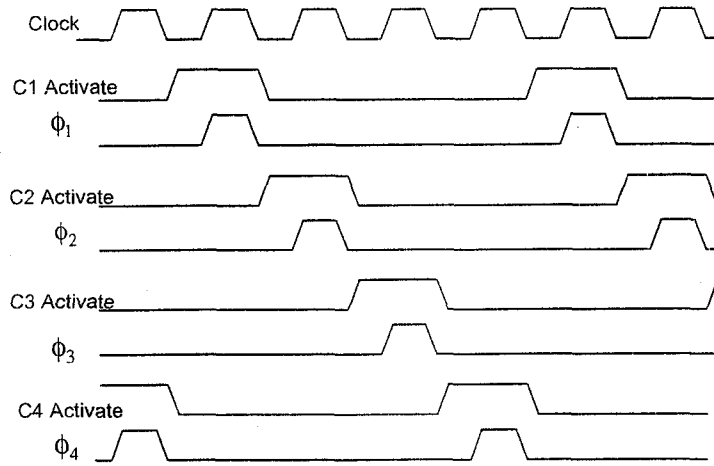
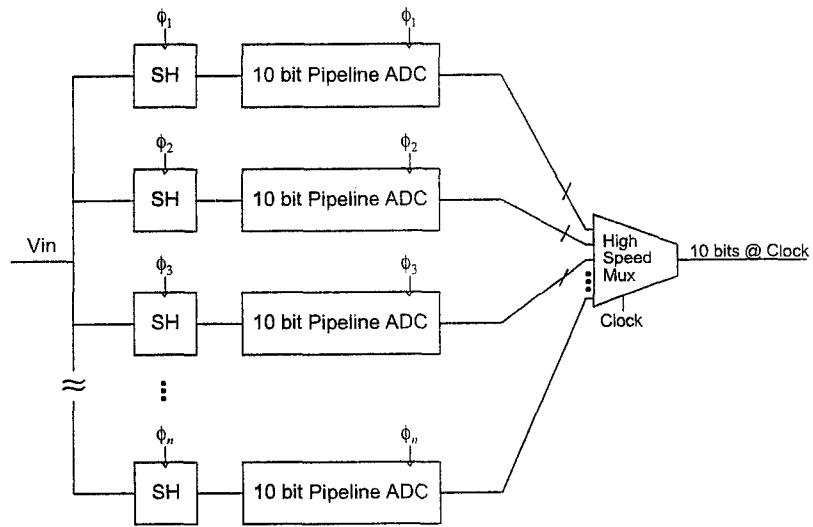


Figure 20 Parallel-pipeline ADC in time-interleaved fashion.

3.9. Oversampling ADCs

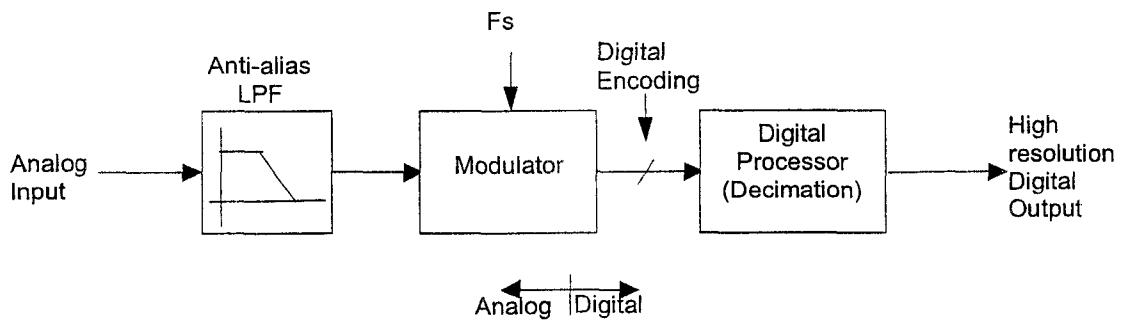


Figure 21 Block diagram for a 1-bit oversampled ADC.

Oversampling A/D converters modulate their analog inputs into short digital words at very high sampling rate. A special kind of filters, called *decimation filters*, resample this code at the Nyquist¹ rate of the signal and increase the word length to maintain resolution.

Sigma-delta, ($\Sigma\Delta$), modulation has been the preferred technique for oversampling conversion and they have been widely used in applications where high accuracy analog circuitry would otherwise be required. Oversampling converters use simple and relatively high tolerance analog components but require fast and complex digital signal processing stages. Recent experience with oversampling converters has shown that their circuits can be designed and scheduled with more assurance than had been possible with the more analog-intensive techniques [14].

3.9.1. Sigma Delta Modulation

A generalized oversampled ADC systems is shown in Figure 21. The block diagram of the oversampled converter shows three main system blocks: the anti-aliasing filter, the analog modulator and digital decimator.

Oversampling ADCs converters achieve high resolution by shifting their quantization noise outside the signal band and then removing it with digital filters. The process that shapes the quantization noise in the sigma-delta modulator that is shown in Figure 22 can be explained as making a prediction of low frequency values of the noise and subtracting it from the signal.

This prediction process works well when its sampling frequency is high with respect to the Nyquist rate, which will result in ADCs with fine resolution. This resolution also improves with the number of levels in the internal quantizer and with the order of the prediction. Very reliable modulators have been built having just two-level quantization and second order prediction.

The major advantage of oversampled ADC system is that the analog circuit complexity can be greatly reduced if the encoding is selected such that the modulator only needs to resolve a coarse quantization (frequently a single bit). Also, if oversampling rates are high, the baseband is a small portion of the sampling frequency. Consequently, constraints on the analog anti-aliasing filter can be relaxed, permitting gradual roll-off, linear phase and easy construction with passive components [15].

Figure 22 illustrates the simplest form of an oversampled interpolative modulator, which features an integrator, a 1-bit ADC and a DAC, and a summer. This topology, known as sigma-delta, uses feedback to lock onto a band-limited input $X(t)$.

¹ *Nyquist rate* is half the input signal frequency. For example, if the input signal is running at a 10MHz frequency, then its Nyquist rate is 5MHz.

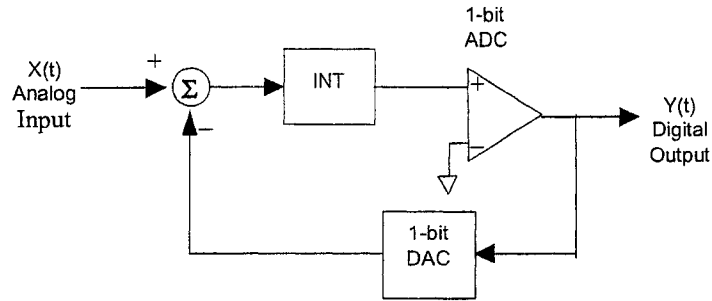


Figure 22 Block diagram of 1-bit Sigma-Delta loop.

Unless the input $X(t)$ exactly equals one of the discrete DAC output levels, a tracking error results. The integrator accumulates the tracking error over time and the in-loop ADC feeds back a value that will minimize the accumulated tracking error. Thus, the DAC output toggles about the input $X(t)$ so that the average DAC output is approximately equal to the average of the input.

The operation of the sigma- delta modulator can be analyzed quantitatively by modeling the integrator with its discrete-time equivalent and the quantization process by an additive noise source as illustrated in Figure 23.

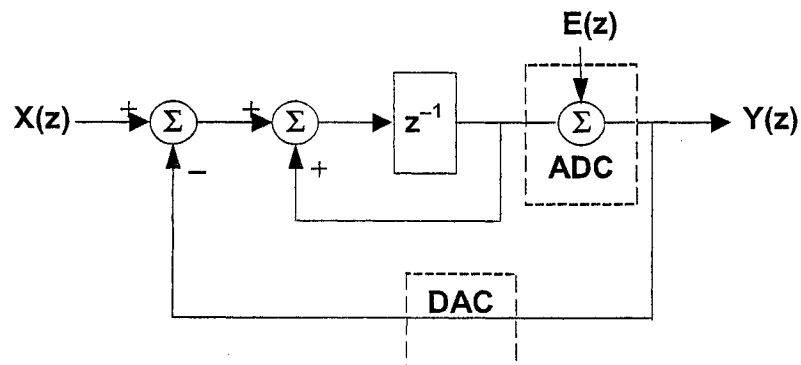


Figure 23 Discrete time equivalent of delta-sigma loop.

3.10. Conclusions

In this chapter, the common architectures of ADCs were presented. Each one of those ADCs has its own characteristics that makes it fit in a certain application. The specifications of those ADCs are always trading with each other. The higher the speed is, the lower the resolution is, and the more the power is. High speed ADCs are always low resolution, while high resolution ADCs are always low speed ones. The designer needs to choose the architecture that best fits its target application.

In a range of Low, Moderate and High, the flash ADC can be described as High speed, High power, Large Area and Low resolution. Two step flash ADCs are Moderate speed, Moderate power, Moderate area and Moderate resolution. The folding ADCs, multistep ADCs and pipeline ADCs are similar to the two step flash

classification although they are not exactly identical. Successive approximation and algorithmic ADCs as well as oversampling ADCs are considered High resolution, Low power, area and speed. Although the multipath ADCs are considered High speed, power and area and Moderate resolution, however, they can't achieve the speed of the flash ADCs.

References

- [1] M. Koen, "High performance analog to digital converter architectures," Bipolar Circuits and Technology Meeting, 1989., Proceedings of the 1989, pp. 35 - 43.
- [2] K. Balasubramanian, "A flash ADC with reduced complexity," *IEEE Transactions on Industrial Electronics*, Vol. 42, NO. 1, February 1995, pp. 106-108.
- [3] T. Wakimoto, Y. Akazawa, and S. Konaka, " Si bipolar 2-GHz 6-bit flash A/D conversion LSI," *IEEE Journal of Solid-State Circuits*, VOL. 23, NO. 6, DECEMBER 1988, pp. 1345-1350.
- [4] J. Corcoran, "High speed sample and hold and analog-to-digital- converter circuits," from the book *Analog Circuit Design* edit by J. H. Huijsing *et al.*, 1993, Kluwer Academic Publishers.
- [5] R. Razavi, Principles of Data Conversion System Design, 1995, by AT&T, IEEE press.
- [6] C. Mangelsdorf, "A 400-MHz input flash converter with error correction," *IEEE Journal of Solid-State Circuits*, VOL. 25, NO. 1, February 1990.
- [7] Y. Akazawa *et al.*, "A 400 Msps 8b flash AD conversion LSI," in *ISSCC Dig. Tech. Papers*, vol. 30, 1987. pp. 98-99.
- [8] B. Razavi and B. Wooley, "A 12-b 5-Msample/s two-step CMOS A/D converter," *IEEE Journal of Solid-State Circuits*, VOL. 27, NO. 12, December, 1992, pp. 1667-1678.
- [9] T. Matsuura, T. Tsukada and S. Ohiba, "An 8b 20MHz CMOS half-flash A/D converter," *IEEE ISSCC* 1988, pp. 220-221.
- [10] H. Lee and B. Song, "A code-error calibrated two-step A/D converter," *IEEE ISSCC* 1992, pp. 38-39.
- [11] R. Roovers and M. Steyaert, "Design of CMOS A/D converters with folding and/or interpolating techniques," *Advanced A-D and D-A Conversion Techniques and their Applications*, 6-8 July 1994, Conference Publication No. 393, pp. 76-81.
- [12] B. Nauta and A. Venes, "A 70-MS/s 110-mW 8-b CMOS folding and Interpolating A/D converter," *IEEE Journal of Solid-State Circuits*, VOL. 30, NO. 12, December 1995, pp. 1302-1308.
- [13] A. Venes and R. Plassche, "An 80-MHz, 80-mW, 8-b CMOS folding A/D converter with distributed track-and-hold preprocessing," *IEEE Journal of Solid-State Circuits*, VOL. 31, NO. 12, December 1996, pp. 1846-1853.
- [14] G. Temes and J. Candy, "A tutorial discussion of the oversampling method for A/D and D/A conversion," *IEEE* 1990.
- [15] S. Nadeem and C. Sodini, "Oversampled Analog-to-Digital Converters," *Analog Circuit Design*, edited by J. Huijsing *et al.*, Kluwer Academic Publishers 1993.

- [16] Ramesh Harjani, Analog-to-Digital Converters, The Circuits and Filters Handbook, *IEEE Press*, pp 2098-2127, 1995.
- [17] W. C. Black, Jr. and D. A. Hodges, "Time interleaved converter arrays," *IEEE J. Solid-State Circuits*, vol. SC-15, pp. 1022-1029, Dec. 1980.
- [18] C. S. G. Conroy, D. W. Cline, and P. R. Gray, "An 8-b 85-MS/s parallel pipeline A/D converter in 1- μ m CMOS," *IEEE J. Solid-State Circuits*, vol. 28, pp. 447-454, Apr. 1993.
- [19] A. Petraglia and S. K. Mitra, "Analysis of mismatch effects among A/D converters in a time-interleaved waveform digitizer," *IEEE Trans. Instrum. Meas.*, vol. 40, pp. 831-835, Oct. 1991.

CHAPTER 4. Pipeline ADCs

4.1. Introduction

Pipelining is an implementation technique whereby multiple operations are overlapped in execution. Today, fast CPUs in particular and digital systems in general are mainly attributed to pipelining.

A pipeline is like an assembly line. In an automobile assembly line, there are many steps, each contributing something to the construction of the car. Each step operates in parallel with the other steps, though on a different car. A pipeline ADC, (PADC), consists of many *stages*, that are usually, but not necessarily, identical. The stages are connected one to the next to form a pipeline. In most of the implementations, the pipeline is preceded by a circuit, called *Sample-and-Hold* (S/H), used to quantize the input to the pipeline, which is an analog signal.

Each stage does some kind of processing on the input signal and then passes a new signal to the next stage. The main function of each stage is to give some information about the input signal to that stage. Each stage quantizes the input signal to a certain value, or bin. The Pipelined ADC is very similar to the multistage ADC, while the main difference between them is that pipeline ADC has S/H circuits between the stages. Each stage consumes one clock cycle to do the operation and all the stages are working at the same time, but each operates on a different sample of the input signal.

A block diagram that shows the operation of the pipeline ADC is shown in Figure 24.

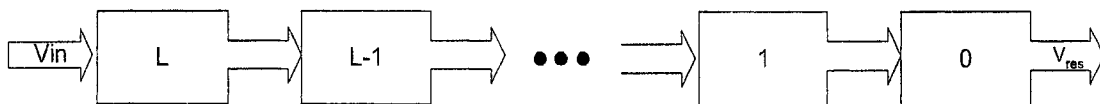


Figure 24 Pipeline ADC.

Usually, the first stage of a pipeline ADC is the S/H as shown in Figure 25. The main function of the S/H is to generate an output same as the input signal but at specified times. Figure 26 shows a sinusoidal input signal as the input to the pipeline, V_{in} and the sampled output, V_o .



Figure 25 Pipeline ADC with SH.

The main operations done by each stage are:

- giving information about the input signal in form of a digital number,
- amplifying, and,
- bounding the signal.

Usually, the last two operations are done together to insure that the next stage in the pipeline will be able to work on the signal passed to it from the current stage. This operation is done by subtracting a specific value from the input signal such that the result is within the range that the next stage can handle. This is called the *input range* of the next stage. A single stage of the PADC is shown in Figure 27.

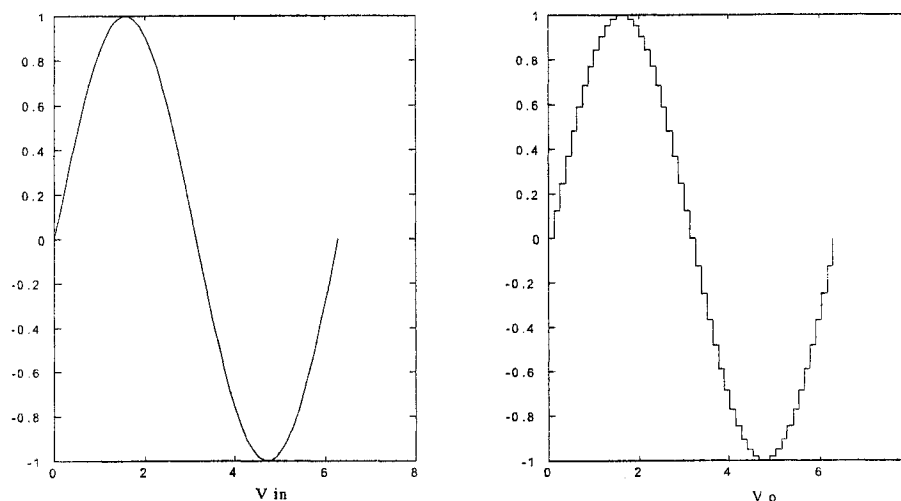


Figure 26 Input and output of the SH stage.

Each stage of the ADC consists of a subADC, subDAC and a subtractor. The subADC is an ADC that generates smaller number of bits than the original ADC. In this design, each subADC generates 1.5 bits and contains two comparators. It is a small flash ADC. The subDAC is a small DAC that takes the output of the subADC to generate an analog output to be used in the subtractor.

Fortunately, all these operations can be done in one circuit called the multiplying ADC, (MADC), [1]. This circuit is shown in Figure 28, where the subADC consists of the switching circuit that contains the comparators and some logic circuits and the subDAC and the subtractor are made of the switches, reference voltages and the operational amplifier.

The clocking scheme used to operate the operational amplifier circuit shown in Figure 28 is shown in Figure 31. This configuration has been chosen because it has smaller feedback ratio, β , than other configurations with the same closed loop gain. The smaller β means the faster the opamp to settle to its final value. A second advantage to this configuration, as will be shown later, is that it is less sensitive to capacitor mismatch, which means more accurate closed loop gain.

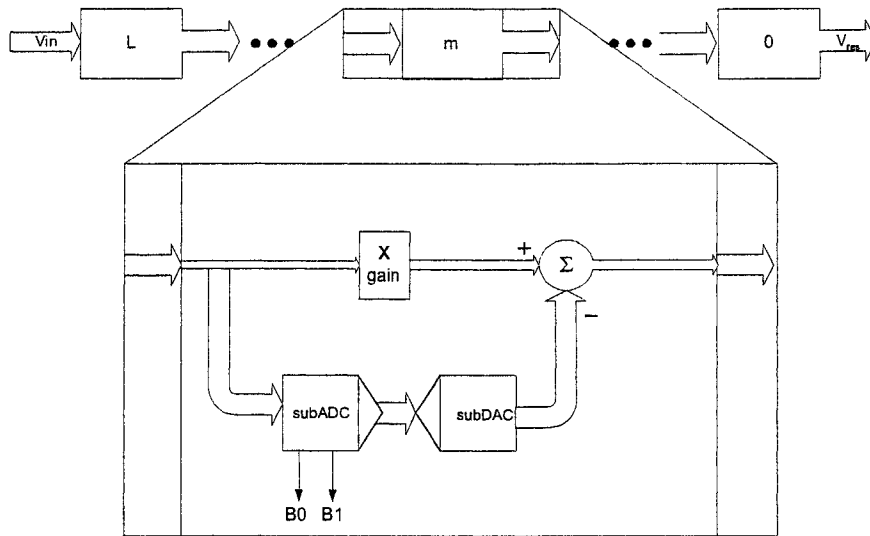


Figure 27 One stage of the PADC.

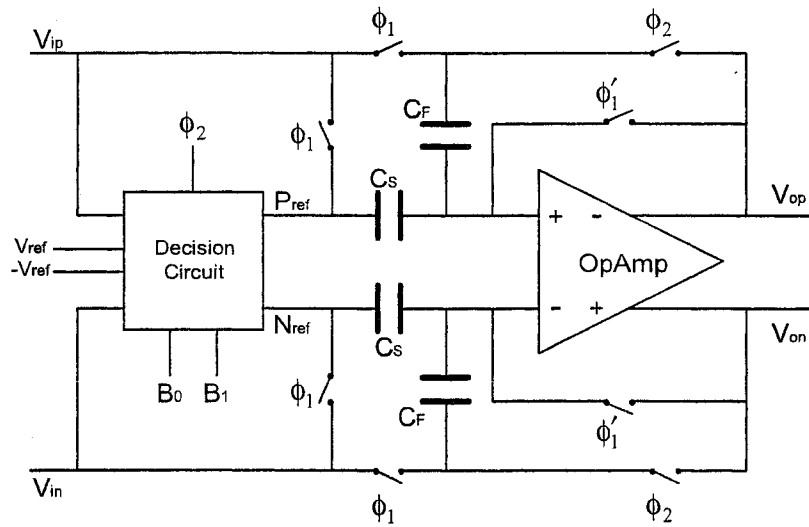


Figure 28 Multiplying ADC (MADC).

The switches shown in Figure 28 can be either a simple NMOS or PMOS transistor or a transmission gate that includes both. Usually, if a transistor is the choice for a switch, it is chosen to be NMOS, since it is stronger than the PMOS. However, simple transistor realization cannot be used as a switch if the input to that switch is varying too much. The reason is that a single transistor needs to have a sufficient excess bias voltage, V_{EB} , that will drive its gate. The overdrive voltage is defined to be:

$$V_{EB} = V_{gs} - V_t \tag{1}$$

Where V_{gs} is the gate-to-source voltage and V_t is the threshold voltage of the transistor.

The source voltage is going to be the input voltage. If the input voltage gets close to the gate voltage, then V_{gs} is very small and might be less than V_t and so, the transistor will not have enough overdrive voltage to be able to turn it ON or OFF.

To overcome this problem, there are two choices available in literature.

- 1) Use a clock-boosting scheme [2].
- 2) Use a transmission gate switch.

The first choice has some disadvantages and mainly used in low voltage applications.

Usually, switches that are connected to the input voltage in Figure 28 are realized using a transmission gate. Whether we use a simple transistor or a transmission gate, the overdrive voltage will still be dependent on the input voltage, although, this dependency is less for a transmission gate switch. This will result in input dependent switching.

To reduce this phenomenon, a special switching behavior is followed. This is depicted in Figure 28 and Figure 31. Two more clocks; ϕ'_1 and ϕ''_1 are used to help in reducing the input dependent switching phenomenon. The operation of this switching will be described along with the explanation of the operation of the MDAC.

The operation of the MADC is described next along with Figure 31 that shows the non overlapping clocks used to drive the MADC switches.

This configuration has been used in [1]. Note that there are 4 capacitors; two of them are called the *sampling* capacitors and the other two are called the *integrating* capacitors. Two of the 4 capacitors; one sampling and another integrating, are connected to the positive side of the opamp, while the other two are connected to the negative side of the opamp. The bottom plates of the sampling and integrating capacitors are connected to the summing nodes of the opamp.

Let's consider the capacitors connected to the negative summing node of the opamp: the sampling capacitor C_{s1} and the integrating capacitor C_{i1} . There are 4 clocks that control the operation of the circuit: ϕ_1 , ϕ'_1 and ϕ''_1 and ϕ_2 and their complements. ϕ_1 , ϕ'_1 and ϕ''_1 are identical except that ϕ'_1 goes down first, and then followed by ϕ''_1 and then ϕ_1 as shown in the Figure 31. ϕ_2 is the nonoverlapping complement of ϕ_1 clock.

This circuit has a reference clock as its input. A clock generator circuit generates the above 4 clocks and their CMOS complements. As ϕ_1 goes into one period, i.e., goes High and then Low, this circuit will go through two modes; *tracking* mode in which the circuit is tracking the input signals on both V_{ip} and V_{in} , and *holding* mode, in which the circuit will hold to a constant value. When the circuit changes from the tracking mode to the holding mode, the circuit is said to be sampling the input signal and the held value is going to represent the signal that is being sampled at this moment. The tracking mode occurs while ϕ_1 is High. At the same time ϕ_1 is High, ϕ'_1 and ϕ''_1 are also High. The circuit will stay in the tracking mode until ϕ'_1 goes Low first, followed by ϕ''_1 and then ϕ_1 , which will signal the end of the tracking mode. In this mode, the top plates of C_{s1} and C_{i1} are connected to the positive input, V_{ip} , while the top plates of C_{s2} and C_{i2} are connected to the

negative input, V_{in} . At the same time the bottom plates of all capacitors are connected together and to the common mode voltage. This is shown in Figure 29.

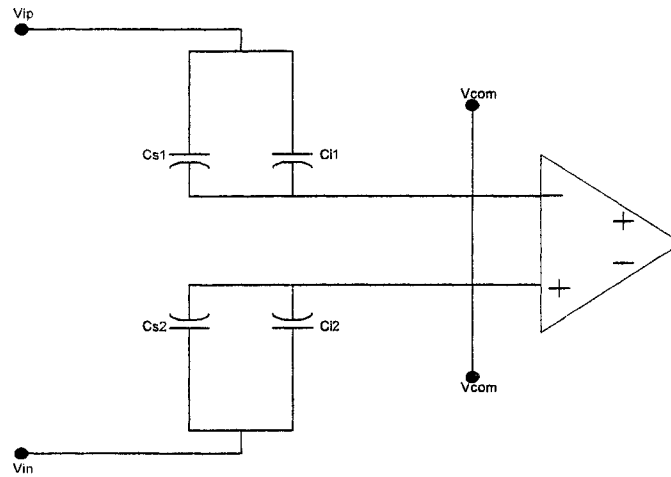


Figure 29 Operational amplifier circuit in tracking mode.

The sampling operation takes place in the transition of ϕ''_1 . It starts when ϕ'_1 goes Low first. This will disconnect the common mode voltage from the capacitors' bottom plates and also the summing input nodes of the opamp. So, the bottom plates of the capacitors are now floating and not connected to anything except themselves. This means that no more charge will be introduced to them. At this moment, if the input voltage change, the bottom plates' voltages may change accordingly in such away to keep the conservation of charge law valid all the time.

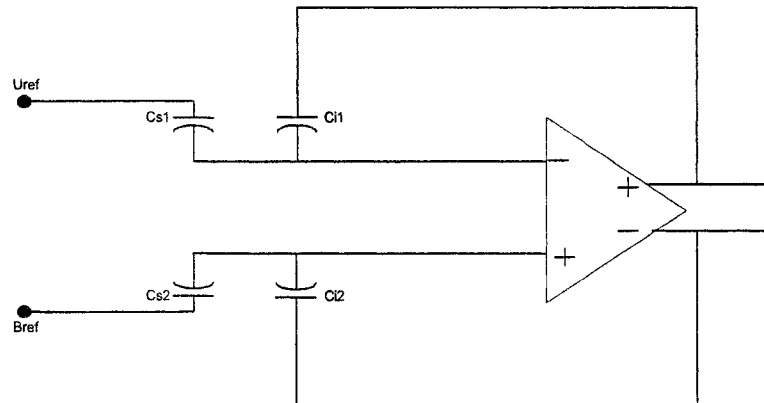


Figure 30 Operational amplifier in the holding mode.

After ϕ'_1 goes Low, ϕ''_1 goes Low, which will disconnect the two summing nodes from each others. This means that the bottom plates of the capacitors are disconnected now, so, no more charge transfer between the two summing nodes will take place any more. After this, the charges on the capacitors will be constant even if the input voltage changes its value, and the input is said to be sampled. Now, we can safely disconnect the

input signal from the top plates of the capacitors. This is done when ϕ_1 goes Low. After this, the capacitors are floating. The top plates of C_{s1} and C_{i1} as well as C_{s2} and C_{i2} are not connected to any thing, while the bottom plates of C_{s1} and C_{i1} are connected together, as well as the bottom plates of C_{s2} and C_{i2} . This condition will stay until ϕ_2 goes High, after which the holding mode starts.

The holding mode starts when ϕ_2 goes High. At this time, the circuit is configured in a negative feedback configuration, in which the positive output terminal is connected to the negative summing input node through C_{i1} and the negative output terminal is connected to the positive summing input node through C_{i2} . That is, the top plates of C_{i1} and C_{i2} are connected to the positive and negative output terminals, respectively, while the bottom plates of C_{i1} and C_{i2} are connected to the negative and positive summing nodes of the opamp, respectively. On the other hand, the top plates of C_{s1} and C_{s2} are connected to two reference voltages, UR and BR, respectively. This configuration is shown in Figure 30. The reason for this will be described shortly. The bottom plates of C_{s1} and C_{s2} are connected to the negative and positive summing nodes, respectively. The holding mode will complete when ϕ_2 goes Low. Mathematically, the above description can be modeled as follows:

While in the tracking mode, the charge stored on the capacitors will be:

$$Q_{cs1} = C_{s1} * v_{cs1} \quad (2)$$

$$Q_{cs2} = C_{s2} * v_{cs2} \quad (3)$$

$$Q_{ci1} = C_{i1} * v_{ci1} \quad (4)$$

$$Q_{ci2} = C_{i2} * v_{ci2} \quad (5)$$

Where Q_{cs1} is the charge stored on C_{s1} and v_{cs1} is the voltage across C_{s1} , which is $(V_{ip} - V_{com})$ where V_{com} is the common mode voltage. Same thing applies to the rest of the equations.

Assume that $v_{ip} = (V_{ip} - V_{com})$ and $v_{in} = (V_{in} - V_{com})$. Equations now can be rewritten as:

$$Q_{cs1} = C_{s1} * v_{cs1} = C_{s1} * v_{ip} \quad (6)$$

$$Q_{cs2} = C_{s2} * v_{cs2} = C_{s2} * v_{in} \quad (7)$$

$$Q_{ci1} = C_{i1} * v_{ci1} = C_{s1} * v_{ip} \quad (8)$$

$$Q_{ci2} = C_{i2} * v_{ci2} = C_{s2} * v_{in} \quad (9)$$

Let us consider one part of the differential circuit, for example the positive part. In the hold mode, the top plate of C_{s1} will be connected to a reference voltage, V_x . V_x might be less than V_{com} , equal to V_{com} or larger than V_{com} . Let v_x be $(V_x - V_{com})$ so, we might have v_x to be zero, a negative value or a positive one. If v_x is zero, this means that the top plates of C_{s1} and C_{i1} are connected to V_{com} , assuming that the common mode voltage of the circuit, which is $(V_{dd} + V_{ss})/2$, is equal to the common mode voltage of the input signal, which is $(V_{ip} + V_{in})/2$. The way this is implemented is by connecting the top plates of C_{s1} and C_{s2} together. Since the summing nodes now are connected only to the bottom plates of the capacitors, all the charge on C_{s1} will be dumped to C_{i1} . At the end of the hold period, the total charge on the bottom plate of C_{i1} will be Q_{ci1} plus Q_{cs1} and the top plate of C_{i1} will sink or source a charge from the output of the opamp accordingly. The reason for the addition operation

is that the bottom plates of C_{s1} and C_{i1} have the same charge sign, i.e., either both of them are positive or both of them are negative for both of them. If C_{s1} is equal to C_{i1} , then, according to equation (7), Q_{cil} is equal to Q_{cs1} and the total charge on C_{i1} is twice as it used to be when the input is sampled. Note that every thing here is with respect to the common mode, but not the absolute value of the input. The voltage across C_{i1} will be:

$$v_{cil} = Q_{cil} / C_{i1} = 2 * v_{in} \quad (10)$$

This means that the output voltage will be $(2 * v_{in} + V_{com})$ but not $2 * V_{in}$.

Now, let's consider another condition where v_x is positive, and less than v_{ip} . In this case some charge will stay on C_{s1} when being in the holding mode, i.e., not all the charge on C_{s1} will be dumped to C_{i1} . This charge will be $(C_{s1} * v_x)$. Thus, the total charge on C_{i1} will be:

$$Q_{cil} + (Q_{cs1} - (C_{s1} * v_x)) \quad (11)$$

So, the charge at the end of the hold mode on C_{i1} will be:

$$(C_{i1} + C_{s1}) * v_{ip} - C_{s1} * v_x \quad (12)$$

The voltage at the end of the hold mode on C_{i1} will be:

$$[(C_{i1} + C_{s1}) * v_{ip} - C_{s1} * v_x] / C_{i1} \quad (13)$$

$$= (1 + C_{s1}/C_{i1}) * v_{ip} - C_{s1}/C_{i1} * v_x \quad (14)$$

Hence, the output voltage will be, assuming $C_{s1} = C_{i1}$,

$$V_{op} = v_{cil} + V_{com} \quad (15)$$

$$= 2 * v_{ip} - v_x + V_{com} \quad (16)$$

So, the overall output voltage will be the difference between the input voltage and the common mode voltage multiplied by the closed loop gain subtracted from it the reference voltage and then added to the common voltage. This analysis reveals that the operation of this circuit is very similar to the large and small signal analysis of a transistor, where the Q point is the synonym of the common mode voltage here. From now on, we will follow this analysis, where we call that as the differential analysis (DA). Now, using the DA, the output voltage can be written as $v_{op} = 2 * v_{ip} - v_x$, where v_{op} is the positive output voltage in the differential analysis form, and it is equal to $(V_{op} - V_{com})$.

Following the same analysis, the relationship between v_{in} and v_{on} is given by:

$$v_{on} = 2 * v_{in} + v_x \quad (17)$$

$$V_{on} = 2 * v_{in} + v_x + V_{com}$$

The differential output is given by:

$$v_{op} - v_{in} = 2 * (v_{ip} - v_{in}) - 2 * v_x \quad (18)$$

$$= 2 * (v_{id} - v_x) \quad (19)$$

Where v_{id} is the differential input voltage.

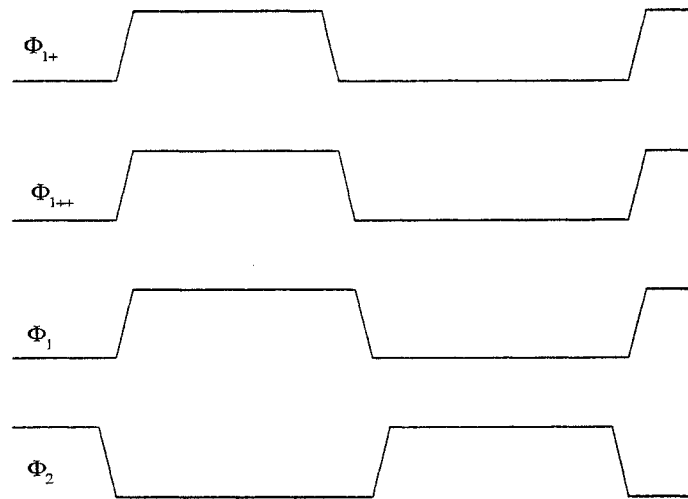


Figure 31 Non overlapped clocks for the operational amplifier circuit.

Before finishing the analysis of the operational amplifier, we need to understand that the two outputs of the opamp perform equation (16) and equation (17) independent of each other. What guarantees the differential operation is the common mode feedback circuit. This implies that equations (16) and (17) best describe the operation of the opamp.

This is important in the actual design since this will determine the DAC values to be added or subtracted from the input signal. Note that the DAC value in both equations: (16) and (17) is not multiplied by the gain of the stage, only the input is.

4.2. Pipeline Building Blocks

4.2.1. Operational Amplifier

Maybe the most important and difficult part of the ADC is the operational amplifier, (OA). Major part of the overall accuracy and resolution is attributed to the accuracy and resolution of the operational amplifier since it is the major source of error in the system. This is because of the direct operations it performs on the signal, and hence, the accuracy of these operations will affect the accuracy of the system directly.

There are many variables that determine the architectures of the OA a designer can choose. Usually, an OA with the following characteristics is required:

- High SNR and SNDR.
- Large SFDR.
- Low power, small area and large accuracy.
- High speed (small slewing and settling times).
- Large Input/Output swing.
- Low voltage.

- Designed in a cheap process.
- High CMRR and PSRR.

In addition to other parameters that might be considered that depend on the application and/or the process of design.

The above parameters are always conflicting and some of them have to tradeoff among each other. It is the responsibility of the designer to determine which of those parameters are important to him and which are not.

In the literature, there are many architectures for the operational amplifier. Each one of those has its own characteristics. Usually, they are categorized into the following categories: one stage, two-stage or more than that. Each one of those categories might also have different architectures. For example, one stage OAs can be simple transistor with an active load, a folded cascode or a telescopic OA. Detailed analysis of the theory of each type and category can be found in [3].

In this section, not all the architectures available in literature will be covered, however, I will cover the one used in my design in details.

In this design, we were targeting 10 bits at 100Msps, using the TSMC 0.25u digital process with 2.5V power supply.

Following are parameters we had:

- CMRR and PSRR are not important.
- Large Input/Output swing.
- Open loop DC gain is greater than 8,000 at nominal conditions of operation.
- Gain-Bandwidth greater than 500MHz.
- High SNR and SNDR.
- Low power.

Based on the above parameters, the folded cascode OA with boosting amplifiers has been chosen. The architecture shown in Figure 32 is considered as a single stage OA. This is mainly chosen to result in a faster OA. As the name suggests, a folded cascode with boosting amplifiers consists of a folded cascode OA with another amplifiers to boost up the gain as explained in [4].

The operational amplifier is going to be used in a pipeline ADC. In pipeline ADCs, stages work in a complementary fashion; i.e., while a stage is sampling, the next and the previous stages are holding. For a specific stage, the operational amplifier output is valid in only one phase of the clock. Since the ADC will be working at 100MHz, the period of the clock is 10ns, which gives the operational amplifier only 5ns to settle to its final value, which is $\frac{1}{2}$ of the period. This takes place in the hold phase. Practically, the time given to the opamp to settle is even less than $\frac{1}{2}$ of the clock period. This is because there are some activities that take place at the end of the hold phase such as the decision taken by the comparators and the selection of the DAC values necessary for the operation of the next stage.

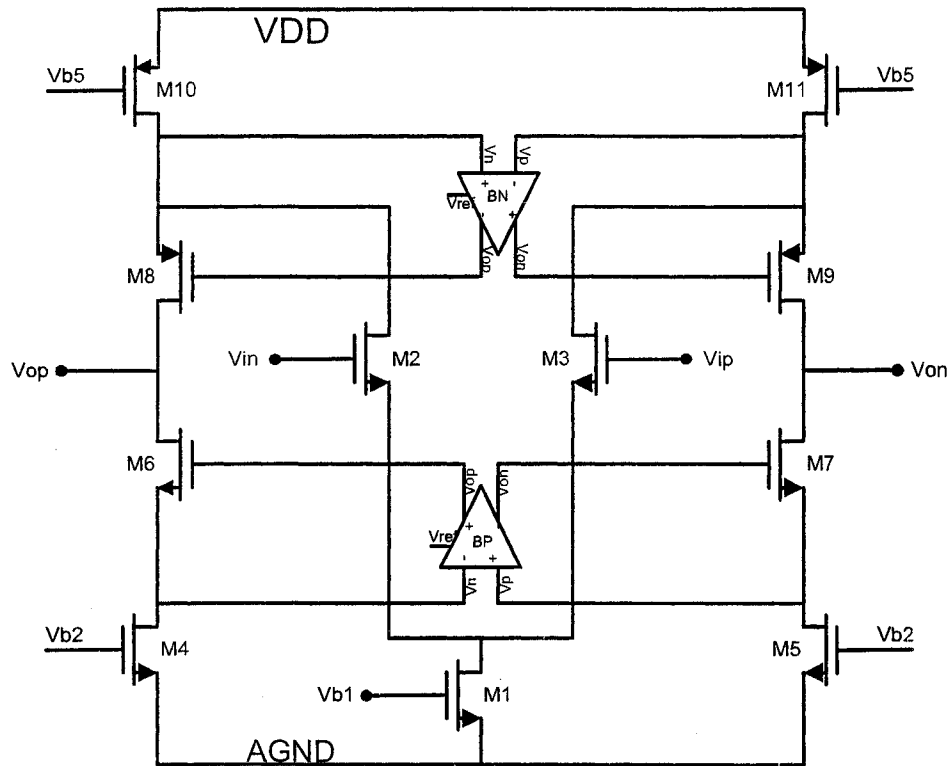


Figure 32 Folded Cascoded Operational Amplifier with gain boosting.

4.2.2. Common Mode feedback circuit

The operational amplifier, as well as other components in the ADC, is designed differentially. This requires a common mode feedback circuit, (CMFB), to guarantee the differential behavior of the OA and keep the quiescent point at the required value. A dynamic CMFB circuit can be used as shown in. A dynamic CMFB circuit was chosen because of the high bandwidth it has. One requirement for any CMFB circuit is that its bandwidth should be at least the bandwidth of the OA, otherwise, it will slow down the operation of the OA. The CMFB circuit shown in has the same bandwidth as the OA since it shares the same transistors used for in the signal path. The capacitors in the dynamic CMFB circuit should be chosen carefully since they increase the load of the OA. One more characteristic of this CMFB circuit is that it is small in area and consumes no power [5].

4.2.3. CMOS Comparator design

The comparator is a simple device that is being used as a decision-making circuit. In its simplest case, the comparator has two inputs; v_p and v_n , and one output; v_o [3]. The decision made by the comparator can be described as follows:

If v_p is greater than v_n , the output voltage, v_o , will be at logic 1, HIGH, while if v_n is larger than v_p , the output voltage, v_o , will be at logic 0, LOW. For CMOS operation, logic 1 or HIGH is close to VDD, while logic 0, or LOW is close to VSS, which is in most cases ground potential or 0V.

To better understand the operation of the comparator, we will assume that one of the input is always held constant, sometimes called the reference voltage, while the other input may be swept in order to see the characteristic of the comparator.

There are many parameters of the comparator that need to be addressed when we design it.

4.2.3.1. Offset voltage

In the definition above, it was mentioned that the output of the comparator would be HIGH if v_p is larger than v_n , and LOW if v_p is smaller than v_n . This means that the decision point where the output changes state occurs when $v_p = v_n$. This decision point is called the trip point, or sometimes the switching point. Ideally, $v_p = v_n$ should be the trip point, but practically, the trip point is displaced away from the $v_p = v_n$ point. That means that if we swept v_p while keeping v_n constant, the output of the comparator will not change state at $v_p = v_n$ point, rather, it will change state at the $v_p = v_n + v_{os}$ point, where v_{os} is called the offset voltage. Ideally, v_{os} is 0V but practically, it might be positive or negative value.

In order to have a good comparator, The offset voltage should be minimized to zero, however, this will make the design very difficult because that will trade with the speed of the comparator. In some designs, the offset voltage can be tolerated to a certain value. This will relax the design of the comparator, and give flexibility to other design parameters to be implemented easily. To clarify this, here is an example.

In [1], comparators are used in each stage as a subADC that will generate the digital value of the input and a controlling signals for the subDAC. A 1.5-bit per stage design where used that is unaffected with the comparators' offsets as long as they are less than $\pm \frac{1}{4} v_{ref}$.

4.2.3.2. Comparator Gain and Metastability

The following question may be raised: if $v_p = v_n$, what should be the output of the comparator, HIGH or LOW? Nobody really knows, even for ideal comparator and the output is not defined for this point. In practical comparators, the situation is a little bit worse. The output of the comparator is not only undefined for one point, (the trip point), but it is also undefined for a range of the input voltage. For the input voltage, this is called the minimum resolvable signal, or V_{mrs} , which means that if the difference between the two inputs is less than V_{mrs} , the output is not defined. When the input difference is less than V_{mrs} , the comparator is called to be in a metastable condition, or this phenomenon is called *metastability* [6].

One way to view the operation of the comparator is that it is amplifying the input difference by multiplying that difference by a certain value called comparator gain. Since the output of the comparator is limited by V_{dd} and V_{ss} , the input difference will not amplified beyond those two values, and that what gives the

logic operation of the comparator. If the gain of the comparator is not high enough, the output will not be a valid logic level and the comparator will be in the metastable state. So, one of the solutions to Metastability is to increase the comparator gain. The larger the gain, the smaller the V_{mrs} . Logically, this is true, since if we need a zero-valued V_{mrs} , we need a gain of infinity to do that, which is impossible to implement. This also means that we will never have a V_{mrs} of 0 value.

V_{mrs} can be defined as: V_{dd}/A , where A is the comparator gain [6]

4.2.3.3. Kickback Noise

In some comparator architectures the input signal is connected to the gate of a transistor, while the drain of that transistor represents the output node of the comparator.

Because of this configuration, there is an overlapping capacitor that connects the gate of the transistor with its drain. Since the outputs of the comparator will rail to either VDD or Vss, part of the output signal will be coupled to the inputs, called the kickback, and hence disturbs them. This is important if we have a reference ladder, since other reference voltages of different comparators will be affected as well, which means that wrong decisions may be taken. In high performance design, such as high speed and/or high accuracy, kickback noise should be minimized. A clear solution to this problem is to isolate the output nodes from the input stage. This technique is explained later, since it is becoming a common use for high performance comparators.

4.2.3.4. Speed

Sometimes, the comparator needs to run at the system full speed. Usually, the comparator takes its decision in steps, in order to overcome the problems it might have. The overall time the comparator takes from the moment it starts looking at the input signals until it takes the decision should be minimized, since that means the operating frequency.

In my system, the static comparator has been chosen over the dynamic one[7], since speed is more important in this design than power. The comparator consists of two stages followed by a regeneration latch. Regeneration has been proved to provide high speed for the comparator [8]. The designer has to pay attention to the value of the boosting capacitors. The second stage of the comparator will source to or sink current from the two boosting capacitors. The output current of the second stage should be sufficient to charge the capacitors to their final values in a half clock cycle. If the values of the two capacitors are large, it will take the second stage of the comparator too long to be able to charge them, and hence, the speed of the comparator will go down.

4.3. Error Sources in Pipeline ADCs

In this section, some error sources affecting typical implementations of pipelined ADCs are discussed. These error sources have historically limited the performance of pipelined ADCs. These error sources may be divided into two categories; noise, which varies from sample to sample, and mismatches, which do not vary

from sample to sample. This distinction has an important impact. Mismatch related errors could be corrected by calibration. Noise related errors, on the other hand, cannot be easily corrected by calibration [9].

The discussion in this section attempts to quantify the effect of some of the error sources on the performance of the ADC. In order to simplify the analysis, it is assumed in each of the sections below that a single error source acts alone in the absence of other errors. For example, when comparator offsets are discussed, noise and gain errors are assumed absent, and the DAC levels are assumed ideal. In a real pipelined ADC, a number of these errors could act simultaneously and this could result in compounding effects not predicted by this analysis.

The effect of these errors on the overall ADC depends heavily on the implementation of each stage. In the following, the effect of errors will be shown on two configurations, the 1-bit per stage and the 1.5-bit per stage.

A 1-bit-per-stage configuration is shown in Figure 33.a), where only one comparator is used to generate the binary output. Figure 33.b) shows the 1.5-bit per stage configuration, where two comparators are used to generate the binary outputs. A logic circuit is used to generate the control signal that will be used in the switching box to generate the correct DAC values.

The main components of each stage are two identical capacitors C_s and C_f , an operational amplifier and comparators. During the sampling phase, the voltage $V(n)$ at the input of the n th stage of the pipeline is sampled onto both C_s and C_f . Near the end of this phase, the comparators compare the $V(n)$ with the threshold voltages. For a 1-bit configuration, V_{th} is calculated as follows:

$$V_{th} = \frac{V_{refp} + V_{refn}}{2} \quad (20)$$

The digital output of the comparator $D(n)$ is:

$$D(n) = \begin{cases} 1 & V(n) \geq V_{th} \\ 0 & V(n) < V_{th} \end{cases} \quad (21)$$

However, for the 1.5-bit per stage, V_{th1} and V_{th2} are calculated as follows:

$$V_{th_1} = V_{refn} + \frac{3}{8} \cdot (V_{refp} - V_{refn}) \quad (22)$$

$$V_{th_2} = V_{refn} + \frac{5}{8} \cdot (V_{refp} - V_{refn}) \quad (23)$$

The comparators of Figure 33.b) is implemented differentially. The outputs of the comparators, D_1 and D_0 are related to the input $V(n)$ as shown in Table 3.

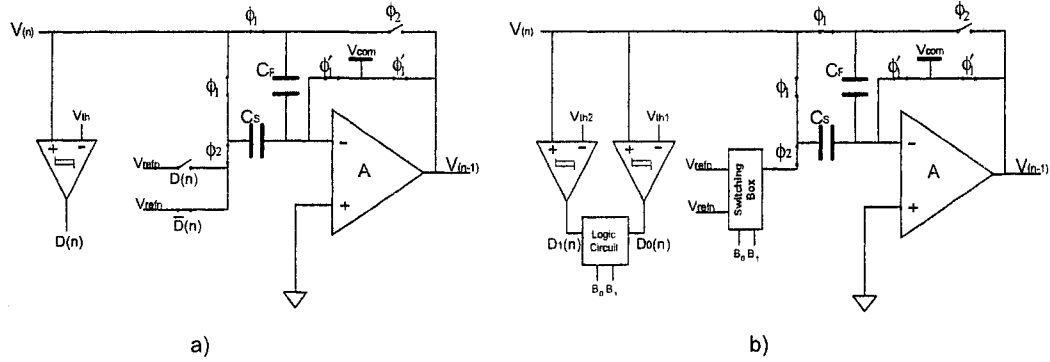


Figure 33 1-bit and 1.5-bit per stage configuration.

During the second phase of the operation of the stage, called the holding phase, the bottom plate of C_F is connected to the output of the op-amp, while the bottom plate of C_S is connected to either V_{refp} or V_{refn} depending on the value of the binary outputs.

Table 3 Digital outputs of the comparators.

Input	Output ($D_1 D_0$)	Required Output ($B_1 B_0$)
$V(n) < V_{th1}$	01	00
$V_{th1} < V(n) < V_{th2}$	00	01
$V(n) > V_{th2}$	10	10

If all the components in the stage are ideal, its output voltage or residue, $V(n-1)$ of the 1-bit per stage is given by:

$$V(n-1) = 2V(n) - D(n)V_{refp} - \overline{D(n)}V_{refn} \tag{24}$$

While the 1.5-bit per stage residue is given by:

$$V(n-1) = 2V(n) - D_1(n)V_{refp} - \overline{D_0(n)}V_{refn} \tag{25}$$

To understand the effect of this error on the overall pipeline, consider the representation of a 4-bit section of an N-bit pipeline ADC. Furthermore, the 3-bit pipeline section is assumed to be ideal with input range from V_{refp} to V_{refn} . For the purpose of illustration, only stage $n=4$ will be assumed nonideal.

If stage 4 is ideal, its output versus its input for the 1-bit and 1.5-bit per stage are shown in Figure 34.a) and Figure 34.b), respectively.

And the overall characteristics of the 4-bit ADC with 1-bit and 1.5-bit per stage are shown in Figure 34.c) and Figure 34.d), respectively. Note that the input and the output ranges are identical.

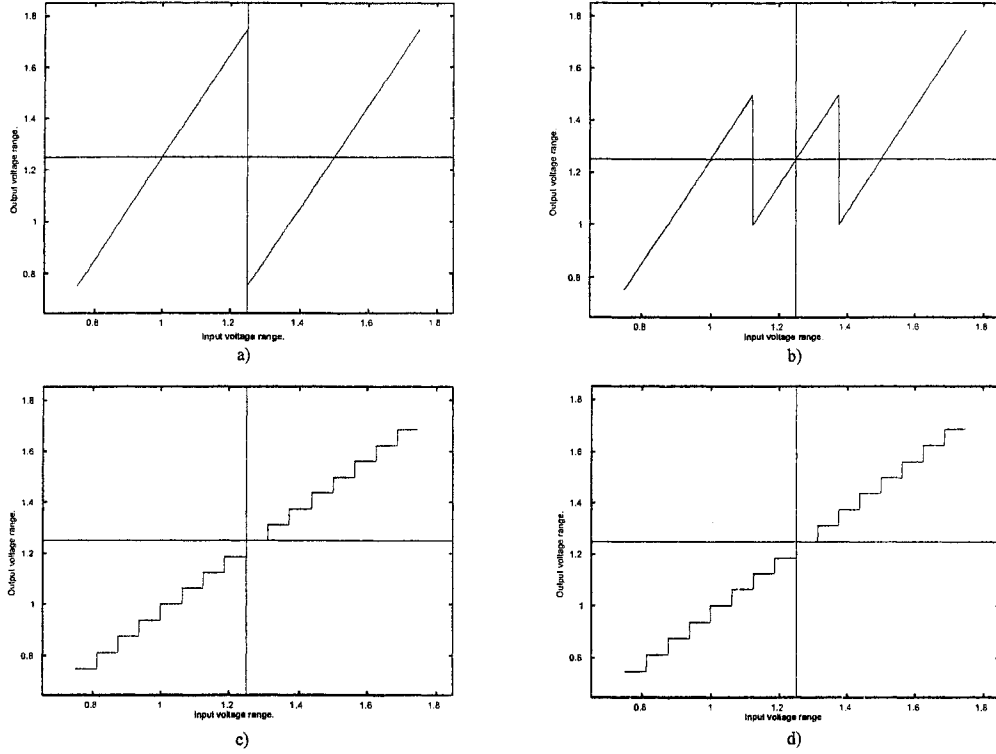


Figure 34 Stage output and ADC output of 1-bit and 1.5-bit per stage ADCs.

4.3.1. Capacitor Mismatch

Nonequal capacitors will result in the following equation for the 1-bit per stage:

$$V(n-1) = \left(1 + \frac{C_s}{C_f}\right) V(n) + \frac{C_s}{C_f} (-D(n)V_{refp} - \overline{D(n)}V_{refn}) \quad (26)$$

And for the 1.5-bit per stage:

$$V(n-1) = \left(1 + \frac{C_s}{C_f}\right) V(n) + \frac{C_s}{C_f} (-D_1(n)V_{refp} - \overline{D_0(n)}V_{refn}) \quad (27)$$

Referring to the overall ADC will have missing codes if the capacitor ratio of stage 4 is less than unity. The reason for this is that the output of stage 4 will not be able to reach the full range, which will be the following stage's full input range. The effect of this is that the 3-bit digital output word $D(3)D(2)D(1)$ will fail to reach all logic 1's before $D(4)$ transitions from a logic 0 to a logic 1.

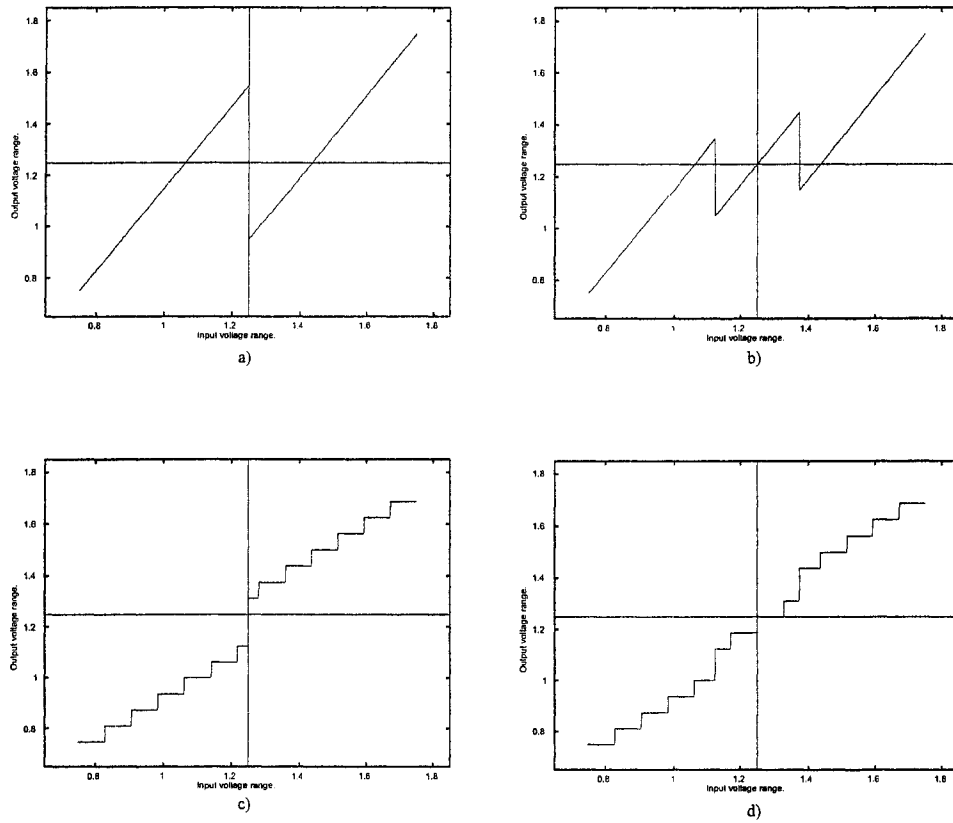


Figure 35 Gain error. $C_s/C_f = 0.6$.

Figure 35 shows the effect of capacitor ratio error on the performance of the ADC for the two configurations. It clearly shows that for the two cases missing codes result at the transition points of stage 4 if the capacitor ratio is less than 1. Figure 36 shows the effect of capacitor ratio is greater than 1. In this case, non-monotonicity exists. Capacitor ratio errors not only result in a slope error in the behavior of the stage, but also, it will affect the DAC values as shown in equation (26) and equation (27).

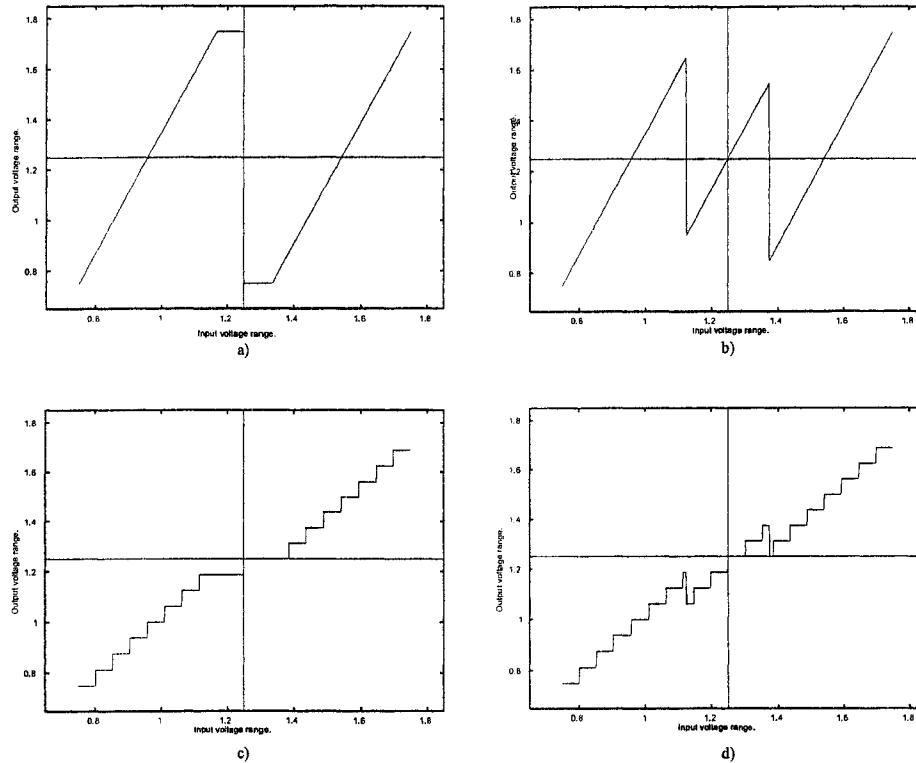


Figure 36 Capacitor ratio error. $C_s/C_f = 1.4$.

4.3.2. Comparator Offsets

The operation of the comparator is fundamental to the operation of the ADC. It comprises the subADC in each stage. The operation of the comparator can be thought of as a subtraction operation of the two inputs and generating a binary output of “1” if the difference is greater than 0 and “0” if the difference is less than 0.

The most critical error in the comparator is its offset. It may affect the overall ADC and may result in missing codes. The offset of the comparator can be modeled as a voltage that is added to one of the inputs but not the other. Practically, it is produced because of a mismatch between the two transistors in the differential pair that constitutes the input stage of the comparator. Thus, when the two inputs of the comparator are close to each other, the comparator may make a wrong decision and the binary output is wrong. This in turn will cause the wrong reference voltage to be subtracted from the input. The result is a residue that is out of range of the next stage of the pipeline when amplified. The above operation and its effect on the overall characteristic are illustrated in Figure 37, where a missing code has resulted.

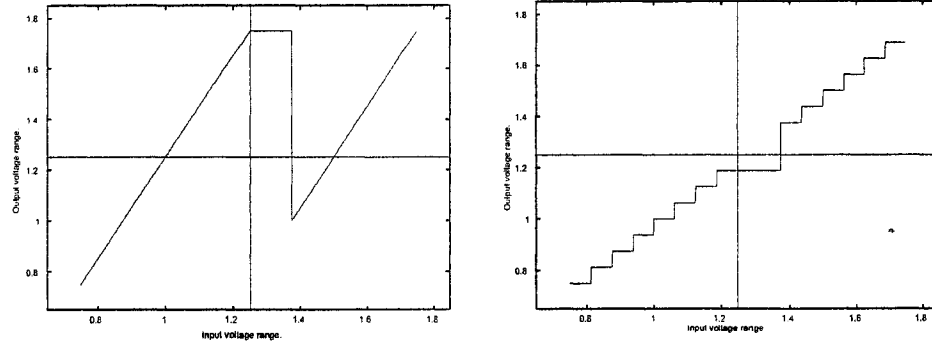


Figure 37 Effect of comparator offset on the 1-bit per stage ADC.

The 1.5-bit per stage ADC is not susceptible to comparator offset errors as long as those errors are less than $\pm \frac{1}{4} v_{ref}$, which is $\pm 125\text{mV}$. Figure 38.a) and Figure 38.b) shows the output of the 4th stage of the ADC and the ADC overall characteristic, respectively with an offset of $+125\text{mV}$ in the comparators' offsets. Those two figures show that nothing has changed in the overall characteristic of the ADC which is target requirement of the design. In Figure 38.c) and Figure 38.d), however, errors in the offsets of 187.5mV were introduced, and that clearly shows that the ADC suffers from missing code due to the saturation of stage 4 that resulted from an error in the offsets greater than 125mV .

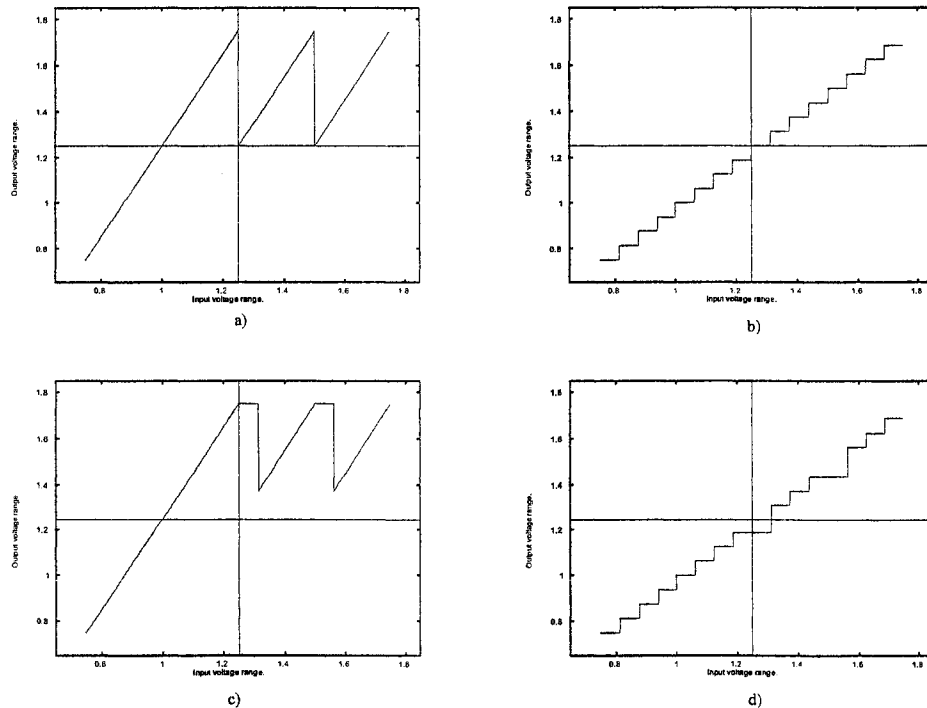


Figure 38 Effect of comparators' offset on the 1.5-bit per stage ADC.

4.3.3. Thermal noise

Thermal noise is caused by the random motion of electrons. All particles at temperatures above absolute zero are in random motion. Since electrons carry charge, the thermal motion of electrons results in a random current that increases with temperature. This noise current is present in all circuits and corrupts any signals passing through. In a pipelined analog to digital converter, the first stage circuit is the most important source of noise. Two noise sources are significant: the sampling switches and the operational amplifier. The noise in the sampling switch comes from the fact that practically when it turns on to it has a finite resistance. The sampling switch is used to sample the input signal onto a sampling capacitor. As this happens, noise from the sampling switch is sampled with it onto the sampling capacitor. This operation is illustrated in Figure 39 where the noise *rms* value is calculated as:

$$\sqrt{V_{noise}^2} = \sqrt{\frac{kT}{C}} \quad (28)$$

Where k is the boltzman's constant = $1.38e-23$, T is the temperature in Kelvin and C is the sampling capacitor. As an example, if $C = C_s = 1pF$, then the *rms* kT/C noise is $64\mu V$.

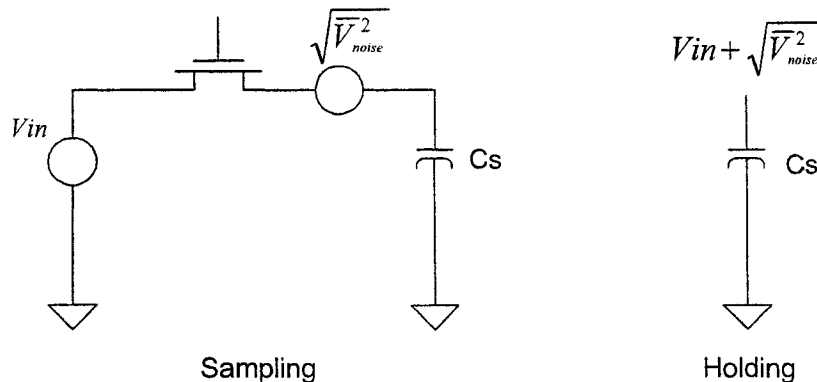


Figure 39 Thermal noise modeling.

This source of thermal noise is commonly referred to as kT/C noise because the noise power is proportional to kT/C where C is the size of the sampling capacitor. The operational amplifier also contributes thermal noise degradation to the signal being processed. The contribution of the sample and hold amplifier is also inversely proportional to a capacitance. In a single stage amplifier, it is inversely proportional to the load capacitance. In a Miller compensated amplifier it is inversely proportional to the compensation capacitance.

When designing an operational amplifier, usually minimum capacitor sizes are required for many reasons. The thermal noise puts lower limit on the size of the used capacitors. For example, for a 12-bit resolution ADC, the thermal noise of the overall ADC should be less than 1LSB. Since there are many sources of errors in the overall ADC, we might give the thermal noise a budget of $1/4LSB$, which will be equivalent to

0.153mV. So, the *rms* thermal noise should be less than that, thus:

$$\sqrt{\frac{1.38e-23 * 300}{C}} \leq 1.53e-4 \quad (29)$$

Or, $C \geq 177 \text{ fF}$. This suggests that for a 12 bit ADC with 2.5V, the minimum size capacitor is 200fF so that the thermal noise is not the major contribution to the overall linearity.

Thermal noise is perhaps the most fundamental source of error in a pipelined ADC. Because it is random from one sample to the next, it is not easily corrected by calibration. Thermal noise can be alleviated by using large components or by oversampling. However, for a fixed input bandwidth specification, both of these remedies increase the power dissipation. Thus, a fundamental tradeoff exists between thermal noise, speed, and power dissipation [3]

4.3.4. Charge injection and Clock feedthrough

4.3.4.1. Charge injection

Charge injection is the injection of charge from a transistor when it turns off into its nodes. Usually, this problem arises when a transistor is used as a switch. In this mode of operation, the transistor operates in the triode region, where V_{gs} usually goes to one of the rails depending on the transistor type. To understand this, we need to analyze a transistor in its triode region of operation. Lets consider an NMOS transistor. When the transistor is turned ON, V_{gs} needs to be HIGH which means that $V_{gs} \gg V_{th}$. Since the transistor is working in its triode region, V_{ds} needs to be very small and ideally, it should be 0. For the purpose of this analysis, we will assume that V_{ds} is very small compared to $V_{gs} - V_{th}$. When the transistor operates in the triode region, an inverted channel occurs which behaves as a conductor. This will create a virtual capacitor that has the gate and the inverted channel as its two plates, and the gate oxide material that is under the gate as its insulator.

The amount of charge per unit area that can be stored in this capacitor can be approximated by:

$$Q'_{ch} = C_{ox} (V_{gs} - V_{TH}) \quad (30)$$

And the total charge stored in the channel will be:

$$Q_{ch} = C_{ox} (V_{gs} - V_{TH}) \times W \cdot L \quad (31)$$

When the transistor turns OFF, Q_{ch} will be dumped to the source and drain of the transistor as shown in Figure 40 [6]. Although the percentage of the total charge that is dumped to the drain is not exactly determined, many people assume that to be 50%. The charge that is dumped to v_{in} is not problematic, since v_{in} is a source-driven node, but the charge injected to the sampling capacitor will cause a voltage change on the capacitor. If we assume that the gate voltage rails to V_{dd} when the switch is ON, and that 50% of the total charge stored in the transistor will be dumped to the capacitor, the change in voltage on the capacitor due to charge injection is:

$$\Delta v_{load} = -\frac{C_{ox}(VDD - v_{in} - V_{TH}) \times W \cdot L}{2C_s} \quad (32)$$

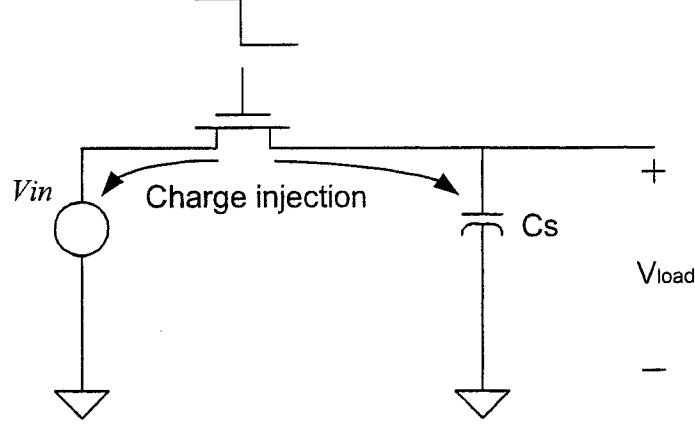


Figure 40 Charge injection for an NMOS switch transistor.

Equation (32) shows that the change in the voltage is signal-dependent which will result in signal dependant distortion of the signal. What makes things even worse is that the threshold voltage is also signal-dependant which will deteriorate the harmonic distortion of the circuit. The overall effect of charge injection on the system is that it adds to the nonlinearity of the system and causes the total harmonic distortion to drop.

4.3.4.2. Clock feedthrough

The clock feedthrough comes from the fact that a coupling exists between the gate of the transistor and its source and drain through two overlapping capacitors: C_{gs} and C_{gd} , where C_{gs} is the gate-to-source overlapping capacitor and C_{gd} is the gate-to-drain overlapping capacitor. As with the charge injection, when the transistor turns ON, the drain of the transistor is driven by the input signal and there is no clock feedthrough. When the clock signal that drives the gate of the switch turns OFF, a capacitive voltage divider exists between the gate-drain capacitance and the sampling capacitor as shown in Figure 41 where the overlapping capacitance is assumed to be half of the gate capacitance.

This will result in a voltage change on the sampling capacitor, C_s , according to the following equation:

$$\Delta v_{load} = \frac{C_{overlap} \cdot \Delta V_{gs}}{C_s + C_{overlap}} \quad (33)$$

Where $C_{overlap}$ is the overlapping capacitance value,

$$C_{overlap} = C_{ox} \cdot W \cdot LD \quad (34)$$

Where LD is the length of that overlaps the drain/source.

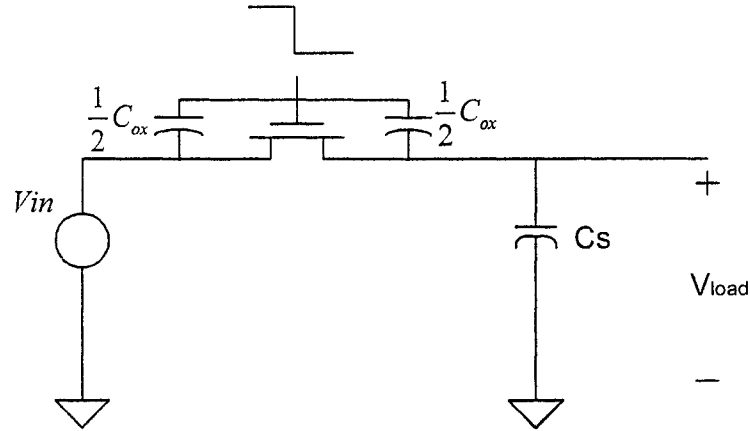


Figure 41 Clock feedthrough. modeling.

4.4. Channel-related errors

Time-interleaved ADCs are vulnerable to three major sources of errors. These are *timing* mismatch, or sometimes called *jitter*, *offset* mismatch and *gain* mismatch among the channels. Those are in addition to the errors that exist in each channel. Other minor errors exist but can be included in those mentioned above. The effect of these errors will be discussed and analyzed in what follows. A simulator of the ADC has been built. Each error is modeled differently in the simulator and the FFT plot of a sinusoidal input quantized with this ADC is plotted. The resolution of the input signal is 12 bits, and the resolution of the ADC is 6 bits. Although the input signal in the simulator can be set to any resolution and so as the ADC, the above numbers were chosen merely for illustration. One more thing we need to know about the simulator is that the input and the output of the ADC and every stage in it ranges between V_{refn} and V_{refp} , and the output saturates to either one of those if the input range tries to exceed its limit

4.4.1. Channel Gain mismatches

For an ideal channel, its gain should be 1 and the output should be a replica of the input. That means if the output is symbolized by y and the input by x , then the relationship between the input and the output should be $y=x$. However, in the actual implementation, the gain of each channel is not 1.

Assuming that the channel gain is the only source of error, the output is related to the input by the following relationship:

$$y = a \cdot x \quad (35)$$

Where a is not necessarily 1.

In the simulator, the gain errors are modeled as a gain mismatch in the first stage only as shown in the following equation:

$$V(n-1) = (1 + ChannelGainError) \cdot 2 \cdot V(n) - D_1(n)V_{refp} - \overline{D_0(n)}V_{refn} \quad (36)$$

Where *ChannelGainError* is the error in the gain of a certain channel. Figure 42 shows the transfer characteristic of 3 ADCs with different gains. Note the clipping of the transfer characteristic due to exceeding the input and/or output ranges. The mathematical analysis of the effect of gain mismatches among the channels in addition to the effect of both offset and timing mismatches are analyzed thoroughly in [10].

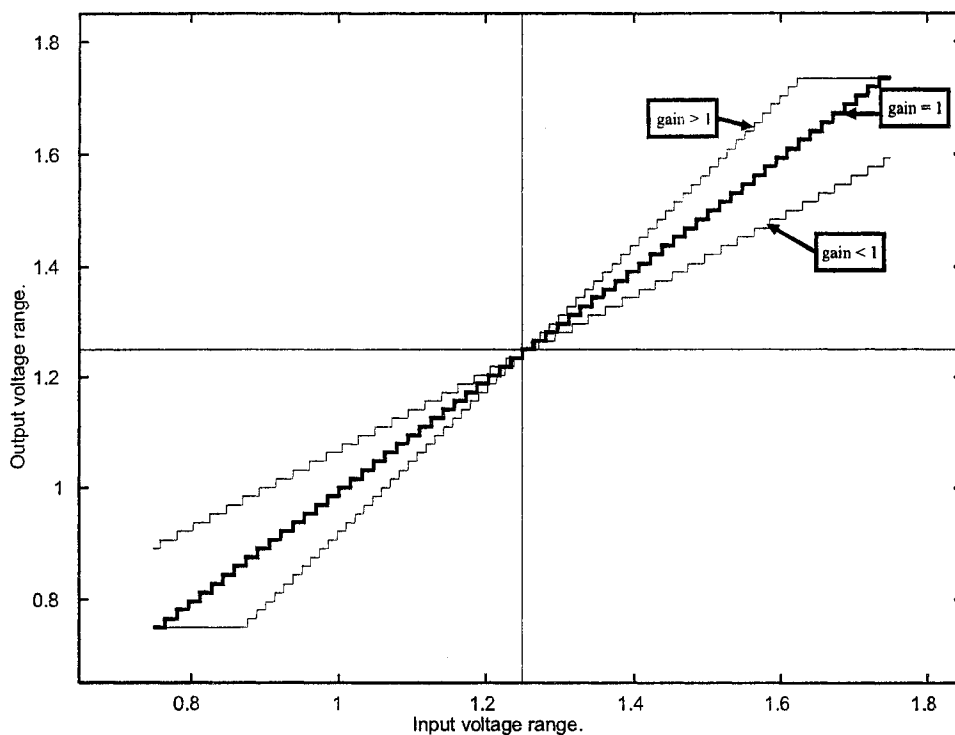


Figure 42 ADCs with different gains of the transfer characteristics.

The FFT plot of a sine wave input with a frequency of 100Hz quantized with the aforementioned ADC without errors is shown in Figure 43.a). Figure 43.b) shows the output of the ADC when it has gain errors. As the figure clearly shows that the effect of gain mismatch on the FFT plot of the output signal is that *side tones* or sometimes called *spurs* at:

$$\frac{fs}{M} \pm fin, 2 \frac{fs}{M} \pm fin, \dots, \frac{(M-1) \cdot fs}{M} \pm fin \quad (37)$$

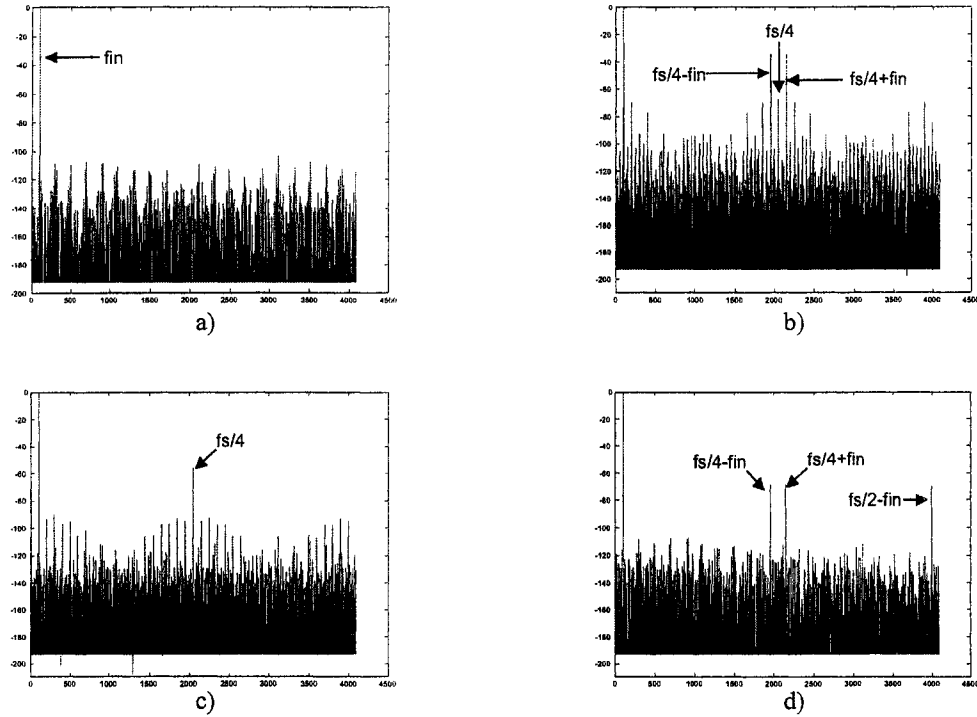


Figure 43 Effect of different errors in the frequency domain. A) No errors. B) Gain errors only. C) Offset errors only. D) Timing jitter only .

4.4.2. Channel offset mismatches

Channels with different offsets are shown in Figure 44. The offset in the transfer characteristic shifts it either to the left or to the right depending on the shift if it is positive or negative.

In the frequency domain, offset mismatch among the channels manifest themselves as spurs at:

$$\frac{fs}{M}, 2 \frac{fs}{M}, \dots \quad (38)$$

This is clearly shown in Figure 43.c).

The offset in each channel is modeled by introducing it to the following equation in the simulator:

$$V(n-1) = (1 + ChannelGainError) \cdot 2 \cdot V(n) - D_1(n)Vrefp - \overline{D_0(n)}Vrefn - offset \quad (39)$$

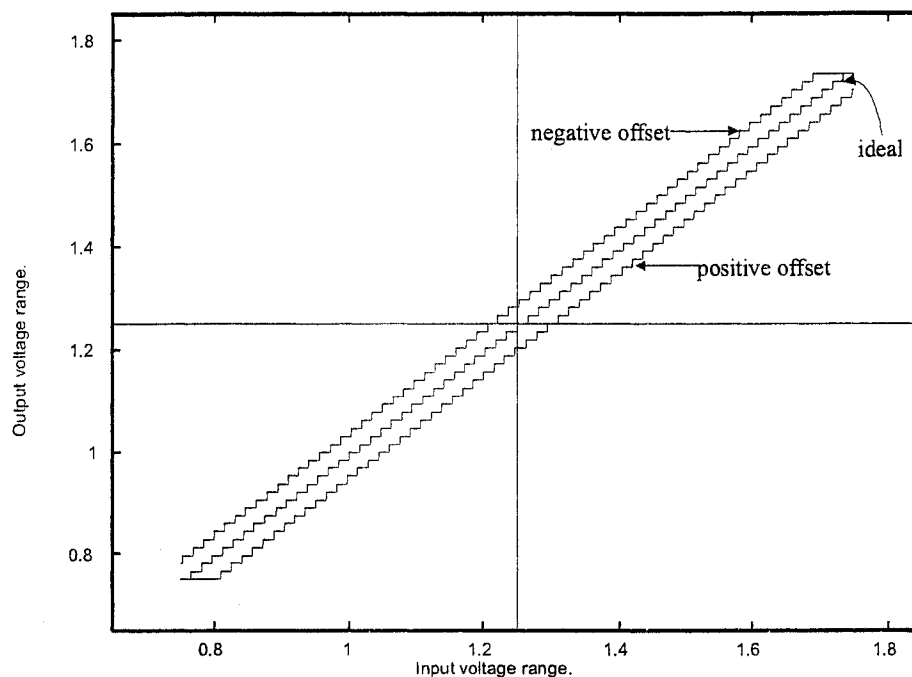


Figure 44 Offset errors in the transfer characteristic of each path.

4.4.3. Timing mismatch and Jitter

Unlike the previous two errors among the channels, timing mismatch is a little bit ambiguous to model. In order to be able to model it, the following procedure was followed.

The input signal, which was a 50 period sine wave, was digitized with a 12-bit resolution. This means that the number of points of the input signal was 2^{12} , which is 4096 points. Thus, each period was sampled ~ 82 samples, which means that the sampling frequency is ~ 82 times the input frequency. A new parameter called *spacing* was introduced to allow for timing mismatch to be modeled. When distributing the input signal to the channels, a number of samples, that is equal to the *spacing* variable, is skipped. So, if the first channel takes the first sample of the input signal, the second channel takes the $(\text{first}+1*\text{spacing})$ th sample, the 3d channel takes the $(\text{first}+2*\text{spacing})$ th sample and so on.

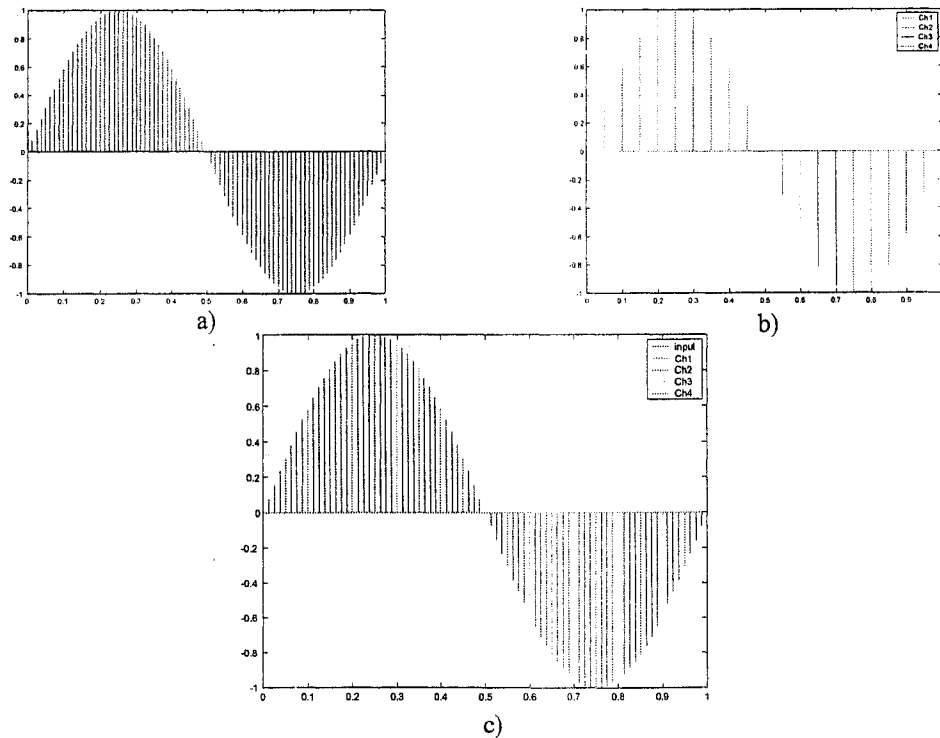


Figure 45 Jitter modeling in the parallel pipeline.

Since we have 4 channels, the total number of points a channel can see will be $(4096/(4*\text{spacing}))$. The jitter modeling using the spacing variable will be shown shortly. To illustrate the distribution of the input to the channels, let's consider one period that is sampled 80 times as shown in Figure 45.a). The *spacing* variable has a value of 4, which means that 4 samples of the input signal will be skipped before a channel takes the input. Assuming that the first channel starts by taking the first sample, the second channel will take the fifth sample instead of the second, the third channel will take the 9th sample instead of the third and so on. This means that the resolution of the input to each channel will be reduced by 2 bits that what it used to be. This is illustrated in Figure 45.b) and Figure 45.c) where the second channel, ch2, is represented by the red samples, the third channel, ch3, is represented by the blue samples ...etc. The jitter in the simulator was modeled such that instead of taking the i th sample of the input to a certain channel, $(i+1), (i+2), \dots, (i + \text{spacing}-1), (i-1), (i-2), \dots, (i - \text{spacing}+1)$ can be taken depending on the amount of jitter.

The effect of the jitter in the frequency domain is shown in the FFT plot of Figure 43.c), where tones around the channel frequency, fs/M , will appear. This is very similar to the gain errors.

The combined effect of the above three errors is shown in Figure 46 which clearly shows that as the sources of errors increase in the system, the performance deteriorates.

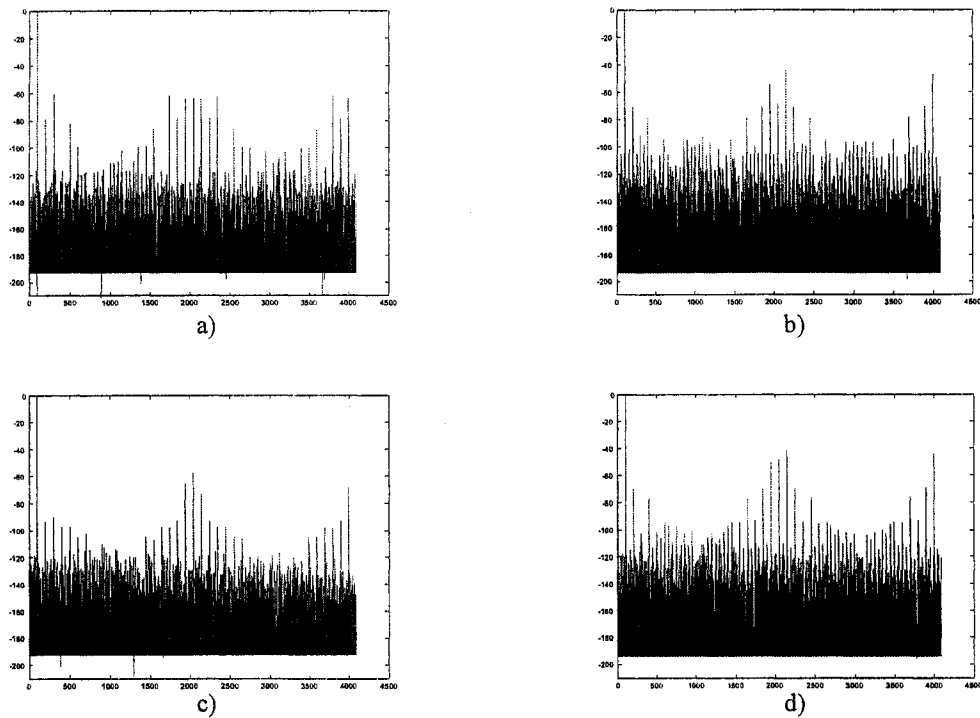


Figure 46 Effects of more than one error on the overall system. A) Gain and offset errors. B) Gain and timing errors. C) Offset and timing errors, and d) Gain, offset and timing errors.

4.5. Conclusions

This chapter talked about pipeline ADCs in particular, its building blocks as well as the sources of errors that exist in them. It also presented the effects of errors in a multipath ADC that may cause certain spurs to show in the FFT plot of the output of the ADC.

This chapter presented the guidelines that a designer needs to follow in order to be able to design the overall ADC. In particular, the design of the operational amplifier and its boosting opamps have been discussed. To be able to work at a 100MS/s, a fully differential folded-cascode with a fully differential boosting amplifiers was chosen, since it can achieve high dc gain as well high speed with minimum power. The operation and the design of the comparator was also presented. It was shown that in order to achieve the target speed, a static comparator was selected. The error sources in the design of comparators have been discussed and analyzed so that future designs of comparators may be robust enough to achieve correct operation of the ADC.

Other sources of errors in the design of an ADC have also been presented and analyzed in this chapter.

References

- [1] S. H. Lewis, H. S. Fetterman, G. F. Gross, Jr., R. Ramachandran, and T. R. Viswanathan, "A 10-b 20-Msample/s analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol. 27, pp. 351-358, Mar. 1992.
- [2] T. B. Cho and P. R. Gray, "A 10 b, 20 Msample/s, 35 mW pipeline A/D converter," *IEEE J. Solid-State Circuits*, vol. 30, pp. 166-172, Mar. 1995.
- [3] D. Cline, PhD thesis, University of California at Berkeley.
- [4] K. Bult and G. Geelen, "A fast-settling CMOS op amp for SC circuits with 90-dB DC gain," *IEEE J. Solid-State Circuits*, VOL. 25, NO. 6, Dec. 1990, pp. 1379-1384.
- [5] G. Nicollini, P. Confalonieri, and D. Senderowicz, "A fully differential sample-and-hold circuit for high-speed applications," *IEEE J. Solid-State Circuits*, vol. 24, pp. 1461-1465, Oct. 1989.
- [6] R. J. Baker, H. W. Li, and D. E. Boyce, CMOS Circuit design, layout and simulation, 1998 by the IEEE.
- [7] K. Y. Kim, N. Kusayanagi, and A. A. Abidi, "A 10-b, 100-MS/s CMOS A/D converter," *IEEE J. Solid-State Circuits*, vol. 32, pp. 302-311, Mar. 1997.
- [8] J.-T. Wu and B. Wooley, "A 100 MHz pipelined CMOS Comparator," *IEEE J. Solid-State Circuits*, VOL. 23, NO. 6, pp. 1379-1385, Dec., 1988.
- [9] I. E. Opris, L. D. Lewicki and B. C. Wong, "A single-ended 12-bit 20 Msample/s self-calibrating pipeline A/D converter," *IEEE Journal of solid-state circuits*, vol. 33, No. 12, December 1998, pp. 1989-1903.
- [10] C. S. G. Conroy, "High speed parallel pipeline A/D converter technique in CMOS," PhD desertion, University of California, 1994.
- [11] W. C. Black, Jr. and D. A. Hodges, "Time interleaved converter arrays," *IEEE J. Solid-State Circuits*, vol. SC-15, pp. 1022-1029, Dec. 1980.

CHAPTER 5. Implementation Of 10-Bit And 100 Ms/S Pipeline ADC

5.1. Introduction

In this chapter, a 10-bit ADC that has been implemented in silicon will be presented. The ADC consists of 9 stages, where each stage provides 1.5 bits of information except the last one which provides 2 bits. Error correction is used in every stage except the last one, which uses 0.5 bit provided by each stage in order to relax the comparator design. Thus, each stage resolves 1 bit of the input signal except the last stage, which provides 2 bits. The overall resolution of the ADC is 10 bits.

5.2. 10-bit Pipeline ADC

Figure 47 shows the block diagram of the overall ADC. It consists of 9 stages; S0 to S8. Each stage generates two bits: B0 and B1. The ‘shift register and adders’ circuit takes the generated 18 bits and generates the 10 bits as will be described later using what is called the digital correction technique. To allow for testability, the ADC can be configured as 3-bit, 4-bit, ..., or 10-bit converter, by using a MUX that connects the last stage to the first, second, ..., or the eighth stage, respectively. Without this configuration, the last stage doesn't need to have an OA, because there is no addition, subtraction or multiplication operation takes place. However, we intentionally provided the OA in a Sample-and-Hold configuration just to pass the output of the stage that precedes it.

For stage L in Figure 47, the output voltage is related to the input according to the following equation.

$$V_{oL} = A_L V_{iL} - D_L V_{xL} \quad (1)$$

Where V_{oL} is the output voltage of stage L , A_L is the gain of stage L , V_i is the input to stage L , D_L is the digital code generated by stage L to represent the input voltage and V_{xL} is the DAC value at stage L that is subtracted or added to the input voltage.

Similarly, the input/output relationship for stage $L-1$, can be given by:

$$V_{o(L-1)} = A_{L-1} V_{i(L-1)} - D_{L-1} V_{x(L-1)} = A_{L-1} V_{oL} - D_{L-1} V_{x(L-1)} \quad (2)$$

That is because $V_{i(L-1)} = V_{oL}$. Substituting equation (2) in equation (1):

$$V_{o(L-1)} = A_{L-1} V_{oL} - D_{L-1} V_{x(L-1)} = A_{L-1} [A_L V_{iL} - D_L V_{xL}] - D_{L-1} V_{x(L-1)} \quad (3)$$

Generally, for any stage, m , the output of that stage with respect to the input voltage is given by:

$$V_{o(m-1)} = (A_m A_{(m+1)} \dots A_L) V_{iL} - (A_m A_{(m+1)} \dots A_{L-1}) D_L V_{xL} - (A_m A_{(m+1)} \dots A_{L-2}) D_{L-1} V_{x(L-1)} - \dots - D_{L-m} V_{x(L-m)} \quad (4)$$

For the overall ADC that has $L+1$ stages, equation (4) can be written as:

$$V_{o0} = V_{res} = -[D_0 V_{x0} + A_0 D_1 V_{x1} + A_0 A_1 D_2 V_{x2} + \dots + (A_0 A_1 \dots A_{L-1}) D_L V_{xL}] + (A_0 A_1 \dots A_L) V_i \quad (5)$$

This equation is a general one that represents the ADC operation and will be called the *ADC characteristic equation* henceforth.

If we assume identical stages, which means that $A_L = A_{L-1} = \dots = A_{L-m} = A$, $V_{xL} = V_{x(L-1)} = \dots = V_{x(L-m)} = V_x$, equation (5) can be written as:

$$V_{res} = -[D_0 V_x + A D_1 V_x + A^2 D_2 V_x + \dots + A^L D_L V_{xL}] + A^{(L+1)} V_i \quad (6)$$

Which also can be written as:

$$V_{res} = A^{(L+1)} V_i - [A^L D_L + A^{(L-1)} D_{L-1} + \dots + D_0] V_x \quad (7)$$

The output of the ADC is the digital code, $D_0 \dots D_L$, while the output voltage which is V_{res} is always neglected and it is called the *quantization error* of the ADC. This error is inherent to all ADCs.

A digital circuit is always associated with the ADC that uses the digital output of the ADC to reconstruct its input, but in the digital domain, i.e., to give the binary representation of the input voltage.

While doing so, the digital circuit assumes ideal values for the stage gain and the DAC values. It also neglects V_{res} . According to the digital circuit, the following equation holds:

$$0 = A^{(L+1)} V_i - [A^L D_L + A^{(L-1)} D_{L-1} + \dots + D_0] V_x \quad (8)$$

Or:

$$A^{(L+1)} V_i = [A^L D_L + A^{(L-1)} D_{L-1} + \dots + D_0] V_x \quad (9)$$

$$V_i = [A^{-1} D_L + A^{-2} D_{L-1} + \dots + A^{-(L+1)} D_0] V_x \quad (10)$$

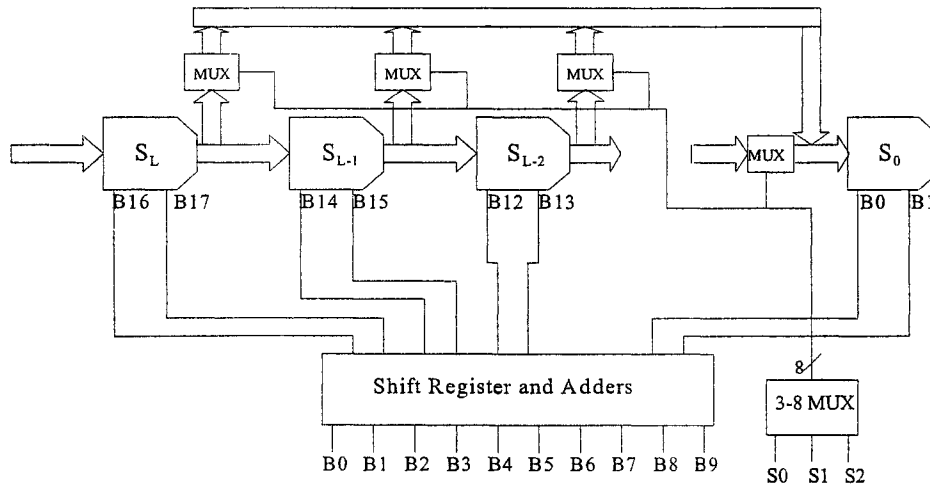


Figure 47 A block diagram of a 10-bit Pipeline ADC.

The bracketed term in equation (10) shows the digital representation of the input signal. Since the gain of each stage is 2, equation (10) says that the digital output of the ADC is formed by accumulating the digital output of stage L divided by 2, or equivalently shifted to the right by 1, and the digital output of the second stage divided by 4, or equivalently shifted to the right by 2 and etc. The circuit called 'Shift Register and Adders' in Figure 47 performs this operation.

5.2.1. Operation of one stage of the ADC

The output of each stage related to its held input is given by:

$$v_{od} = 2 v_{id} - v_x \quad (11)$$

Where v_{od} is the differential output, v_{id} is the differential input, and v_x is a reference voltage.

A one stage of the pipeline ADC is shown in Figure 48 which was implemented to perform equation (11).

The ADC was implemented in 0.25 μ m digital CMOS process, which has a power supply of 2.5V. The common mode voltage is 1.25V. Each of the input signals can range from 0.95 to 1.55 with an overall differential input range of 1.2V. The MADC described in the previous chapter performs the subADC, the DAC and generates the binary code shown in Figure 48.

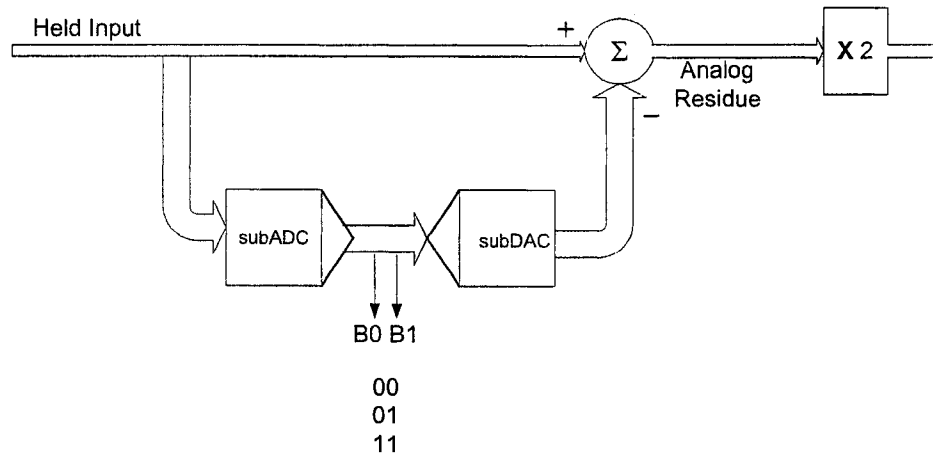


Figure 48 Block diagram of One-Stage Pipeline ADC.

Figure 49 shows the analog residue plot shown in Figure 48 with respect to the held input. The differential input to the ADC ranges from $-0.6V$ to $+0.6V$. The input to the comparator is also differential, which compares the negative input, v_{in} , with the v_{th1} , and the positive, v_{ip} , with the v_{th2} . V_{ref} equals $+0.3V$ in our implementation, this means that the differential input should be swept from $-2v_{ref}$ to $2v_{ref}$. Looking at one of the inputs will give more insight to the analysis and thus, the analysis that follows will consider only one input, mainly v_{ip} . Figure 49 shows the analog residue vs. the held input swept from $-v_{ref}$ to $+v_{ref}$.

In our implementation, the input of a stage ranges from 0.95V–1.55V and it determines the linear range of operation of that stage. This means that V_{refn} and V_{refp} are 0.95V and 1.55V, respectively. Each stage has to make sure that when it does any operation on the signal, the result is still within the linear range, otherwise, that stage or its successor will saturate and go out of its linear region. This will cause the ADC to give erroneous information about the input signal.

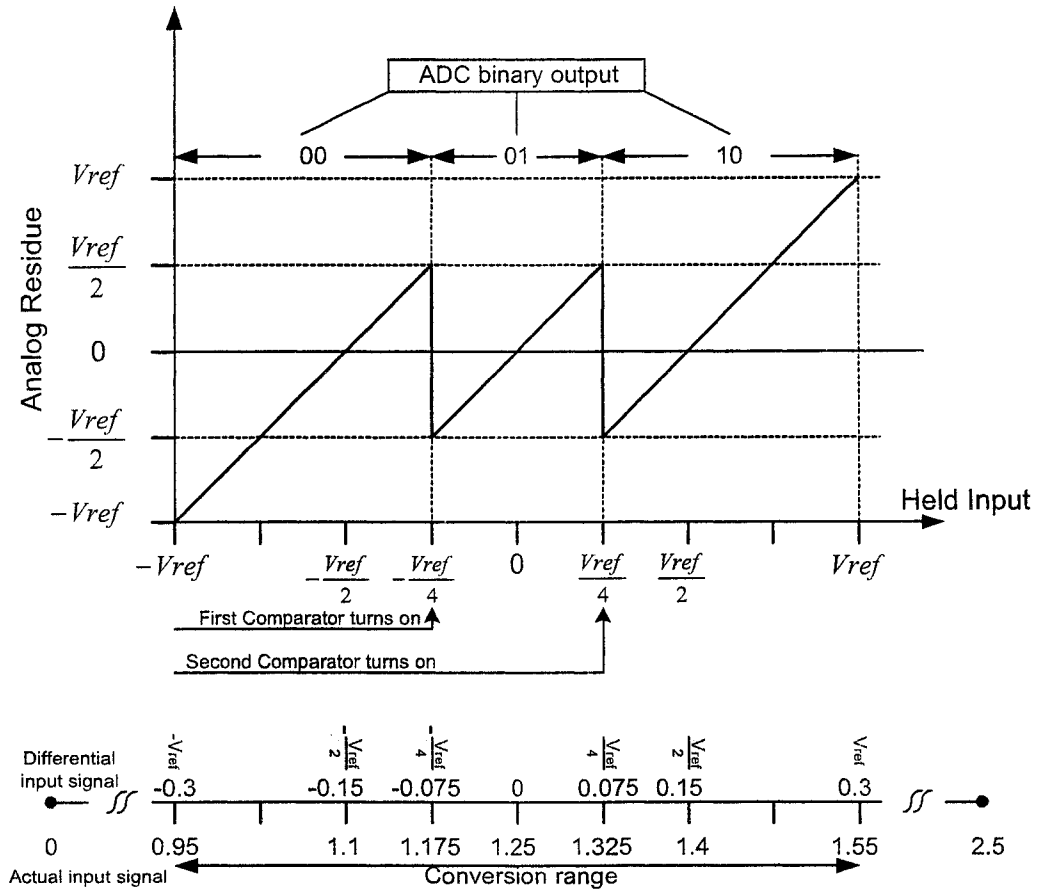


Figure 49 Analog Residue vs. Held Input for an ideal ADC.

So, it is very important to know when an error in the operation of a stage can occur so that it can be corrected before it goes out-of-range. When doing so, it is also important to know the direction of movement of the input signal as shown in Figure 49.

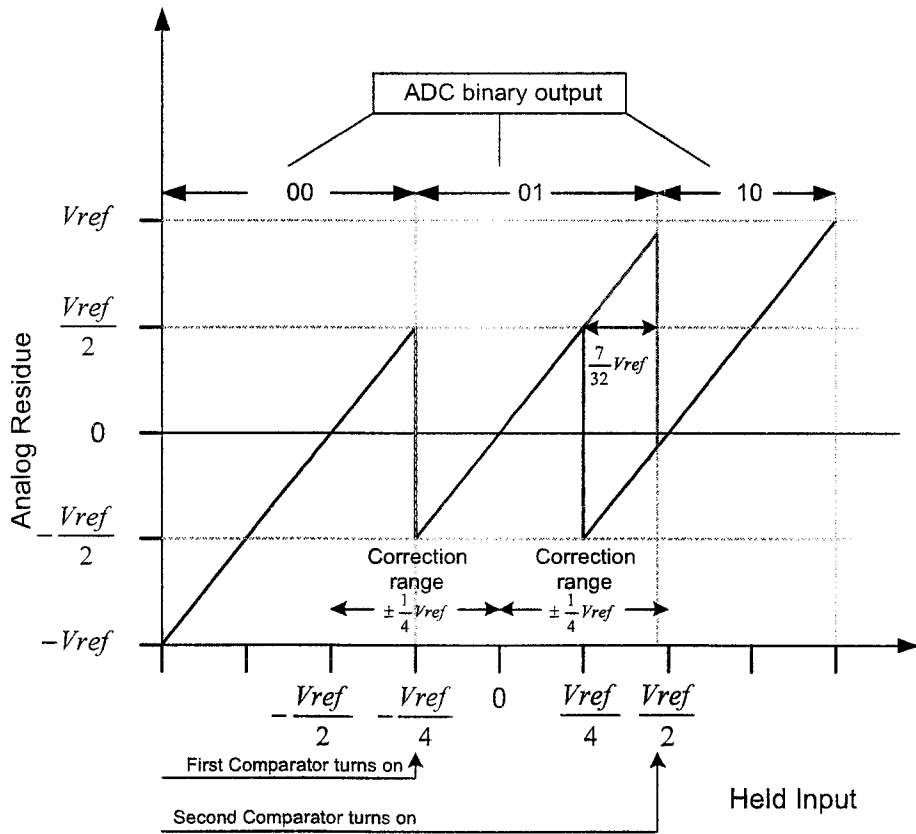
Figure 49 shows the mapping of v_{ip} . This figure can be described as follows. As v_{ip} starts at 0. It either moves up in the positive direction towards v_{ref} , or down in the negative direction towards $-v_{ref}$. As the input signal is swept from 0 to $-v_{ref}$, the output starts to follow the input signal until it reaches $-v_{ref}/4V$, where v_{ref} is added to boost up the output so that it stays in its linear region. When the input is swept from 0 to v_{ref} , the output starts to follow the input signal until it reaches $v_{ref}/4V$, where v_{ref} is subtracted to boost down the output.

The value of the input signal where boosting up or down takes place is called a *trip point* or *threshold voltage*. In this design, there are two trip points: $v_{th1} = -v_{ref}/4$ and $v_{th2} = +v_{ref}/4V$.

Two comparators were used to determine the two trip points. The first comparator is responsible for comparing the input signal to v_{th1} , while the second comparator is responsible for comparing the input signal with v_{th2} .

In practice, however, the comparators might include some offset. This means that the first comparator, for example, will change its output when the input signal passes $v_{th1}+v_{off1}$, rather than v_{th1} , where v_{off1} is the offset voltage inherent to the first comparator.

Our goal is to make the ADC generates correct output even when offsets exist in the comparators of each stage. This operation is illustrated in Figure 50.



- Ideal Residue Plot.
- - - Residue Plot with Second Comparator having $-7/32 V_{ref}$ offset.

Figure 50 Ideal and Nonideal transfer characteristic of a stage.

Figure 50 shows the ideal and nonideal analog residue plots. The figure also shows that the second comparator has an offset of $-7v_{ref}/32V$ and still the output is within the linear range. From Figure 50, the maximum offset that can be tolerated is $\pm \frac{1}{4}v_{ref} = \pm 125mV$ after which a stage goes out of its linear region.

This scheme of operation in which a stage works is called *error correction*. The error correction technique corrects errors in the comparators, which relaxes their design too much. This large amount of error that can be tolerated simplifies the design of the ADC, too. First, simple but fast comparators can be designed and second, comparators in the first stage can be connected directly to the input signal rather than connecting them at the output of the first stage. This, for sure, will introduce errors, but as long as the errors are less than the tolerable offset voltage, the error correction technique will take care of it. The impact of this way of connection will reduce the required number of stages by 1 to achieve the target resolution.

Each stage quantizes the input signal to one of three ranges, which means that it gives $\log_2(3) = \log_{10}(3)/\log_{10}(2) \approx 1.6$ bit. Due to the fact that each stage multiplies the analog residue by 2, only 1 bit of the 1.6 bit given by each stage is being used. In literature, however, this architecture is always referred to as 1.5bit per stage. The rest of the information is used as redundancy in order to allow for the previous scenario of operation for each stage to happen.

5.2.2. Operation of the subADC

The subADC of a stage consists of the comparators configuration in addition to the switching circuit. The main function of the switching circuit is to generate the voltages that are used to boost up or down the input signal so that it stays within the linear region of operation of that stage.

In our circuit, we always add a value to the input signal, either $+v_{ref}/2$ or $-v_{ref}/2$. Since we have two inputs to the circuit, v_{in} and v_{ip} , at the same time, and those two inputs are differential, we need to have two reference voltages available at the same time, too. In order to do so, the comparators sense the inputs and based on that they generate the reference voltages. $Pref$ is the reference voltage connected to v_{ip} , which is $-v_{ref}$, while $Nref$ is the reference voltage connected to v_{in} , which is $+v_{ref}$.

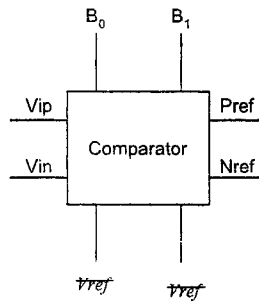


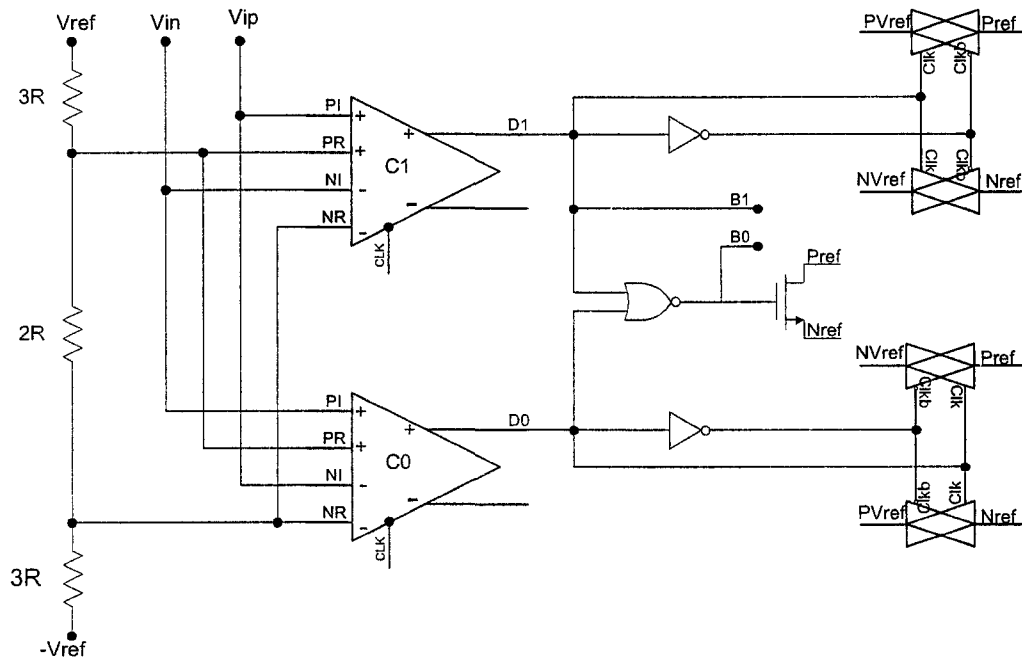
Figure 51 Block diagram of the Comparators' circuit.

Figure 51 shows the block diagram of the comparator circuit, where it has two inputs; v_{ip} and v_{in} and two reference voltages. It generates the two digital bits, B0 and B1 in addition to the two reference voltages: P_{ref} and N_{ref} . The complete schematic of Figure 51 is shown in Figure 52 where the outputs of the comparators are used to generate the digital outputs: B0 and B1, in addition to controlling the switches in order to generate the reference voltages.

5.2.3. Operation of the comparators

The threshold voltages of the comparators are set using a resistor string. The first threshold voltage, V_{th1} , is $-v_{ref}/4$ from the common mode voltage in a $-v_{ref}$ to $+v_{ref}$ input range. For this design, in general:

$$V_{th1} = V_{refn} + \frac{3}{8} \cdot (V_{refp} - V_{refn}) \quad (12)$$



Input	Output (D1D0)	Required Output (B1B0)	Pref	Nref
$v_{ip} < V_{th1}$	01	00	Nvref	PVref
$V_{th1} < v_{ip} < V_{th2}$	00	01	0	0
$v_{ip} > V_{th2}$	10	10	PVref	Nvref

Figure 52 Switching circuit that generates the digital data and reference voltages.

Where V_{refp} and V_{refn} are the maximum and minimum values the input can take, respectively. With $V_{refp} = 1.55V$, $V_{refn} = 0.95$, $V_{th1} = 1.175V$. Similar to equation (12), V_{th2} is set according to the following equation:

$$V_{th2} = V_{refn} + \frac{5}{8} \cdot (V_{refp} - V_{refn}) \quad (13)$$

With $V_{refp} = 1.55V$, $V_{refn} = 0.95V$, $V_{th2} = 1.325V$.

For the sake of clarity, let's consider a single ended input, say v_{ip} , to the switching circuit. The trip point of the first comparator, C_0 , is V_{th1} and it is V_{th2} for the second one, C_1 . According to Figure 52, C_1 generates the MSB bit, while C_0 generates the LSB bit. The output of C_0 is Low as long as v_{ip} is greater than V_{th1} and High otherwise. The output of C_1 is Low as long as v_{ip} is less than V_{th2} and High otherwise. Table 4 shows the digital output that is made of D_1D_0 , where D_1 and D_0 are the positive outputs of C_1 and C_0 , respectively.

Table 4 Digital outputs of the comparators.

Input	Output (D_1D_0)	Required Output (B_1B_0)
$v_{ip} < V_{th1}$	01	00
$V_{th1} < v_{ip} < V_{th2}$	00	01
$v_{ip} > V_{th2}$	10	10

Going back to Figure 52, NVref corresponds to V_{refn} , while PVref corresponds to V_{refp} . Pref and NVref are the positive and negative DAC values, respectively, that are going to be connected to the bottom plates of the sampling capacitor when the operational amplifier is configured in the hold mode.

Considering one side of the DAC, say Pref, if V_{refp} is connected to Pref, then an addition operation will take place, while if V_{refn} is connected to Pref, a subtraction operation will take place. Based on the value of the input signal, either addition or subtraction of v_{ref} should be performed. This decision is made based on the outputs of the comparators. The configuration of the comparators shown in Figure 52 is implemented so that minimal logic is used to turn ON or OFF the switches. As can be seen from Table 4 a NOR gate is needed only when the two outputs of the comparators are 00, which implies that the DAC values need to be 0.

The circuit configuration shown in Figure 52 has the advantage that a minimal logic is used to generate the signals that will result in the correct DAC values. From the circuit in the same figure and according to Table 4, it is also shown that no extra logic is needed to generate the required output, B_1B_0 of the switching circuit. According to Table 4

$$B_1 = D_1, \quad B_0 = \overline{(D_1 + D_0)} \quad (14)$$

So, B_1 is directly taken from the output of the MSB comparator, while B_0 is taken after the NOR gate that is used to generate the signal when the DAC value is 0.

As can be seen in Table 4 and Figure 52, each stage generates 2 bits, however, it is contributing only one bit to the overall resolution of the ADC. This is accomplished by using *digital error correction*, which overlaps one bit of each stage with the next stage.

But how does the error correction work? This is described in the next section.

5.2.4. Digital error correction

Digital error correction is a technique used to prevent comparator offsets from limiting the resolution of an analog to digital converter. In this technique, the comparator offsets may not be zero. Instead, the ADC is designed in a way that is tolerant to comparator offsets. Without digital error correction, the comparator offset must be no more than the least significant bit of the ADC. With digital error correction, larger offsets can be tolerated. This technique is attractive because it allows the use of simplified comparators. This can potentially save hardware and power. This technique also allows analog to digital converters to achieve resolutions that would not be possible without it.

To show how this works, consider the following example. Assume that the ADC is ideal and consists of 2 stages only: S1 and S0, where S1 represents the most significant stage. Suppose the input to S1 is $+7V_{ref}/32V$. According to Figure 53, the ideal output of S1 will be $-1/16V_{ref}$ with a digital code of 10, while the second stage will have a residue output of $-1/16V_{ref}$ with a digital code of 00. The nonideal first stage, S1, with an offset of $+7/32V_{ref}$, the output of S1 will be $7/16V_{ref}$ with a digital code of 01 and the second ideal stage, S0, will have an output of $-1/16V_{ref}$ with a digital code of 10.

The final digital code generated by the ideal ADC will be:

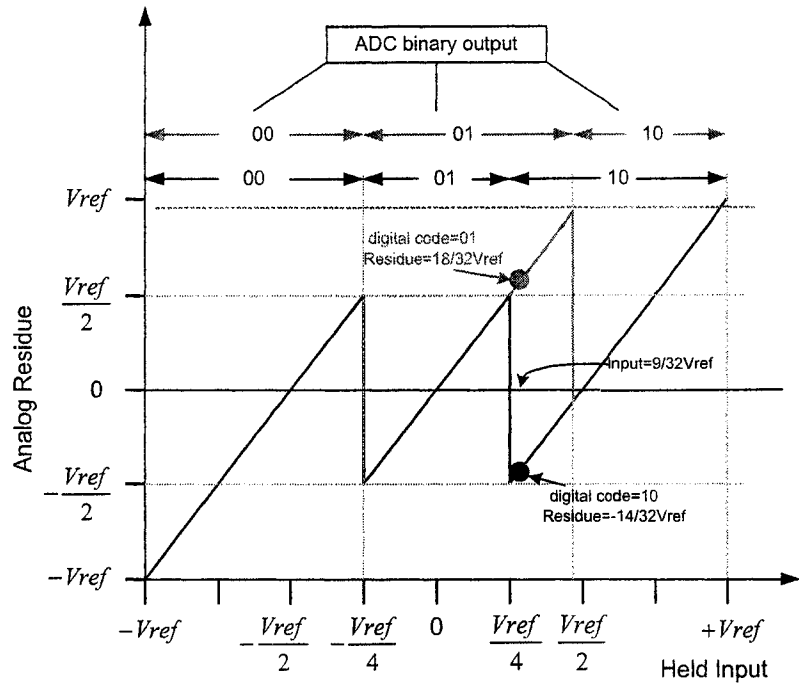
$$\begin{array}{r} \\ 1 0 \\ + 0 0 \\ \hline 1 0 0 \end{array}$$

The nonideal ADC with an offset of $+7/32 V_{ref}$ will generate the following digital equivalent:

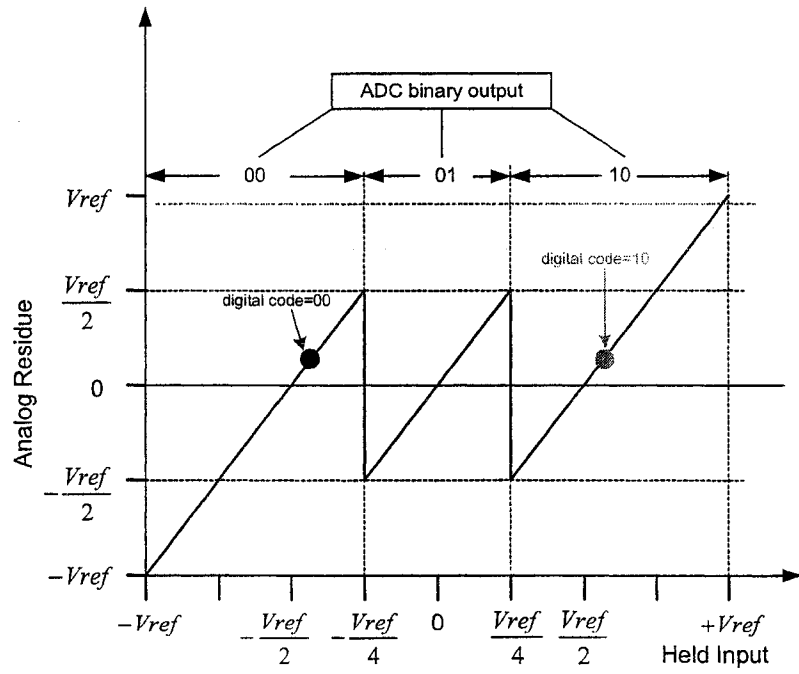
$$\begin{array}{r} \\ 0 1 \\ + 1 0 \\ \hline 1 0 0 \end{array}$$

We can clearly see that both of the digital outputs of the ideal and nonideal ADCs are the same.

Figure 54 shows the quantized output of an ideal 4-bit ADC that doesn't have offsets in the comparators. It clearly shows that there are 16 quantized outputs and the DNL is 0LSB. Figure 54 also shows the residue of the first stage in the ADC, which agrees with Figure 49.



First Stage



Second Stage

Figure 53 First stage with offset and ideal second stage.

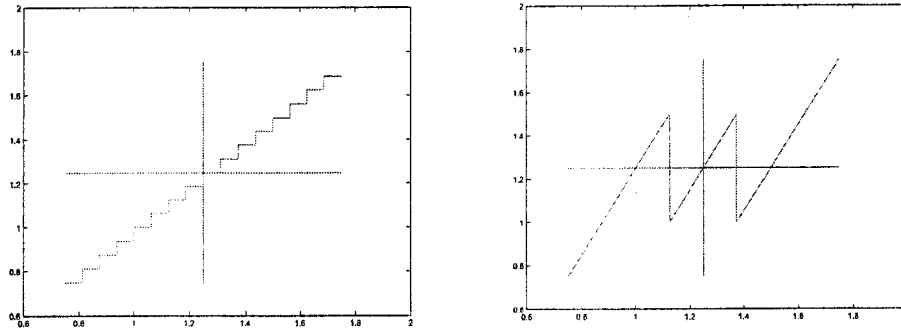


Figure 54 Overall transfer characteristic and residue plots of an ideal ADC.

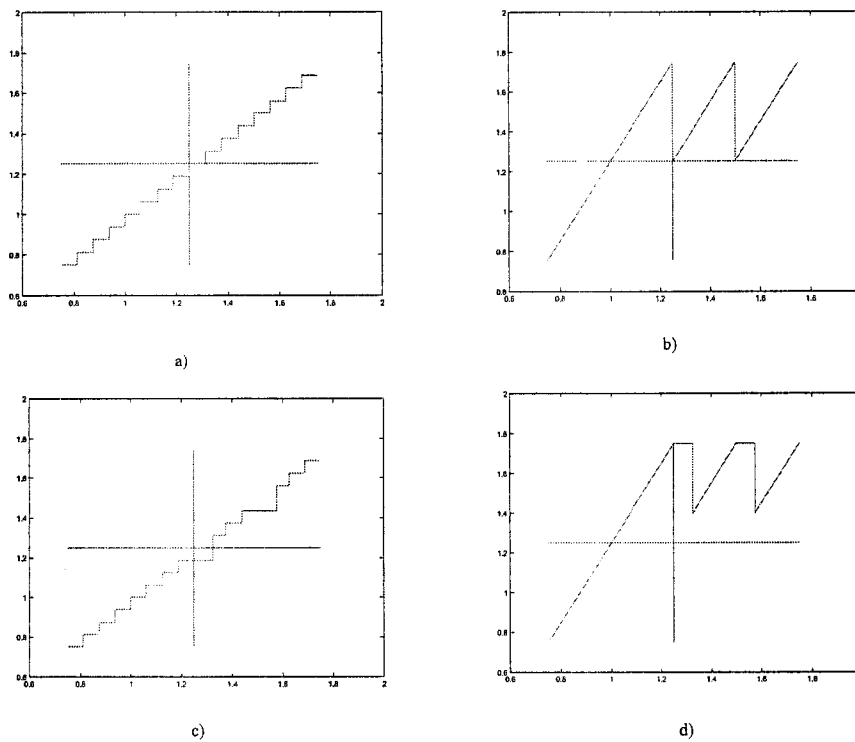


Figure 55 Overall transfer characteristic and residue plots of a non-ideal ADC.

Figure 55.a) shows the overall characteristic of a non-ideal 4-bit ADC that has its last 3 stages ideal and the first stage is not ideal. Comparing Figure 55.b) to Figure 54, the error introduced in the comparator offset is $v_{ref}/4$, which is 125mV and the ADC is still vulnerable to this error. Actually, $v_{ref}/4$ is the maximum error that can be tolerated using this technique. The reason for this is that an error greater than this amount will result in the stage saturation as shown in Figure 55.d) which will result in errors in the transfer function as shown in Figure 55.c).

5.2.5. Implementation of one stage

Each stage of the ADC consists of an operational amplifier, four capacitors and a switching circuit as shown in Figure 56. The switching circuit performs two operations; generating the digital equivalent value of the input signal through B0 and B1, and preparing the appropriate DAC values, v_{ref} or $-v_{ref}$, to be added to the input signal.

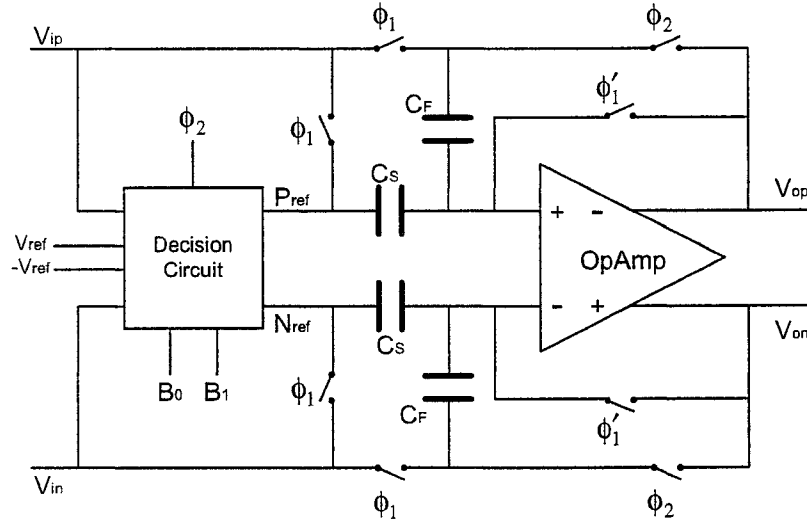


Figure 56 One stage of the ADC.

Nowadays, this configuration of the circuit to implement one stage is the choice for many ADCs. The reason for this is its higher feedback ratio, which relaxes the design of the operational amplifier. This architecture is presented in [1].

The operation of one stage is best described by looking at two consecutive stages. The two stages are configured as shown in Figure 57 and the timing diagram for these stages is shown in the same figure.

Each stage operates in one of two modes: Sampling or Holding. The Sampling mode occurs when ϕ_1 , ϕ_1' , ϕ_1'' clocks are High. This will cause the capacitors to be connected to the input signals: v_{in} and v_{ip} from one plate, while the second plate of every capacitor is connected to the other plates of the other capacitors and to the common mode voltage. This will cause the capacitors to track the differential values of the input signals by storing an equivalent charge based on the value of the capacitors. The Holding mode starts after ϕ_1 goes Low and when ϕ_2 goes High. In this mode the reference voltages are connected to the sampling capacitor, C_s , while the feedback capacitor C_f is connected to the output as shown in Figure 57.

Right before the end of the sampling phase, the comparators take their decision when the 'Comparator Reset/Latch signal goes Low. As the diagram in Figure 57 shows, the output of the comparators should be ready before the current stage, which includes the comparators, changes to the hold mode. Actually, not only the comparators outputs, should be ready by that time, but also the DAC values generated by the switches should

also be ready. In summary, before the hold mode of the current stage starts, the comparators should generate the binary outputs and drive the switches to generate the appropriate DAC values.

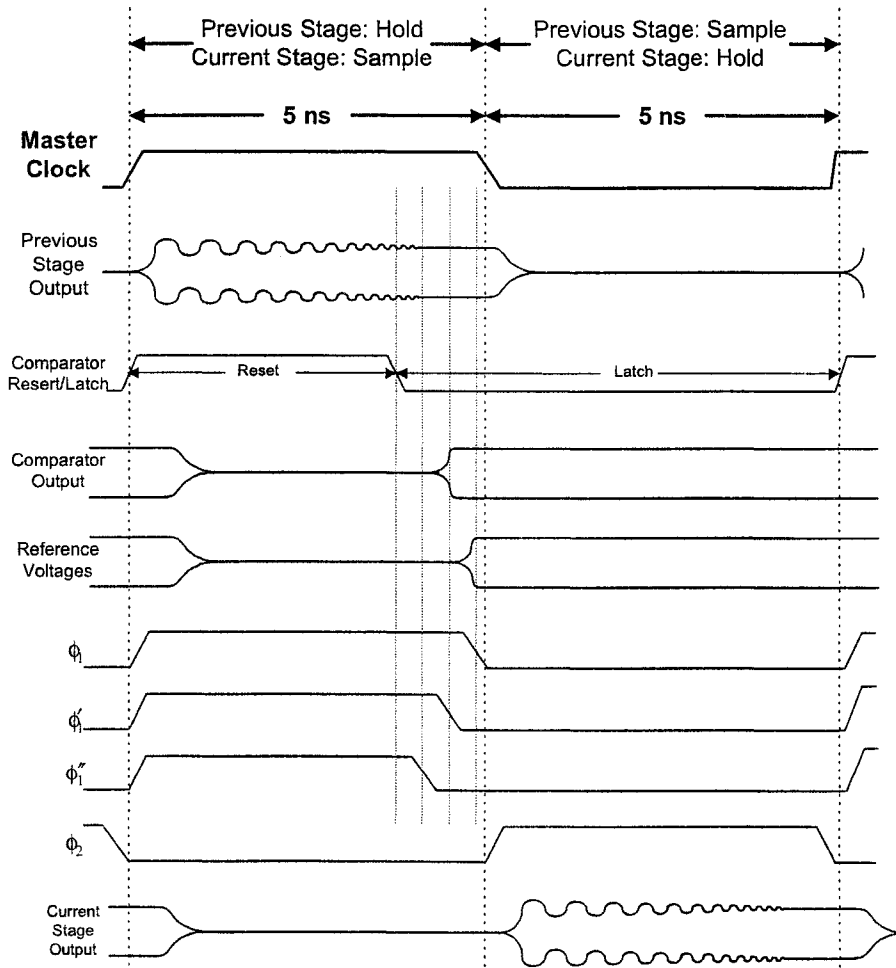
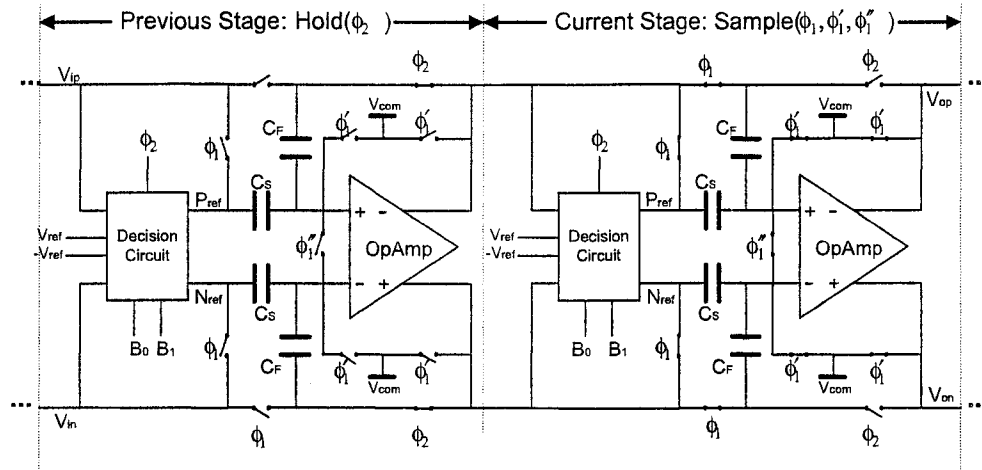


Figure 57 Two stages of the ADC configured in complementary fashion.

An issue might arise at this point. What if the comparators make wrong decision due to not enough settling time? The answer is: No problem. The digital error correction will guarantee that the final digital equivalent value will be the same as illustrated in the example above because this can be viewed as an offset error in the comparators which is tolerable using the redundancy. As long as the amount of error in addition to the offset exists in the comparator is less than the tolerable offset, the algorithm will correct for it.

5.2.6. Design of the operational amplifier

Probably, the most important component in the ADC is the operational amplifier, since it imposes a great effect on both speed and resolution of the ADC. Not only it is the most difficult, but it is also the most power consuming and the source of major errors in the overall system.

Understanding the specifications of the operational amplifier and calculating them is a vital step before starting the design. Specifications such as DC gain, unity gain bandwidth and slewing rate are among many other important ones. The importance of one specification over the others depends heavily on the application in which the opamp is going to be used. For example, for an operational amplifier to be used in an ADC, speed and accuracy may be considered as the most important specifications. The following section describes the analysis of an operational amplifier that is used in a switched capacitor ADC.

Although there are many factors that affect its operation, the accuracy of the operational amplifier is measured mainly by its dc gain. An operational amplifier configured in a closed loop feedback configuration is shown in Figure 58.

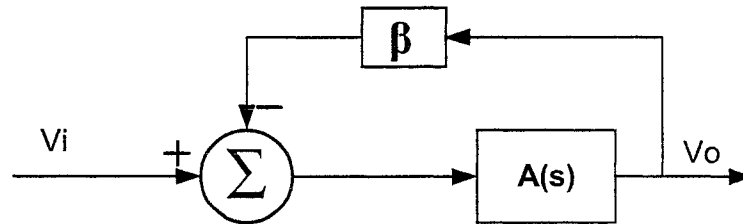


Figure 58 Feedback model of operational amplifier.

The transfer function of the closed loop gain is given by:

$$T(s) = A_{cl}(s) = \frac{A(s)}{1 + \beta \cdot A(s)} \quad (15)$$

Where β is the feedback factor and $A(s)$ is the open loop gain of the opamp.

In the design of this ADC, a single stage fully differential folded cascode was used. The open loop gain of the opamp can be given by:

$$A(s) = \frac{W_u}{s} \quad (16)$$

Where, w_u is the unity gain frequency in radians.

$$T(s) = A_{CL}(s) = \frac{w_u}{s + \beta w_u} = \frac{1}{\beta} \cdot \frac{1}{1 + \left(\frac{s}{\beta w_u} \right)} \quad (17)$$

Which means that the closed-loop opamp has a dc gain, at $s = 0$, equals to $1/\beta$ and it has a -3dB frequency given by:

$$w_{-3\text{dB}} = \frac{1}{\beta w_u} \quad (18)$$

The transfer function $T(s)$ relates the output to the input as:

$$T(s) = A_{CL}(s) = \frac{V_o(s)}{V_i(s)} \quad (19)$$

For a step input, $V_i(s) = V_s/s$, and thus,

$$V_o(s) = \frac{V_s}{s} \cdot \frac{1}{\beta} \cdot \frac{1}{1 + \left(\frac{s}{\beta w_u} \right)} = \frac{V_s}{\beta} \left[\frac{1}{s} - \frac{1}{s + \beta w_u} \right] \quad (20)$$

Where, V_s is the magnitude of the step input.

Taking the inverse Laplace transform to find the time domain response:

$$v_o(t) = \frac{V_s}{\beta} \left(1 - e^{-\frac{t}{\tau}} \right) \quad (21)$$

$$\text{Where, } \tau = \frac{1}{\beta w_u}$$

One can clearly see that for fast opamps, τ needs to be small which requires both large β , or feedback ratio, and large unity gain frequency.

Equation (21) also states that since the settling time is finite, there will be a settling error that is equal to $e^{-t/\tau}$.

For example, if a 1.0% accuracy is required, then one must allow $e^{-t/\tau}$ to reach 0.01, which is achieved at a time of 4.6τ . For settling within a 0.1 percent accuracy, the settling time needed becomes approximately 7τ .

The above analysis assumes that the opamp has an infinite dc gain, which is not the actual case. In practice, however, the opamp has a dc gain, A_0 , at $s = 0$, thus it can be modeled as:

$$A(s) = \frac{A_0}{1 + \frac{s \cdot A_0}{w_u}} \quad (22)$$

Substituting equation (22) in equation (15) will give the closed-loop gain, $A_{CL}(s)$ as:

$$A_{CL}(s) = \frac{1}{\beta'} \cdot \frac{1}{1 + \left(\frac{s}{\beta' w_u} \right)} \quad (23)$$

$$\text{Where, } \beta' = \frac{1}{\frac{1}{A_0} + \beta}$$

So, the error due to finite opamp gain can be approximated by:

$$err = \frac{1}{A_0 \beta} \quad (24)$$

So, for a 1 percent error gain, $err < .01$, or equivalently, for $\beta = 0.5$, the required dc gain, A_0 is > 200 .

Now, lets go back and derive the specifications from the above equations.

When excited with a step input, the opamp goes in two regions. In the first region, the opamp will slew if the output current is not enough to charge the output capacitors in exponential fashion as equation (21) states.

This is usually the case in most of the implementations, since the opamp will be designed with minimum power consumption.

The second region of operation will be the settling region where the opamp will settle to its final output. In a switch capacitor design, the opamp will be given a time to finish both regions of operations. This is usually going to be half a period when its output is valid. Assuming that the total time given to the opamp is t , then

$$t = t_{sl} + t_{ss} \quad (25)$$

Where, t_{sl} is the time needed for the opamp to finish slewing, and t_{ss} is the time needed by the opamp to finish settling.

As a rule of thumb, t_{sl} is usually set to 20% of t , and t_{ss} is set to 80% of t . So, for example, if we want to design an opamp that uses a resetting architecture and runs at 100MHz, then $t = 5ns$, $t_{sl} = 1ns$ and $t_{ss} = 4ns$. This means that the opamp should be able to finish slewing in 1ns and finish settling in 4ns.

If we assume we want a settling error to be 0.1 percent, then $7\tau = 4ns$ or equivalently, $\tau = 0.57143ns$. Using equation (22) with $\beta = 0.5$, $w_u = 3.5\text{Grad/s}$ which is approximately 560MHz.

Now let us consider a practical example. Suppose we want to design a 10 bit pipeline ADC that runs at 100MHz. The first thing to decide is how many bits each stage should resolve. If the 100MHz speed is too tough to achieve in a certain process, like CMOS, then we need to consider low number of bits per stage, since the higher the resolved bits in each stage the higher the required gain of that stage will be, which means the lower the feedback ratio. This will result in τ being large. The minimum number of bits per stage is 1, and let's say that we decided that we want 1 bit per stage ADC.

The second step is to determine β . This depends on the configuration of each stage that will achieve the required gain. For a 1-bit per stage, the required gain is 2. There are some configurations that achieve this gain with $\beta = 0.5$ and some others achieve it with $\beta = 0.33$. Clearly, for higher speed, we want to consider the one with $\beta = 0.5$. So, now, we have $\beta = 0.5$ determined. The next step is to find the required unity gain of the operational amplifier and its dc gain. The unity gain requirement can be derived from the speed of the ADC, which is 100MHz. As derived above, the opamp needs to have at least a unity gain of 560MHz.

There are many sources of errors in the ADC. For simplicity, let's consider that the finite dc gain of the opamp and not enough settling time are the only sources of errors. In general, all the sources of errors should contribute to less than half an LSB. Since we have only two sources, each should contribute at most one quarter of LSB. This, in turn, means that the dc gain of the opamp should be accurate to more than 12 bits for our 10 bit ADC and so as the settling error of the opamp. So, the required accuracy of the finite gain as well as the settling time should be less than $1/2^{12}$, which is 0.0244 percent.

To find the required unity gain of the opamp, the 4ns should be equal to 8.4τ , making τ equal to 0.48ns. Using equation (21), the unity gain frequency, $f_u = 670\text{MHz}$.

The dc gain of the operational amplifier is found using equation (24), where $err = 0.0244$ percent. Thus, A_0 should be at least 8192.

The operational amplifier is a standard fully differential single stage folded cascode with boosting amplifiers as shown in Figure 59. The operation of the boosting amplifiers is best described in [2]. The boosting amplifiers, shown in Figure 59 as BN and BP, are also fully differential folded cascode opamps. The operational amplifier that uses the boosting amplifiers is called the main opamp, while the boosting amplifiers are always referred to as boosting amplifiers. Single stage design has been considered to give better frequency response in addition to the fact that it is more stable over temperature, process and power supply variations. Although it has a worse frequency response than the regular differential opamp, the folded cascode boosting opamp was chosen over the regular differential opamp because it can be designed with higher gain. The two different boosting amplifiers were used instead of single ended design because they give higher gain, in addition to the fact that they are more area efficient since we need only two of them instead of four in the case of single ended design. The boosting amplifiers are of two types: the BN has an NMOS differential input stage, while the BP has a PMOS differential input stage. As shown in Figure 59, the inputs of the BN boosting amplifier comes from the drains of M10 and M11 transistors, which are supposed to be biased in the saturation region and have a drain-to-source voltage, $V_{ds} < -0.5\text{V}$. This means that the inputs to the differential pair of BN are going to be around 2V, hence an NMOS differential input stage is required. The bottom boosting amplifier, BP, has its inputs coming from the drains of M4 and M5 which are supposed to be biased at $V_{ds} < 0.5\text{V}$, hence a PMOS differential input stage is required.

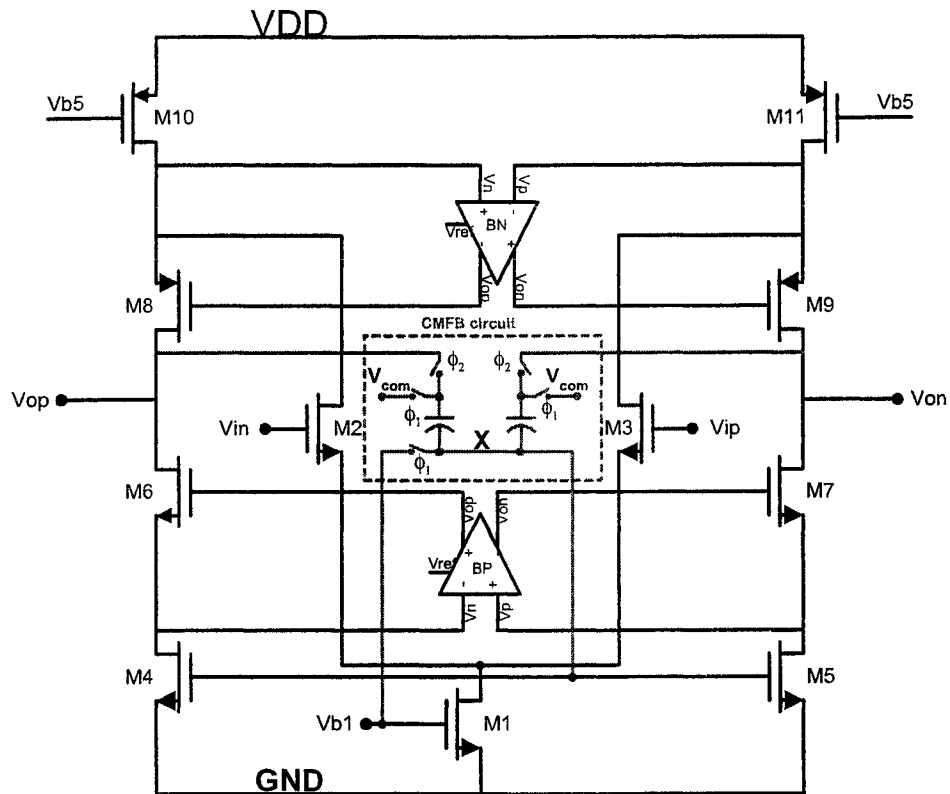


Figure 59 Main operational amplifier with boosting opamps and CMFB circuit.

The NMOS type boosting amplifier, BN, with its continuous time common mode feedback circuit, CMFB, is shown in Figure 60. It is very similar to the main opamp with the exception that it doesn't have boosting amplifiers and that the tail current source that consists of M1 and M1x transistors is cascoded so as to increase the source voltage of transistors M2 and M3. This is to decrease the excess bias voltage of those transistors in order to guarantee that they are in the saturation region of operation when they have a common mode voltage input applied to their gates. The CMFB circuit consists of all transistors Mc1-Mc9. The main function of the CMFB circuit is to set the common mode voltage of the output nodes, Vop and Von, to the biasing voltage of transistors M8 and M9.

5.2.6.1. CMFB circuit design of the main amplifier

The common mode voltage of the opamp can be controlled by many transistors. Using one side of the opamp, any one of transistors M1, M4 and M10 can be used to control the common mode voltage of the opamp. In this design, M4 was chosen. The relationship between the voltage at the gate of M4 and the common mode voltage is inverted. As the voltage at the gate of M4 increases, the common mode voltage drops and vice versa.

To maximize the output swing of the operational amplifier, a switched capacitor CMFB circuit is utilized to keep the common mode output voltage at the required level. The CMFB circuit is shown in red in Figure 59 and consists of 2 capacitors and couple of switches. The two capacitors have the same value which should be chosen such that it is not too large to load the main opamp or too small to be affected by the charge injection of the switches. The sizes of the switches should also be chosen carefully so that they won't have great effect on the capacitors. The operation of the CMFB circuit is as follows. The CMFB circuit works in two phases. In the sample phase of the opamp, the output of the opamp are disconnected from the CMFB circuit and V_{com} is connected instead, while, when being in the hold mode, the capacitors are disconnected from V_{com} and, then, connected to the output of the main opamp. V_{com} represents the required common mode voltage of the operational amplifier and it is set in this design to $1.25V$. The second side of the capacitors are connected to the biasing voltage of the transistors used at nominal conditions. This will be illustrated soon.

At nominal conditions and without the CMFB being connected to the opamp, transistors M4 and M5 are designed to be biased with V_{bj} . With V_{bj} biasing M1, M4 and M5, the common mode of the output voltage of the main opamp is around V_{com} .

The two capacitors average the output of the opamp, with node X being set by V_{bj} in the sampling phase. If the common mode voltage of the output of the main amplifier comes to be similar to one set by design, then node X in the hold mode will also be similar to V_{bj} . If the common mode voltage of the outputs increases, the voltage at node X will increase to more than V_{bj} , which will increase the biasing voltage of the gates of M4 and M5 and thus, decreasing the common mode voltage of the output. If the common mode voltage of the outputs of the main opamp is less than that set by design, the voltage at node X will drop to below V_{bj} and thus increasing the common mode voltage of the outputs of the main opamp, and thus, the output of the main opamp will be kept close to the voltage set by design.

5.2.6.2. CMFB circuit design of the boosting amplifiers

Designing the CMFB circuit for the boosting amplifiers is a straightforward process. The output of the each boosting amplifier doesn't need to swing too much, thus, a continuous time CMFB circuit can be used. The first step is to design the boosting amplifier without the CMFB circuit such that the common mode output of the opamp is around $V_{dd}/2$. Once this is finished, part of the output current is generated by the CMFB circuit using transistors Mc8 and Mc9. For example, suppose that after designing the opamp without the CMFB circuit, the W/L ratio of M4 and M5 comes to be 4. If we assume that one quarter of the output current will be provided by the CMFB circuit, then W/L of both M4 and M5 will be reduced to 3. With the CMFB circuit being not connected, half of the current in M11 will be in M5, so if Mc1 is made $1/4^{\text{th}}$ of M11, then $1/4^{\text{th}}$ of the current in M11 will be in Mc1. If V_{ref} equals the common mode voltage of V_{on} and V_{op} , then Mc2-Mc7 are designed such that its current in Mc4 is the same as the current through both of Mc2 and Mc3 together. This means that the current through Mc4 is $1/2$ of the current in Mc1, or equivalently, the current in Mc4 is $1/8^{\text{th}}$ of that of M11.

Since the current in the path of M9 and M7 is $\frac{1}{2}$ of M11, then the current of Mc4 is $\frac{1}{4}$ th of that in M9 or M7. So, $\frac{1}{4}$ th of the current of M9 or M7 will be provided by the CMFB circuit, and the rest is provided by M5 which will be $\frac{3}{4}$ th of the current. This is why the W/L ratio of M5 was reduced from 4 to 3 to represent the $\frac{3}{4}$ portion of the current.

V_{ref} in Figure 60 is set externally to the biasing voltage of M8 and M9 of the main opamp. This gives us the opportunity to test the main amplifier with or without the boosting circuits, since this voltage will be fed to the V_{ref} or to the transistors directly.

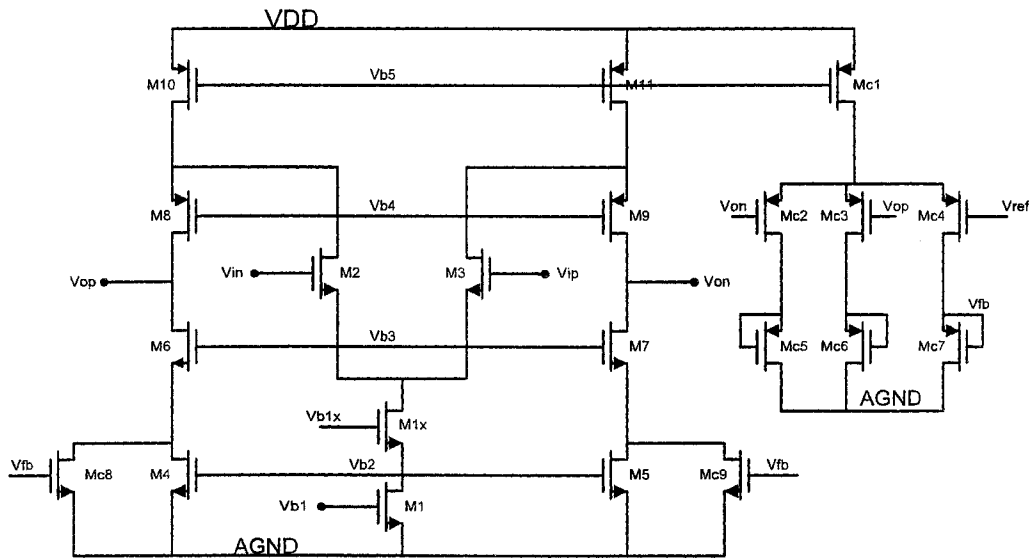


Figure 60 Boosting amplifier with NMOS differential input stage.

The BP boosting amplifier is the same as the NMOS type with the exception that a PMOS differential input stage is used in addition to an NMOS CMFB circuit instead of the PMOS one used above.

5.2.7. Comparator implementation

For high speed comparator design, regeneration of the output should be used [2].

The comparator circuit that implements regeneration is shown in Figure 61. This comparator consists of two-stage preamp in order to decrease the minimum resolvable signal by increasing the gain of the preamp, which will increase the resolution of the overall comparator. The differential amplifier in the dashed box provides the difference circuit that amplifies the difference between V_{ip} and V_{rp} and also V_{rn} and V_{in} . Since this is a differential amplifier, then if V_{ip} is greater than V_{rp} , this should also guarantee that V_{rn} is greater than V_{in} . If V_{ip} is greater than V_{rp} , then Node B is at higher voltage than node A. This is because, if V_{ip} is higher than V_{rp} , the current in M3 transistor is larger than the current in M4 transistor due to the larger excess bias voltage on M3 than it is on M4. Those two currents in M3 and M4 will pass through the load; M0 and M1 respectively. Larger current in M0 than current in M1 means larger voltage drop on M0 than M1, which, in turn, means that the

voltage at node A is less than the voltage at node B. Same thing applies to the bottom differential opamp in the box. If V_{in} is smaller than V_{rn} , Node B will be pushed further up and node A will be pushed further down, and hence this differential configuration will enhance both the speed by helping the upper differential amp to push the node voltages up and down, and the resolution by increasing the dynamic input range by two.

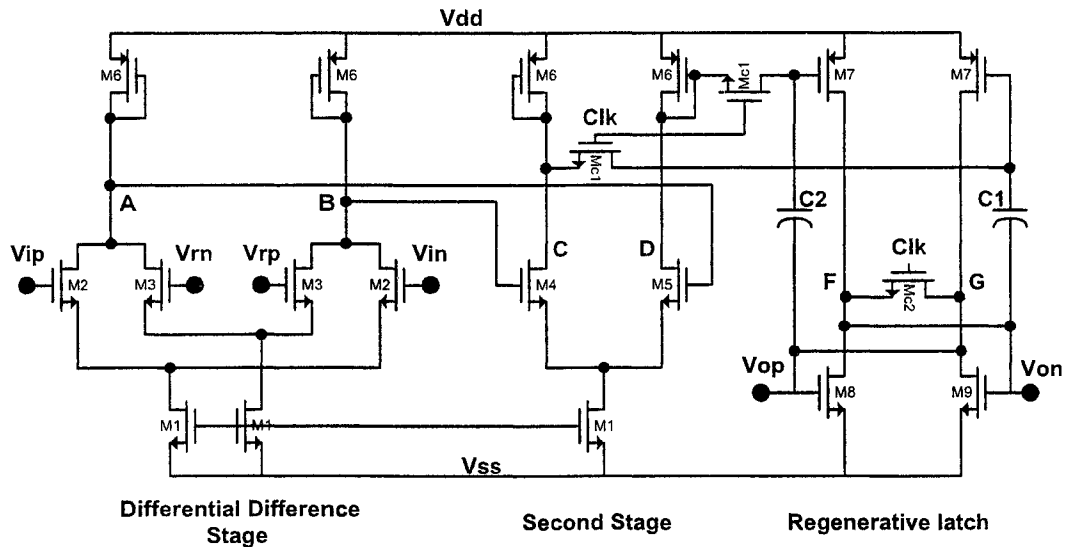


Figure 61 Static comparator with latch.

Same analysis applies to the second stage of the preamp. If node B is higher than Node A, this will cause the voltage at node C to be higher than that at node D. When the Clk is High, the bottom plates of the two boosting capacitors, C1 and C2, are connected together while the upper plates are connected to Nodes F and G, which are also connected to nodes C and D respectively. This due to the fact that the two transistors; M19 and M20, makes a short circuit and the regenerative latch is disabled.

So, when Clk is High, and V_{ip} is higher than V_{rp} , B will be higher than A, C will be higher than D, and so, the top plate of C1 will be at higher voltage than the top plate of C2.

When the clock is turned off, M19 and M20 transistors turn off and disconnect nodes C and D from nodes F and G respectively, while transistor M18 enables the regenerative latch. Since the top plate of C1 is at higher voltage than that of C2, then the excess bias of M12 transistor is larger than that of M13, which means that larger current will be going into M15 than that of M14. Since M14 and M15 behave as the loads of M13 and M12, respectively, V_{ds} of M15 will be larger than that of M14, and since the latch is in a positive feedback, that will push V_{ds} of M15 further to increase, while V_{ds} of M14 to decrease. V_{ds} of M15 will rail to V_{dd} while V_{ds} of M14 will rail to V_{ss} . One important issue in the design of this comparator that will affect the speed of the comparator is the size of the boosting capacitors. The size of the capacitors is chosen such that its kT/C effect is less than the accuracy required to be provided by the comparator, so, it should be larger than C_{min} , where C_{min} is determined from the kT/C requirement. The upper limit of the boosting capacitor, M_{ax} is determined from the

speed of the comparator. The second stage of the preamp will source current to or sink current from any of the boosting caps. When the Clk switches from Low to High, and stays in the High period, the current sourced to or sinked from the caps should be able to reach its steady state before the Clk changes to Low, otherwise, the comparator might make wrong decision. So, the maximum output current of the second stage of the preamp as well as the time in which the voltage at the top plates of the capacitors will settle will determine the maximum size of the capacitors. Larger capacitor means it needs more time for the second stage of the preamp to be able to charge it, which means slower operation of the comparator.

One more thing regarding the operation of the comparator. The comparator should follow the output of the opamp of its stage. But, it should shut off just before the opamp does so, or putting it differently, it should shut off right when the next stage starts the holding mode. This results in that the clock of the comparator should follow Phi2 of the stage where the comparator belongs, but it should shut off a little earlier than Phi2. So, the clock generator will generate another signal like Phi2, but it shuts off earlier than Phi2.

5.2.8. Capacitor design in TSMC 0.25u Process

The TSMC process is a digital process that doesn't include a high precision capacitor. We implemented the capacitors in our design by using the four layers of metal; Metal2, Metal3, Metal4 and Metal5, as a sandwich capacitor. We depend on the parasitic capacitance between each two layers to make our capacitors. A capacitor has Metal2 and Metal4 connected with each other, while Metal3 and Metal5 are also connected with each other to make the second plate of the capacitor.

Especial layout techniques have been taken into consideration to increase the matching of the capacitors. The input/output relationship of the operational amplifier circuit in each stage is given by:

$$v_{od} = \left(1 + \frac{C_s}{C_f} \right) v_{id} - \frac{C_s}{C_f} v_x \quad (26)$$

Each stage of the ADC has a gain of 2. The gain of each stage is represented by the first bracketed term in equation (26), which shows that C_s/C_f should have a value of 1. In order to get the required accuracy of a specific stage, C_s should be matched with C_f to the accuracy of the ADC or better. So, from gain accuracy stand point, absolute values of C_s and C_f are not as important as matching the capacitors with each others, which means that the two capacitors of every stage have to be matched to the accuracy of that stage or better.

To achieve this, especial layout techniques were followed such as common centroid and interdigitization. These two techniques were used together as shown in Figure 63, where one of the capacitors is called A, while the other is called B. Each capacitor is divided into eight smaller ones so that they can be interdigitized with those of the second capacitor. This procedure was followed in case there is a horizontal, vertical, or diagonal gradient in the process, the effect will be reduced.

5.3. Layout

5.3.1. Operational amplifier layout

The simulation shows that the amplifier is slewing in $\sim 1.5\text{ns}$ which is very close to the calculated one. However, it was very clear that the speed is limited by the GBW of the opamp. In order to reduce the external parasitics of the operational amplifier and to reduce other sources of errors such as opamp offset, special attention should be paid to the layout.

The common centroid techniques were used wherever it was possible in order to have better matching, and metal overlapping was avoided as much as it could be in order to reduce the parasitics. For example, every two corresponding transistors in the main opamp have been laid out as common centroid so that they are matched together. Figure 62 shows the layout of the two transistors: M10 and M11 in the main opamp shown in Figure 59. Since the number of transistors doesn't completely agree with the common centroid requirement, dummy transistors were added as shown in the same figure.

The sampling as well as the integrating capacitors are also laid out in common centroid fashion to increase the matching. Special attention was paid to the overlapping of the metal wires connecting the layers that make the capacitors so that they contribute of equivalent parasitic capacitance. The reason for this is that overlapping was not completely avoided.

Figure 63 shows the way the capacitors are laid out. This layout is less sensitive to the gradient effects of the process in the horizontal direction, vertical direction or diagonal direction.

The complete layout of the operational amplifier is shown in Figure 64.

5.3.2. Stage layout

Each stage consists of a one opamp, comparators clock generator, CMFB and some switches. The layout of each stage was constructed such that the analog part that is made of the opamp, comparators and CMFB circuit is separated from the digital part that is made of the clock generator circuit. This is shown in Figure 62. As mentioned above, especial attention was made to avoid any overlapping of wires as much as possible.

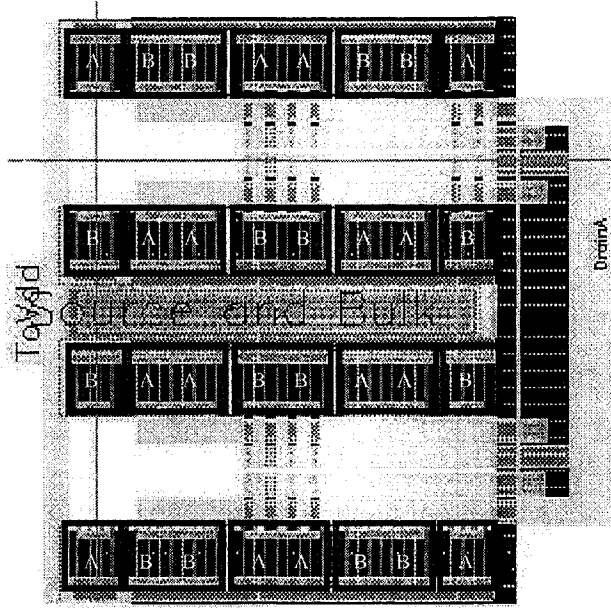


Figure 62 PMOS transistor laid out in CC fashion.

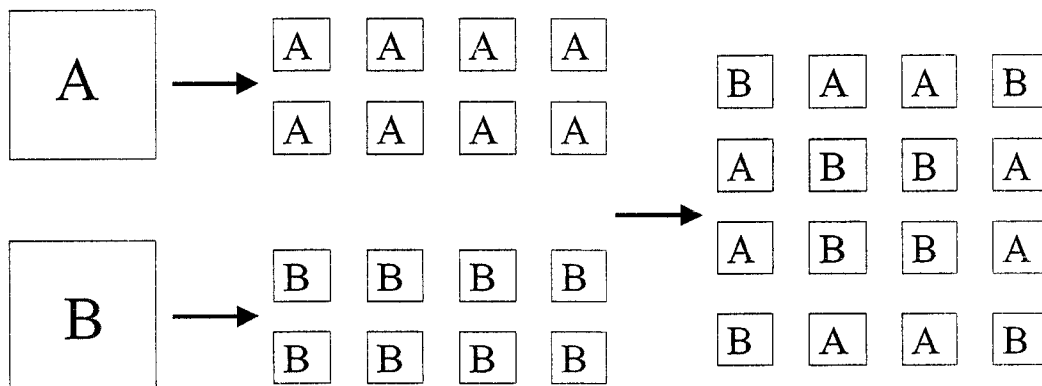


Figure 63 Common Centroid layout.

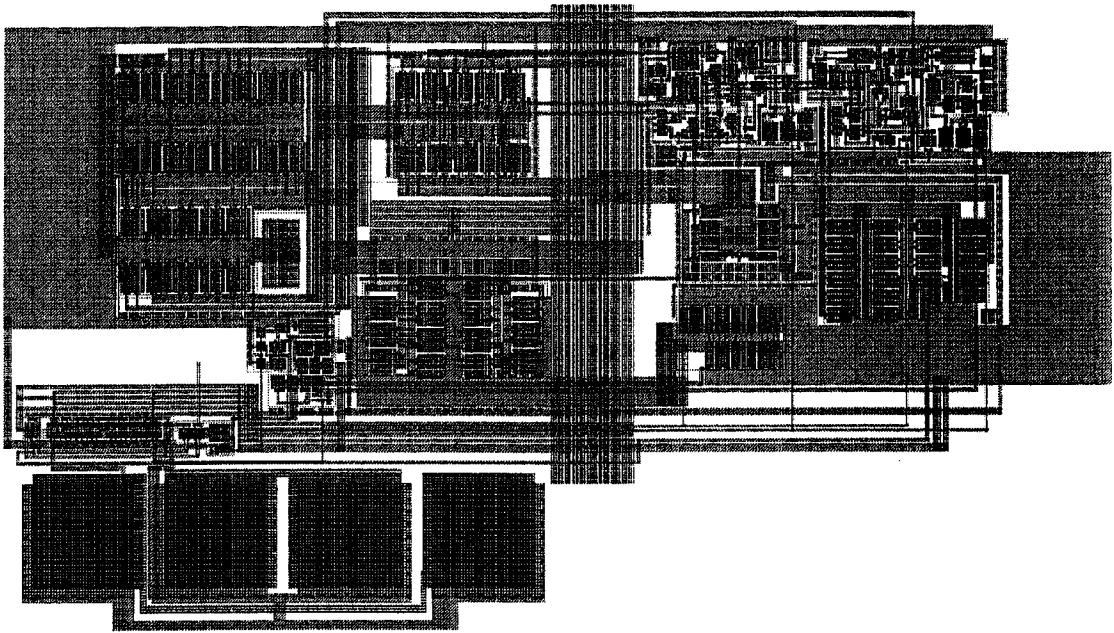


Figure 64 layout of the operational amplifier.

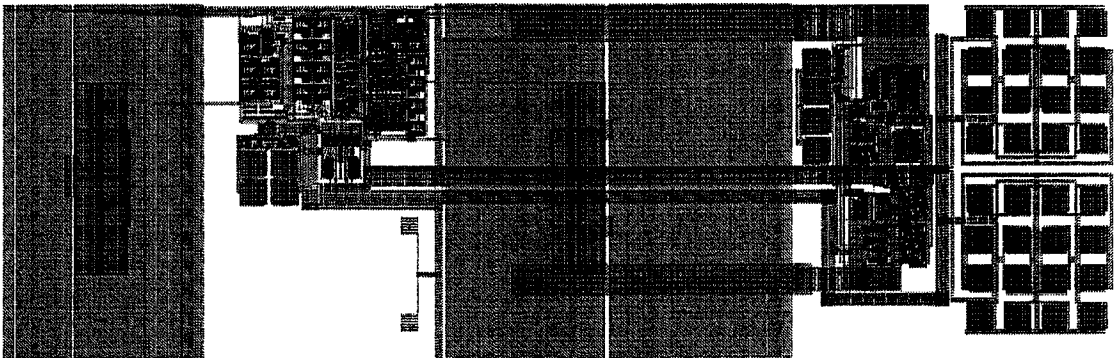


Figure 65 layout of a single stage without error correction circuit.

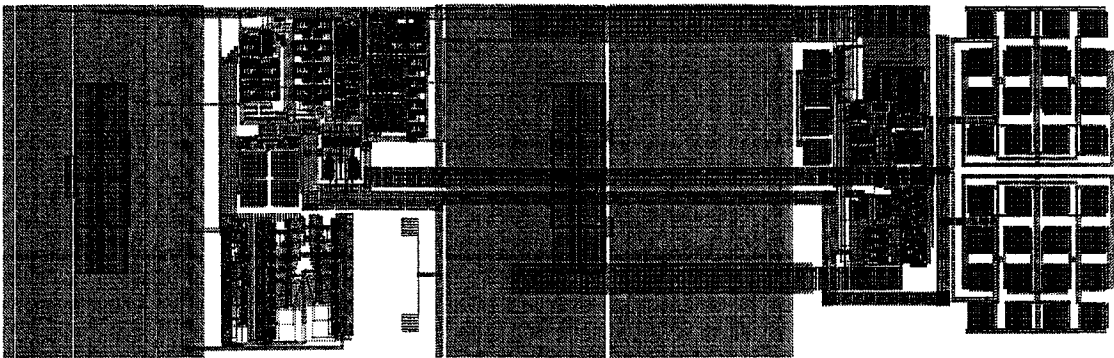


Figure 66 layout of a single stage with the error correction circuit.

5.3.3. Overall Layout.

The ADC has been carefully laid out in order to enhance matching, reduce parasitic and reduce the coupling between the digital and the analog parts.

To enhance the matching, symmetry of differential signal paths was followed wherever possible. For example, instead of connecting two differential signals as shown in Figure 67.a), the actual layout was done as shown in Figure 67.b).

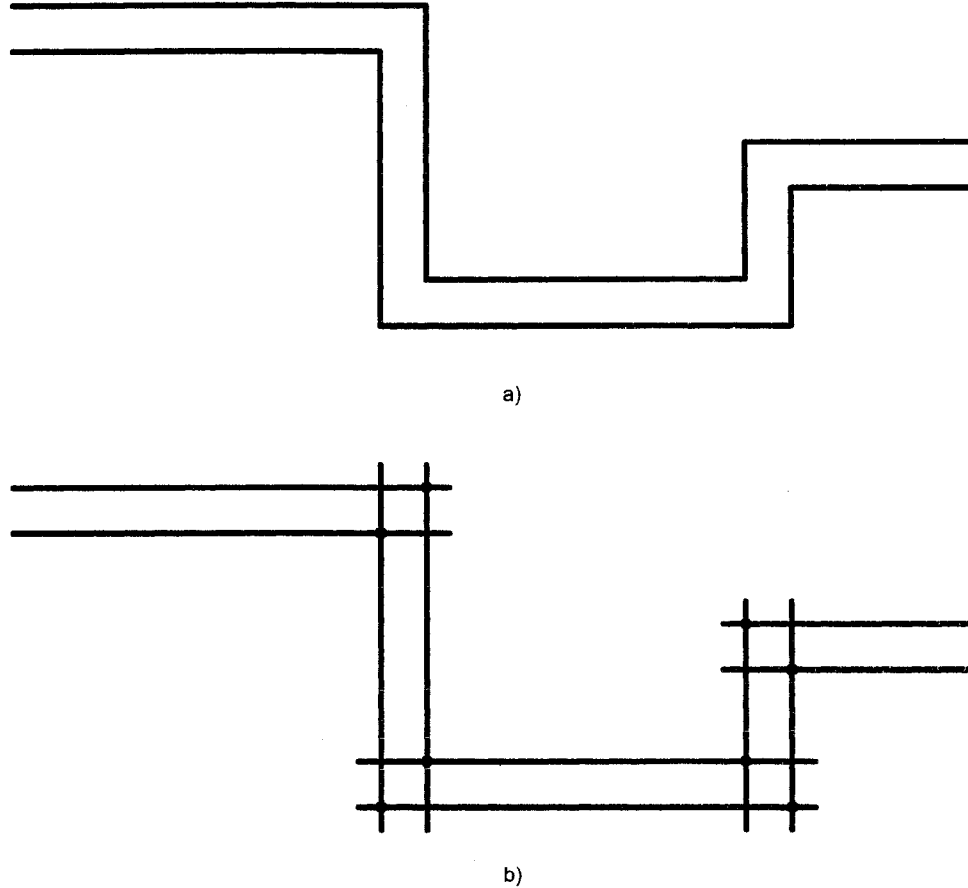


Figure 67 Layout matching technique.

Parasitics have been reduced in the layout by making sure of using common centroid as well as digitization techniques whenever possible. This is particularly important in this design because there are many big transistors especially those in the main operational amplifier that need to be matched with each other. The capacitors of each stage are also laid out in a common centroid fashion to enhance their matching. In particular, the first stage of the ADC which is a single-ended-to-differential (STD) circuit has 3 capacitors that needed to be matched with each other. This new technique is illustrated in Figure 68 where the three capacitors are A, B and C. Each one is divided into 12 unit capacitors. These unit capacitors are distributed as shown in Figure 68

and Figure 69 to enhance the matching. This way, the parasitics will be common to the three capacitors. The overall layout of the STD circuit is shown in Figure 70.

Critical differential signals are also separated from each other by ground lines in order to reduce the mutual coupling between them.

In order to decouple the analog portion from the digital ones, the power supplies of the two parts are separated from each other by a distance of more than 100 μ m as shown in Figure 71. This is believed to be the best way to keep the digital noise away from affecting the analog circuits. Additional decoupling capacitors are also added underneath the power supply rails.

There were 2 chips that have been fabricated; The first one is the ADC only, while in the second one digital correction circuits were added to the first 4 stages.

The two layouts are shown in Figure 72 and Figure 73.

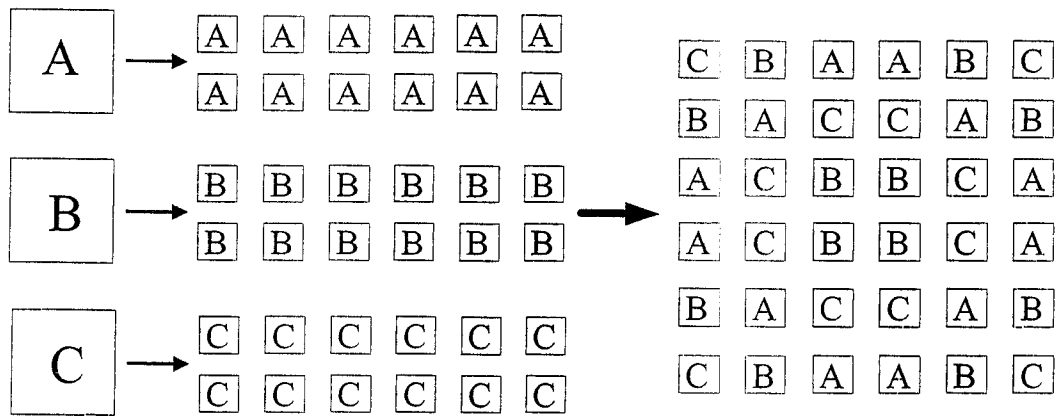


Figure 68 Common-centroid layout for 3 capacitors.

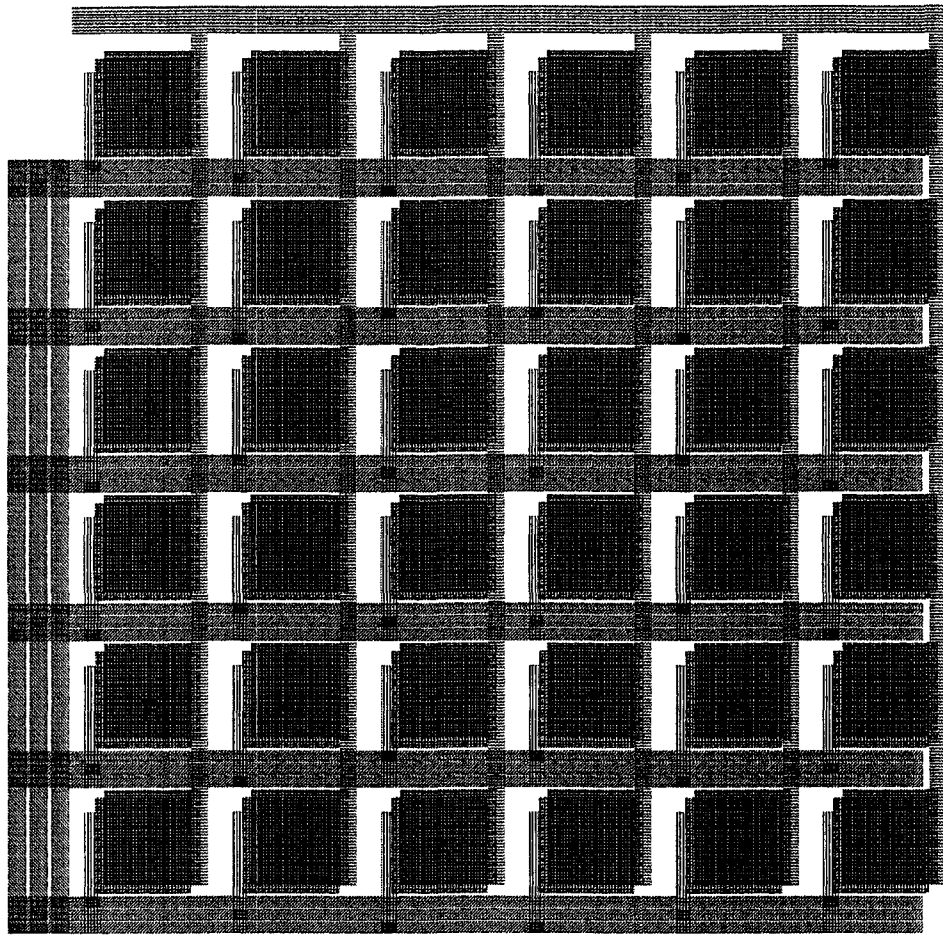


Figure 69 layout of 3 capacitors as common centroid.

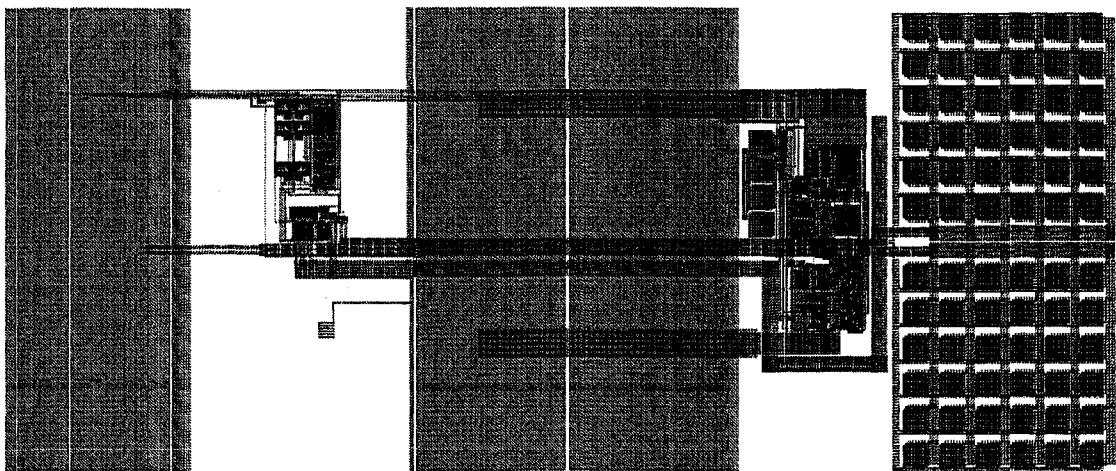


Figure 70 Single-ended-to-differential (STD) stage.

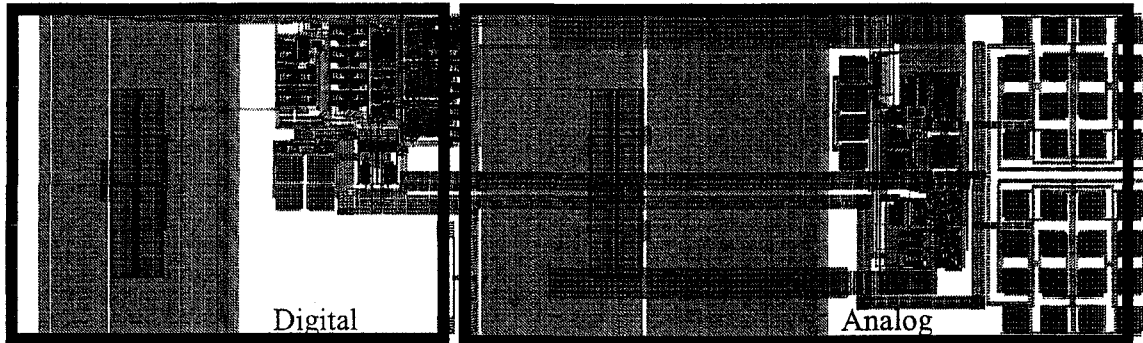


Figure 71 Layout of the MSB stage in the ADC. This figure shows the separation between the analog circuits and the digital ones.

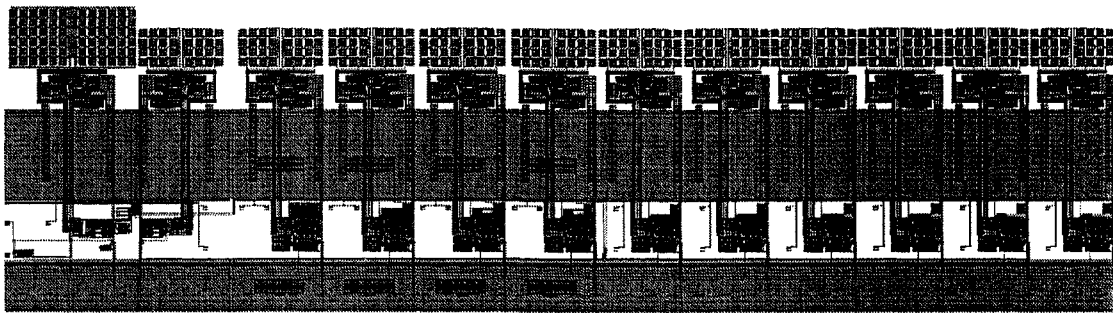


Figure 72 ADC top level layout without correction circuit.

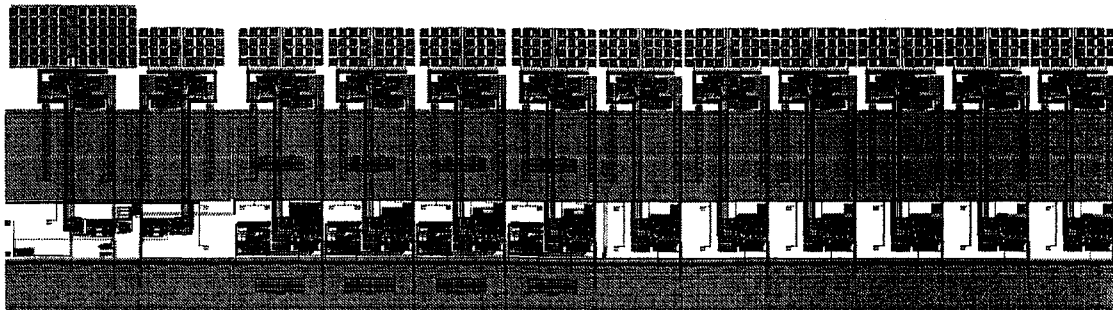


Figure 73 ADC top level layout with correction circuit.

5.4. Testing Results

The ADC has been fabricated and tested using the board whose layout is shown in Figure 74 and a photo of the board is shown in. The board has been designed using the EAGLE software to draw its schematic and then generate the layout. The test setup for the ADC is shown in Figure 75. A logic analyzer is needed to gather the digital data coming out of the ADC for further analysis. As shown in the figure, the ADC generates 18 binary outputs, which are the binary outputs coming out from the 9 stages of the ADC.

The digital portion of the ADC was functioning correctly. This portion generates the clocks that are provided to the different ADC stages.

The analog portion of the ADC saturates and doesn't respond to the input.

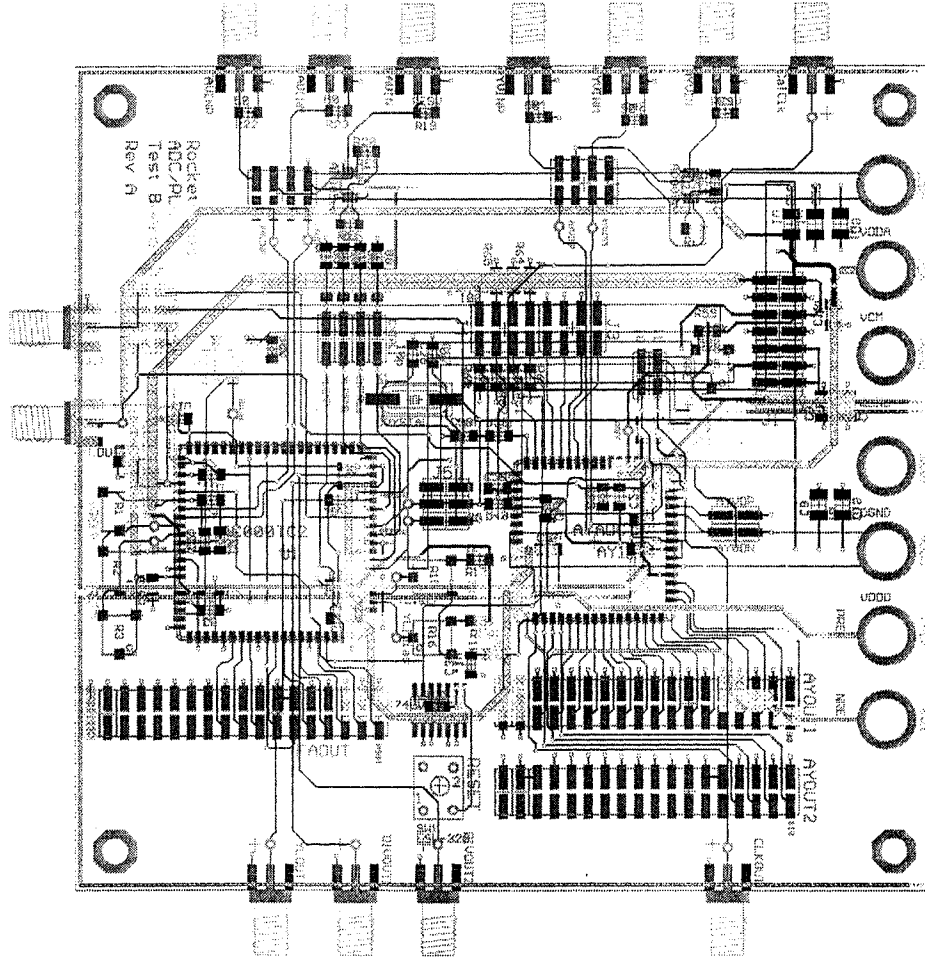


Figure 74 Board layout for chip testing.

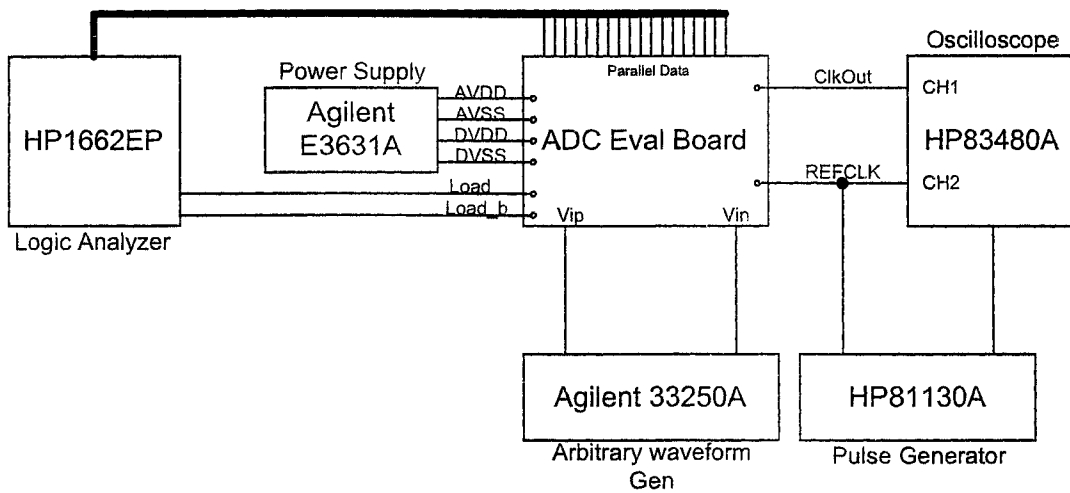


Figure 75 Test setup for the ADC.

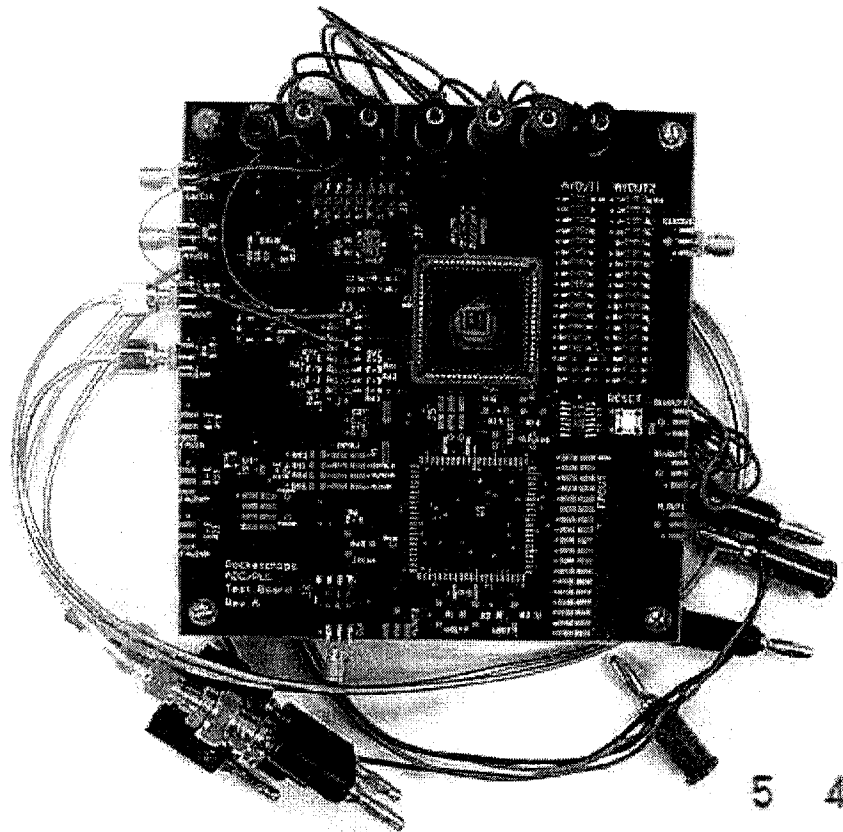


Figure 76 Board photo.

Extensive simulations were carried out under different conditions and found out that the ADC is robust enough to work under all conditions. Figure 77 and Figure 78 show the simulation out of the ADC under the ss corner with 125° temperature and the outputs of the boosting amplifiers initialized to $0.0V$. Each one of the figures show the input of the ADC being swept from $-V_{ref}$ to $+V_{ref}$, the output of the first stage, the output of the second stage and the output of the third stage change according to equation (11). This simulation shows that even when the operational amplifier was saturating because of the initial condition on the boosting amplifiers, it recovered its status and works fine after that. This suggests that the problem might be process related.

Probing measurements suggest that the causes of this failure are:

- The biasing circuit was not able to provide enough current to the operational amplifier because simple current mirrors were used. The solution to this problem is to use cascaded current mirrors whenever possible because they have more output impedance than the regular ones, although it must be noted that this on its own does not explain the analog failure. Bandgap voltage reference should also be built on chip so that the biasing voltages and current can be designed to track voltage, temperature and process variations better than the current design.

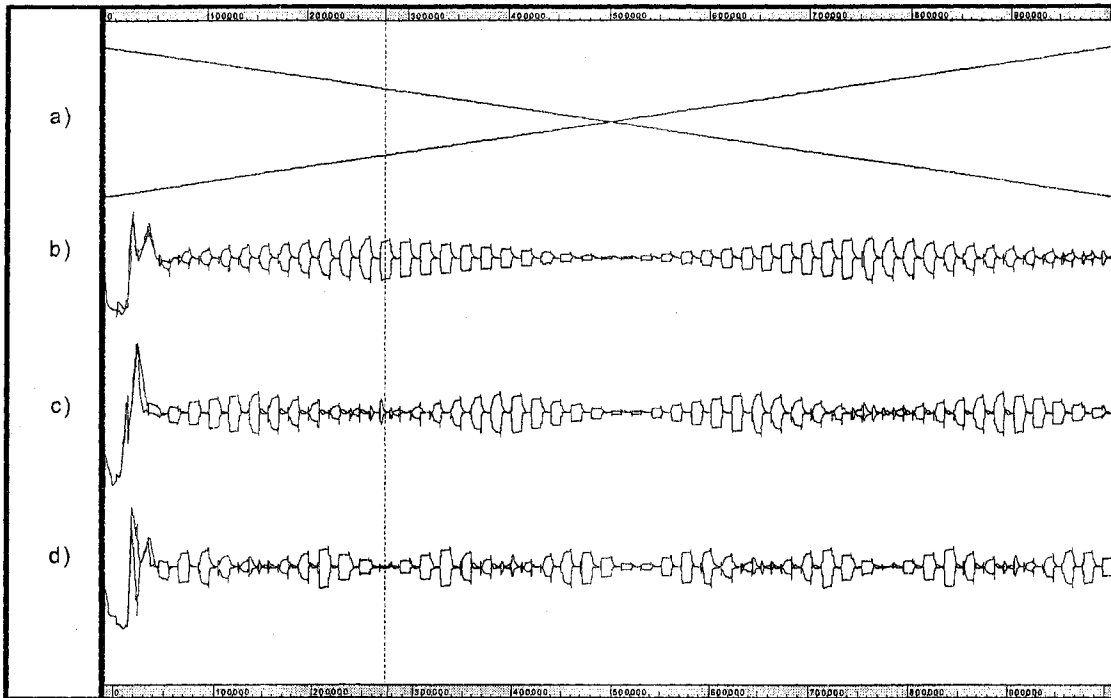


Figure 77 Simulation of the ADC. a) The input of the ADC. b) The output of the first stage of the ADC. c) The output of the second stage of the ADC. d) The output of the third stage of the ADC.

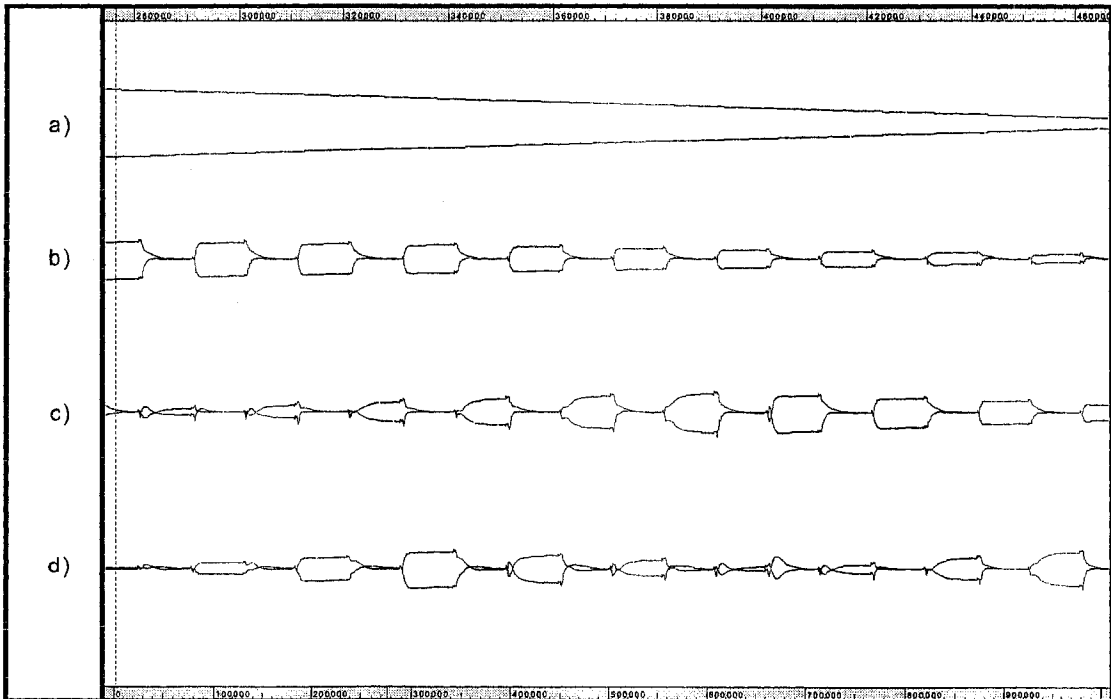


Figure 78 A zoomed in simulation of Figure 77. a) The input of the ADC. b) The output of the first stage of the ADC. c) The output of the second stage of the ADC. d) The output of the third stage of the ADC.

- Switches that control the configuration of each stage were not functioning properly. This caused the inputs to the operational amplifier to have undetermined value that caused it to saturate.

Consultation with experienced designers suggests that each switch must have guarding around it for proper operation.

5.5. Conclusions

Although some of the design techniques followed in the implementation of a 10-b, 100MS/s ADC have been presented in the previous chapter, this chapter presents the steps followed in the implementation in more details.

In this chapter, the operation of one ADC stage, the interaction between two consecutive stages and the simulations of operation of the entire ADC have been presented. The operation of the subADC, the mapping of the digital code coming out of the comparators to the correct one and the operation of the comparators used in this design have also been presented.

It has shown that the 1.5-bit per stage architecture is tolerable to offset errors in the comparators as long as they are less than $\frac{1}{4} V_{ref}$.

A fully differential folded cascode operational amplifier was used in every stage. Fully differential folded cascode operational amplifiers have also been used to boost the dc gain of the main operational amplifier without sacrificing the speed. As in most fully differential circuits, a common-mode feedback circuit was used in the main opamp as well as in the boosting amplifiers to define the common mode value of the outputs. A switched-capacitor type common-mode feedback circuit was used for the main operational amplifier since this type of CMFB circuits has less impact on the output swing. For the boosting operational amplifier, a continuous time CMFB circuit was used, because the output of those amplifier doesn't have large swing.

To enhance the performance of the ADC, special layout techniques were followed. In particular, common centroid techniques for both capacitors and transistors were used wherever possible. Matching and shielding of signal routing were also used. Analog circuits were separated from the digital circuits by more than 100u to prevent the digital noise from reaching the analog circuits. The overall ADC has been laid out in a straight line to reduce the effect of the process gradient on the performance.

Decoupling capacitors underneath the power buses were used to minimize the noise on the power supply. The power traces have been sized according to the maximum current going through them in order to avoid electromigration issues.

The last section of this chapter presented simulation results as well as the layout of the board used to test the chip.

References

- [1] S. H. Lewis, H. S. Fetterman, G. F. Gross, Jr., R. Ramachandran, and T. R. Viswanathan, "A 10-b 20-Msample/s analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol. 27, pp. 351-358, Mar. 1992.
- [2] K. Bult and G. Geelen, "A fast-settling CMOS op amp for SC circuits with 90-dB DC gain," *IEEE J. Solid-State Circuits*, VOL. 25, NO. 6, DECEMBER 1990, pp. 1379-1384.
- [3] J.-T. Wu and B. Wooley, "A 100 MHz pipelined CMOS Comparator," *IEEE J. Solid-State Circuits*, VOL. 23, NO. 6, pp. 1379-1385, DECEMBER, 1988.

CHAPTER 6. ADC Error Correction And Calibration

6.1. Introduction

The accuracy and resolution of any ADC is usually limited to a certain number of bits. This means that the equivalent value of the digital output of the an M -bit ADC can be within $\pm\frac{1}{2}$ LSB from the actual analog input, where the value of the LSB is equal to the input range divided by 2^M . This limitation is usually imposed by many parameters.

For example, in a stage of a switched capacitor pipeline ADC, the gain-of-2 is usually implemented using a capacitor ratio. The value of this capacitor ratio is exactly 1.0 or 2.0, depending on the implementation. However, the actual ADC will not have the exact value. This will cause an error in the ADC that will reduce its accuracy and resolution.

A capacitor ratio is usually implemented using 2 capacitors. If the ratio is needed to be exactly 1.0, the two capacitors must exactly be the same. Due to limitations in the process in which the ADC is designed, and the way the two capacitors are laid out, the exact values of the two capacitors will not be the same, which will result in a ratio that is not exactly 1.0.

Generally speaking, most of the errors in an ADC come from errors in the components that are used to make up the ADC. To reduce the effects of the errors in the components of the ADC, the designer has one of two options:

- a. Choose a good process that has good precision characteristics and speed. For example, all of the very high-performance ADCs that have frequency of operation in the GHz range are designed in GaAs or SiGe process. Such kinds of ADCs are used in oscilloscopes, digitizers and high performance tools. However, such kinds of ADCs consume huge amount of power.
- b. Use CMOS process with a mechanism to correct for those errors. This mechanism can be implemented within the ADC such that the sensitivity of the ADC to component errors is minimized or minimized as long as the errors are guaranteed not to exceed certain values. This is usually referred to as *error correction* or *error calibration*.

In this chapter, The second option will be discussed. Error correction and error calibration will be used interchangeably throughout this chapter.

Error correction techniques can be divided into two categories: based on hardware and operation. Based on hardware, error correction can be either analog or digital, while based on operation it might be either background (sometimes called continuous) or foreground calibration.

6.2. Analog versus digital calibration

Analog calibration refers to the calibration when analog hardware is added to the ADC so that some components of the ADC are calibrated either directly by changing their values according to a specific algorithm or changing their configuration to reduce the effect of their errors. Such kind of components might include capacitors or DAC values. One way to implement this type of calibration is using trimming methods. Another way of doing this is to use capacitor-averaging [4] where the effect of capacitor mismatch is reduced by adding another opamp and clock phase. Sometimes, analog calibration includes the measurement of the individual errors, storing them in memory during the calibration cycle, and reading them from memory to estimate residue errors during normal conversion [5][6]. Therefore, in the analog calibration the errors are subtracted in the analog domain using a calibration DAC.

In the digital calibration, however, errors are calibrated by calibrating the digital codes or the digital output of the ADC [4].

There are many ways for calibration in literature, however, three of them will be covered in this chapter:

6.2.1. Over-range and under-range stages

In this design, over-range stages are used in all pipelines or in some of them [1][2]. Any stage transfer characteristic should remain linear in the $-1.5V_{ref}$ to $+1.5V_{ref}$ range so that errors can be corrected. The input range is $-1.0V_{ref}$ to $+1.0V_{ref}$, which means that the operational amplifier has to have its linear range larger than its input range. This means that this architecture allows the output of a stage to deviate from its ideal range and go out of it. The succeeding stage of a stage that went out of range should be able to recover from that since its linear input range can extend to $-1.5V_{ref}$ to $+1.5V_{ref}$. The correction algorithm used is called “*accuracy bootstrapping*” and it corrects one stage at a time. The basic idea in this algorithm is to use the rest of the ADC to measure the actual values of the all the parameters used in the ADC, store them and then use them to generate the binary equivalent of an input. The measured values are saved in small digital look-up tables in each stage of the converter. No tuning of analog components is necessary as long as the coefficients are computed accurately.

6.2.2. 1.5-bit per stage

In this design [7], redundancy is introduced such that with errors in the reference voltages exist, the output of a stage doesn't go out of range (as long as the errors are less than a specific value). This architecture is going to be used in the implementation of the ADC in this thesis.

6.2.3. Stage gain < 2

Gain errors, comparator offsets and DAC errors can be tolerated by using gain < 2. The architecture that implements this algorithm [8] requires digital multipliers and will result in a complicated digital circuit. The output range of the opamp is not completely used. The above errors can be avoided by increasing the input range of each stage beyond the nominal output range of the previous stage. This guarantees that the residues and thus the quantization error would remain limited. This has been achieved by reducing the gain of each stage to less than 2.

6.2.4. Pros and cons for the above designs

6.2.4.1. Stages with over-range and under-range

- It needs calibration cycle to measure the errors in each stage and then use this information to give correct results.
- It uses less logic in each stage. The calibration algorithm allows for some of the stage to have overrange and not all of them. So, most of the stages of the pipeline have 1 comparator, while the overrange stages have 2 comparators.
- It needs digital circuit that does the calibration in the calibration cycle. This might affect the noise of the system and hence degrades the overall SNR of the ADC.
- corrects for different errors in the system, like gain errors, comparator offsets and DAC errors.
- operational amplifiers need to be overdesigned. The input signal range should be $-1.0 V_{ref}$ to $+1.0 V_{ref}$, while the opamp should allow the input range to be $-1.5V_{ref}$ to $+1.5V_{ref}$, so that if errors occur, the opamp should stay in the linear region.

6.2.4.2. 1.5-bit per stage

- It mainly corrects for comparator errors.
- It requires redundancy in each stage except the last one.
- Doesn't require stand-alone digital system to do the calibration. This means that it is less noisy.
- Correction is done at the same time the input signal is processed.
- Opamps don't need to be overdesigned to tolerate errors.

6.2.4.3. Stage gain < 2

- One comparator per stage was used.
- Gain of 1.93 was used which resulted in a very complicated digital logic that will generate the digital code.
- Corrects for gain, DAC, offsets and reference errors in the system.

- Not all the dynamic range of the opamp was used due to gain less than 2.

6.3. Capacitor Error-Averaging

Capacitor error averaging is another technique for achieving a precise gain of 2 in the residue amplifier [5]. In this technique, calibration is not used. Instead, the residue is amplified twice, each time with a different feedback capacitor. By averaging these two results together, the error introduced by capacitor mismatch can be averaged out. The disadvantage of this technique is that it requires the use of two amplifiers rather than one for each sample and hold stage and an extra clock cycle is needed since the signal is amplified twice.

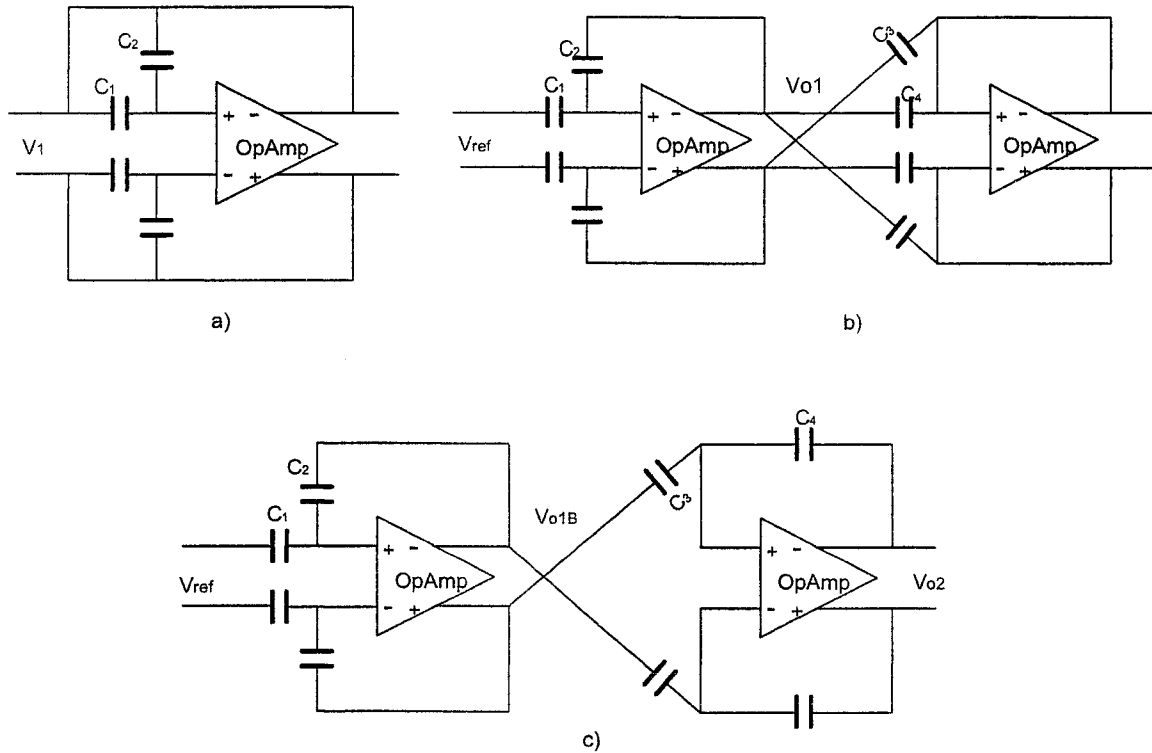


Figure 79 Capacitor mismatch error-averaging technique: a) sampling phase, b) hold phase 1, and c) hold phase 2.

Figure 79 illustrates what happens during each phase. During the sampling phase, the input signal V_{in} is sampled onto capacitors C_1 and C_2 . These two capacitors do not match perfectly. Therefore, using either capacitor as the feedback capacitor will result in a gain that is too large or too small. Therefore, there are two hold phases, one where C_1 is used for feedback and the other where C_2 is used for feedback. During the first hold phase, C_2 is used for feedback, and the output of the first amplifier is sampled onto capacitor C_4 while its inverse is sampled onto capacitor C_3 . During the second hold phase, C_1 is used for feedback, and the new output is connected to C_3 .

Now, the above figure is analyzed to get the output voltage V_{o2} as a function of the input voltage V_I and the reference voltage V_R . From the sampling phase and hold phase 1, the following expression is obtained.

$$V_{o1} = \left(\frac{C_1 + C_2}{C_2} \right) V_1 - \frac{C_1}{C_2} V_R \quad (1)$$

From the sampling phase and hold phase 2, the following expression is obtained.

$$V_{o1B} = \left(\frac{C_1 + C_2}{C_1} \right) V_1 - \frac{C_2}{C_1} V_R \quad (2)$$

Finally, from the hold phases 1 and 2, the following expression is obtained.

$$V_{o2} = \left(\frac{C_4 + C_3}{C_4} \right) V_{o1} - \frac{C_3}{C_4} V_{o1B} \quad (3)$$

Combining equations (1), (2) and (3), the following equation for V_{o2} is obtained:

$$V_{o2} = \left(1 + \frac{C_1}{C_2} \right) V_1 - \frac{C_1}{C_2} V_R + \frac{C_3}{C_4} \left(\frac{C_2^2 - C_1^2}{C_1 C_2} \right) V_1 + \frac{C_3}{C_4} \left(\frac{C_1^2 - C_2^2}{C_1 C_2} \right) V_R \quad (4)$$

Writing C_1 and C_2 in terms of a nominal capacitance C and error δC ,

$$C_1 = C - \frac{1}{2} \delta C \quad (5)$$

And

$$C_2 = C + \frac{1}{2} \delta C \quad (6)$$

For $\delta C \ll 1$, the following approximation can be made:

$$\frac{1}{1 + \frac{\delta C}{C}} \approx 1 - \frac{\delta C}{C} \quad (7)$$

And so,

$$V_{o2} = \left(2 - \frac{\delta C}{C} + \frac{2C_3}{C_4} \frac{\delta C}{C} \right) V_1 - \left(1 - \frac{\delta C}{C} + \frac{2C_3}{C_4} \frac{\delta C}{C} \right) V_R \quad (8)$$

From this equation, one can see that for $C_3 \approx 0.5C_4$, the mismatch effect cancels quite good.

6.4. Continuous calibration

In this design [3], a one-bit per stage pipeline A/D converter was used in the overall architecture. The architecture utilizes an extra stage that is calibrated outside of the pipeline and is periodically substituted for a pipeline stage requiring calibration.

The 1-bit ADC within a stage of the converter is simply a comparator that senses the difference between the analog input to the stage and a threshold voltage, V_{th} . The 1-bit DAC is formed with a pair of switches that connect the output to one of the two reference voltages V_{refp} and V_{refn} . A gain-of-2 amplifier in each pipeline stage was used. The sample and hold, 1-bit DAC, subtraction, and gain functions in a pipeline stage are readily merged into a single switched-capacitor CMOS circuit block. The architecture is the same as the one used in [6]. The operation of a stage is summarized as follows: During the *sampling* phase, the input voltage $V(I)$ at the input of the first stage of the pipeline is sampled onto both C_s and C_f , where C_s is the sampling Capacitor and C_f is the feedback capacitor. Near the end of this phase, the comparator compares $V(I)$ with the threshold voltage of the comparator and makes the decision.

During the multiply-and-subtract phase or sometimes called the *hold* phase, the bottom plate of C_f is connected to the output of the opamp while the bottom plate of C_s is connected to the appropriate reference voltage, V_{refp} or V_{refn} depending on the result of the comparison done by the comparator.

The output of the comparator in the n th stage of the pipeline is $D(n)$ and it is given by:

$$D(n) = \begin{cases} 1 & \text{if } V(n) \geq V_{th} \\ 0 & \text{if } V(n) < V_{th} \end{cases} \quad (9)$$

Assuming ideal stages in the converter, the output of the n th stage is given by:

$$V(n+1) = 2 V(n) - D(n) \cdot V_{refp} - \overline{D(n)} \cdot V_{refn} \quad (10)$$

Error sources of errors in a 1-bit-per-stage ADC such as comparator offset, charge injection from the sampling switches, finite op-amp gain and capacitor mismatch might generate missing decision levels and missing codes and affects the overall transfer characteristic. For example, if the effects of finite opamp gain A_0 and nonequal capacitors are taken into account, the analog output of a pipeline stage $V(n+1)$ is more generally given by:

$$V(n+1) = K \left[\left(1 + \frac{C_s}{C_f} \right) V(n) + \frac{C_s}{C_f} \cdot \left(-D(n) \cdot V_{refp} - \overline{D(n)} \cdot V_{refn} \right) \right] \quad (11)$$

Where K is given by:

$$K = \frac{A_0}{1 + \frac{C_s}{C_f} + A_0} \quad (12)$$

The ultimate target of this calibration algorithm is to get rid of the missing codes and the missing decision levels. The calibrated transfer characteristic has no missing codes or missing decision levels, and therefore the absolute value of the converter's differential nonlinearity (DNL) is guaranteed to be less than 1 LSB. To guarantee that the converter has enough decision levels to achieve the specified resolution, extra stages are added to the pipeline. The basic idea of the proposed calibration technique is that a given level of

performance can be achieved simply by guaranteeing that for analog inputs to a stage around V_{th} , the stage's output range exactly matches the resolvable input range of the remaining stage in the pipeline.

The calibration algorithm performs two things:

1. Removal of the missing decision levels.

This is not absolutely necessary. The reason for that is that the missing decision levels can be removed by adding extra stages. But, when does the missing decision levels occur? They occur when the input range to any stage in the pipeline is exceeded. It suggested that a capacitor ratio less than unity to be used in stages likely to require calibration. The primary advantage of this approach is that an extra capacitor is not required. Another advantage is that the input range of the remaining stages in the pipeline may be completely accessed for at least some input values, thereby possibly leading to less reduction in the number of decision levels[3].

2. Removal of the missing codes:

Missing codes are avoided in two steps. First, the residue of a stage for an input just less than the stage's comparator threshold voltage, V_{th} , is adjusted until it reaches the full-scale input voltage of the following stage in the pipeline. The second calibration step ensures that the converter's transfer characteristic has no missing codes when $D(n)$ transitions to a logic 1. This is accomplished by ensuring that stage n 's residue reaches the most negative resolvable input voltage of the remaining stages when the analog input to stage n is just larger than V_{th} .

Together, these two steps guarantee that the digital outputs of the subsequent stages will transition from all logic 1's to all logic 0's when $D(n)$ transitions from a logic 0 to a logic 1 due to an infinitesimal increase in $V(n)$ from $(V_{th}+\epsilon)$ to $(V_{th}+\epsilon)$, where ϵ is a small voltage. The two steps were implemented as follows:

- a. When $D(n) = 0$, the output of stage n which is the input of stage $n+1$ is adjusted to $(1+\beta)V_{refp}$ such that $D(n+1)D(n+2) \dots D(n)$ will reach logic 1's and cause the analog residue of the LSB stage in the pipeline to reach the ideal full-scale output, V_{refp} . This will be accomplished by changing the threshold voltage of the comparator from V_{th} to V_{tha} .
- b. The second step is to use V_{tha} in order to change V_{refp} to V_{refpa} when $D(1)$ is 1 such that the output voltage of the stage $V(2)$ reaches the following stage's most negative resolvable input voltage $(1+\gamma)V_{refn}$. This will force the last stage in the pipeline to produce the output V_{refn} .

The value of β of stage n is calculated from the following equation assuming that stage $n+1$ has a capacitor ratio α

$$\beta|_{K=1} = \frac{1-\alpha}{1+\alpha} \left(1 - \frac{V_{refn}}{V_{refp}} \right) \quad (13)$$

The actual value of γ can be determined by setting the input to stage $n+1$, which is $V(n+1)$, to V_{refn} and using equation (3) with $D(n) = 0$.

6.5. Proposed Single Path calibration algorithm

6.5.1. Overview

It has been noticed that adjusting the DAC values can cancel the gain errors. Since the DAC values are adjusted, DAC errors are of no significant in the ADC.

Throughout this work, the two configurations: 1.5bit-per-stage and 1-bit-per-stage were analyzed and simulated. However, for the purpose of analysis, the 1.5bit-per-stage configuration will be considered.

A 1.5-bit per stage pipeline ADC consists of multiple of stages where each stage resolves 1.5 bit of the overall digital representation of the input signal. The 1.5 bit comes from the fact that each stage quantizes the input to one of three levels, where each level has its own binary representation. A typical schematic of a stage of a 1.5-bit per stage ADC is shown in Figure 80 with its residue plot in Figure 81. Figure 81 shows the output of the stage vs. its input.

In a CMOS ADC, all signals are evaluated with respect to a common mode signal, hence, the x-axis of Figure 81 is the input voltage with respect to the common mode signal. The input of the ADC is usually differential, however, in order to simplify the analysis, single-ended design is considered in this section

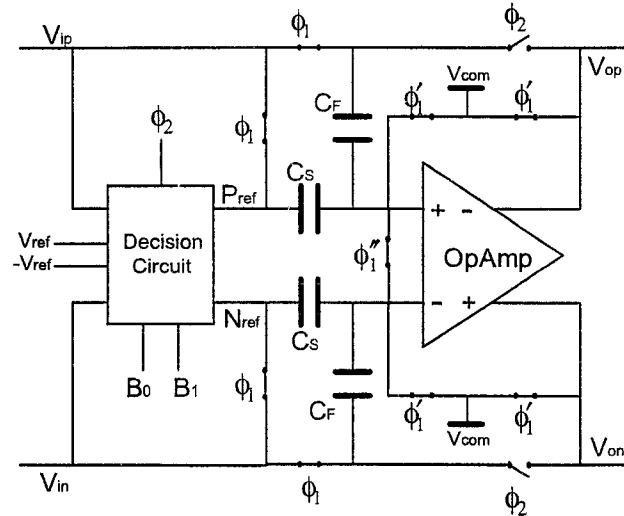


Figure 80 1.5-bit-per-stage stage of an ADC.

The decision circuit shown in Figure 80 generates the digital output of the stage by comparing the input signals to threshold voltages. There are two threshold voltages:

$$V_{thn} = V_{refn} + \frac{3}{8}(V_{refp} - V_{refn}) \quad (14)$$

$$V_{thp} = V_{refn} + \frac{5}{8}(V_{refp} - V_{refn}) \tag{15}$$

Where V_{thn} is the negative threshold voltage, V_{thp} is the positive threshold voltage, V_{refn} and V_{refp} are the negative and positive reference voltages of the circuit. Equations (14) and (15) give the value of the threshold voltages in absolute values, however, in differential form, $V_{thp} = -V_{thn} = 0.125$, when $V_{refp} = -V_{refn} = 0.5$.

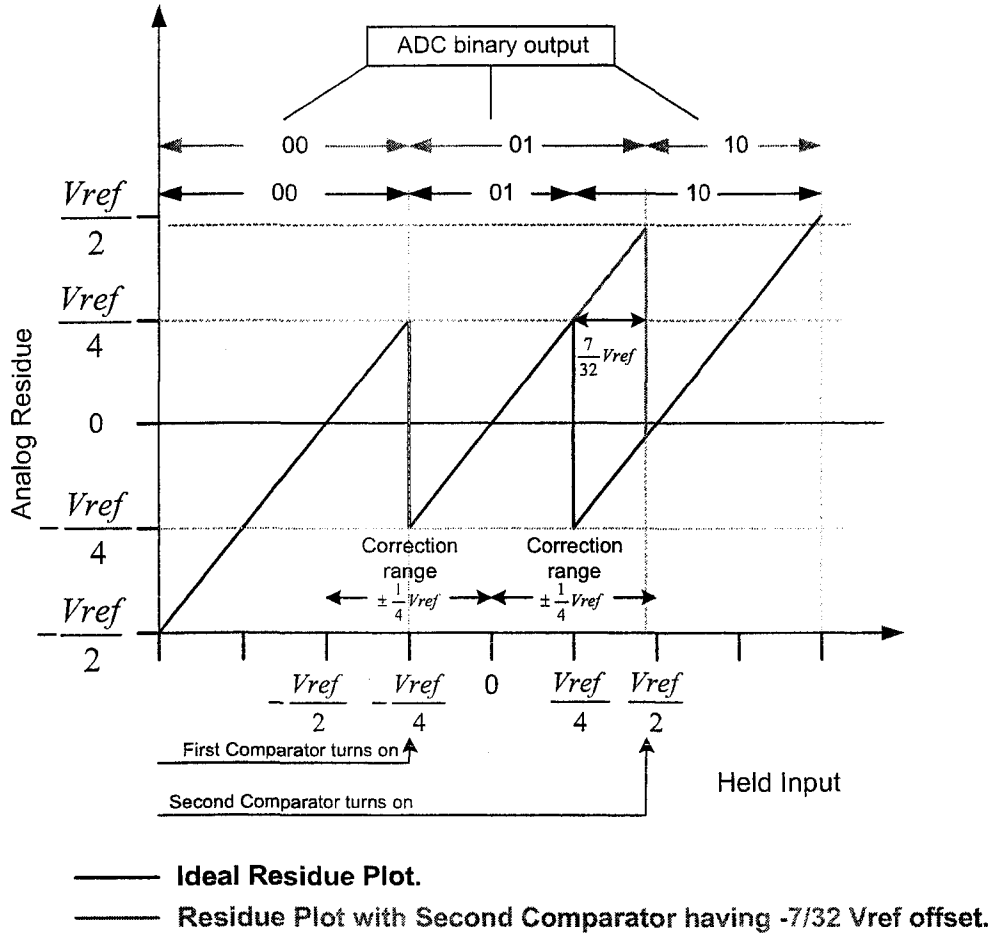


Figure 81 Residue plot of a 1.5-bit-per-stage ADC.

The digital output of the decision circuit is given by:

$$B_0 = \begin{cases} 0 & V_{ip} < V_{thn} \\ 1 & V_{ip} \geq V_{thn} \end{cases} \tag{16}$$

$$B_1 = \begin{cases} 0 & V_{ip} < V_{thp} \\ 1 & V_{ip} \geq V_{thp} \end{cases} \tag{17}$$

The two outputs given in equations (16) and (17) are used to generate the stage binary output shown in Figure 81. Those relations are given by the following equation:

$$\begin{aligned} D_0 &= B_0 \overline{B_1} \\ D_1 &= B_1 \\ D &= D_1 D_0 \end{aligned} \quad (18)$$

Where D is the binary data of each stage. This is shown as the 'ADC binary output' in Figure 81.

Due to the symmetry of the pipeline architecture, the input to each stage should range from $-V_{ref}$ to $+V_{ref}$. Since the input of a certain stage is connected to the output of its previous stage, this means that the output of each stage should also be in the same range, i.e., the output should range from range from $-V_{ref}$ to $+V_{ref}$. If the input value is greater than $\frac{1}{2}$ of the input range, say $V_{ip} = 3/4 V_{ref}$, then multiplying it by 2 will cause the output of the stage to go out of its region. To overcome this problem, each stage not only multiply by 2, but also add, subtract, or do nothing, such that the output is guaranteed to be in the specified region.

The residue plot in Figure 1.b) shows that there are three distinct regions. When $D_1 D_0 = 00$, $+V_{ref}$ is added, when $D_1 D_0 = 01$, nothing is added or subtracted, When $D_1 D_0 = 10$, $-V_{ref}$ is added, and the output is always within the specified region.

Without loss of generality, a 4-stage pipeline ADC will be considered in this section. This ADC consists of 4 stages, where the last stage resolves 2 bits, while the first three stages each resolves 1 bit. The ADC is shown in Figure 82.

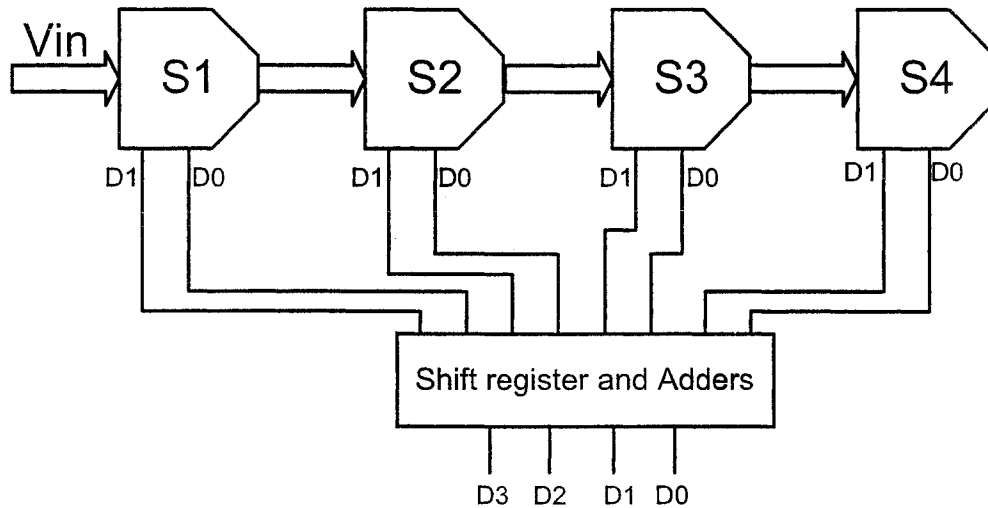


Figure 82 Four-stage pipeline ADC.

If we assume that V_{in} , the input voltage of stage 1 as shown in Figure 82, is $V(1)$, then the output of stage 1 is the input of stage 2, hence,

$$\begin{aligned}
 V_o(1) &= V(2) \\
 V_o(2) &= V(3) \\
 V_o(3) &= V(4)
 \end{aligned}
 \tag{19}$$

For an ideal stage, say stage n , shown in Figure 80, its output is related to its inputs according to the following equation:

$$V(n+1) = 2 \cdot V(n) - B_1 \cdot V_{refp} - \overline{B_0} \cdot V_{refn}
 \tag{20}$$

Where B_0 and B_1 are those given by equations (16) and (17), respectively.

For an ideal ADC the overall transfer characteristic and the residue of stage n are shown in Figure 83.

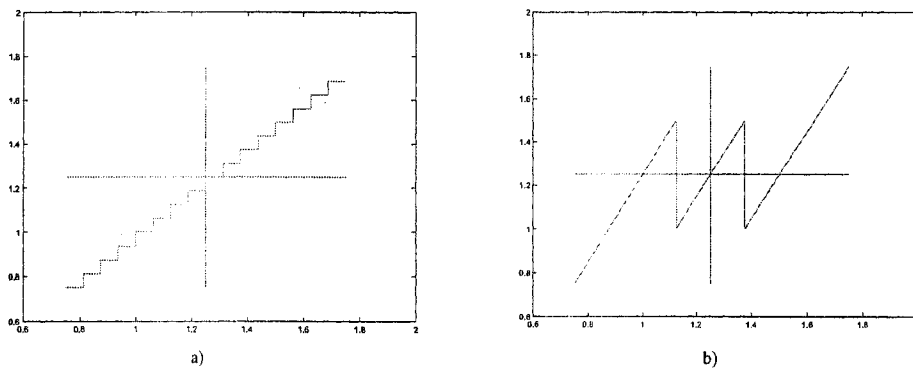


Figure 83 Overall Ideal characteristic and residue of the ADC.

6.5.2. Comparator offsets

The 1.5-bit-per-stage configuration shown in Figure 80 is not affected by comparator offsets as long as they are less than $\pm \frac{V_{ref}}{4}$. Figure 84 shows the ADC transfer characteristic with comparator offsets. In Figure 84.a) and Figure 84.b) the offset is exactly $\frac{V_{ref}}{4}$, while in Figure 84.c) and Figure 84.d) an offset $> \frac{V_{ref}}{4}$ is introduced. One can clearly see the effect of the offsets on the ADC if those errors are greater than $\frac{V_{ref}}{4}$.

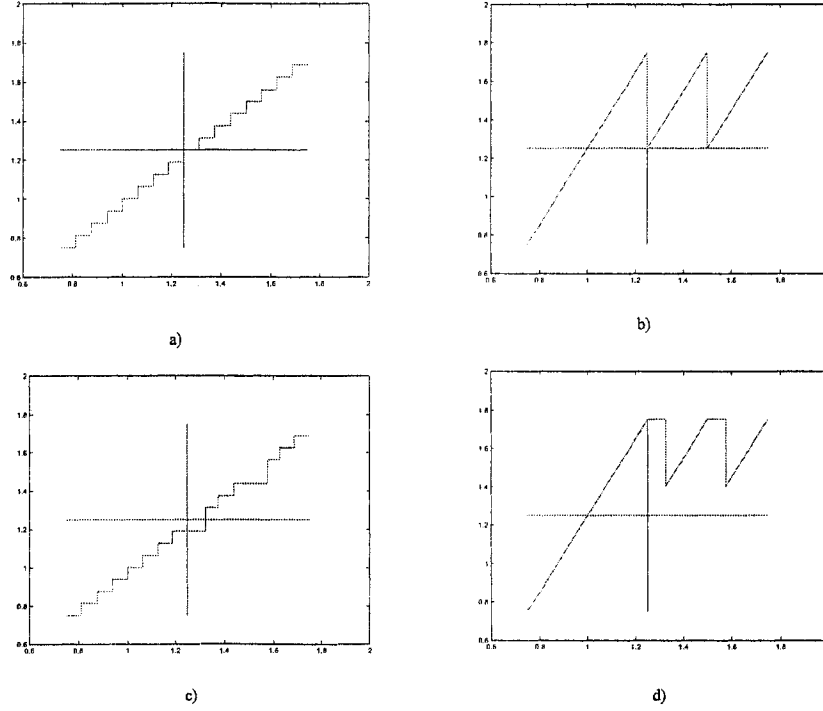


Figure 84 Effects of Comparator offsets on the ADC behavior.

6.5.2.1. Gain errors

In this design, gain errors mainly come from two sources: Opamp finite gain and the capacitor mismatch. Other sources of errors can be included in the capacitor ratio and the op-amp gain [3]. In the following analysis, only the gain errors resulted from the capacitor ratio will be considered.

With capacitor ratios modeled, equation (20) can be rewritten as:

$$V(n+1) = \left(1 + \frac{C_s}{C_f}\right) \cdot V(n) + \frac{C_s}{C_f} (-B_1 \cdot V_{refp} - \overline{B_0} \cdot V_{refn}) \quad (21)$$

To simplify the analysis, let us consider the 4-stage pipeline shown in Figure 82 with only the first stage, stage 4, is having the errors while the rest of the pipeline is considered ideal. This assumption will be relaxed in the suggested algorithm.

First, we need to analyze the effect of this capacitor ratio on the ADC behavior.

6.5.2.2. Effects of Capacitor mismatch on the ADC

There are two effects on the ADC behavior with capacitor ration $1+\delta C$, where δC is the error in the capacitor ratio.

- a. Step width. If we start sweeping the input signal to the ADC from V_{refn} and increasing it, then the digital code will start from $B_0 = 1$ and $B_j = 0$. For that input value, equation (21) can be written as:

$$V(n+1) = \left(1 + \frac{C_s}{C_f}\right) \cdot V(n) - \frac{C_s}{C_f} \cdot V_{refn} \quad (22)$$

According to Figure 81, when the transition occurs, the value of the output, which is $V(n+1)$ in equation (23), will be $-\frac{V_{ref}}{4}$, which is the threshold voltage of the last stage. The step width, will be the value of the input of the first stage that caused the output of the last stage to reach $-\frac{V_{ref}}{4}$.

In our 4-stage ADC, the width of the first step will be $(V(1) - V_{refn})$ that makes $V(5) = -\frac{V_{ref}}{4}$.

Substituting that in equation (22) will result in:

$$V(5) = \left(1 + \frac{C_s}{C_f}\right)_4 \cdot V(4) - \left(\frac{C_s}{C_f}\right)_4 \cdot V_{refn} \quad (23)$$

$$V(4) = \left(1 + \frac{C_s}{C_f}\right)_3 \cdot V(3) - \left(\frac{C_s}{C_f}\right)_3 \cdot V_{refn} \quad (24)$$

$$V(3) = \left(1 + \frac{C_s}{C_f}\right)_2 \cdot V(2) - \left(\frac{C_s}{C_f}\right)_2 \cdot V_{refn} \quad (25)$$

$$V(2) = \left(1 + \frac{C_s}{C_f}\right)_1 \cdot V(1) - \left(\frac{C_s}{C_f}\right)_1 \cdot V_{refn} \quad (26)$$

Note that the subscript in equations (23) – (26) means the stage number. If we assume that all the stages are ideal except the first one, then equations (23) – (26) can be rewritten as:

$$-\frac{V_{ref}}{2} = 2 \cdot V_d(4) - V_{refnd} \quad (27)$$

$$V_d(4) = 2 \cdot V_d(3) - V_{refnd} \quad (28)$$

$$V_d(3) = 2 \cdot V_d(2) - V_{refnd} \quad (29)$$

$$V_d(2) = \left(1 + \frac{C_s}{C_f}\right)_1 \cdot V_d(1) - \left(\frac{C_s}{C_f}\right)_1 \cdot V_{refnd} \quad (30)$$

Where the subscript d in equation (27) – (30) means the differential value.

Substituting equation (27) – (29) into equation (30) to find $V_d(1)$.

$$V_d(1) = \frac{1}{8 + 8 \cdot \left(\frac{C_s}{C_f}\right)_1} \cdot \left(-\frac{V_{ref}}{4} + V_{refnd} \cdot \left(8 \cdot \left(\frac{C_s}{C_f}\right)_1 + 7 \right) \right) \quad (31)$$

Thus the width of the first step will be $V_d(1) - V_{refnd}$.

Figure 85 shows the effect of the capacitor ratio value on the width of the first step. The relationship is almost linear. As the ratio increases, the width decreases.

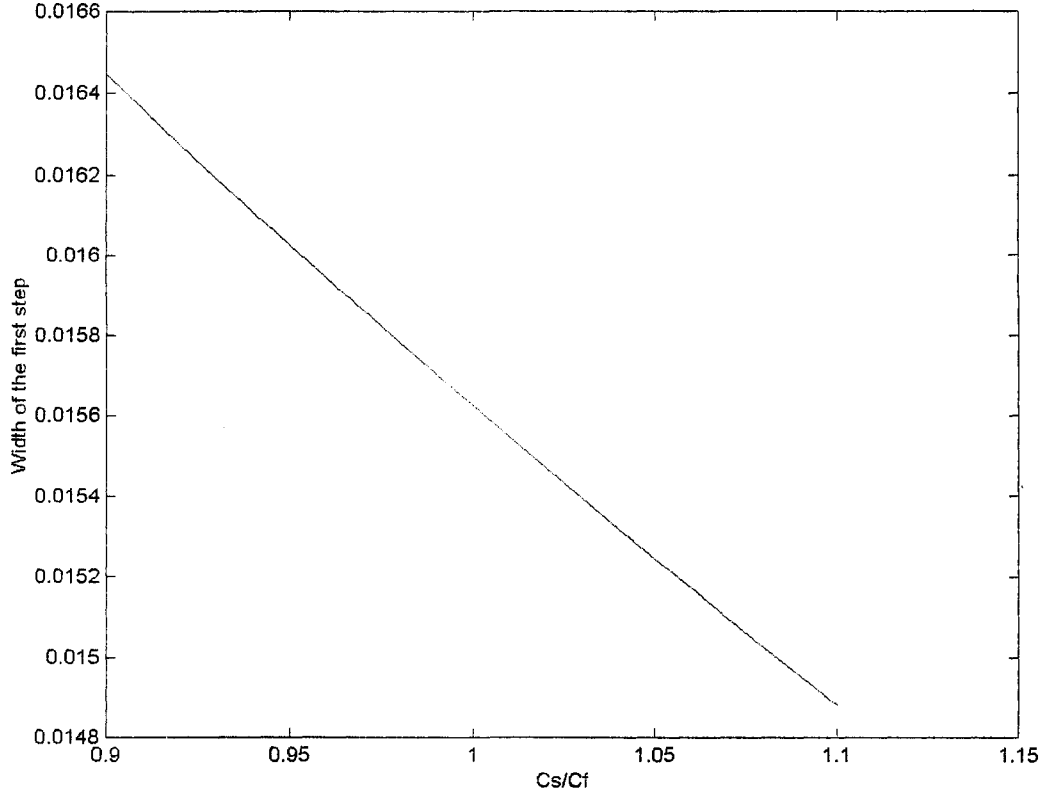


Figure 85. Effect of gain error on step width.

- b. Slope of the overall characteristic. If we assume that $C_s/C_f = 1.0 + \delta C$, then errors in the capacitor ratio will cause the slope of the ADC characteristic to be $2 + \delta C$.

The above two errors can be quantified by the following equation.

$$V_{od} = (((((2 + \delta C) \cdot V_d(1) - (1 + \delta C) \cdot V_{n1d}) \cdot 2 - V_{n2d}) \cdot 2 - V_{n3d}) \cdot 2 - V_{n4d} \quad (32)$$

Consider the input voltage to the ADC is V_{refn} . This represents the 0 input and the digital output of the ADC is 0000. The output of the ADC, which is the output of stage 4, will also be V_{refn} if the ADC is ideal. Increasing the input gradually will also increase the output gradually. By the time the input is increased by 1 LSB, the output of the ADC will reach its maximum, which is V_{refp} . The digital representation of the input signal is still 0000. Mathematically, the output of the ADC for an input less than $(V_{refn} + 1 \text{ LSB})$ is given by equation (32)

Where the subscript d stands for *differential* which means that V_{od} is the differential value of the output of the ADC, $V_d(1)$ is the differential input of the ADC, V_{nd} is the differential value of the negative reference voltage of stage 1, and so on.

Equation (32) is derived from equation (22) above when the digital output of the ADC is 0000. Equation (24) clearly shows the effects of capacitor errors on the ADC characteristic.

- $\delta C \cdot V_d(1)$ contributes to a slope error.
- $\delta C \cdot V_{nd}$ contributes to a horizontal shift in the first step of the ADC overall characteristic.

For negative δC , the overall ADC characteristic will suffer missing codes and for positive δC , the overall ADC characteristic will suffer nonmonotonicity as shown in Figure 86.a) and Figure 86.b) respectively.

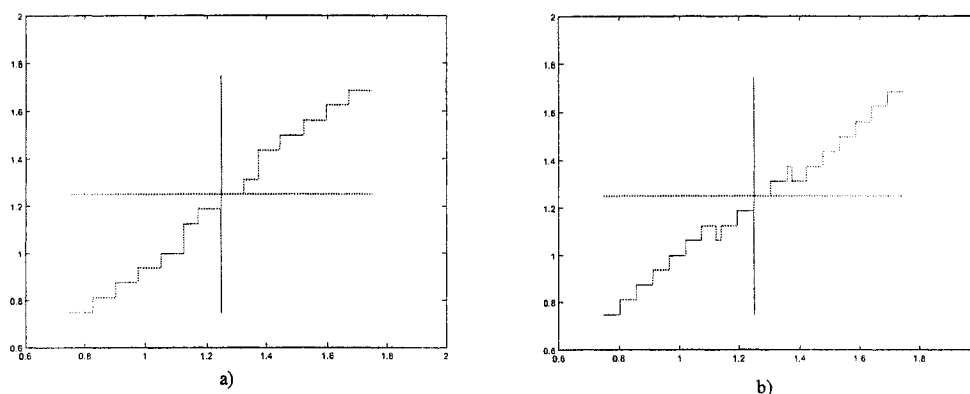


Figure 86 Effect of negative and positive capacitor errors on the overall ADC characteristic.

Comparing Figure 83.a) to Figure 86.a) leads us to the fact that a missing code results in the digital code of the last step of the ADC overall characteristic when D_1 is 00 to be less than that of the ideal by at least 1 as illustrated in Figure 87. This is mainly due to the negative δC which caused the gain of the ADC to be less than 2 and made the ADC unable to reach that of Figure 83.a).

6.5.3. Correction Algorithm

It is observed that changing the DAC values will reduce the effect of capacitor ratio errors and inherently will get rid of DAC errors as well.

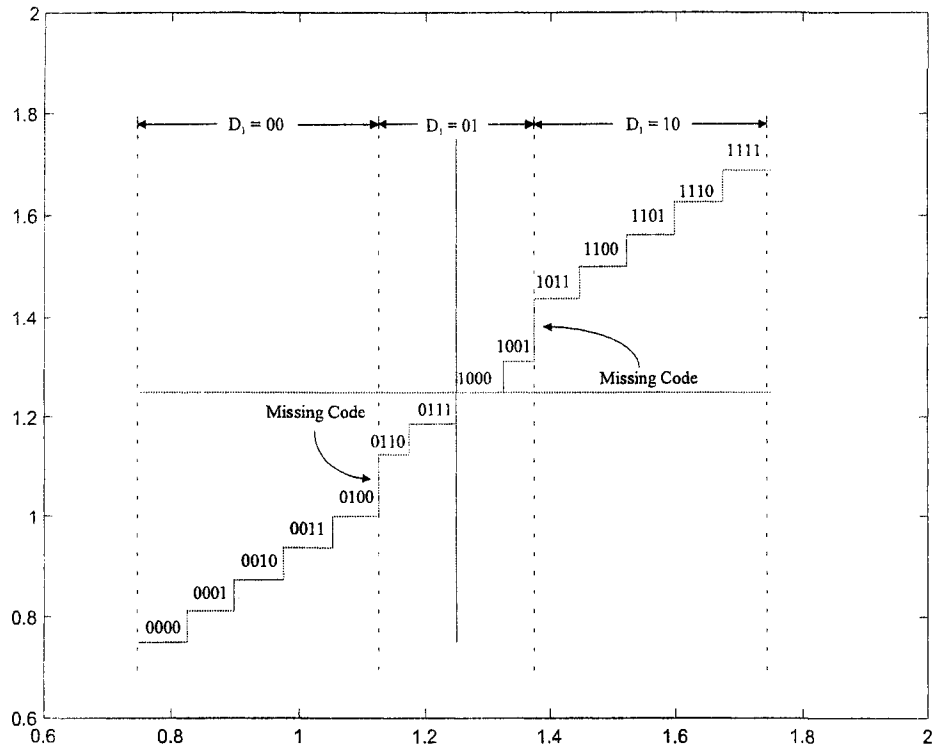


Figure 87 Illustration of the missing code.

A missing code results due to the negative δC which caused the gain of the ADC to be less than 2 and the transition gap size to be less than the ideal V_{ref} . The actual size of the transition gap is $\frac{C_s}{C_f} \cdot V_{ref}$. The missing code can be recovered if the actual size of the transition gap is changed to V_{ref} . As equation (33) suggests, this can be implemented by increasing V_{refn} such that $\frac{C_s}{C_f} \cdot V_{refn}$ equals to the ideal V_{ref} . Similarly, nonmonotonicity occurs when δC is positive which leads to a gain of more than 2 and a transition gap size more than the ideal V_{ref} . The effect of this error can be reduced by reducing V_{refp} such that $\frac{C_s}{C_f} \cdot V_{refp}$ equals the ideal V_{ref} .

In the appendix, an algorithm is presented that corrects for capacitor mismatch by changing the DAC values. It assumes that the last 2 stages of the pipeline are ideal. The algorithm corrects the ADC stage by stage starting by the least significant one before the ideal last two stages. So, for the 4-stage ADC mentioned above, the algorithm assumes that stages 3 and 4 are ideal and starts by correcting stage 2. Once stage is corrected, it uses the corrected stage 2 and stages 3 and 4 to correct stage 1.

When correcting a stage, the algorithm starts by calculating the gain of that stage. Based on that, δC will be found. This value is then used to calculate the DAC offset needed to correct for this error.

The error correction algorithm has a self-correction mechanism, where after adding the offsets to the DAC values it checks if it decreases the DNL. If so, it exits, otherwise, it tries to reduce the offset by half, and add it again and check for better DNL. The algorithm exits the added offset deteriorates the DNL, or, the offset becomes less than a tolerance which will leave the DAC values unchanged.

Figure 88.a) and Figure 88.b) show the overall characteristic of the ADC after correction where missing codes and nonmonotonicity have been cancelled. The downside of this correction is that the width of the first step of the overall characteristic is affected. When correction for missing codes occurred, the width of the first step is reduced, while it is increased when correction for nonmonotonicity took place.

These two effects will determine how much correction can be introduced to the system.

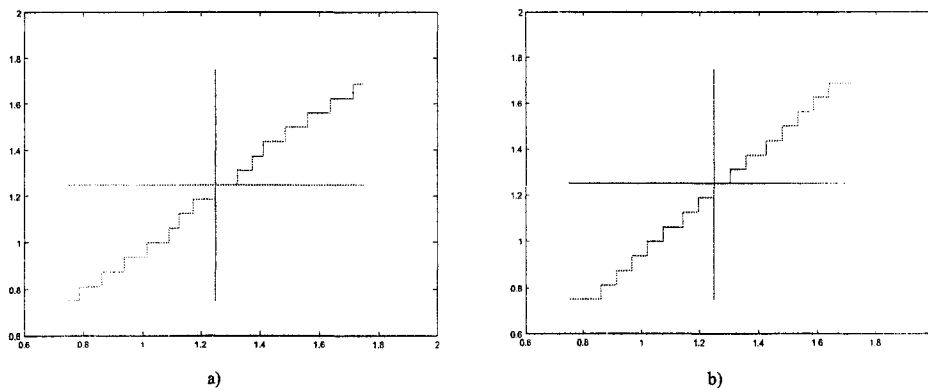


Figure 88 Overall characteristic of an ADC with capacitor error after correction.

```

//      For an N bit ADC: stage 0 to stage N-1, where stage 0 is the first stage in
//      the pipe that is connected directly to the input

set Tolerance = 0.1;    // in LSB
// Assuming that the last 2 stages are ideal and the last stage generates 2 bits

set n = 0;

//      The algorithm starts by correcting one stage at a time. It starts by the first Nonideal LSB
//      stage which is stage 0, since the last 2 stages: stage N-1 and stage N-2, are ideal.

while (n <= N-3) {

    // generate the ADC overall characteristic
    GenerateADCOverall(n);

    //Calculate the gain of the MSB stage.
    set StageActualGain = CalculateStageGain();

    // calculate the value of Cs/Ci which is the acutal capacitor ratio.
    set CsCi = CalculateCsCi();
    if (Gain = 2) exit(0); // done....
    else if (Gain < 2) {
        set FirstStepWidth = StepWidth(1); // Find the width of the first step
        if ( MissingCodeExists() ) {

            // set the width of the first step to 0.5LSB
            set UltimateFirstStepWidth = 0.5;
            set Found = FALSE;
            while ((UltimateFirstStepWidth > Tolerance) and !Found) {

                // Calculate DACOffset
                set DACOffset = (FirstStepWidth-UltimateFirstStepWidth);
                InitializeADC( DACOffset() );
                GenerateADCOverall(n);
                if ( !MissingCodeExists() ) set Found = TRUE;

                // decrease the width of the first step
                else set UltimateFirstStepWidth = UltimateFirstStepWidth / 2;
            }
            if ( !Found) exit(1); // Algorithm fails to correct
        }
        else {
            set NewStepWidth = CalculateStepWidth();
            set DACOffset = (NewStepWidth + FirstStepWidth)/2.0-NewStepWidth;
            InitializeADC( DACOffset() );
            // done
        }
        else { // Gain > 2
            set Error = (1.0 - 2.0/StageActualGain)/(2.0*LSB); // 1/(2.0*LSB) is NumOfLevels/2
            set DACOffset = -Error;
            InitializeADC( DACOffset() );
            // done
        }
    }
    set n = n+1;
} // done correction algorithm

```

Figure 89 Correction algorithm for gain errors.

The algorithm that corrects for the gain errors in the ADC is shown in Figure 89. It assumes the last two stages are ideal. The last two stages of the ADC generate 3 bits, since the last stage generates 2 true bits².

The algorithm shown in Figure 89 starts by the first stage of the ADC and tries to correct the gain error in this stage. It formulates a small ADC that consists of this stages as the first stage, stage $N-2$ and stage $N-1$. Only the first stage is not ideal in this ADC. The algorithm starts by finding the gain of this ADC. If the gain is found to be greater than 2, the algorithm calculates the amount of offset in the DAC values to be subtracted by calculating the overall error introduced to the system due to the gain error. This overall error is calculated as follows.

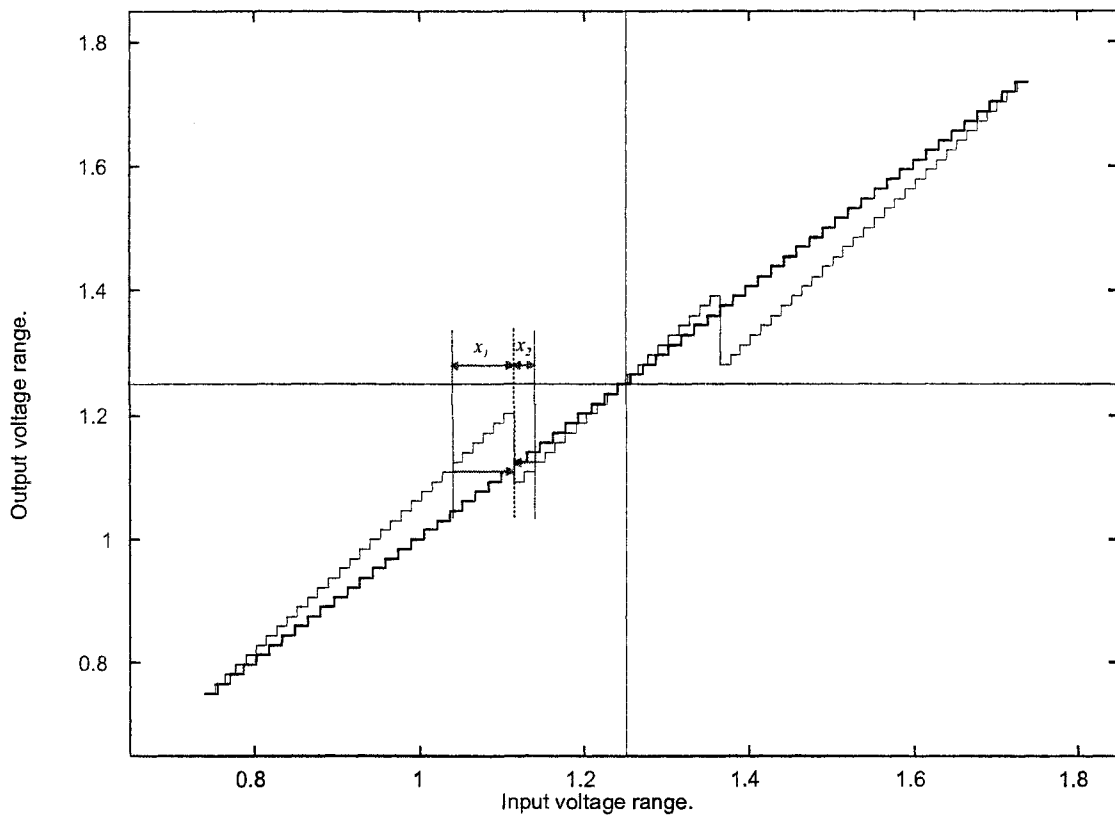


Figure 90 Calculation of gain errors.

² The reason for this is that the last stage has no redundancy in it. It has three comparators and quantizes to 4 levels.

x_1 and x_2 in Figure 90 show the shift introduced due to gain errors. The solid line in Figure 90 represents the overall characteristic of the ADC with δC greater than 0, while the thick line represents the overall characteristic of an ideal

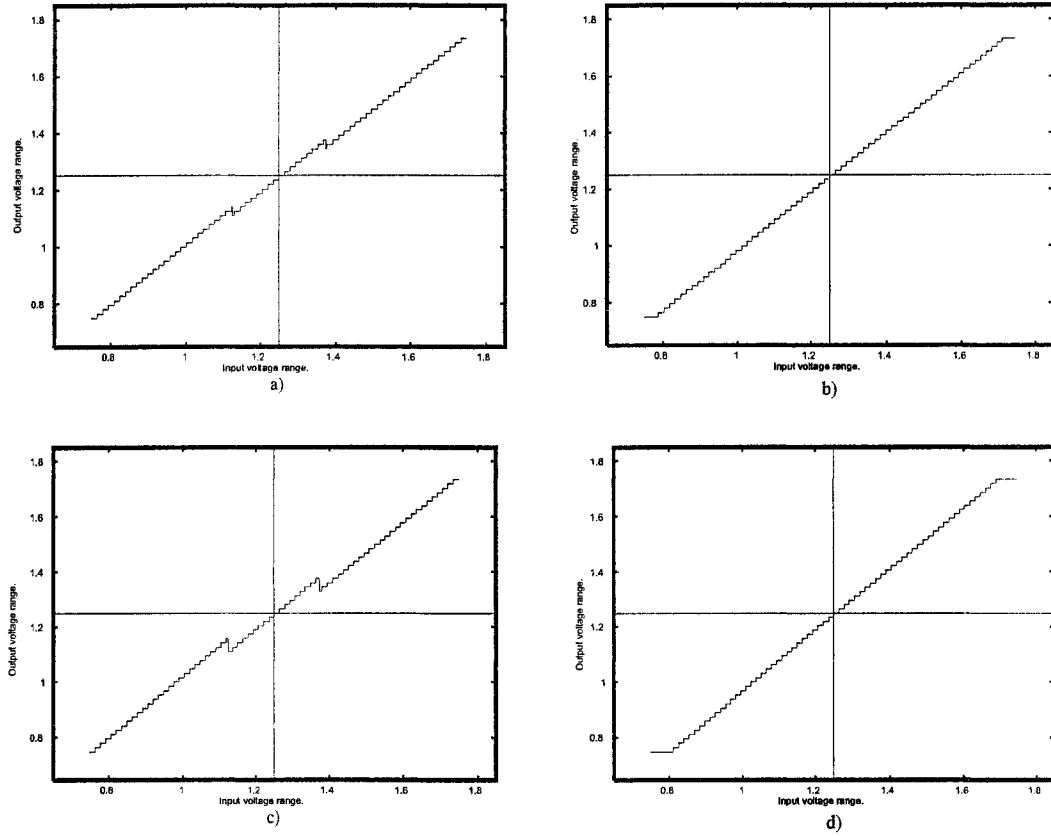


Figure 91 Overall characteristic of a 6-bit ADC before and after applying the correction algorithm.

a) and b) $C_s/C_i = 1.1$, c) and d) $C_s/C_i = 1.2$.

In order to get rid of the nonmonotonicity, the DAC offset that needs to be subtracted should be the addition of x_1 and x_2 . The two values are calculated as follows:

$$x_1 = \frac{3}{4} \cdot \frac{m}{2} \cdot \left(1 - \frac{2.0}{2.0 + \delta C} \right) \quad (33)$$

$$x_2 = \frac{1}{4} \cdot \frac{m}{2} \cdot \left(1 - \frac{2.0}{2.0 + \delta C} \right) \quad (34)$$

Where x_1 and x_2 are measured in LSB, m is the number of levels or steps in the ideal ADC, which is 2^N , where N , is the number of bits the ADC can resolve. Note that

$$x_1 + x_2 = \frac{m}{2} \cdot \left(1 - \frac{2.0}{2.0 + \delta C} \right) \quad (35)$$

Which is the same error that will result in a 1-bit per stage architecture. Figure 91 shows the overall characteristic of a 6-bit ADC before and after applying the algorithm. Note the increase in the offset as the gain error increases after correcting it.

For negative δC , the stage gain will be less than 2, and missing codes might result. Although the same criterion used above for a positive δC can be used here, but that might result in the first step of the overall transfer characteristic being missed. To get rid of this, the algorithm tries to average the step width that is affected by the gain error with the first step. However, if a missing code exists, the algorithm tries first to introduce the missing code by increasing the DAC values. Once there is no missing code, the algorithm tries to average the width of the step that was missing with the first step or the minimum width step if no missing code exists.

After the first stage is calibrated, the next one in line is calibrated until all the stages are calibrated.

As an illustration of this algorithm, a 5-bit ADC is simulated, which has the first 3 stages not ideal, while the last two are. A gain less than 2 resulted in a DNL of 0.4297 LSB as shown in Figure 92.a). After applying the algorithm, the DNL comes out to be 0.3477 LSB and the output is shown in Figure 92.b). The nonlinearity has been reduced. For a gain greater than 2, nonmonotonicity has been resulted as shown in Figure 93.a). After applying the algorithm the DNL comes out to be 0.3828 with a monotonic output as shown in Figure 93.b).

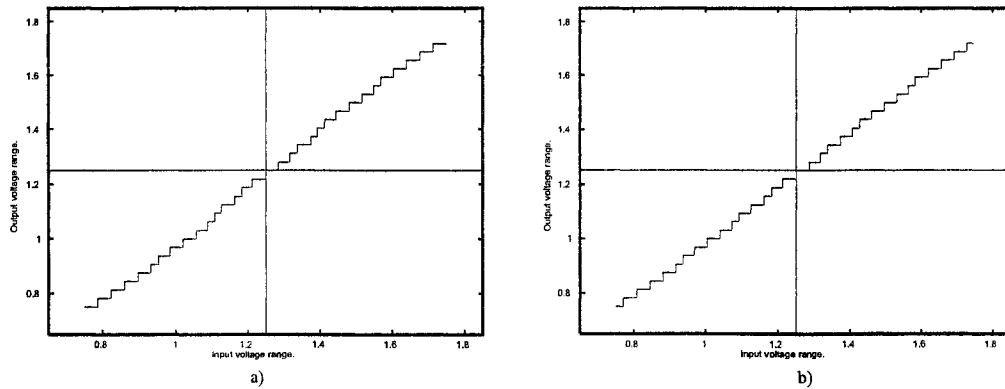


Figure 92 5-bit ADC with stage gain = 1.85: a) before calibration, and b) after calibration.

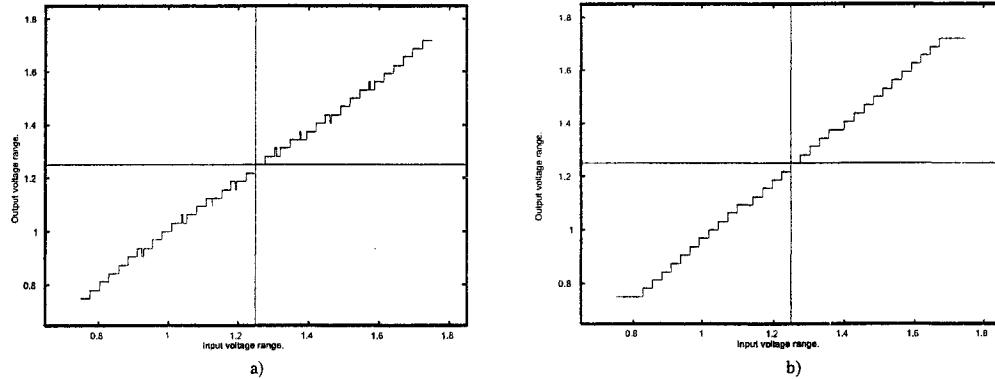


Figure 93 5-bit ADC with stage gain 2.15: a) before calibration, and b) after calibration.

As an example, a 10-bit ADC is considered. A sinewave signal is applied to its input. The FFT plot of the sinewave input is shown in Figure 94.a). Figure 94.b) shows the FFT plot of the quantized signal at the output of the ADC when there is no errors introduced. It shows the ADC exhibits an SNDR of 70dB. With gain errors randomly chosen for the different stages of the ADC, the output is distorted as shown in Figure 94.c). The SNDR dropped to 42dB. After applying the correction algorithm, the SNDR increased to 54dB. This means that the correction algorithm resulted in a gain of 12dB.

The correction algorithm doesn't suppress the tones that resulted from the errors in the gain and DAC values of each stage, however, it tries to average those tones and make them look like a uniform noise.

The algorithm works in both direction. It can start from the MSB stage and goes to the LSB stage or doing the opposite by starting from the LSB stage and going to the MSB one. If the first approach is chosen, the DNL is always measured after a stage being corrected. If it resulted in a better DNL, the algorithm moves to the next stage, otherwise, it will try to reduce the amount of offset introduced and check the DNL again, until either a tolerance value is met or a better DNL is achieved.

There are many other ways and techniques that can be used to do the measurement of most of the errors in the ADC. Code-error measurement technique, [11][12], can be used to simultaneously measure all the nonlinearity errors in a stage that result from many sources. Custom microcontrollers can also be used to do the measurement as in [2] and [13].

The algorithm shown in Figure 89 assumes that the actual gain of the stage under calibration is known. An algorithm proposed here in section 6.5.4 will show how to find the actual gain and DAC offsets of each stage from one linear sweep of the input and recording of the digital output. This algorithm should be performed before the algorithm in Figure 89. As mentioned before, the outputs of this algorithm are the actual gain and DAC values of each stage. This linear sweep can be performed by a DAC with a better accuracy and resolution than the ADC under measurement. When doing so, an image of the ADC is generated, which means that the algorithm in Figure 89 can be used without performing an actual measurement to the real ADC.

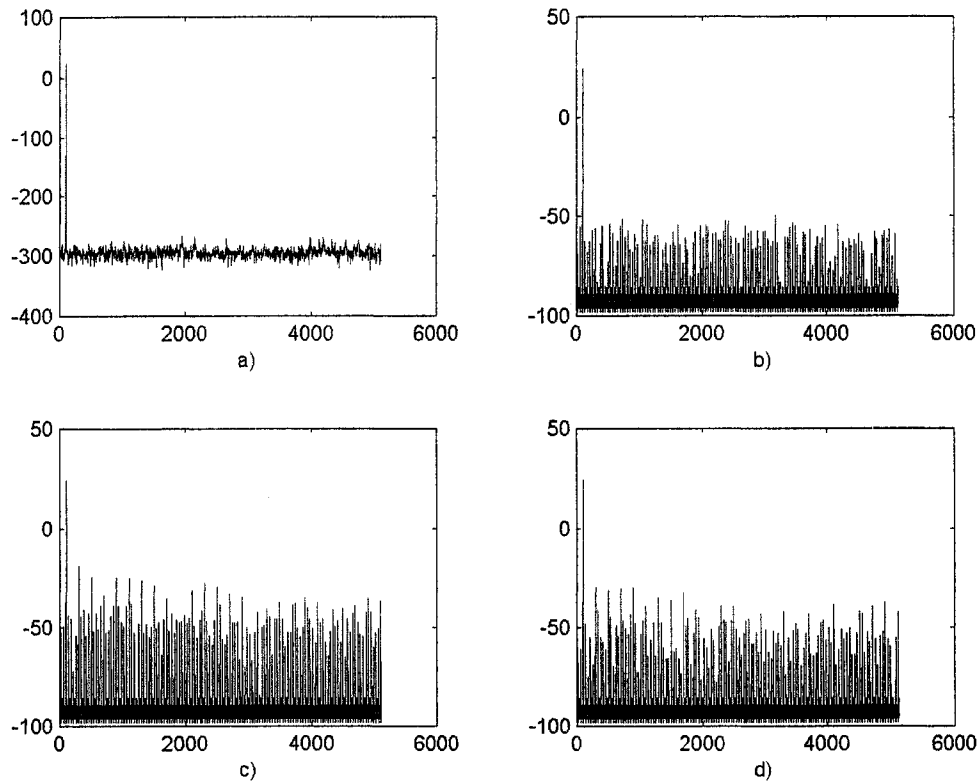


Figure 94 FFT plot of the output of 10-bit ADC. a) The FFT of the input signal. b) The FFT of the quantized output when the ADC doesn't have any error. c) The FFT of the output when the ADC is having gain errors. d) The FFT of the output after the gain errors are corrected by the algorithm.

One way to implement this algorithm is to use a DAC that has a better accuracy than the ADC under design. This scheme is implemented in [5] and has the disadvantage of extra DAC but it is simple and straightforward.

6.5.4. Gain and DAC measurement algorithm

In this section, an algorithm that measures the actual gain of every stage of the ADC as well as its DAC values is presented.

Lets assume that we have an n -bit ADC that has n identical stages; S_1 to S_n , where S_1 is the MSB stage. The input of the ADC is the input to stage S_1 . Using a DAC that has a better accuracy than the ADC under measurement, the input to the ADC is swept from $-V_{refnd}$ to $+V_{refnd}$, where V_{refnd} is the differential value of the minimum allowable input. For example, for an ADC whose input can be swept from 0.75 to 1.75 and a common-mode value of 1.25, then V_{refnd} is -0.5 . At the same time, the digital equivalent of the each input value is recorded.

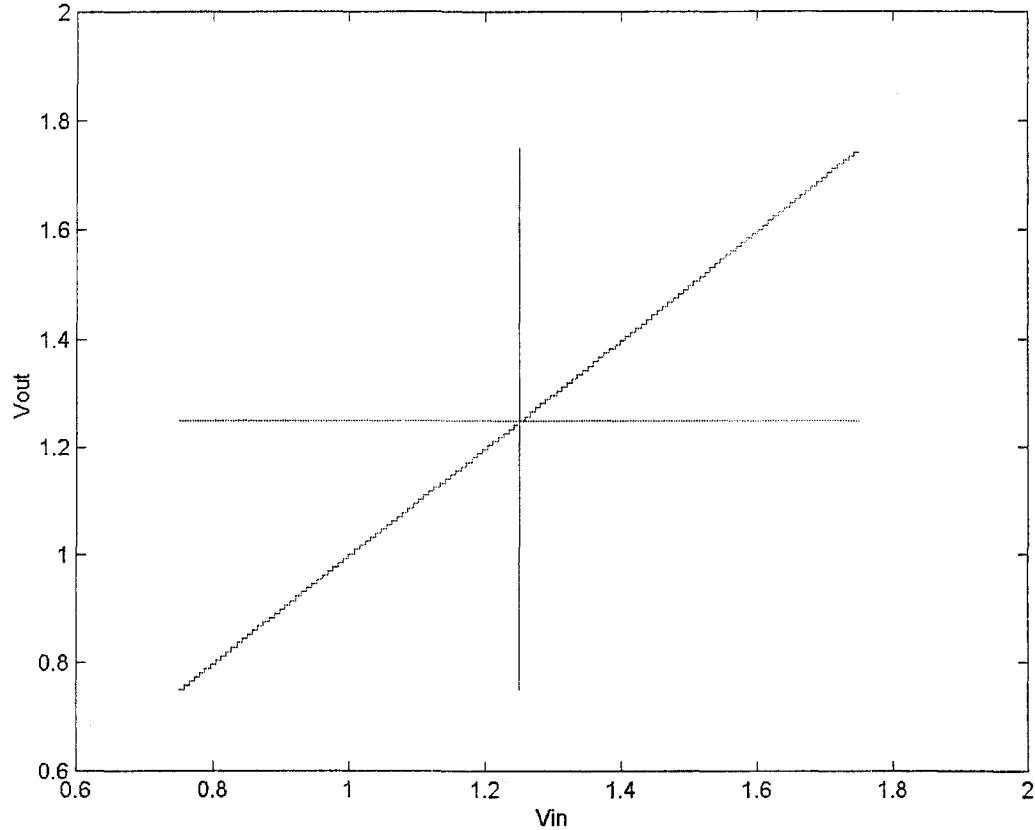


Figure 95 Input/Output characteristic of an 8-bit ADC.

The DNL of the ADC is always measured after constructing the digital equivalent of the input and then referring that value to the input of the ADC and then plot it versus the input value as shown in Figure 95. This way, the digital equivalent output of the input is always normalized so that it also ranges from $-V_{refnd}$ to $+V_{refpd}$. DNL measurement now becomes a straightforward process that can be done by comparing the width of each step of the output with 1LSB, where the difference between the step width and 1LSB value is the DNL for that step.

The goal of this algorithm is to use one linear sweep of the input from $-V_{refnd}$ to $+V_{refpd}$ and be able to measure the gain as well as the DAC values of each stage. This can be done by looking at the output of each stage as the input is being swept. For example, in an 8-bit ADC, the output of S_1 vs. the input of the ADC is shown in Figure 96.a) and the output of S_2 vs. the input of the ADC is shown in Figure 96.b).

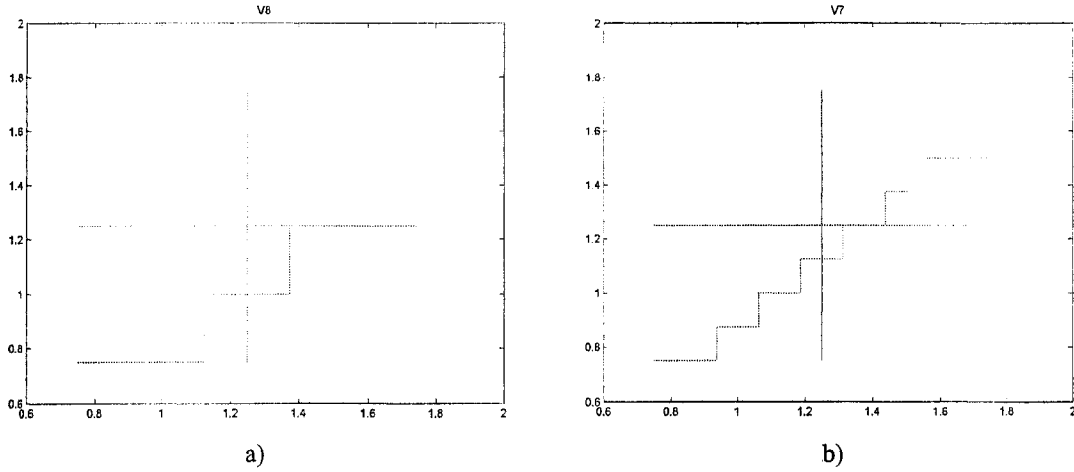


Figure 96 a) Output of S_1 vs. ADC input. b) Output of S_2 vs. ADC input.

This can be done using the following equation.

$$V_{n+1} = A_n \cdot V_n + (A_n - 1) \cdot \left(-B_1 \cdot V_{refpd} - \overline{B_0} \cdot V_{refnd} \right) \quad (36)$$

The algorithm starts by the first stage; the MSB stage, S_1 . Substituting $n=1$ in equation (36) will result in:

$$V_2 = A_1 \cdot V_1 + (A_1 - 1) \cdot \left(-B_1 \cdot V_{refpd} - \overline{B_0} \cdot V_{refnd} \right) \quad (37)$$

As the input is being swept from $-V_{refnd}$, B_0 and B_1 are both 0, and V_{refnd} is the first DAC value of S_1 . Equation (37) now becomes:

$$V_2 = A_1 \cdot V_1 - (A_1 - 1) \cdot V_{refnd} \quad (38)$$

The ADC is configured such that the digital output of the first stage is coming out from the comparators that are connected to the input of the ADC directly. Hence, they carry no information about the DAC or gain values of the first stage of the ADC. This means that Figure 96.a) is not useful. However, Figure 96.b) is the one that contains the DAC and gain information, since it is directly related to equation (38).

If we assume that the first transition of Figure 96.b) happens when the input to the first stage is V_{11} , while the second transition occurs when the input is V_{12} , and using equation (38), we can easily calculate the gain of the first stage. When the input is V_{11} , equation (38) can be written as:

$$V_{21} = A_1 \cdot V_{11} - (A_1 - 1) \cdot V_{refnd} \quad (39)$$

When the input is V_{12} , this will result in the following equation:

$$V_{22} = A_1 \cdot V_{12} - (A_1 - 1) \cdot V_{refnd} \quad (40)$$

V_{21} and V_{22} are the digitized outputs which can be easily deduced. Subtracting equation (39) from equation (40) will result in the following equation:

$$V_{22} - V_{21} = A_1 \cdot (V_{12} - V_{11}) \quad (41)$$

From which the gain of the first stage, A_1 , can be found as:

$$A_1 = \frac{V_{22} - V_{21}}{V_{12} - V_{11}} \quad (42)$$

Any DAC error in the DAC values of the first stage will be common to the first and second transitions of Figure 96.b), so, the subtraction in equation (41) will cancel out DAC errors.

Notice that when the first transition in Figure 96.b) occurs, the input to the second stage, which is the output of the first stage is equal to the first threshold of the second stage. Substituting that in equation (39), the actual value of the first DAC can be found.

Similarly, the value of the second DAC can be found, but using the last transition, rather than the first one.

In general, equation (42) gives the gain of the subADC that is made of the first stage up to stage m , where m is the stage for which the gain is being calculated, i.e.:

$$A_{Gm} = A_1 A_2 A_3 \cdots A_m = \frac{V_{(m+1)2} - V_{(m+1)1}}{V_{m2} - V_{m1}} \quad (43)$$

Where A_{Gm} is the total gain at the output of stage m . To find the actual gain of stage m , equation (36) is used to find A_{Gm} and also to find $A_{G(m-1)}$. Then, the gain of stage m will be:

$$A_m = \frac{A_{Gm}}{A_{G(m-1)}} \quad (44)$$

As an example, consider an 8-bit ADC with the last 7 stages are considered non-ideal. The actual values as well as the values calculated using the above equations for the gain of every stage as well as the DAC values are shown in Table 5. It is very clear from the table that the calculated values are very close to the actual ones.

Table 5 Actual and calculated values of each stage's gain and DAC values.

Stage Num	Actual Gain	Calculated Gain	Actual DAC1	Calculated DAC1	Actual DAC2	Calculated DAC2
2 (LSB)	1.875000	1.818182	-0.375000	-0.388889	0.437500	0.444444
3	2.125000	2.133333	-0.625000	-0.621324	0.562500	0.558824
4	1.968750	1.968750	-0.468750	-0.467742	0.531250	0.530242
5	1.937500	1.936508	-0.453125	-0.453390	0.468750	0.469280
6	2.093750	2.092213	-0.515625	-0.516182	0.507813	0.508208
7	2.035156	2.035260	-0.496094	-0.496097	0.492188	0.492313
8 (MSB)	1.970703	1.971126	-0.505859	-0.505699	0.503906	0.503840

6.6. Multipath Calibration

Different approaches were proposed in literature to correct for errors in a parallel ADC.

6.6.1. Gain error randomization

In this architecture, an extra channel with a FIFO register that has extra memory to generate a randomized channel selector was used. Although channel randomization can help to eliminate tones associated with the input referred offsets of individual channels, chopper stabilization techniques were used to reduce the offset at the input of each channel and thus reducing the power of noise due to input-referred offset.

6.6.2. Channel normalization

Channel normalization [13] was used to linearize the transfer characteristic of each channel using the accuracy-bootstrapping algorithm. The channels were linearized with respect to an ideal characteristic, which is the average of the characteristics of all channels.

6.6.3. Hardware sharing.

In this approach, the resources are shared across multiple channels. Use of a pipeline approach for each of the parallel channels, allows circuitry, such as bias circuits and resistor strings, to be shared over all the channels. Fixed pattern noise effects due to inter-channel mismatches are minimized by appropriate auto zeroing and by the use of a common resistor string DAC for all the channels.

6.6.4. Digital Calibration

6.6.4.1. A Digital Background Calibration Technique

The background calibration is done here by adding a calibration signal to the ADC input and processing both signals simultaneously [9].

The summary of the algorithm is that it forces all the ADC channels in the array to have the same desired gain value (and therefore to match each other). This was achieved by using an adaptive system to calibrate the gain of one ADC as shown in Figure 97. The key blocks were: a pseudorandom number generator (RNG), a 1-bit DAC, the ADC under calibration, a digital multiplier with variable gain determined by the adaptive loop, and a digital accumulator. The sequence generated by the pseudo-RNG is binary and approximately white. It has zero mean and is uncorrelated with the input signal. During calibration, the random number is converted to an analog noise through the 1-bit DAC and is added to the input of the ADC. The same random number is then subtracted at the output, and the difference is taken as the ADC final output. Then ϵ is multiplied by N , scaled by a small negative number ($-\mu_{\text{gain}}$), and accumulated to determine the gain through feedback. In practice, $\mu_{\text{gain}} > 0$; therefore, the feedback is negative.

The sequences ϵ_1 and ϵ_2 are the outputs of the gain calibration system. Each sequence contains the input signal and the offset of the associated ADC, but the gain mismatch terms are eliminated. A variable offset O is added to the gain-corrected output of ADC_2 , and the result is subtracted from the ADC_1 output. The difference is scaled by μ_{offset} and accumulated to determine O . If the step size μ_{offset} is small, the average offset correction converges to a value that makes the average accumulator update equal to zero.

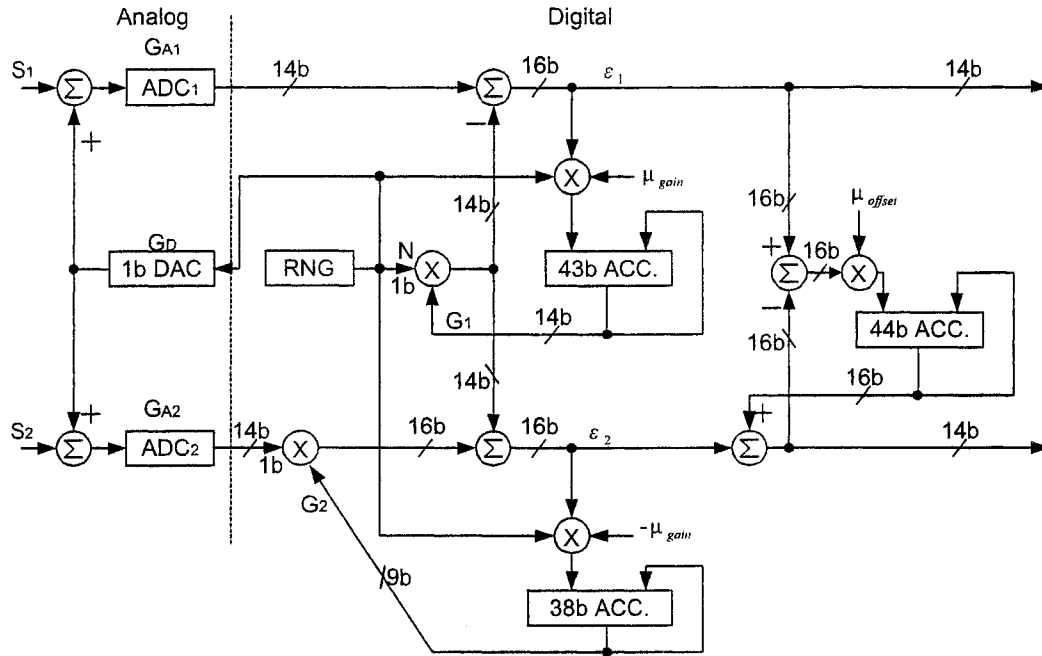


Figure 97 Adaptive digital background calibration system.

After convergence, the average offsets of the interleaved channels are equalized. Again, the random component of the offset arising from noise in the adaptive system can be made arbitrarily small by reducing the step size μ_{offset} .

6.6.4.2. An Analog Background Calibration Technique

Figure 98 shows a block diagram of a time-interleaved ADC system that uses adaptive calibration to overcome gain and offset mismatches [10]. On the left of the diagram, a front-rank sample-and-hold amplifier (SHA) operating at the sample rate of the ADC array is used to eliminate the effect of timing mismatches between the time-interleaved ADC's.

Three high-speed time-interleaved ADC's are used in Figure 98. At any time, two of the three high-speed ADC's operate in a ping-pong mode, allowing a data-conversion rate that is double that of each individual ADC. Meanwhile, the other ADC is selected to be in a calibration mode. In the calibration mode, the selected ADC and the reference ADC are fed identical inputs for many conversions, and the gain and offset of the selected ADC is adjusted to match that of the reference ADC. The gain and offset adjustments are made using a simplified version of the least mean square (LMS) algorithm. Once the calibration cycle is completed, another ADC is selected for calibration and the most recently calibrated ADC replaces it in the ping-pong conversion mode. Each of the three high-speed ADC's is selected sequentially and swapped out for calibration while the other two high-speed ADC's process the input. The rate at which a new ADC is selected for calibration is f_{swap} .

The background calibration allows the offset-correction and gain-correction adjustments to track low-frequency variations (such as $1/f$ noise). In steady state, each of the ADC's has a gain and offset that matches the reference ADC. Therefore, fixed-pattern noise and modulation products from gain and offset mismatches are eliminated.

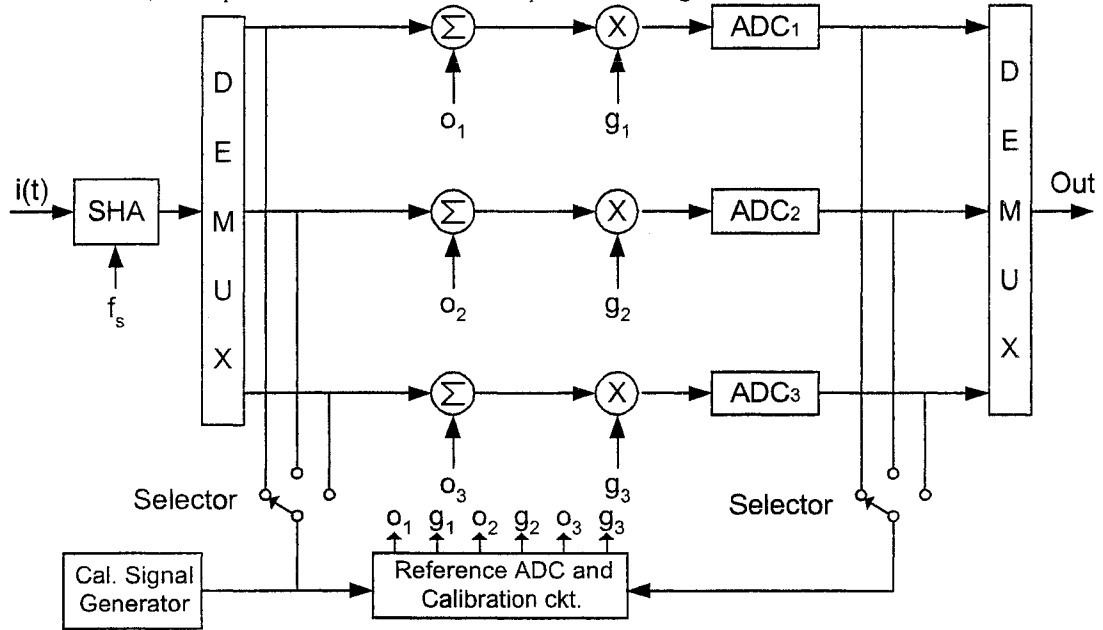


Figure 98 Adaptively calibrated ADC.

This architecture requires $M+1$ time-interleaved ADC's and one reference ADC to increase the conversion rate by a factor of M . This approach becomes more area-efficient as the number of channels increases. A key advantage of this approach is that the ADC array does not have to stop processing the input during calibration because only one of the channels is calibrated at a time. The ADC calibration signal is independent of the input and is supplied by an on-chip signal generator. For the same matching performance, using an independent calibration channel gives a shorter convergence time than when the input and calibration signal are processed together. This topology, which includes a reference ADC, allows for a simplified LMS calibration loop that can be implemented with analog circuits.

References

- [1] E. G. Soenen and R. L. Geiger, "An architecture and an algorithm for fully digital correction of monolithic pipelined ADC's," *IEEE Trans. Circuits Syst. II*, vol. 42, pp. 143-153, Mar. 1995.
- [2] E. Opris, L. D. Lewicki and B. C. Wong, "A single-ended 12-bit 20Msample/s self-calibrating pipeline A/D converter," *IEEE Journal of Solid-State circuits*, vol. 33, No. 12, December 1998, pp. 1989-1903.
- [3] J. M. Ingino, and B. A. Wooley, "A continuously calibrated 12-b, 10-MS/s, 3.3-V A/D converter," *IEEE Journal of Solid-State circuits*, vol. 33, No. 12, December 1998, pp. 1920-1931.
- [4] C. S. G. Conroy, D. W. Cline, and P. R. Gray, "An 8-b 85-MS/s parallel pipeline A/D converter in 1- μ m CMOS," *IEEE J. Solid-State Circuits*, vol. 28, pp. 447-454, Apr. 1993.
- [5] B. S. Song, M. F. Tompsett and K. R. Lakshmikummar, "A 12-bit 1-Msample/s capacitor error-averaging pipelined A/D converter," *IEEE Journal of Solid-State Circuits*, VOL 23, NO. 6, DECEMBER 1988.
- [6] H. S. Lee, D. A. Hodges, and P. R. Gray, "A self-calibrating 15-bit CMOS A/D converter," *IEEE J. Solid-State Circuits*. vol. SC-19, pp. 813-819, Dec. 1984.
- [7] S. H. Lewis, H. S. Fetterman, G. F. Gross, Jr., R. Ramachandran, and T. R. Viswanathan, "A 10-b 20-Msample/s analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol. 27, pp. 351-358, Mar. 1992.
- [8] A. N. Karanicolas, H.-S. Lee, and K. L. Bacrania, "A 15-b 1-Msample/s digitally self-calibrated pipeline ADC," *IEEE J. Solid-State Circuits*, vol.28, pp. 1207-1215, Dec. 1993.
- [9] D. Fu, K. Dyer, S. Lewis, and P. Hurst, "Digital background calibration of a 10-b 40 MS/s parallel pipelined ADC," in *Proc. Int. Solid-State Circuits Conf.*, Feb. 1998, pp. 140-141.
- [10] K. Dyer, D. Fu, S. Lewis, and P. Hurst, "Analog background calibration of a 10-b 40 MS/s parallel pipelined ADC," in *Proc. Int. Solid-State Circuits Conf.*, Feb. 1998, pp. 142-143.
- [11] S. H. Lee and B. S. Song, "A code-error calibrated two-step A/D converter," in *ISSCC Dig. Tech. Papers*, Feb. 1992, pp. 38-39.
- [12] —, "Digital ω -domain calibration of multistep analog-to-digital converters," *IEEE J. Solid-State Circuits*, vol.27, No. 12, Dec. 1992, pp. 1679-1688.
- [13] V. Navin, "An analysis of a digitally self calibrated parallel pipelined analog-to-digital converter," M.Sc. thesis. Iowa State University, 1996.

- [14] M. K. Mayes and S. W. Chin, " Monolithic low-power 16b 1Msample/s self-calibrating pipeline ADC," , in *Proc. Int. Solid-State Circuits Conf.*, 1996, pp. 312-313.

CHAPTER 7. VCO-Based ADCs

7.1. Introduction

ADCs (Analog-to-Digital Converters) are used in order to convert an analog signal to a digital one so that it can be processed by a digital processor. There are many different architectures for ADCs, where each one has its own characteristics. For example, pipeline ADCs are generally suitable for moderate speed in the range of 10-200MHz, with a moderate resolution, in the range of 10-14 bits and they usually consume moderate power.

Flash ADCs are commonly used in high-speed low-resolution applications. Although they are power hungry ADCs, but they can achieve very high speed in the range of 1GHz.

Sigma-Delta based ADCs are usually used in applications where high resolution, in the range of 12-16 bits, is needed. The main limitation of this type of ADCs is its speed. Typical current Sigma-Delta ADCs can run at a clock rate up to 20MHz.

Other types of ADCs include subranging, algorithmic and successive approximation ADCs in addition to the folding and interpolating ADCs.

These architectures have been studied extensively and researched for a long time and they reached the saturation point.

It is always preferable to design an ADC in a digital process, so that it can be integrated with the rest of the digital circuits that made up the digital processor. However, in application where a high resolution ADC is needed, this might be a challenge. High performance ADCs often requires special analog processes, which may have higher power supply requirement. This will usually require the need to design the ADC as a stand-alone chip, which will make it more expensive and more difficult to integrate with the rest of the system.

In this chapter, a new ADC architecture is proposed that can easily be implemented in a digital process and at the same time has outstanding performance compared to the rest of the architectures [1].

7.2. Architecture

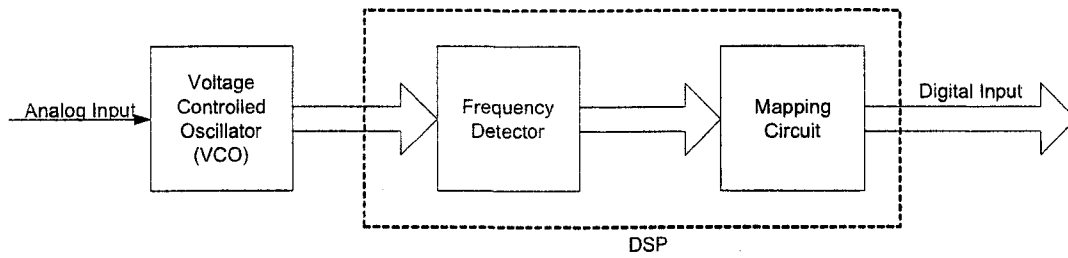


Figure 99 Block diagram of the VCO-based ADC.

The proposed ADC architecture is shown in Figure 99 and uses a voltage-controlled oscillator (VCO) to convert an analog input signal into frequency.

A VCO is a circuit that takes an analog input voltage and generates a sinusoidal signal whose frequency is proportional to the value of the input signal. A VCO is usually characterized by its frequency vs. voltage relationship as shown in Figure 100.

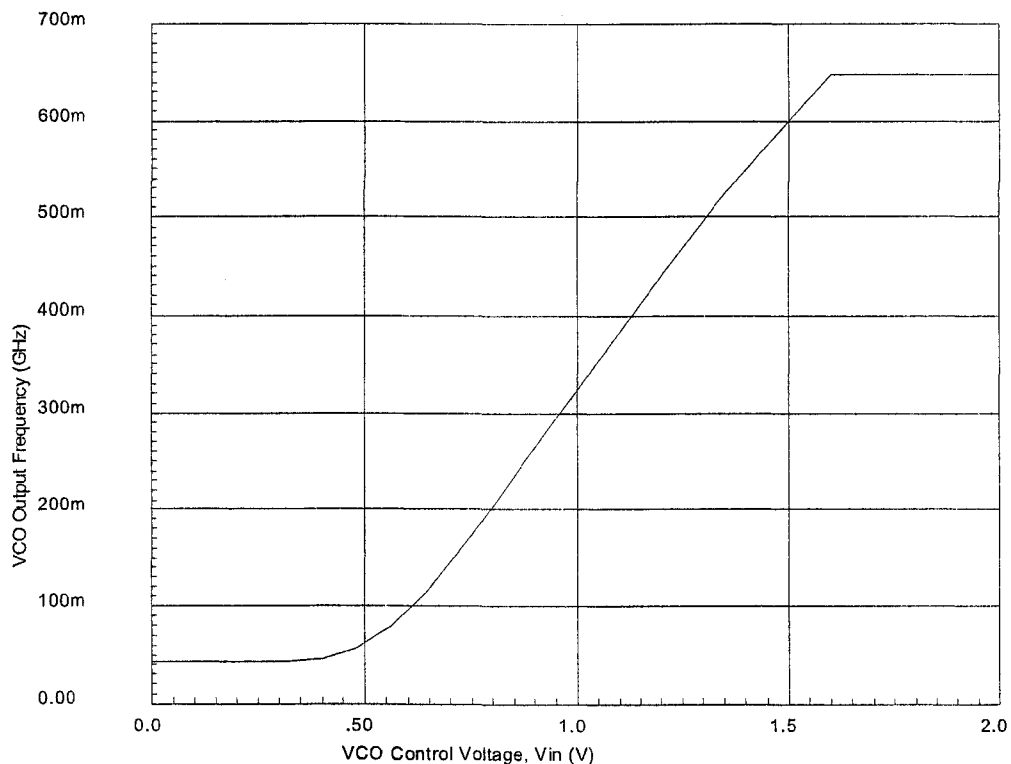


Figure 100 VCO characteristic curve; Frequency vs. voltage.

Figure 100 shows the frequency of a VCO designed in a digital process that has a power supply of 1.8V. The input voltage of the VCO is swept from 0 up to 1.6V.

Figure 101 shows the input and the output of the VCO. The input signal is a sinusoidal one and is shown in the upper portion of the figure. The output of the VCO is shown in the bottom portion of Figure 101 and follows the curve of Figure 100, which indicates that as the input magnitude increases, the frequency of the VCO output increases. As the magnitude of the input signal starts decreasing, the frequency of the VCO output signal will start decreasing as well.

A typical VCO generates multiple outputs that have the same frequency but phase shifted from each other by a constant value. This phase shift is one period time divided by the number of outputs the VCO generate. For example, in a 10-stage VCO, there are 20 outputs, where each output is shifted in time from the next output by one period time divided by 20.

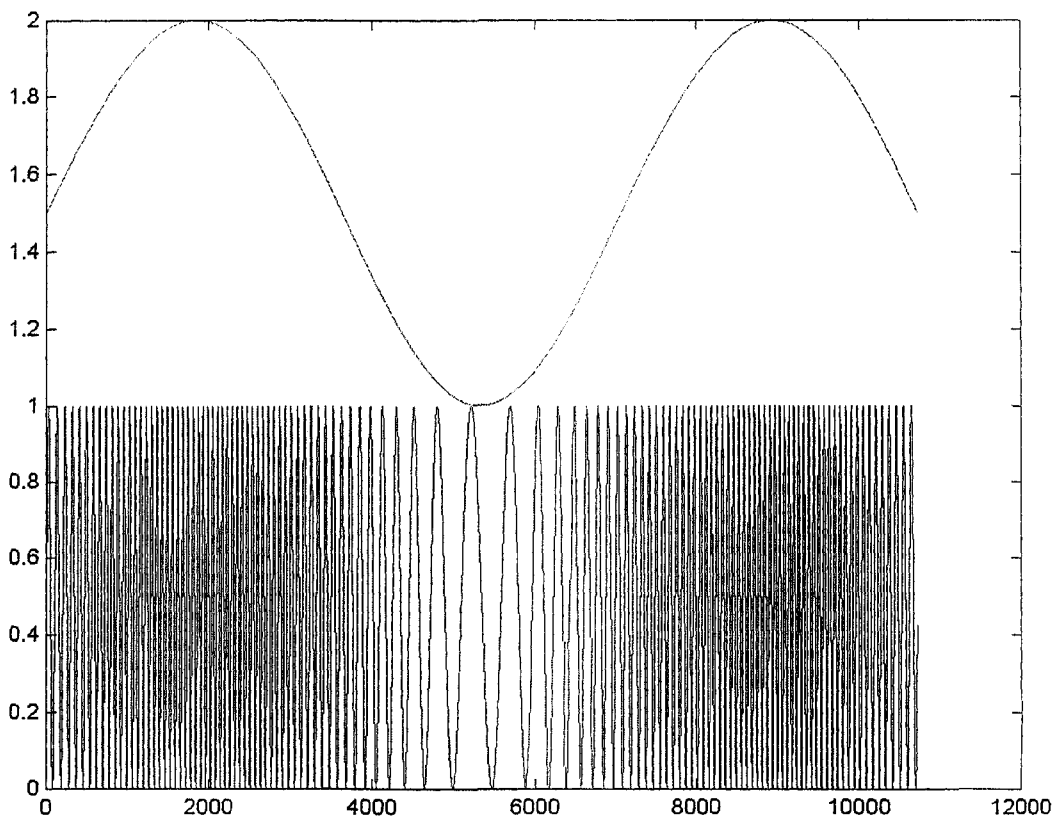


Figure 101 Input and Output of a VCO. The top curve represents a sinusoidal input, while the bottom curve represents the output of the VCO, which is modulated by the input signal.

As shown in Figure 99, the VCO is followed by a frequency detector (FD) that uses the outputs of the VCO to estimate its frequency. The FD is a synchronous one. Its output is synchronized with a clock. A typical implementation of the FD takes a large number of clock cycle to generate an output [2][3][4]. This is due to the method of measuring the frequency or the period of the VCO output signal. Usually, these FDs use a high-speed counter and registers to do the time or frequency detection. The principle of operation is simple, a high-speed

counter will count as long as the output of the VCO is high. Then the number that the counter carries indicates how many clock cycles it took the counter to go through until the output of the VCO is no longer high. Knowing this number as well as the period of the clock that drives the counter will make it easy to measure the frequency of the output signal of the VCO. Other techniques use ADCs to measure the actual period of the signal coming out of the VCO.

7.3. Frequency Detector Circuit

7.3.1. Introduction and overview

Frequency Detector circuits are usually slow running in the couple of MHz ranges. Most of the circuits in literature depend on the use of a counter to measure the frequency of the incoming clock or on a single shot circuit.

This circuit describes a very fast frequency detector (FD) circuit that runs in the couple of hundreds of MHz.

The current implementation of the circuit has a reference clock with a known frequency, and a set of clocks whose frequency, F is unknown. All the clocks inside this set have the same frequency but phase shifted from each others by T_s , where T_s is equivalent to $\frac{T}{2 \cdot N}$, where T is the period of the unknown clock and $2 \cdot N$ is the number of clocks in each period, T .

In case there is only one clock whose frequency is unknown, a DLL (Delay Locked Loop) can be used to generate these clocks.

7.3.2. Proposed solution

The input clock set that has the unknown frequency is called CLK<2N-1:0>. CLK<1> is shifted from CLK<0> by τ , CLK<2> is shifted in phase by τ from CLK<1>, and so on.

The FD samples the first N clocks, CLK<N-1:0> at the rising edge of the reference clock and save them. On the next rising edge of REFCLK, the same set of clocks is sampled again. The new sampled clocks are compared with the previous sampled clocks.

Each clock of the set is sampled by a Master-Slave FlipFlop as shown in Figure 102. Figure 102 shows an FD that has $N = 4$. There are two sets in the FD. At the rising edge of REFCLK, the first set samples the input clocks, CK0-CK3. At the falling edge of REFCLK, the second set samples the output of the first set to save it. At the next rising edge of REFCLK, the input clocks are sampled again, and the output of the first set is compared with the output of the second set.

The basic principle behind the FD is that if the frequency of the input set, F , is the same as the frequency of REFCLK, F_R , the output of DFF1 should match the output of DFF2, the output of DFF3 should

match the output of DFF4, and so on, all the outputs of the DFFs from the second set should match the outputs from the first set.

However, F is not the same as F_R , at the second rising edge of REFCLK, some of the outputs of the first set of the DFFs will be different than these of the outputs of the second set of the DFFs. The number of these DFFs that will be different depends on the difference between F and F_R .

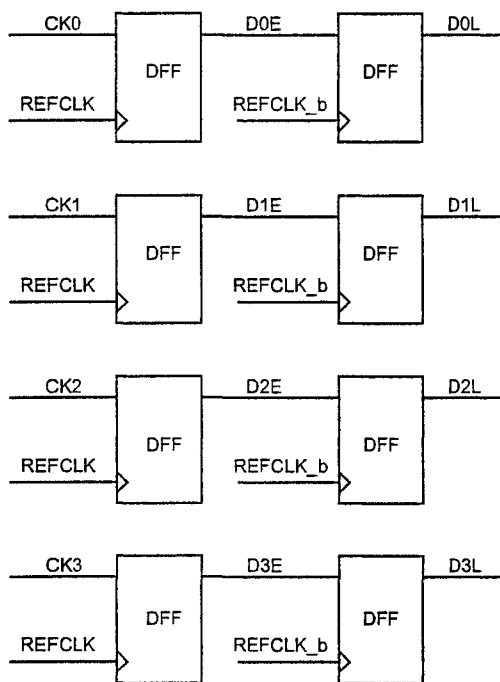


Figure 102 Frequency Detector with $N = 4$.

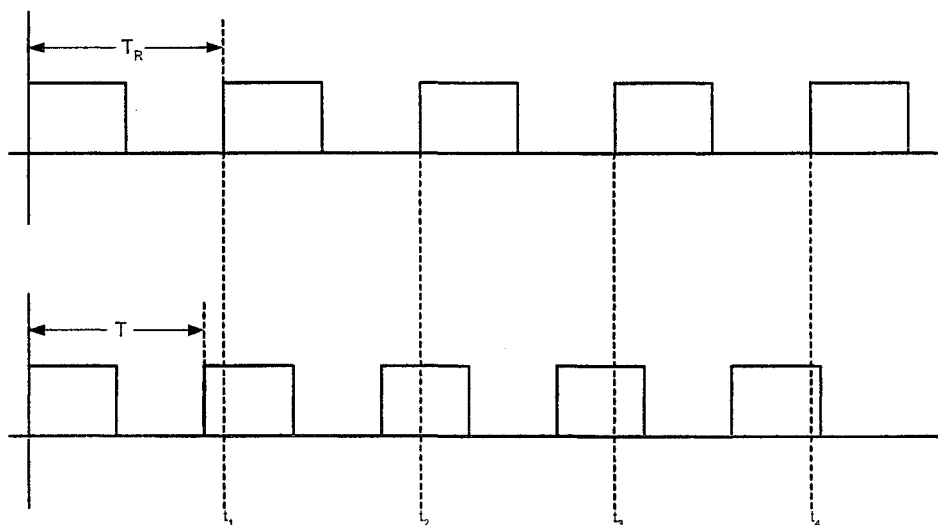


Figure 103 CK0 and REFCLK.

To illustrate the principle, let T be the period of one of the unknown clocks, say CK0, so $T = 1/F$, and T_R is the period of REFCLK, so $T_R = 1/F_R$. Lets assume that the two clocks; CK0 and REFCLK start at the same time as shown in Figure 103.

The first sampling happens at t_1 , while the second sampling takes place at t_2 , and so on. As shown in Figure 103, $t_2 = t_1 + T_R$, $t_3 = t_2 + T_R = t_1 + 2 \cdot T_R$, ... and so on.

If $T < T_R$, then at t_1 , CK0 leads REFCLK by $\tau = T_R - T$. If $T > T_R$, then CK0 lags REFCLK by $\tau = T - T_R$.

The relationship between the clock set and REFCLK as well as among all the signals of the clock set is shown in Figure 104.

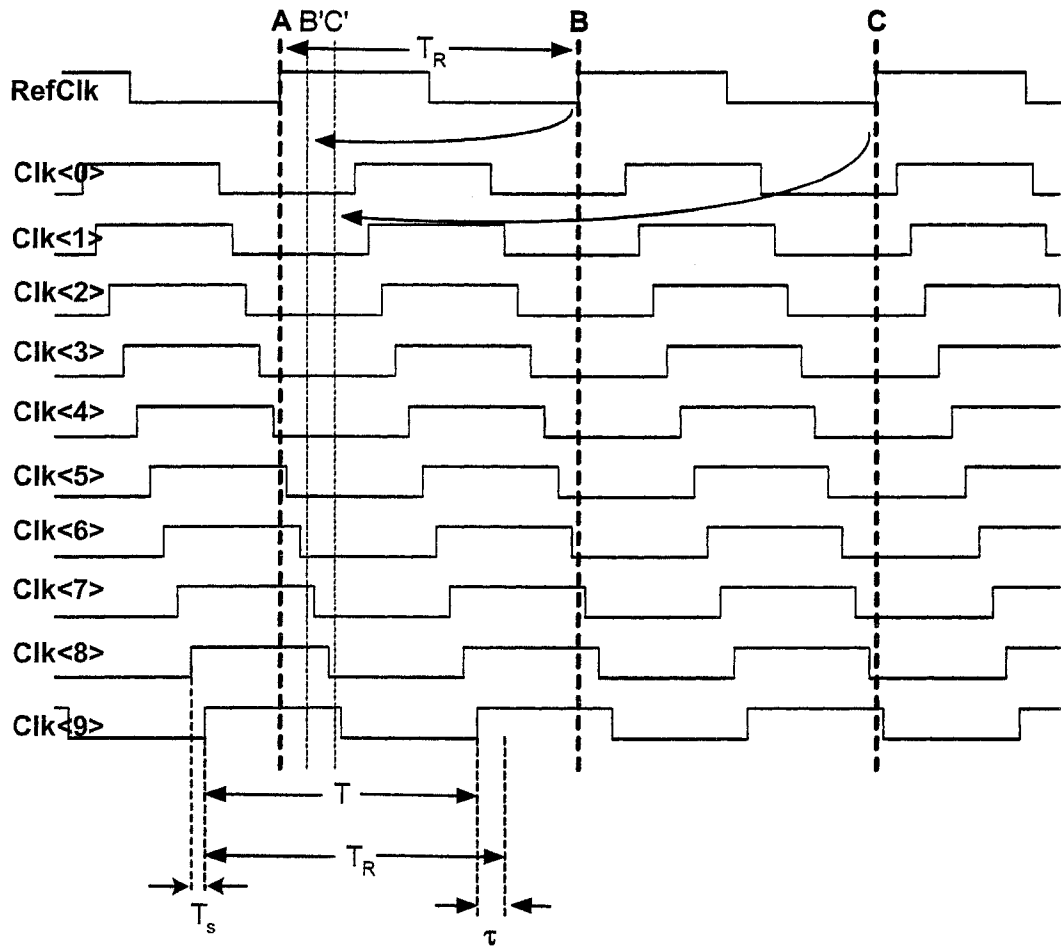


Figure 104 Relationship among all the clocks for N=10.

As shown in Figure 104, CK0, CK1, ..., CK(N-1), will have the same period, but shifted in time by T_s from each other. Mathematically,

$$REFCLK = F_R(t) = \sum_{m=-\infty}^{\infty} [u(t - m \cdot T_R) - u(t - (m+1) \cdot T_R)] \quad (1)$$

Where $u(t)$ is the unit step function shown in Figure 105. Similarly, the rest of the clocks can be written as:

$$CK0(t) = F_0(t) = \sum_{n=-\infty}^{\infty} [u(t - n \cdot T) - u(t - (n+1) \cdot T)] \quad (2)$$

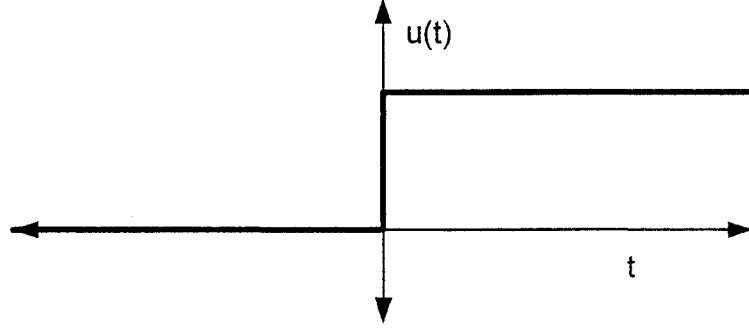


Figure 105 Step function, $u(t)$.

$$CK1(t) = F_1(t) = F_0(t - T_s) = \sum_{n=-\infty}^{\infty} [u(t - T_s - n \cdot T) - u(t - T_s - (n+1) \cdot T)] \quad (3)$$

$$CK2(t) = F_2(t) = F_1(t - T_s) = F_0(t - 2 \cdot T_s) = \sum_{n=-\infty}^{\infty} [u(t - 2 \cdot T_s - n \cdot T) - u(t - 2 \cdot T_s - (n+1) \cdot T)] \quad (4)$$

$$\text{Where } T_s = \frac{T}{2 \cdot N}.$$

Figure 104 shows an example for $N = 10$. $CK10$ is the complement of $CK0$ and is not shown here. Sampling at t_A will result in $F_0(t_A) = F_1(t_A) = F_2(t_A) = F_3(t_A) = F_4(t_A) = 0$, while $F_5(t_A) = F_6(t_A) = F_7(t_A) = F_8(t_A) = F_9(t_A) = 1$.

At t_B , $F_0(t_B) = F_1(t_B) = F_2(t_B) = F_3(t_B) = F_4(t_B) = F_5(t_B) = F_6(t_B) = 0$, while $F_7(t_B) = F_8(t_B) = F_9(t_B) = 1$. As can be seen, two clocks have changed their status from 1 at t_A to 0 at t_B . If the difference between T and T_R is a little bit bigger, more than 2 clocks will change their status.

Mathematically, sampling the clocks at t_A will result in the following equations:

$$CK0(t_A) = F_0(t_A) \quad (5)$$

$$CK1(t_A) = F_1(t_A) = F_0(t_A - T_s) \quad (6)$$

$$CK2(t_A) = F_2(t_A) = F_1(t_A - T_s) = F_0(t_A - 2 \cdot T_s) \quad (7)$$

Sampling the clocks at t_B will result in the following equations:

$$CK0(t_B) = F_0(t_B) = F_0(t_A + T_R) \quad (8)$$

$$CK1(t_B) = F_1(t_B) = F_0(t_B - T_s) = F_0(t_A + T_R - T_s) \quad (9)$$

$$CK2(t_B) = F_2(t_B) = F_1(t_B - T_s) = F_0(t_B - 2 \cdot T_s) = F_0(t_A + T_R - 2 \cdot T_s) \quad (10)$$

Assuming that $T_R - T = \tau$, Equations (8)–(10) can be rewritten as:

$$CK0(t_B) = F_0(t_A + T_R) = F_0(t_A + T + \tau) \quad (11)$$

$$CK1(t_B) = F_1(t_A + T_R) = F_1(t_A + T + \tau) \quad (12)$$

$$CK2(t_B) = F_2(t_B) = F_2(t_A + T_R) = F_2(t_A + T + \tau) \quad (13)$$

However, all the clocks are periodic, which means

$$F_i(t_A + T) = F_i(t_A) \quad \forall i: 1, N \quad (14)$$

Thus:

$$CK0(t_B) = F_0(t_A + \tau) \quad (15)$$

$$CK1(t_B) = F_1(t_A + \tau) \quad (16)$$

$$CK2(t_B) = F_2(t_A + \tau) \quad (17)$$

From equations (15)–(17), one can conclude that sampling at t_B is the same as sampling the clocks at $t_A + \tau$.

Looking at Figure 104, sampling at B is the same as sampling at B' , sampling at C is the same as sampling at C' , also, the difference between B' and A is τ , and the difference between C' and A is $2 \cdot \tau$.

If $\tau = 0$, then at every sampling time; A , B or C in Figure 104, the logical values of the clocks will always be the same.

Equations (15) – (17) are very important because they give a better way of looking at the operation of the FD. Instead of looking at the clocks at every rising edge of REFCLK, we can now freeze the clocks and look at them with τ increments as shown in Figure 106. So, if we look at the clocks at time A and then look at them again at B , we would find that clocks <5> and <6> have changed their values from A to B as shown in Figure 107.a). From B to C , clocks <7> and <8> will change value as shown in Figure 107.b).

As the value of τ gets bigger, the number of clocks that change status will increase when moving from time A to time B , or equivalently, when moving from time A to time B' .

In order to detect a difference in the status of the sampled clocks, the outputs of the two sets in Figure 102 are taken to XOR circuits, as shown in Figure 108.

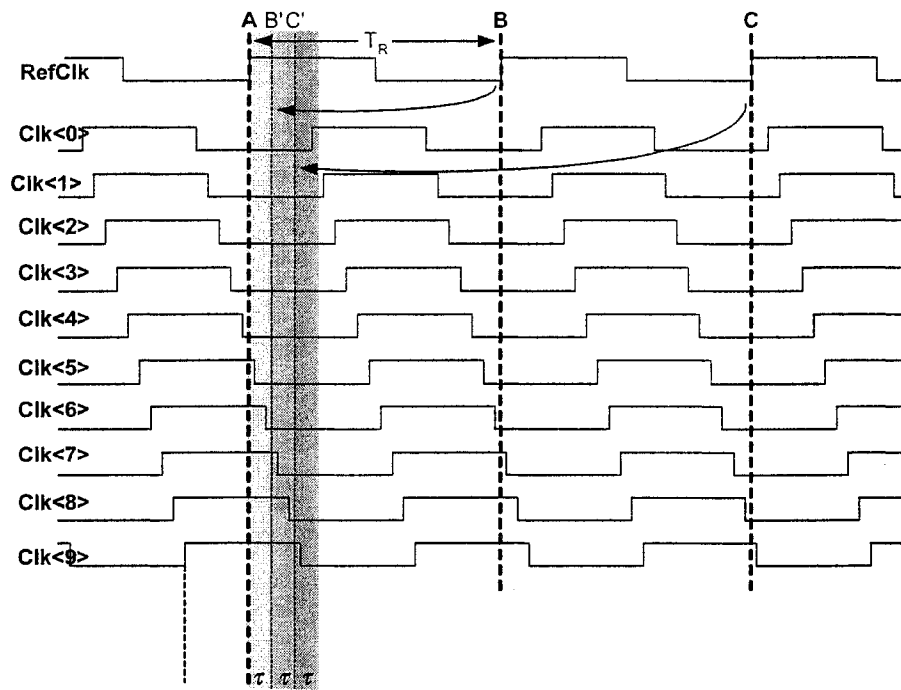


Figure 106 Looking at the clocks with a window of size τ gives a better view of the operation of the FD.

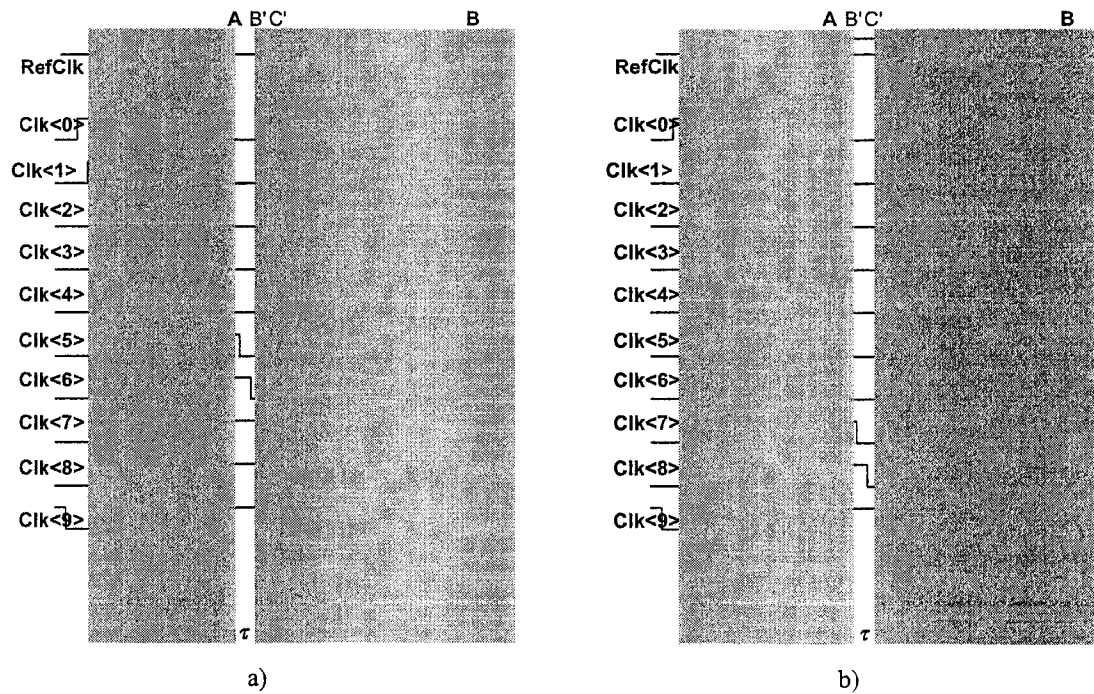


Figure 107 Illustration of the overlaying the window on top of the clocks.

If the output of an XOR goes High, this means that the corresponding clock has changed its status between the two clocks of REFCLK.

If $|\tau| = T_s$, then in each time window, *only* one of the clocks will change its value. If $|\tau| = 2 \cdot T_s$, then in each time window, *only* two of the clocks will change their values. In general, if $|\tau| = a \cdot T_s$, where a is a constant, then in each time window, a clocks will change their values.

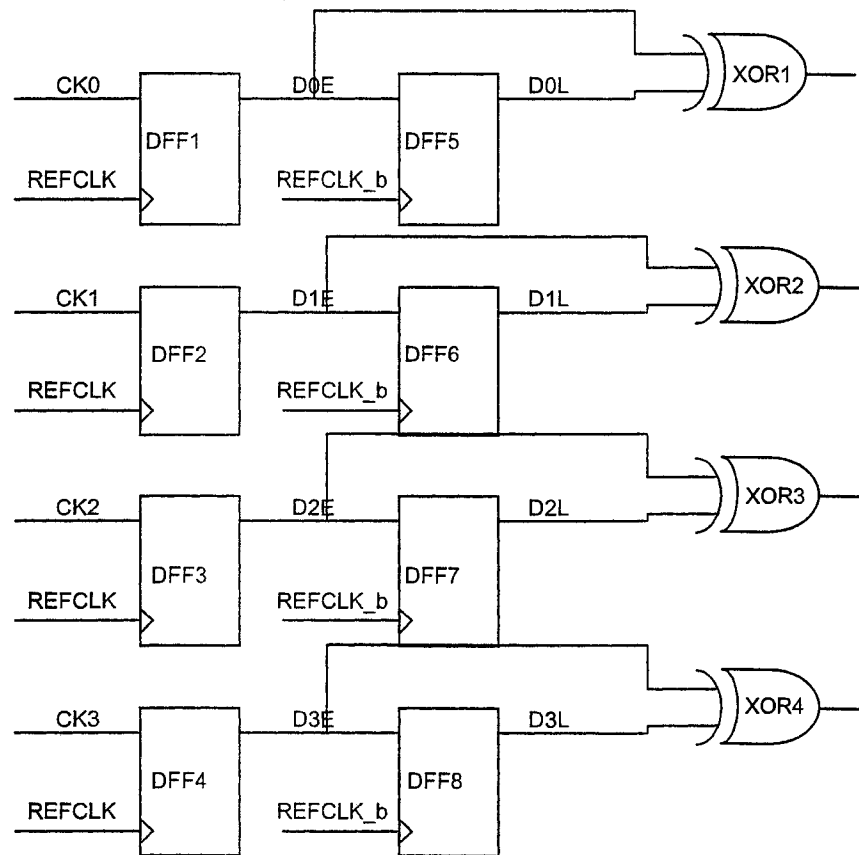


Figure 108 Frequency Detector circuit.

Because of the digitization process that is taken place at every edge of the REFCLK as well as the accumulation of the phase from one clock cycle of REFCLK to the next, if $a < 1$, it will require multiple clocks of REFCLK to detect the a changing clock. This number of clock cycles of REFCLK depends on the value of a . For example, if $a = 0.5$, then one of the clocks will change its value every other cycle of REFCLK, if $a = 0.25$, one clock will change its value every 4 cycles of REFCLK.

In general, if we add the number of clocks that change their values every cycle of REFCLK over M cycles, we will find that number to be $a_T = a \cdot M$, i.e., the total number of clocks that change their values of M cycles of REFCLK will be $a_1 + a_2 + a_3 + \dots + a_M$ where the subscript indicates the cycle number of REFCLK.

However, $a_1 = a_2 = \dots = a_M = a$, hence $a_T = a \cdot M$. This is specially important when $a < 1$, because we need more than one clock cycle of REFCLK to detect a clock change.

So, if M clock cycles are used to find a , then:

$$a = \frac{a_T}{M} \quad (18)$$

Where a_T is the addition of the number of clock changing their states in every clock cycle of REFCLK for M clock cycle of REFCLK.

The above equations show the relationship between the number of clocks that change value between two consecutive clock cycles of REFCLK. It was shown how this number is related to T_s by the following equation:

$$\tau = a \cdot T_s \quad (19)$$

However, T_s is a function of the period of the clocks, which is not known. Note, however, that:

$$T_R = T + \tau \quad (20)$$

Substituting equation (19) in equation (20) will result in:

$$T_R = T + a \cdot T_s \quad (21)$$

But,

$$T_s = \frac{T}{2 \cdot N} \quad (22)$$

Where N is the number of clocks in each half a period of the VCO clocks.

Substituting (22) in (21) will result in:

$$T_R = T + \frac{a \cdot T}{2 \cdot N} = \frac{a + 2 \cdot N}{2 \cdot N} \cdot T \quad (23)$$

Hence,

$$T = \left[\frac{2 \cdot N}{2 \cdot N + a} \right] \cdot T_R \quad (24)$$

Since T_R , a and N are known, then T , the unknown period of the clocks, can be found using equation (24).

7.3.3. Implementation

The implementation of this FD is simple and straightforward. The input to the FD is a set of clocks and a reference clock, REFCLK, as shown in Figure 109. The FD is made mainly from two sets of DFFs. The first, S_1 , samples the clock at the rising edge of REFCLK, while the second set samples the output of the first set at the falling edge of REFCLK. At the next rising edge of the REFCLK, the clocks are sampled again and

compared with those sampled at the falling edge of REFCLK in the previous cycle. This comparison is performed using XOR gates. If the two values are different, the output of the XOR will be High for half a cycle. The output of the XOR circuits will then be taken to an adder to generate the number of clocks that have changed their state during one clock cycle.

Note that in one cycle, the maximum number generated by the adder will be $a = N$, (recall that N clocks are used in this FD). This is true because this will occur when tall the XOR outputs are one. If $(T_R - T) > N \cdot T_s$, then the output of the XOR will not be correct.

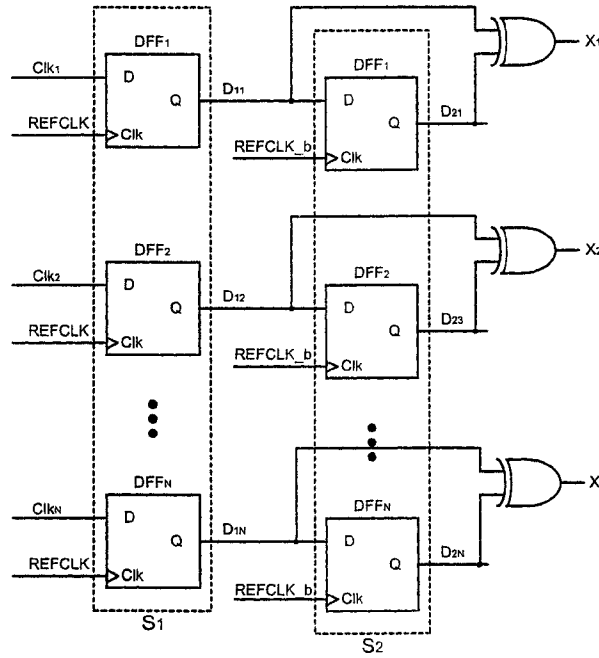


Figure 109. Frequency detector implementation using 2 sets of DFF banks.

Thus, the range of frequency this circuit can correct is:

$$T = \left[\frac{2 \cdot N}{2 \cdot N \pm N} \right] \cdot T_R \tag{25}$$

$$\frac{2}{3} \cdot T_R \leq T \leq 2 \cdot T_R \tag{26}$$

Or equivalently:

$$0.5 \cdot f_{REFCLK} \leq f_{clock} \leq 1.5 \cdot f_{REFCLK} \tag{27}$$

Equation (27) defines the range of operation of the FD for which its output, which is the number of “1” at the output of the XOR circuits, is correct.

To increase the range of operation of the frequency detector shown in Figure 109, the circuit can be modified as shown in Figure 110.

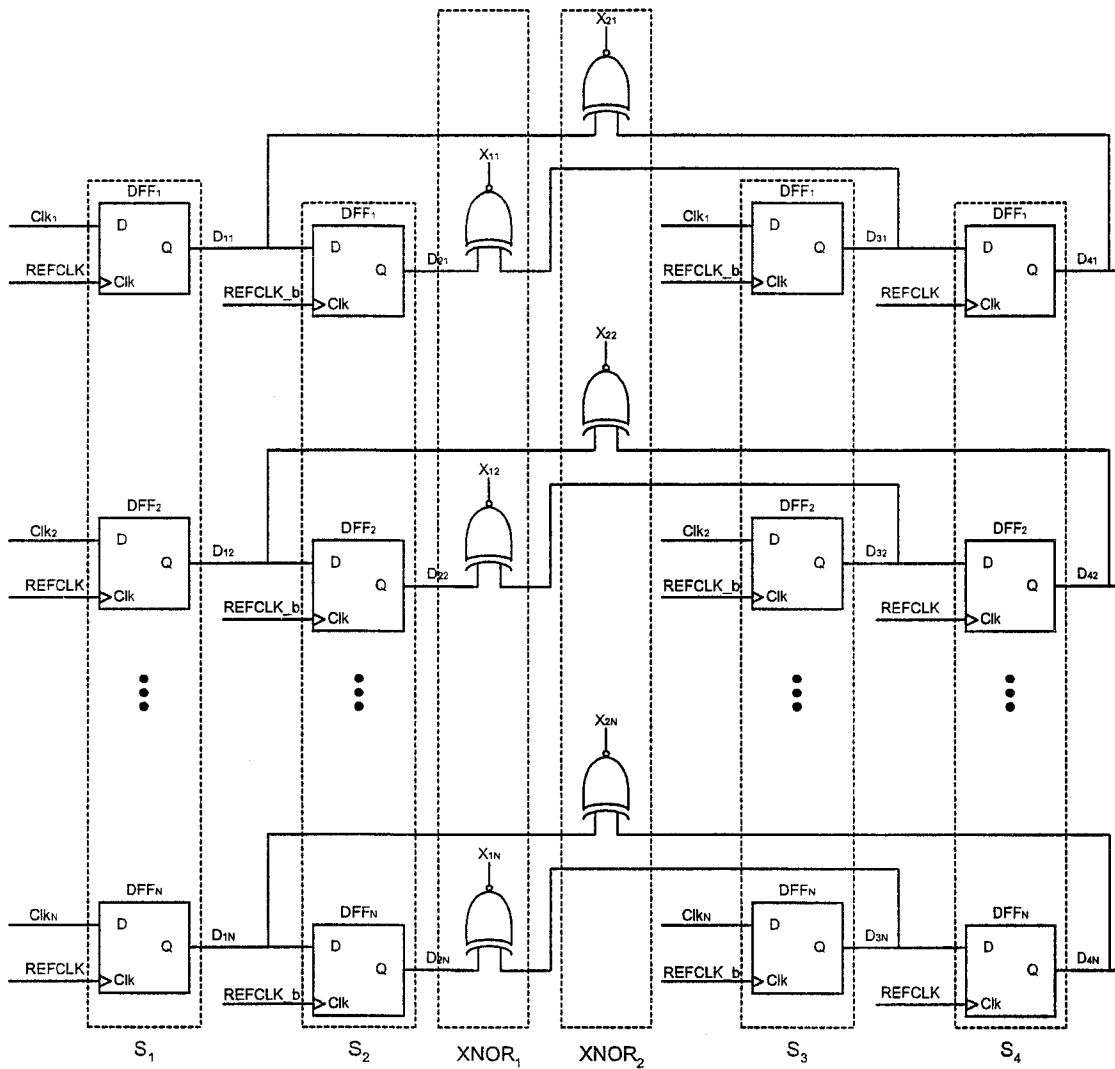


Figure 110. Frequency Detector implementation using 4 sets of DFF banks.

The new circuit shown in Figure 110 consists of 4 sets of DFFs; S_1 , S_2 , S_3 and S_4 and two sets of XNOR circuits; $XNOR_1$ and $XNOR_2$.

In this configuration, the inputs of S_1 and S_2 are connected to the clocks directly, however, S_1 samples the clocks at the rising edge of REFCLK, while S_3 samples the clock at the falling edge of REFCLK. S_2 samples the outputs of S_1 at the falling edge of REFCLK, while S_4 samples the outputs of S_3 at the rising edge of REFCLK.

The idea of the circuit shown in Figure 110 is that the measurement of the number of clocks that changed their status will be done every half a cycle, rather than every full cycle of REFCLK, as the circuit of Figure 109 does.

To illustrate the operation of the new FD, consider that t_0 , t_1 , t_2 , and t_3 , t_4 are the times for the first rising edge of REFCLK, the first falling edge of REFCLK, the second rising edge of REFCLK, and the second falling edge of REFCLK, consecutively as shown in Figure 111.

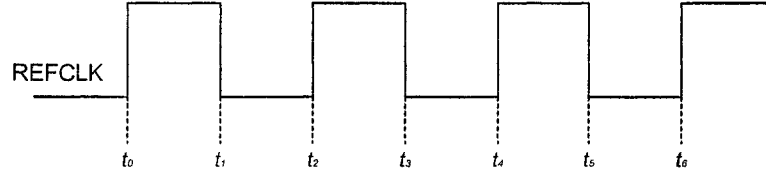


Figure 111 REFCLK sampling times.

S_1 samples the N clocks at t_0 and t_2 while S_3 samples the N clocks at t_1 and t_3 . Similarly, S_2 samples the outputs of S_1 at t_1 and t_3 while S_4 samples the outputs of S_3 at t_2 and t_4 .

The idea behind the new FD is that the measurement is performed every half a cycle rather a full cycle as done by the circuit of Figure 109. If $T = T_R$, then we should expect that what is sampled at the rising edge of REFCLK to be the complement of what is being sampled on the falling edge of REFCLK. That is why an XNOR circuit is used in the FD of Figure 110 while an XOR is used in the circuit of Figure 109.

The maximum number of clocks that can change during one cycle of REFCLK in the new FD will be twice that generated by the FD of Figure 109.

Hence,

$$T = \left[\frac{2 \cdot N}{2 \cdot N \pm 2 \cdot N} \right] \cdot T_R \quad (28)$$

$$\frac{1}{2} \cdot T_R \leq T \leq \infty \quad (29)$$

Or equivalently:

$$0.0 \leq f_{clock} \leq 2 \cdot f_{REFCLK} \quad (30)$$

7.3.4. Derivation of Maximum Error in the FD Measurement

The FD measures the frequency of the clocks by measuring the number of T_s intervals that passed between two consecutive cycles of REFCLK.

If the difference is less than one T_s , then one T_s interval will not be noticed in one cycle, and this why we usually take M cycles to do the measurement.

Over M cycles of REFCLK, if the difference between T and T_R is less than T_s/M , it will not be detected by the FD.

This means that the maximum undetectable error will be T_s/M ,

However, T_s is dependent on the unknown period of the clocks. In this case, it is better to normalize the error with respect to the unknown period. Hence,

$$MaxError = e_{\max} = \frac{\frac{T_s}{M}}{T} \quad (31)$$

Using equation (22),

$$e_{\max} = \frac{\frac{T/2 \cdot N}{M}}{T} = \frac{1}{2 \cdot N \cdot M} \quad (32)$$

So, to reduce the maximum error, either N or M or both are increased. Increasing N means more phases in one period. Since these phases are generated by a VCO, this will require more stages of the VCO to be used. This will require more area and power. Increasing M means taking more clock cycles of REFCLK, which means slower operation of the FD.

7.3.5. Example

In this example, five different values of the frequency of the clocks are investigated in order to show the operation of the FD as well as to validate equation (27). Those five values include the following:

1. $f > 1.5 \cdot f_{REFCLK}$.
2. $f < 0.5 \cdot f_{REFCLK}$.
3. $f_{REFCLK} < f < 1.5 \cdot f_{REFCLK}$.
4. $0.5 \cdot f_{REFCLK} < f < f_{REFCLK}$.
5. $f \approx f_{REFCLK}$.

In all cases, the frequency of REFCLK, f_{REFCLK} , is set to 250MHz. This is equivalent to having a period of 4ns. The number of REFCLK clock cycles used in all of the cases is $M = 10$ and the number of stages in the VCO, N , is 10. Each of the figures below shows the output of the XOR circuits shown in Figure 108, so, if one cycle is considered, the outputs of the XOR circuits will indicate the number of clocks that changed their values since the previous cycle. This number is printed on the top portion of each figure.

The first case is shown in Figure 112. The exact value of the frequency of the clocks is 450MHz, which is more than $1.5 \cdot f_{REFCLK}$. According to (18), the average number of clocks that changed their status in one clock will be:

$$a = \frac{5 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4}{10} = 4.1 \quad (33)$$

Using equation (25),

$$T = \left[\frac{20}{20 + 4.1} \right] \cdot 4n = 3.32ns \quad (34)$$

Or equivalently,

$$f = 301MHz \tag{35}$$

Which is not correct.

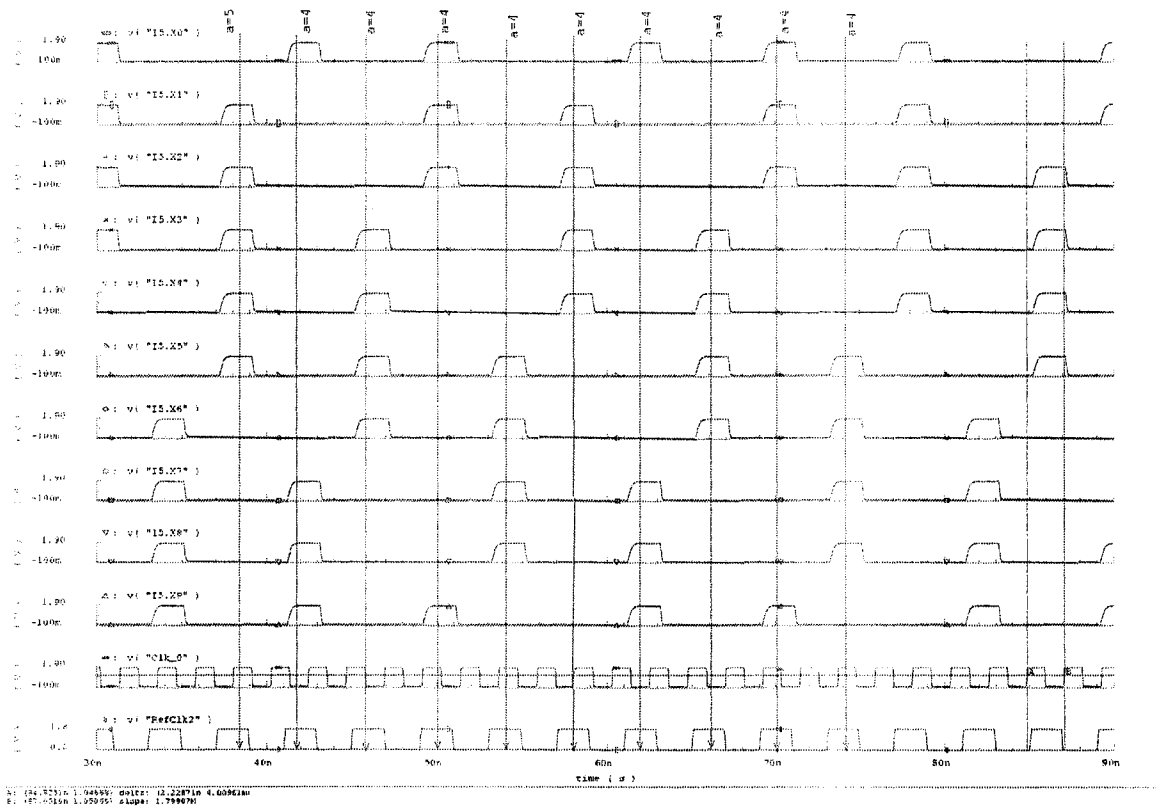


Figure 112 $f = 450MHz$.

The second case is shown in Figure 113, where the actual frequency is 109.26MHz, which is less than $0.5 \cdot f_{REFCLK}$.

From the figure, the measured frequency is:

$$f = \frac{20 - 7.3}{20} \cdot 250 MHz = 158.75MHz \tag{36}$$

We can also see that the measured frequency is far from being correct.

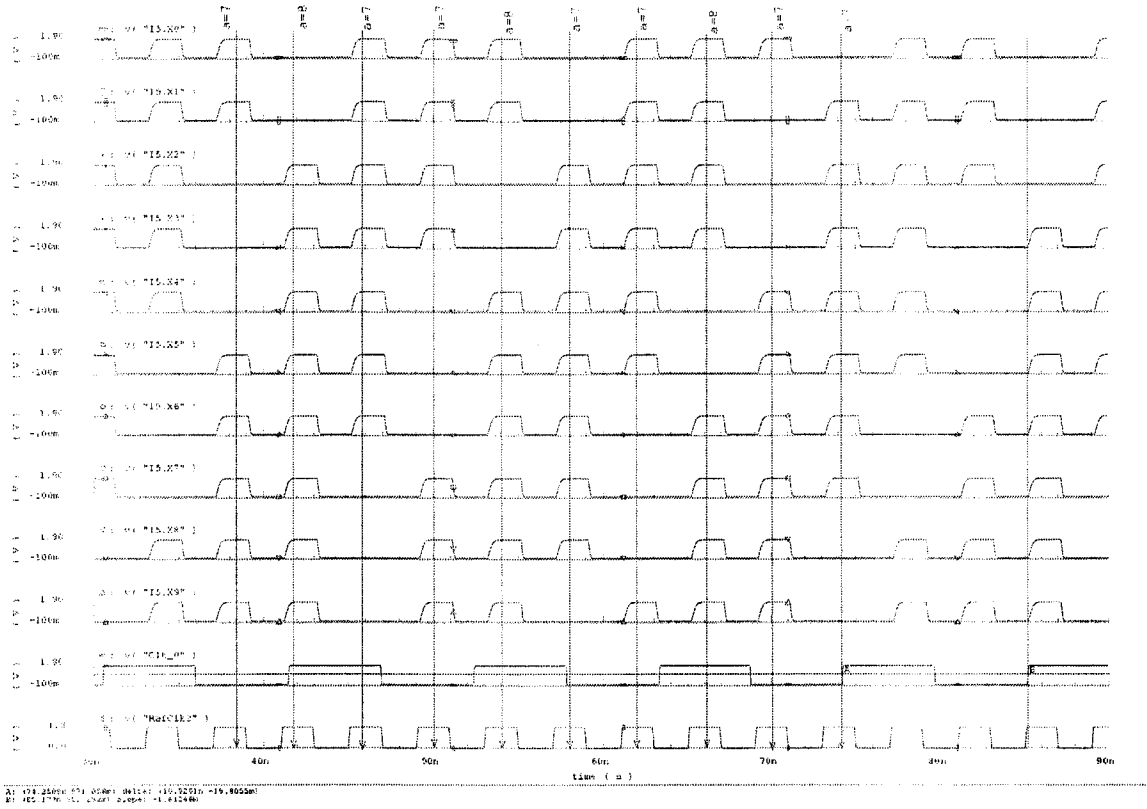


Figure 113 $f = 109.26\text{MHz}$

In the third case, the unknown frequency is within the range of correct operation for the FD. The exact value of the frequency, f , is 314.59MHz and the clocks are shown in Figure 114.

From the figure, the measured frequency is:

$$f = \frac{20 + 5.1}{20} \cdot 250 \text{ MHz} = 313.75\text{MHz} \quad (37)$$

The error in the measurement is equal to:

$$\text{error} = e = \frac{314.59 - 313.75}{314.59} = 2.67 \times 10^{-3} \quad (38)$$

From equation (32), the maximum error is:

$$e_{\max} = \frac{1}{200} = 5 \times 10^{-3} \quad (39)$$

Comparing equation (38) with equation (39), one can clearly see that the actual error is less than the maximum error.

The clock waveforms for the fourth case are shown in Figure 115. The exact value of the unknown frequency is 190.16MHz.

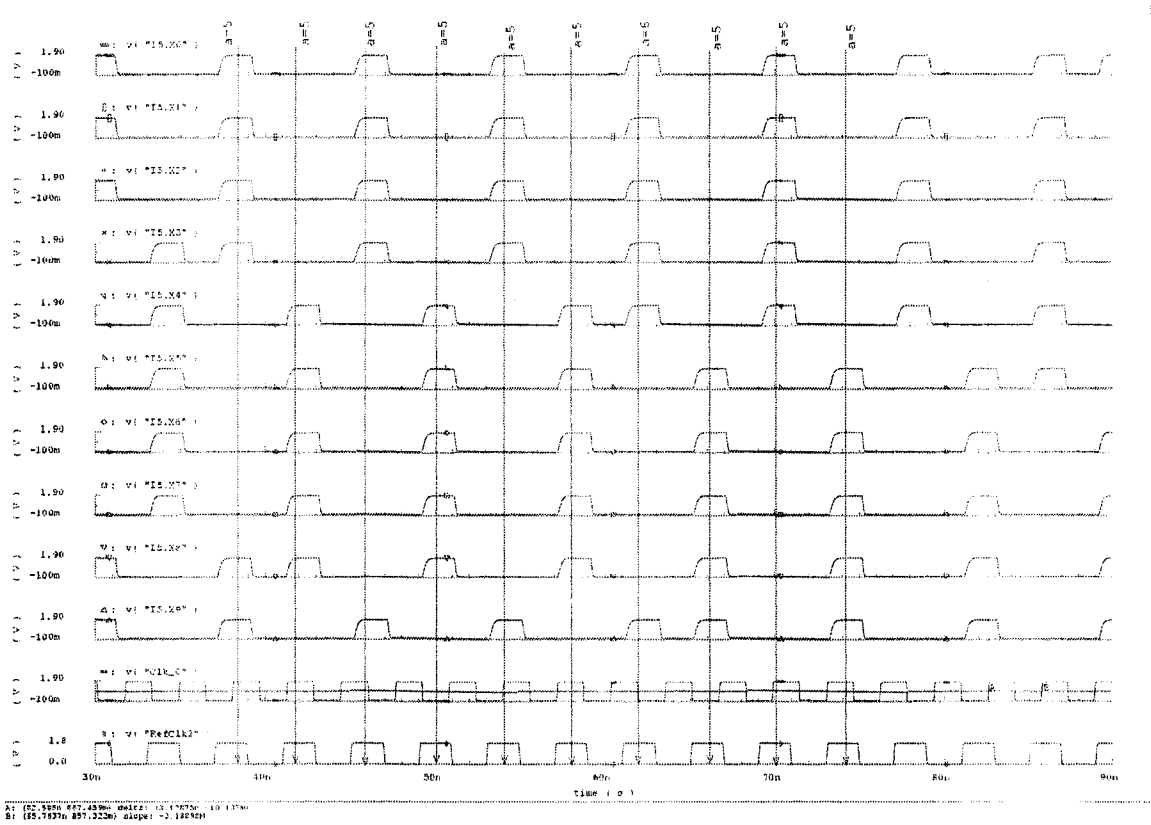


Figure 114 $f = 314.59\text{MHz}$ which is less than $1.5 \cdot 250\text{MHz}$

From the figure, the measured frequency is:

$$f = \frac{20 - 4.8}{20} \cdot 250 \text{ MHz} = 190 \text{ MHz} \quad (40)$$

The error in the measurement is equal to:

$$\text{error} = e = \frac{190.12 - 190}{190.12} = 6.31 \times 10^{-4} \quad (41)$$

Which is much less than the maximum error.

The last case is presented here to show that as the two frequencies get close to each other, less and less “1”s will be coming out of the XOR circuits. In this case, the unknown frequency is chosen to be very close to f_{REFCLK} . The exact value of the unknown frequency is 245.27MHz. Looking at Figure 116 we can see the outputs of the XOR are “0” for most of the time.

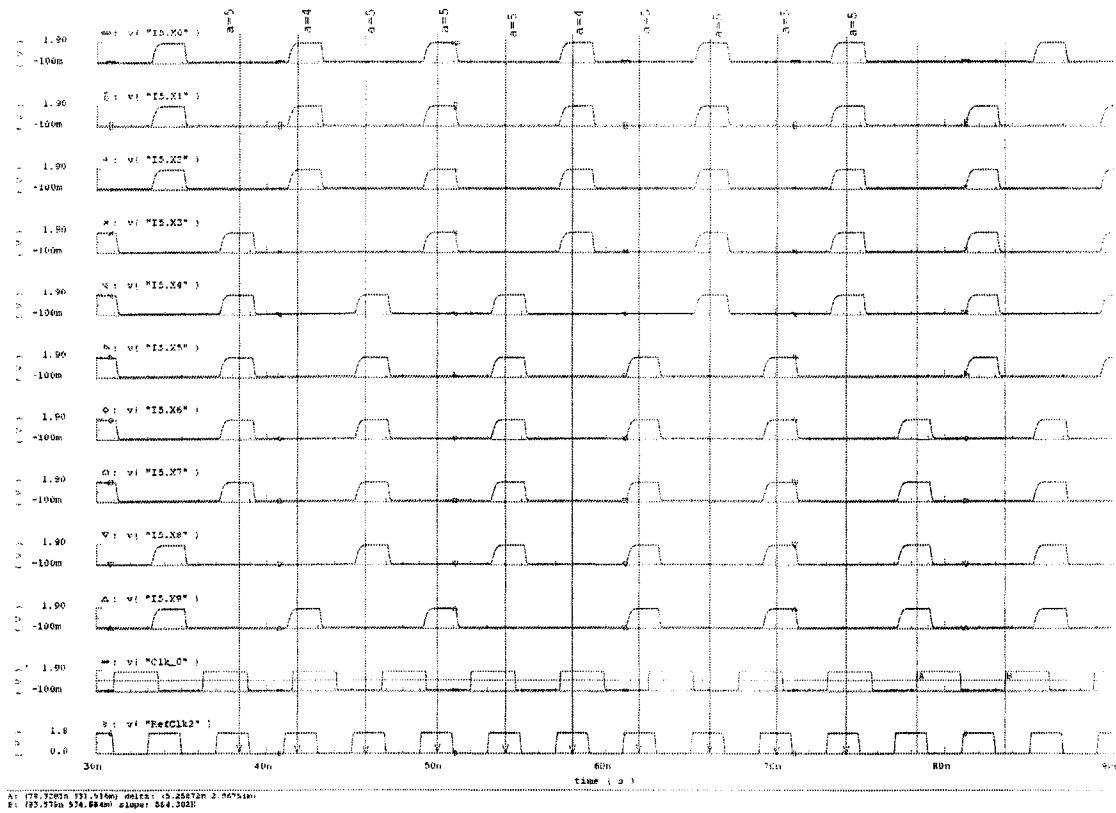


Figure 115 $f = 190.12\text{MHz}$

From the figure, the measured frequency is:

$$f = \frac{20 - 0.4}{20} \cdot 250 \text{ MHz} = 245 \text{ MHz} \quad (42)$$

The error in the measurement is equal to:

$$\text{error} = e = \frac{245.27 - 245}{245.27} = 1.1 \times 10^{-3} \quad (43)$$

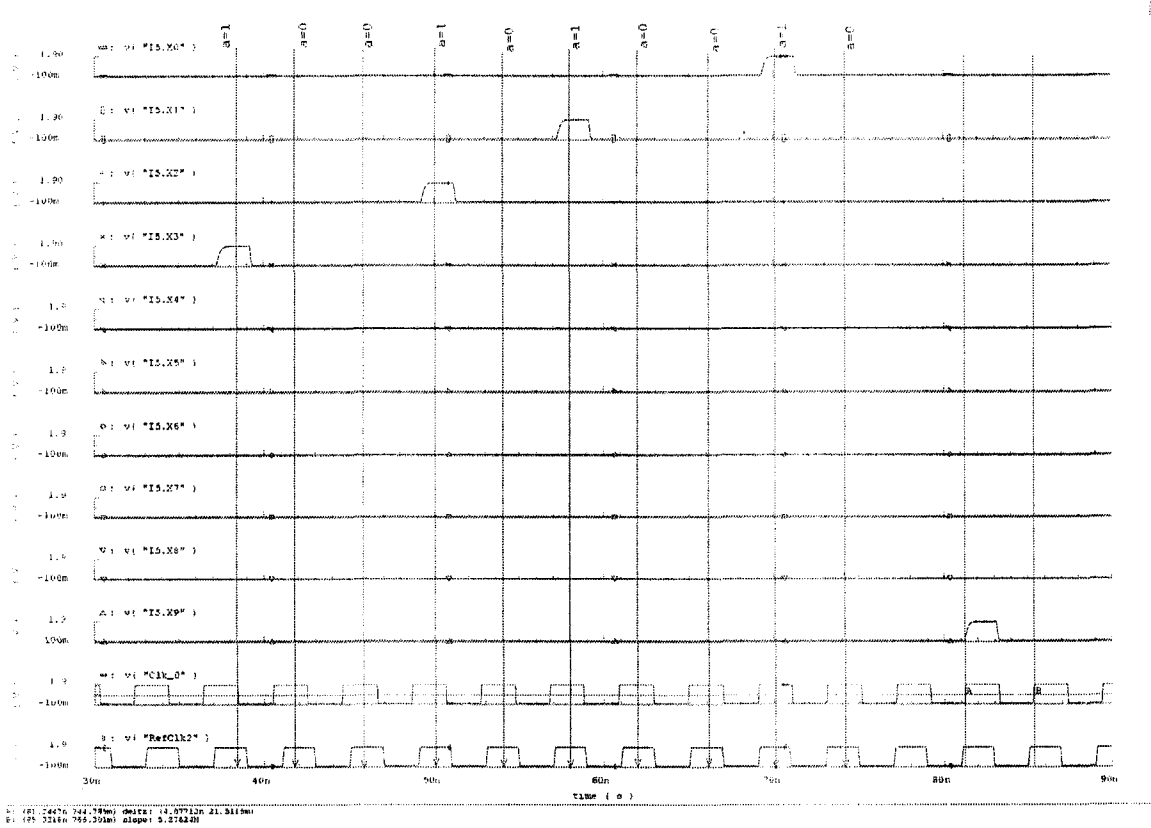


Figure 116 $f=245.27\text{MHz}$

Table 6 Input/Output of the mapping circuit with $f_{\text{REFCLK}} = 250\text{MHz}$

Input frequency (MHz)	Output
125	0 00 00 00
126.953	0 00 00 01
128.906	0 00 00 10
248.047	0 11 11 11
250	1 00 00 00
251.953	1 00 00 01
373.047	1 11 11 11

7.4. The mapping Circuit

The mapping circuit is a digital one. Its function is to map the frequency measured by the FD to a digital representation. For example, for a 7-bit ADC using the first FD architecture shown in Figure 109, with a

250MHz frequency of REFCLK, the frequency range of the FD will be from 125MHz to 375MHz. The mapping circuit input/output are shown in Table 6.

7.5. ADC Overall Picture

To put things in perspective, the following example shows how the VCO-based ADC works.

Assuming that the input range is from 0.75V to 1.1V, which will cause the VCO range to be from 125MHz to 375MHz with $f_{\text{REFCLK}} = 250\text{MHz}$ based on Figure 100. Lets assume that $N = 10$ and $M = 10$, where N and M are as given by equation (32).

Lets assume further that the exact value of the input voltage is 1.0154V. This will cause the VCO to have a frequency of 314.59MHz. Using the FD of Figure 109, the detected frequency is 313.75MHz. Using the mapping circuit explained above with its characteristic table shown in Table 6, this will correspond to a digital code of 1100001.

The measurement error of this ADC is 0.005, which is approximately $\frac{1}{2}$ LSB.

7.6. Summary

In this design, a new novel architecture for the FD [6] is used that is capable of producing the value of the frequency of the input signal at every cycle of the clock. This will make the design of an ADC running at a rate of more than 500MHz feasible.

The FD is a very important block in the ADC. The FD not only sets the speed of the ADC, bit it also affects its resolution. The resolution of the new ADC is determined by the resolution of the FD as well as the number of phases generated by the VCO. Using the new FD, a 500MHz 10-bit ADC is easily achievable in any 0.18u CMOS process with low power consumption. Current simulations show that the power consumption for such ADC is in the vicinity of 270mW, with the circuits not optimized for low power yet. The implementation of the FD is well explained in [6].

Table 7 shows a comparison between different types of ADC architectures explained in this dissertation. It clearly shows that the new ADC architecture is capable of playing a great role in the 10-bit range of resolution but at a much higher conversion rate than its counterparts with comparable power consumption. Alternatively, compared with high speed architectures such as flash ADCs and equivalent resolution, the proposed ADC consumes less power and occupies a much smaller die area.

The main advantages of the new approach are the following:

- I. It can be implemented using any typical digital process. This means that the chip will be easier to scale down as the process scales down, inexpensive, less power consumption, faster and easier for system integration.
- II. VCOs are inherently monotonic. This means that the new design will also be monotonic. Missing codes and nonmonotonicity will not be of an issue.

- III. This is a new architecture. It will open up a new area for research and development that will enable the new ADC to have higher performance in the future.
- IV. The new architecture uses a VCO to convert the voltage into frequency. A signal that is carrying frequency information is easier to transfer inside a system without losing the information than a voltage signal. This means that the speed of the new ADC can be increased using time interleaving (sometimes called multipath).
- V. The VCO is widely used in industry and literature. It has been well studied which will make it easier to design and implement.
- VI. The new ADC trades speed with resolution once it has been implemented. To increase the resolution by one bit, twice the number of the reference clock cycles is needed. This means that the conversion rate drops to half of the original speed. This makes the new design flexible and easy to reconfigure at different speed and/or resolution.

Table 7 Comparison between the different types of ADCs presented in this dissertation

ADC Type	Speed	Resolution	Power	Area
Flash	High	Low	High	High
Two-Step	Medium	Medium	Medium	High
Folding	Medium	Medium	Medium	Medium
Successive Approx. and Algorithmic	Low	High	Low	Low
Pipeline	Medium	Medium	Medium	Medium
Sigma-Delta	Low	High	Medium	Medium
VCO-Based	High	Medium	Medium	Low

References

- [1] Ahmed A. Younis, Marwan M. Hassoun and Moises E. Robinson, "VCO-based ADCs," Patent pending 2001.
- [2] J. S. Lee, W. K. Jin, D M. Choi, G. S. Lee and S. Kim, "A wide range PLL for 64x speed CD-ROMs and 10x speed DVD-ROMs," *IEEE Transactions on Consumer Electronics*, Vol. 46, Issue 3 August 2000, pp. 487-493.
- [3] E. Raisanen-Ruotsalainen, T. Rahkonen and J. Kostamovaara, "An integrated Time-to-Digital converter with 30-ps single-shot precision," *IEEE Journal of Solid-State Circuits*, VOL. 35, NO. 10, OCTOBER 2000, pp. 1507-1510.
- [4] P. Dudek, S. Szczepanski and J. Hatfield, "A high-resolution CMOS Time-to-Digital converter utilizing a vernier delay line," *IEEE Transactions of Solid-State Circuits*, Vol. 35, NO. 2, February, 2000, pp. 240-247.
- [5] J. Park and W. Kim, "An auto-ranging 50-210Mb/s clock recovery circuit with a time-to-digital converter," *IEEE ISSCC*, 1999, pp. 350-351.
- [6] Ahmed A. Younis and Micheal Nix, "Frequency and Time Detection Aparatus," Patent Pending 2001.

CHAPTER 8. High Speed Receiver Design

8.1. Introduction

With the large growth of the internet, the demand on high-speed transmission systems is continuously increasing. Many of the transmission systems such as repeaters, routers, hubs, and switches, nowadays require transmission speeds of gigabits per second with very stringent jitter, power and noise requirements.

In this chapter, novel techniques have been used to implement a high performance receiver to be used in a 1.25-3.125Gb/s transceiver. The transceiver was designed and implemented in a 0.18 μ m CMOS digital process with 1.8V power supply as a Rocketchips part. When running at 3.125Gb/s, the transceiver consumes 208mW of power and has less than 2.6ps of random jitter. Special techniques have been used in the receivers to enhance its jitter tolerance. At 3.125Gb/s, the jitter tolerance of the receiver was found to be 1UI up to 4MHz of input modulation and drops to more than 0.6UI at 20MHz of input modulation.

This chapter focuses more on the blocks that have been investigated during this research, which includes the phase detector of the fine loop, the VCO and the GM cell. It was also the goal of this research to integrate all the blocks of the receiver together and perform the top level simulations to validate the robustness of the system.

The next section will present the architecture of the receiver, its components and the main function of each one. In section 8.3, new techniques such as new VCO layout and buffer separation to increase the performance of the receiver are presented and discussed. These techniques resulted in a very robust system that by far exceeds most of today's standard requirements, such as SONET and infiniband.

To guarantee the robustness, the simulation of the receiver has to exercise different conditions and scenarios and to make sure that the receiver is still working fine under all these conditions. The top level simulation of the analog portion of the receiver is presented in section 8.4. The development of this receiver as it started from lower speeds up to the current speed and all the changes it went through as well as the techniques used to enhance its performance are presented in section 8.5. Measurement results of the receiver as well as the transmitter are presented in section 8.6. The conclusion of this chapter as well as its summary are presented in section 8.8.

8.2. Architecture

The receiver (deserializer) accepts a high-speed differential serial data stream, recovers the clock and data, and outputs a 10-bit or 20-bit parallel word and recovered clock at 1/10th or 1/20th the input data rate. The receiver is made of two main components; a deserializer and a clock and data recovery (CDR) circuit. The deserializer receives a high-speed stream of data, samples it, and then converts it into a lower speed parallel data. The clock recovery circuit is necessary to make sure that when the deserializer reads the data, it reads it correctly by insuring that the sampling time occurs in the middle of the bit period.

The CDR typically consists of a phase-frequency detector (PFD), a charge pump (CP), a loop filter (LF), a voltage controlled oscillator (VCO), and a divider. One typical problem of the PFD is its dead zone [1]. This will usually manifest itself as output jitter, as well as a reduction in the jitter tolerance of the overall system. In this design, a new approach was followed to solve of this problem. This was achieved by using two loops - a coarse loop and a fine loop. The coarse loop is used to bring the frequency of the VCO close to the nominal value, and the fine loop makes sure that the sampling clock is centered in the middle of the bit period. The nominal frequency of the VCO is the frequency at which the VCO will be able to sample the data correctly. By itself, this frequency does not guarantee correct sampling of the data, but it is a necessary condition for correct sampling. The second condition for correct sampling is that the phase of the sampling clock needs to be such that it is sampling the data in the middle of the bit period. The coarse loop is shown inside the long-dashed rectangle in Figure 117 while the fine loop is shown in the dashed rectangle of the same figure.

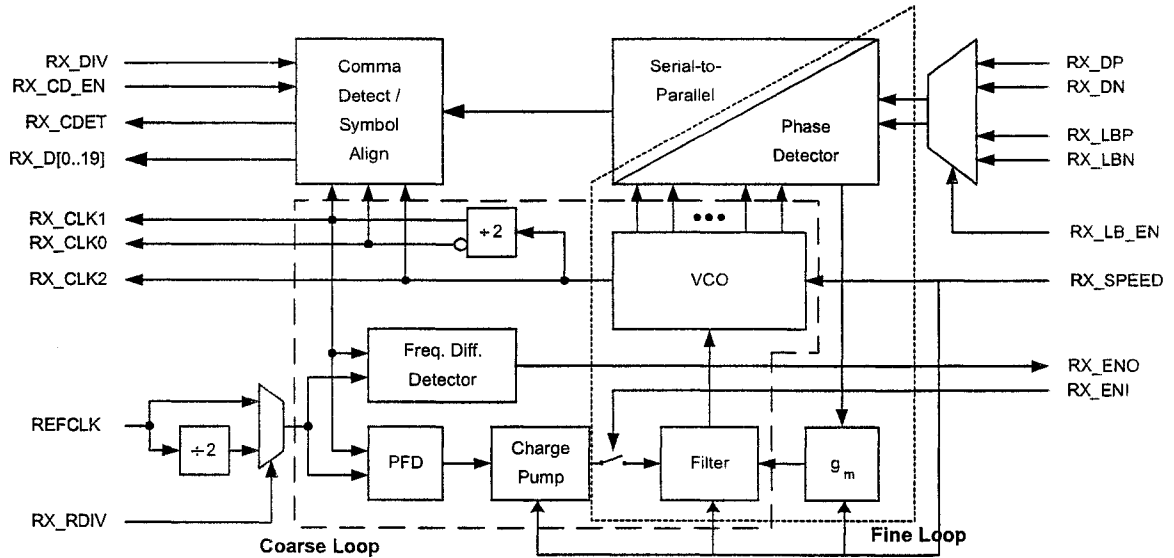


Figure 117 Receiver Block Diagram

8.2.1. The Deserializer

The deserializer covers all the digital functions in the receiver. This digital portion is responsible for producing 10/20-bit parallel data output and a comma detection, and provides clocks for latching the output as well. The deserializer block diagram is shown in Figure 118 and includes the following cells:

1. **Input sampler:** It receives 10 NRZ serial data as well as 20 clocks from the CDR circuit. It samples the data and then retimes it.
2. **Pipe control:** It produces a comma detect signal and latches the state of its input clock when a comma is detected. It also generates a 4-bit word that indicates the position of the comma in the 10-bit word.
3. **Bit alignment:** It generates one 10-bit data stream aligned to the comma.
4. **Phase alignment:** It produces either a 10-bit or a 20-bit word.
5. **Frequency difference detector:** It detects the frequency difference between REFCLK and VCOCLK. Depending on the difference of the two frequencies the ENABLE output will be either high or low. When the difference is less than 2%, the ENABLE output will be low. When the difference is more than 3.5%, the ENABLE output will be high. If the difference is between 2% and 3.5%, then the output will retain the previous state.

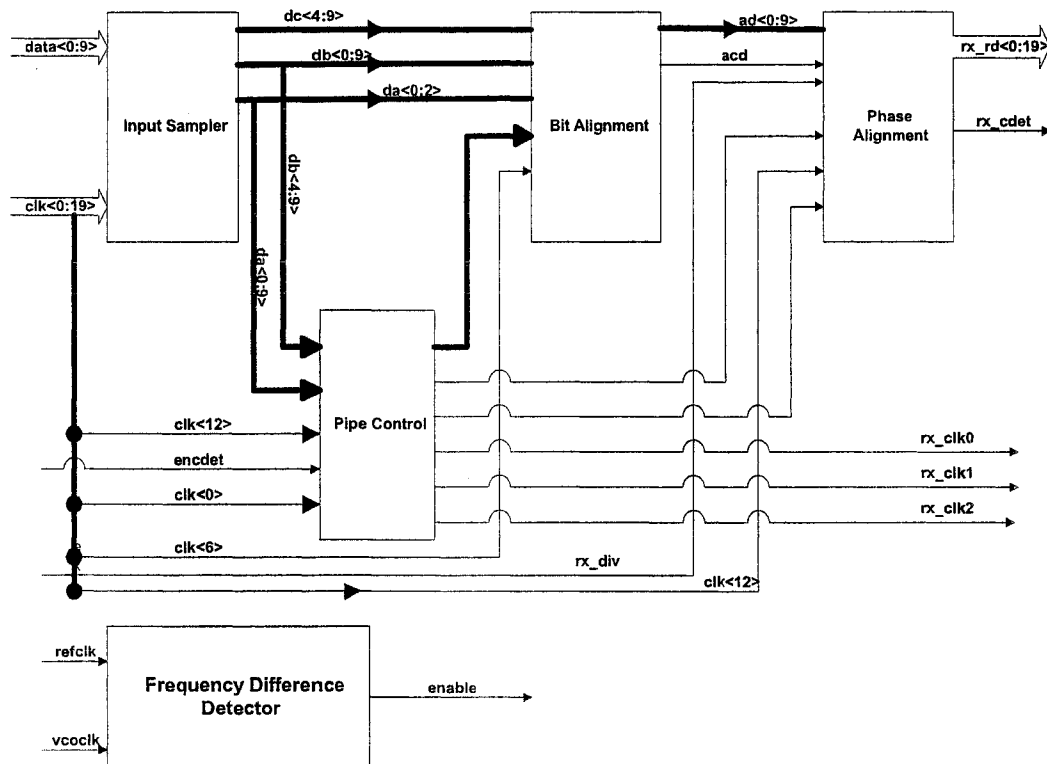


Figure 118 Block diagram for the digital portion of the receiver.

8.2.2. The Coarse Loop

The coarse loop is used whenever the VCO frequency deviates for more than 3.5% from its nominal value. The nominal values for the VCO are: 125MHz for operation at 1.25Gb/s, 250MHz, at 2.5Gb/s; and 312.5MHz, at 3.125Gb/s. The coarse loop employs a phase-frequency detector and as such it can adjust the VCO frequency over a wide range of values using a nominal reference clock signal. The main function of the coarse loop is to adjust the VCO frequency so that its value is within the lock range of the fine loop.

The design of the coarse loop is similar to a conventional PLL. It consists of a phase-frequency detector (PFD), a charge pump (CP), a 2nd order loop filter (LF), a VCO and a divider, as shown in Fig. 1(b) inside the solid rectangle.

In this application, a 10-stage VCO was used. It generates a set of 20 clocks that are time-shifted by 1/20th of the VCO period. Only one clock is used as an input to the PFD that compares this clock with REFCLK. Based on the difference between these two clocks, the PFD generates up and dn signals that drive the CP, which will in turn sink/source a current from/to the LF. The LF averages this current on a capacitor to generate a voltage signal that adjusts the frequency of the VCO.

To characterize the closed-loop response of the coarse loop, a 5% frequency step is introduced to the reference clock. This step should produce a 5% change in the output frequency if the coarse loop locks. The corresponding response of the coarse loop is then recorded, as shown in Figure 119, under the condition that the damping resistance is reduced from its normal value so that the loop is under damped. This allows the natural frequency as well as the gain of the VCO to be easily calculated.

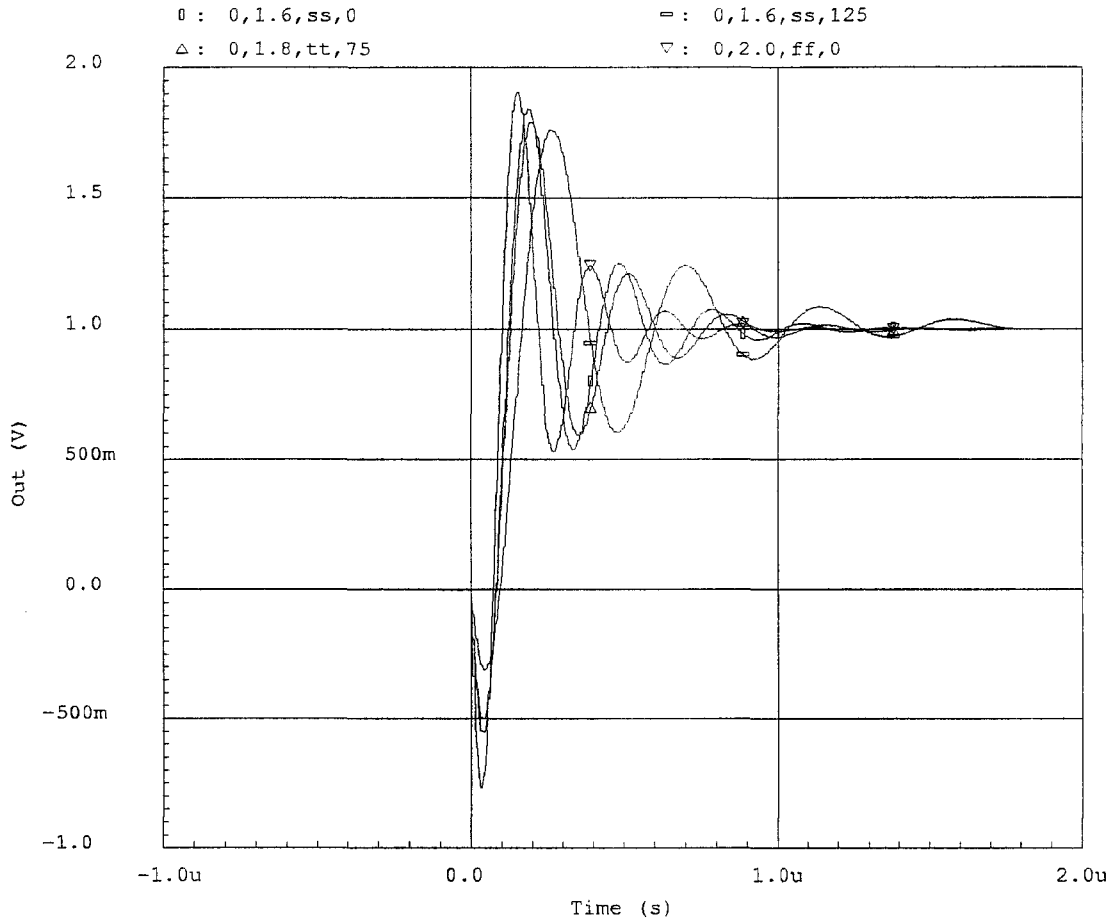


Figure 119 Simulation Results for the Coarse Loop at 125MHz.

8.2.3. The Fine Loop

The fine loop employs the 2x oversampling method as indicated in Figure 120. This method is similar to that used by [2] and [3]. The fine loop uses the input data stream as a frequency reference and it employs only a phase detector so that the lock range is small.

The fine loop also employs an analog phase correction method, which is unique to this design [4]. The fine loop VCO operates at 1/10th of the input serial data rate. For example, when the data rate is 2.5Gb/s, the VCO outputs will run at 250MHz. The phase detector has 10 data output pairs and 10 phase output pairs where the former provide full digital signal levels and the latter provide analog signals.

A block diagram of the fine loop is given in Figure 117 inside the dashed rectangle. The basic elements of the fine loop are: (1) a VCO, (2) a phase detector (PD) circuit, (3) a transconductance circuit (Gm), and (4) a loop filter (LF) circuit. The VCO and LF of the fine loop are shared with the coarse loop.

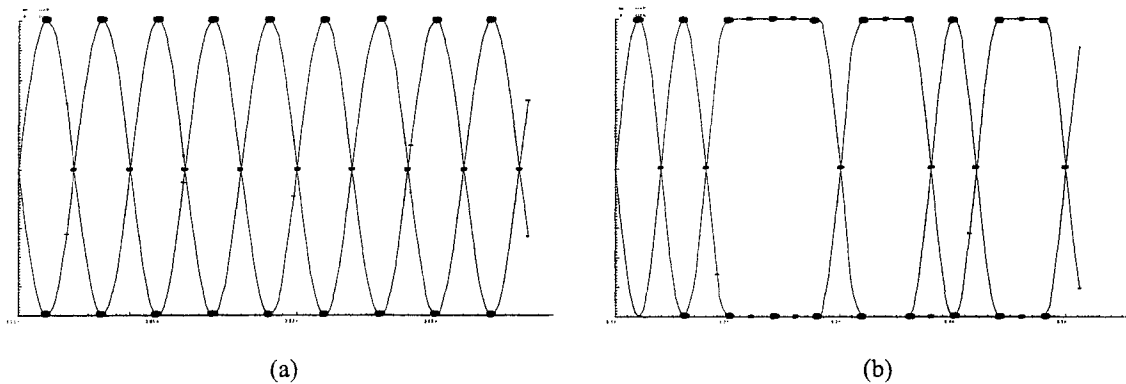


Figure 120 (a) 1010 pattern. (b) 8B10B pattern. Large dots denote Data Samples and small dots denote Phase Samples.

The fine loop is used to track incoming phase variations in the received coded Non Return to Zero (NRZ) signal. This tracking occurs after the coarse loop has brought the VCO close to the correct frequency. The fine loop is a type that integrates phase error and forces it toward zero. The coded NRZ can have a run length as large as 5, which means that there can be 5 bit times between transitions of the incoming signal or 5 bit times between phase detector updates.

A positive phase error means that the VCO is lagging behind the input. If the transitions are perfect ramps between $-V$ volt and $+V$ volt, then the samples taken at or near transitions measure the time error in the VCO. The time error is proportional to the phase error. The Gm circuit converts this error into a proportionate current that feeds the loop filter, which will in turn, increase or decrease the VCO frequency.

To characterize the closed-loop response of the fine loop, a 1% frequency step is introduced to the data rate. This step should produce a 1% change in the output frequency if the fine loop locks. The corresponding response of the fine loop is then recorded, as shown in Figure 121, under the condition that the damping resistance is reduced from its normal value so that the loop is under damped. From the transient response, the natural frequency f_n and damping factor, ζ , can be easily determined.

Figure 121 gives the step response of the fine loop for two different data rates. Figure 121(a) shows the step response of the fine loop when the data rate is 1.25Gb/s, while Figure 121(b) shows the step response of the fine loop when the data rate is 2.5Gb/s.

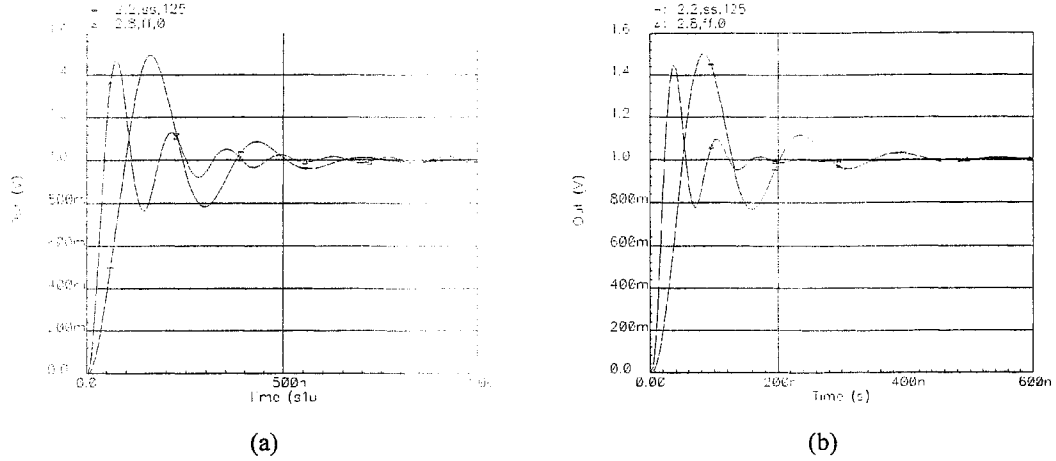


Figure 121 Transient response of the Fine Loop for (a) 1.25Gb/s data rate and (b) 2.50Gb/s data rate.

8.2.4. The Gm Circuit

A simplified version of the Gm circuit is shown in Figure 122. The transconductance circuit converts a differential voltage signal into a single-ended current. It contains one differential pair that converts voltage to current, source degeneration resistor (R) to reduce the overall gain and improve the linear range, and wide-swing cascoded current mirrors. The output stage provides a low current and a high impedance output. The overall g_m of the circuit is:

$$g_m(s) = \left[\frac{g_{m1}}{\left[(1 + R \cdot g_{m1}) \cdot A \cdot \left[1 + \frac{s}{g_{m3}} \cdot (C_{gs3} + C_{gs5}) \right] \right]} + \frac{g_{m1}}{\left[(1 + R \cdot g_{m1}) \cdot A \cdot B \cdot \left[1 + \frac{s}{g_{m3}} \cdot (C_{gs3} + C_{gs5}) \right] \cdot \left[1 + \frac{s}{g_{m7}} \cdot (C_{gs7} + C_{gs8}) \right] \right]} \right] \quad (1)$$

Where g_{m1} is the transconductance of each of the input transistors (mp_{1a} , mp_{1b}), A and B are the NMOS and PMOS current mirror gains, R is the degeneration resistor, which is more correctly modeled as $Z = R || 1/sC$, g_{m3} is the transconductance of the NMOS diode-connected mirror transistors, and C_{gsx} is the gate to source capacitance of device x .

Equation (5) has two elements summed up together. The first element represents the negative path and the second element represents the positive path.

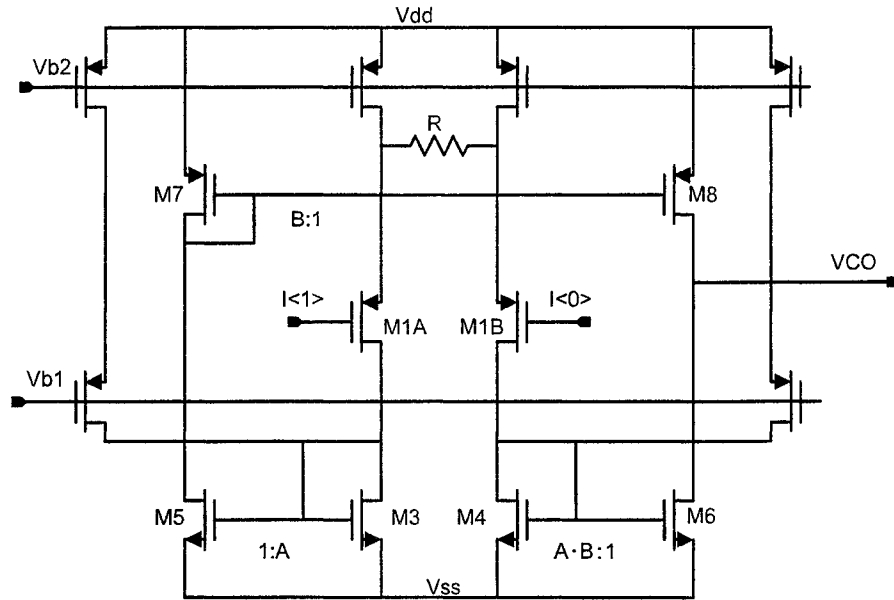


Figure 122 Simplified schematic of the transconductance circuit (G_m).

8.2.4.1. Figures of Merit

When designing the G_m circuit, the designer has to take care of some important characteristics:

- Input linear range: Input voltage range that the G_m circuit can handle without a significant amount of distortion due to non-linear effects.
- g_m value: The gain of the transconductance circuit. This is important because it will affect the loop dynamics.
- Bandwidth: This sets a limit on the output signal frequency. The bandwidth of the G_m cell has to be high enough so that it does not interfere with the fine loop dynamics and it should be low enough so that it will not pass unnecessary noise to its output, which will deteriorate the receiver performance.
- Input offset voltage: The voltage required at the input to get an output current of 0A. Ideally, this should be 0V, however, due to mismatches in transistors, it will not be 0V. The offset of the G_m will appear as a phase error, which will increase the BER.
- PSRR: Power Supply Rejection Ratio, since the G_m circuit is one of the most critical circuits on the fine loop signal path.

8.2.4.2. Designing for Figures of Merit and Simulation Results

The linear range is determined by the size of the differential pair, source degeneration resistor, bias current, and current ratio. The plot in Figure 123 shows the transconductance and linear operating range of the

Gm circuit at 1.25Gb/s data rate. Each curve represents a different corner that was run on the circuit with the input being swept from -1.8V to 1.8V.

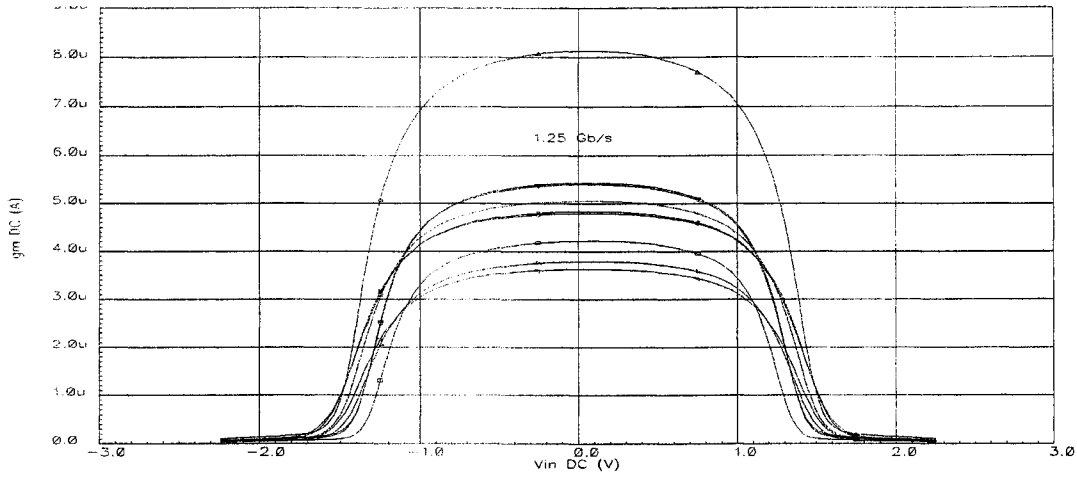


Figure 123 Linear range and g_m value of transconductance circuit for the 1.25Gb/s operation.

The Gm circuit bandwidth must be greater than the fine loop bandwidth, so that the loop dynamics do not change. Simulation results show that the bandwidth of the Gm circuit is in the range of 50MHz to 95MHz, depending on different simulation corners. Figure 124 shows the ac response of the Gm circuit, output voltage (Y-axis) over frequency (X-axis). Each curve represents a different corner that was run on the circuit. Differential ac signals with a magnitude of 0.5V were placed on the Gm inputs.

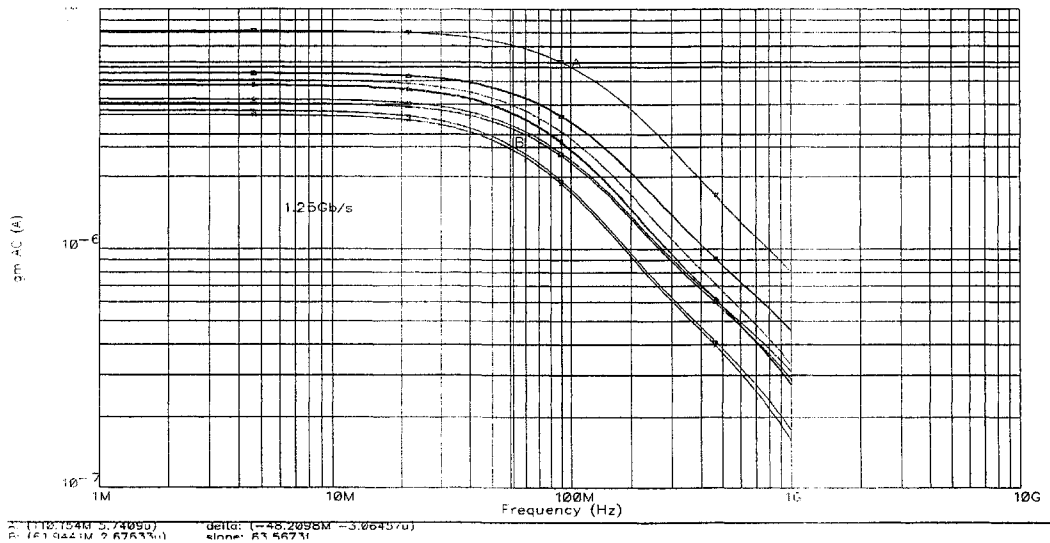


Figure 124 AC response of Gm circuit at different corners.

To determine the offset voltage, feedback was applied to the Gm input. This made the output current of the Gm approximately zero. Then the difference between the Gm inputs was measured over temperature. This

measurement corresponds to the systematic input offset voltage. Simulations were run over different corners, and the results are shown in Figure 125. Additionally, there will be offset due primarily to input device mismatch.

Top-level simulations show that the Gm input offset voltage is less than 3 mV, which is negligible.

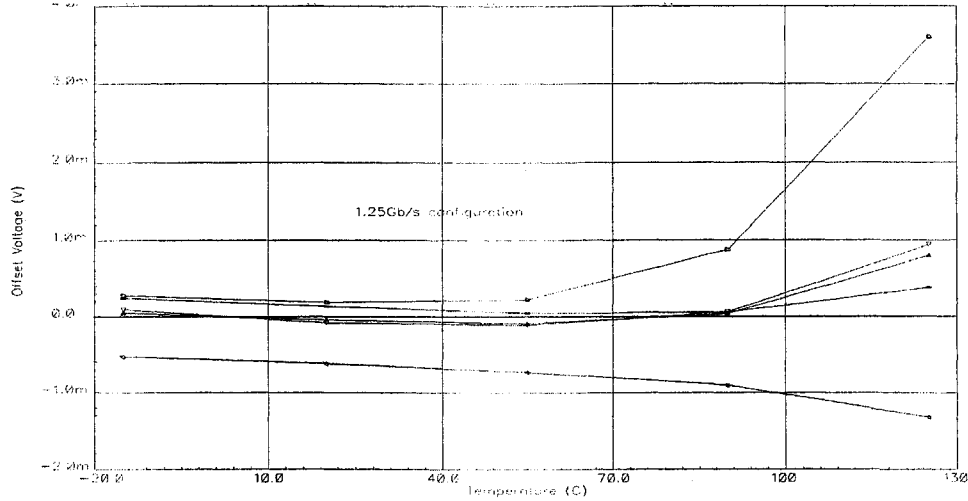


Figure 125 Gm systematic input offset voltage vs. temperature for the 1.25Gb/s operation.

8.3. Performance Enhancement

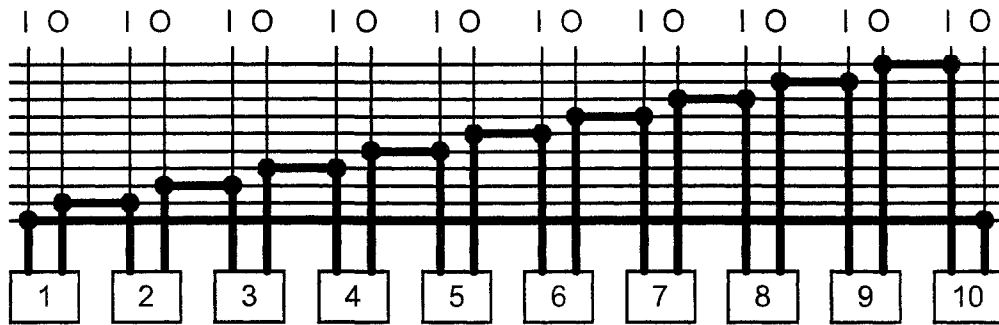
The performance of the receiver has been enhanced by employing the following techniques:

8.3.1. VCO Jitter Minimization

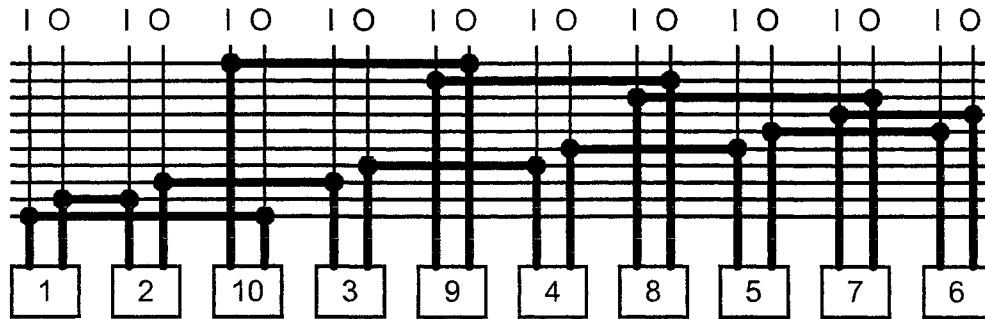
As the major jitter contributor to the overall system, special techniques were followed in the design and layout of the VCO blocks. Mismatch among the loads of the VCO delay lines will result in jitter. Traditionally, a VCO delay line is laid out such that its delay cells are arranged in ascending or descending order and the output of a certain delay cell goes to the input of the next delay cell, except for the last delay cell in the line whose output goes to the input of the first delay cell. Figure 126(a) shows a 10-stage VCO that follows the conventional layout. Each stage is represented by a rectangle that has the cell ID and has one input and one output. In the actual design, each stage has been implemented differentially, i.e., each stage has two differential inputs and two differential outputs. Single-ended representation is used in here just to clarify the concept. In the conventional approach, the VCO is laid out as follows: The output of each stage is directly connected to the input of the next stage so that the output signal path is short for nine stages and long for one stage - as shown in Figure 126(a). Figure 126(b) shows the new VCO layout in which the delay cells are rearranged differently. Here, the signal path is more uniform among the cells and hence will have less mismatch. Assuming that the x dimension is much larger than the y dimension in the wire routing, (which is the

case in this design), Figure 126(a) shows a mismatch ratio of 10:1, while Figure 126(b) shows a mismatch ratio of 2:1.

This new arrangement is basically a folding operation of the delay line of the VCO. The line has been folded on top of itself between cells 6 and 7. This folding operation results in that cell 7 goes in between cells 5 and 6, cell 8 goes in between cells 4 and 5, cell 9 goes in between cells 3 and 4, and cell 10 goes in between cells 2 and 3. Details of the folding process are explained in a patent application [5]. Because the delay line has been folded on top of itself only once, this arrangement is also called single-fold. More folding is also possible and will result in better matching.



(a)



(b)

Figure 126 (a) Traditional VCO layout and (b) New VCO layout.

8.3.2. Buffer Separation

Mismatch in the VCO clocks will result in a nonuniform sampling times, which will in turn reduce the jitter tolerance of the receiver. One cause of this mismatch in the VCO clocks was determined to be the nonuniform loading on those clocks. To reduce the mismatch among the VCO clocks, the nature of the load of those clocks has been studied more carefully. It was found that there are some parts of the load that are sensitive to clock mismatch, while other parts are not. Fortunately, the sensitive parts of the loads are the same for all the

clocks, while the nonsensitive parts are not. This led to the separation of the load into two parts; sensitive and insensitive, where each part is provided by a separate set of clocks from the buffers in the VCO. The buffers in the VCO have been split into two sets, where each buffer is sized according to the size of its load [6] as shown in Figure 127. This technique caused a matched loading to the VCO clocks that are used for the sensitive parts of the loads, which enhanced the jitter tolerance of the receiver.

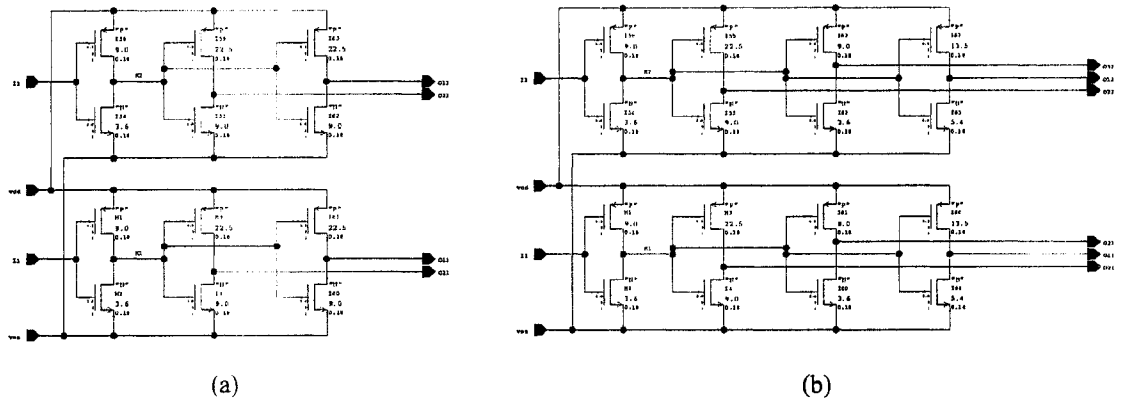


Figure 127 Buffer separation. (a) Original design, and (b) new design.

8.3.3. Power Supply Noise Reduction

A decoupling capacitor structure has been used to eliminate oscillations on the power supply that are caused by the combination of power supply inductance and on-chip capacitance; and to provide an instantaneous current for device switching needs.

From the point of view of power supply noise, the circuit-blocks under the entire transceiver were grouped into various circuit types based on a) how much noise the individual circuit-blocks generate, b) how susceptible they are to noise, and c) how close they are to each other in terms of signal flow and layout. Each of these groups is provided with an individual decoupling filter structure as well as a power supply Kelvin connection.

An array of NMOS gate capacitors in parallel was used as the decoupling capacitor structure. They were placed under the power buses to avoid occupying any extra space.

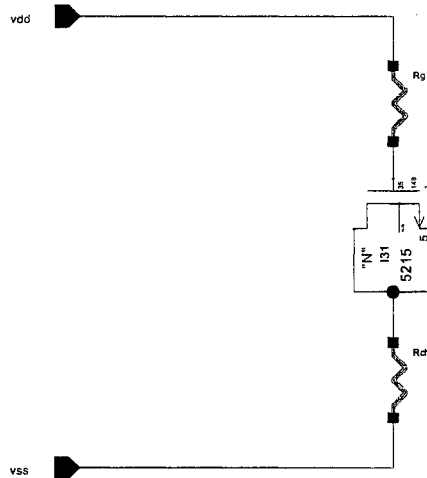
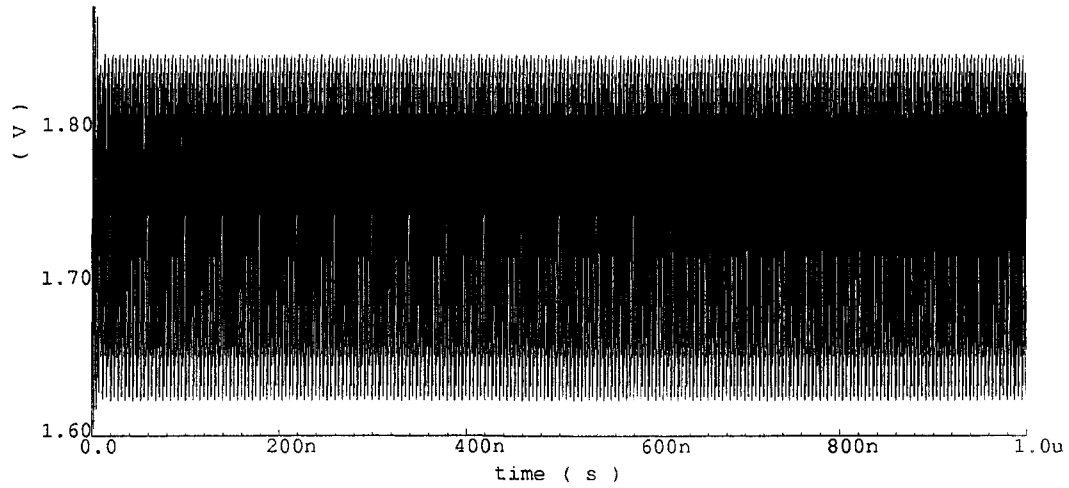


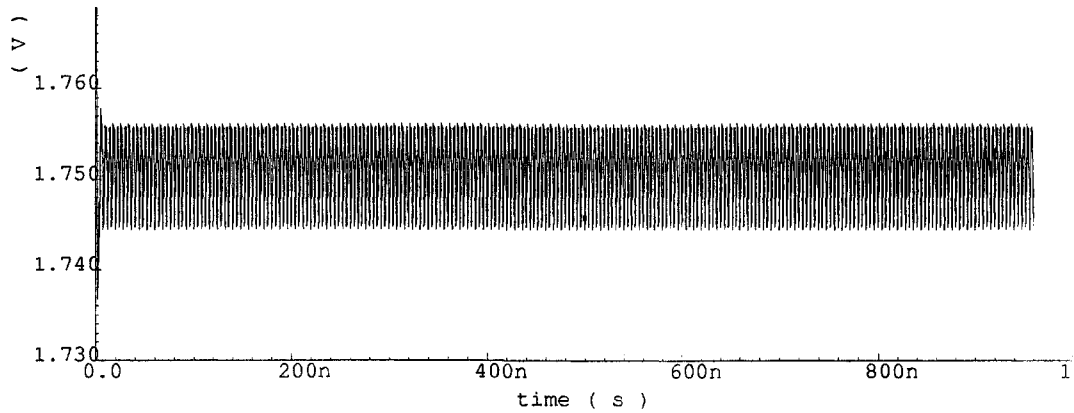
Figure 128 Decoupling Filter Structure

A parasitic gate resistance, (R_g), and a parasitic channel resistance, (R_{ch}), will come in series with each of the transistors [7] as shown in Figure 128. The transistor models do not represent these parasitic resistances reliably; hence these resistances have been inserted as parasitic resistances in series with the transistors used as decoupling capacitances. Other parasitics, such as the metal routing resistance as well as the metal-metal capacitance between the two power supplies, have been added to the simulation.

Figure 129(a) shows the simulation results for one of the circuit groups before adding the decoupling capacitor structure; it shows a noise of 200mV on the power supply. Figure 129(b) shows the results after the addition of the decoupling capacitor structure, which reduced the noise range to less than 15mV.



(a)



(b)

Figure 129 Power supply simulation (a) without the decoupling structure, and (b) with the decoupling structure.

8.4. Top Level Receiver Analog Simulations

A startup simulation is shown in Figure 130. This shows the general operation of the receiver dual coarse-fine loop operation. The top two curves on the same figure show the differential inputs to the Gm, and the bottom one shows the VCO input voltage. At start-up the control voltage to the VCO is at $\sim 0V$. The VCO is oscillating, but not at the correct frequency. This causes the inputs to the Gm to vary. During this time, the coarse loop is controlling/changing the VCO control voltage. After a little more time, the coarse loop locks in frequency to the reference clock. When this happens, the input to the Gm does not vary as much as before, but there can be a large phase difference between the VCO and the incoming data. This is where the fine loop

comes in. At M1, the cursor shown in the plot, control of the VCO voltage is taken from the coarse loop and is given to the fine loop.

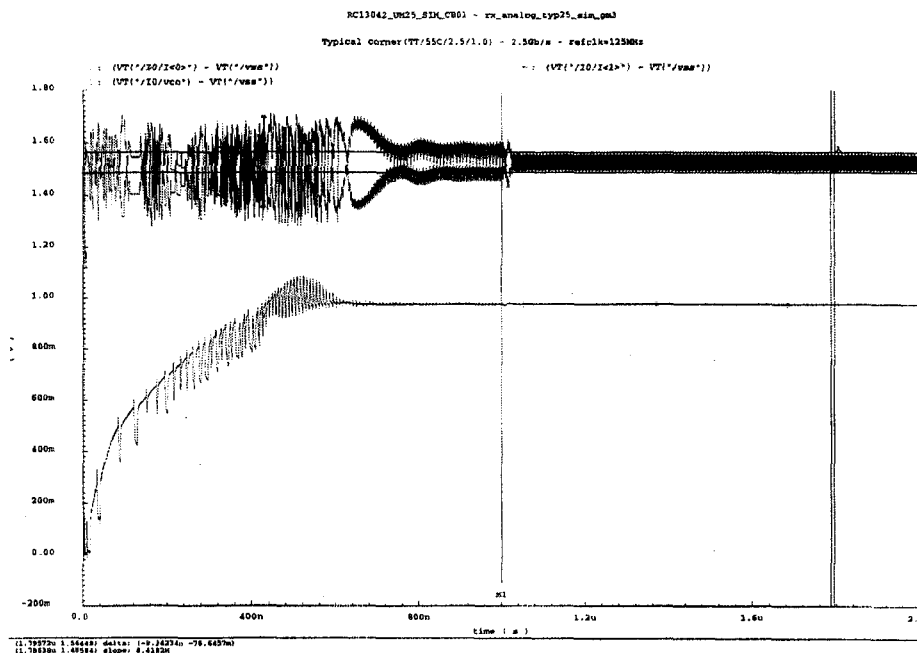


Figure 130 Top-level receiver analog simulation for the 2.5Gb/s operation.

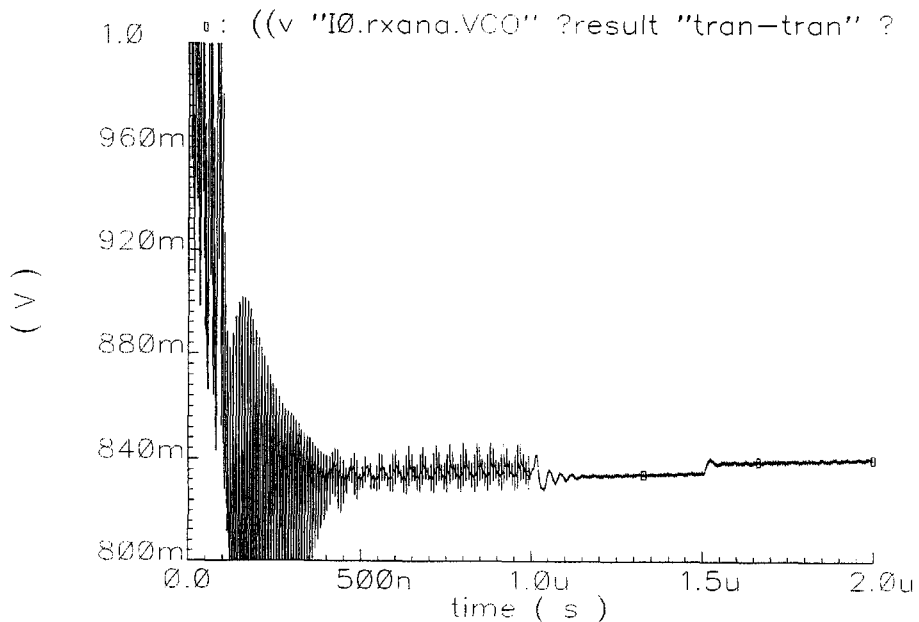


Figure 131 Top-level receiver analog simulation for the 2.5Gb/s operation. At 1.0us the control is switched from the coarse loop to the fine loop. At 1.5us, a 1% frequency step is made to test the response of the system.

The fine loop changes the VCO control voltage so that the incoming data and the VCO are not only at the same frequency, but also have the correct phase relationship. After a short period of time after the fine loop is enabled, the offset of the differential input to the Gm is gone, indicating a phase locked condition.

Frequency step response simulations are shown in Figure 131, which shows that the fine loop will lock again after a step in the frequency of the input data is introduced. Some simulations were done with a phase-modulated input, which produces small oscillations in the control voltage.

8.5. Receiver Development

The current design of the receiver has been developed from an existing architecture developed originally at Rocketchips Inc. and extended to operate at 3.125Gb/s in a 0.18 μ m CMOS process that is using 1.8V power supply. The changes and new techniques this author made and used are highlighted in the following paragraph.

This low voltage operation made the design challenging and required many changes to be done to the original design to make it running at 3.125Gb/s speed. In particular, the Gm circuit in the fine loop was modified to make sure it has the right bandwidth for the 3.125Gb/s operation. The charge pump was modified to operate with lower power supply. Many of the cascoded transistors were removed and the design has to guarantee proper operation of the switches with this lower power supply value. The LATCH suffered from the headroom issue and it was redesigned to overcome this problem. Originally, the LATCH had an NMOS input stage that got replaced with a PMOS input stage. The rest of the circuits in the LATCH had changed accordingly. The bandwidth of the samplers was also increased to more than 1.56125GHz so that the samplers can sample the input data correctly. The VCO is still running at 1/10th of the data rate, which means that its nominal frequency has increased to 312.5MHz, so, it has to be reoptimized at this speed. The jitter tolerance of the VCO was to increase, so, a new VCO layout technique was used to reduce the mismatch among the VCO clocks, and a buffer separation technique resulted in a matched load for the VCO clocks, which further enhanced the matching among the VCO clocks and ultimately resulted in a better jitter tolerance. In this work top level simulations for the analog portion of the receiver was performed to validate the robustness of the design.

At 1.25Gb/s the VCO was running at 250MHz and generating 10 phases that are passed to the phase detector that has 5 phase samplers as well as 5 data samplers. Each one of the samplers use one of the VCO phases in a time interleaved fashion. Because of this operation, the VCO is running at 1/5th of the data rate. For this design, the bandwidth of the samplers must be more than 612.5MHz, which is half of the bit rate. Figure 132 shows the implementation of the phase detector that has 5 cells. Each cell is made of 2 samplers, a LATCH, an XOR and an analog mux as shown in Figure 133. The 5-stage VCO used in this design is shown in Figure 134.

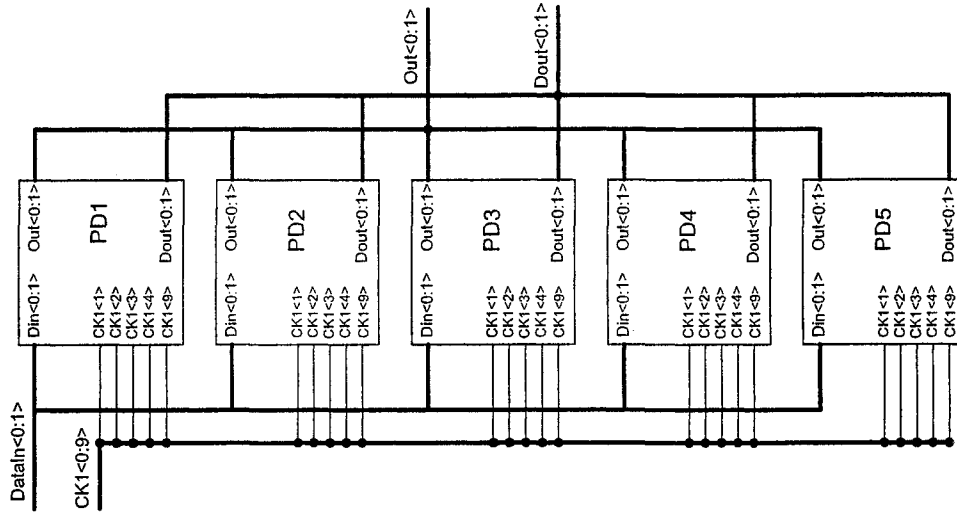


Figure 132 The phase detector is made of 5 cells, each has two samplers, a LATCH, an XOR and an analog MUX

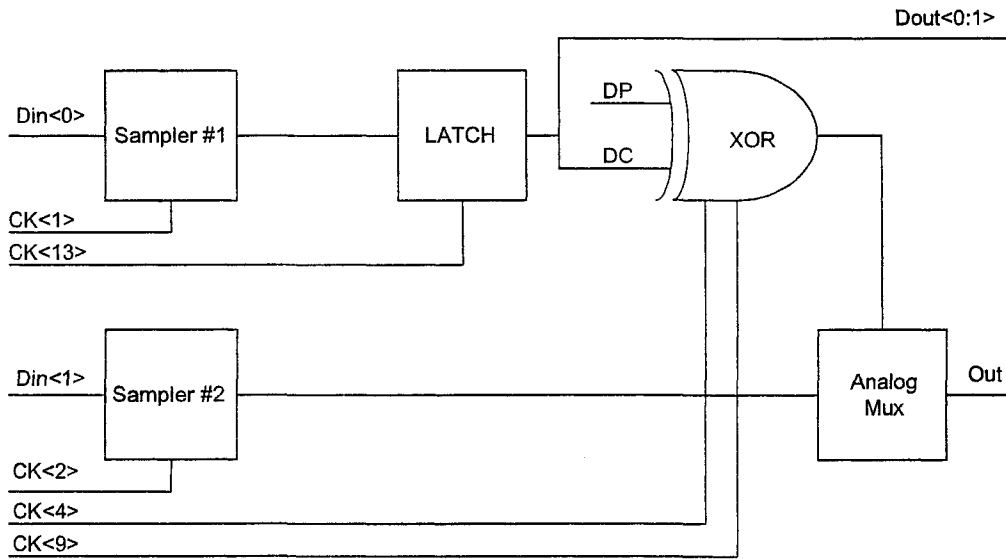


Figure 133 Phase detector implementation

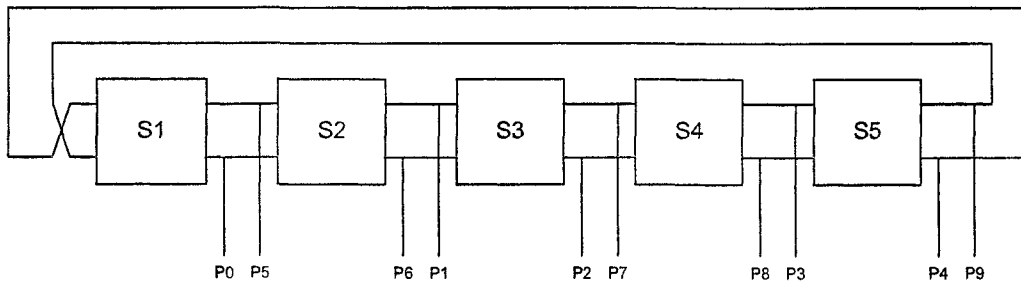


Figure 134 Five stages ring oscillator

A 2.5Gb/s receiver used the same architecture but it has been designed in a 0.25 μ CMOS process using 2.5V power supply. In this design, 10 phase samplers and 10 data samplers are used that required the VCO to generate 20 phases, but still running at 250MHz. Because the number of stages of the VCO has increased, more power has been consumed. The bandwidth of the samplers has increased to 1.25GHz to accommodate the 2.5Gb/s data rate, but they were still running at 250MHz.

Although still using the same architecture, the 3.125Gb/s operation of the design was not easy to achieve. To overcome the bandwidth constraint using the 0.25 μ CMOS process, the 3.125Gb/s was designed in 0.18 μ CMOS process. Although this is a faster process, the new power supply, which is 1.8V, makes the design of many circuits of the receiver a challenging one. Not only the speed requirement was increased to 3.125Gb/s, but also the jitter requirements were becoming more stringent. The Gm circuit in the fine loop was modified to make sure it has the right bandwidth for the 3.125Gb/s operation. The charge pump was modified to operate with lower power supply. Many of the cascoded transistors were removed and the design has to guarantee proper operation of the switches with this lower power supply value. The LATCH suffered from the headroom issue and it was redesigned to overcome this problem. The bandwidth of the samplers was also increased to more than 1.56125GHz so that the samplers can sample the input data correctly.

The VCO is still running at $1/10^{\text{th}}$ of the data rate, which means that its nominal frequency has increased to 312.5MHz. This requires more power and a little increase in the area.

In order to increase the performance of the receiver, two techniques have been used to enhance the matching of the VCO phases.

8.5.1. VCO layout.

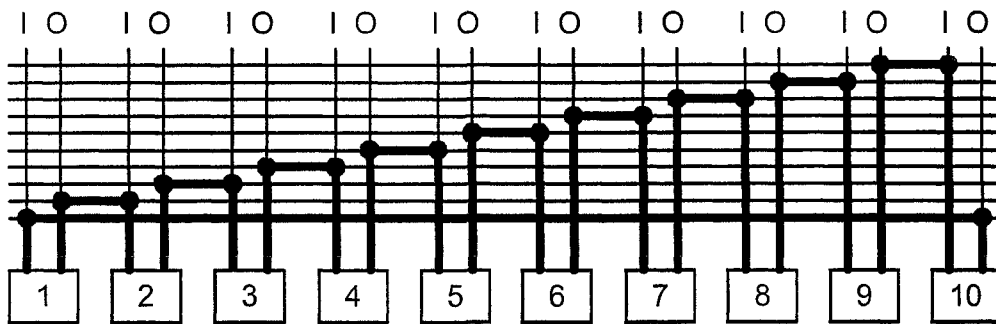
Mismatch among the loads of the VCO delay lines will result in jitter. Traditionally, a VCO delay line is laid out such that its delay cells are arranged in an ascending or descending order and the output of a certain delay cell goes to the input of the next delay cell, except for the last delay cell in the line whose output goes to the input of the first delay cell. Figure 135(a) shows a 10-stage VCO that follows the conventional layout. Each stage is represented by a rectangle that has the cell ID and has one input and one output. In the actual design, each stage has been implemented differentially, i.e., each stage has two differential inputs and two differential outputs. Single-ended representation is used here just to clarify the concept. In the conventional approach, the VCO is laid out as follows: The output of each stage is directly connected to the input of the next stage so that the output signal path is short for nine stages and long for one stage - as shown in Figure 135(a).

There are many techniques in the literature to match the signal paths. One way is to add dummy lines to the short signal paths to match them with the long one. Although this will result in a matched total capacitance of the line, however, the actual capacitance, or some times called the distributed capacitance, that the signal sees while traveling from one node to another is not going to be the same. In addition to that, the resistance seen by the traveling signal is not going to be matched as well. Another approach is the ring layout as

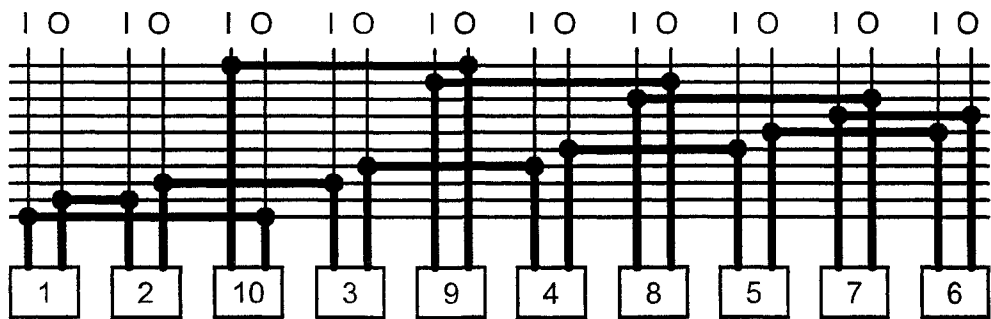
shown in Figure 136. Although this will have better matching of the signal path, the layout of the VCO cells or stages will be different and dependent on the position of that stage. This may cause another source of mismatch to the signals.

Figure 135(b) shows a new VCO layout in which the delay cells are rearranged differently than the conventional ones. Here, the signal path is more uniform among the cells and hence will have less mismatch. Assuming that the dimension is much larger than the y dimension in the wire routing, (which is the case in this design), Figure 135(a) shows a mismatch ratio of 10:1, while Figure 135(b) shows a mismatch ratio of 2:1.

This new arrangement is basically a folding operation of the delay line of the VCO. The line has been folded on top of itself between cells 6 and 7. This folding operation results in that cell 7 goes in between cells 5 and 6, cell 8 goes in between cells 4 and 5, cell 9 goes in between cells 3 and 4, and cell 10 goes in between cells 2 and 3. Details of the folding process are explained in a patent application [5]. Because the delay line has been folded on top of itself only once, this arrangement is also called single-fold. More folding is also possible and will result in better matching.



(a)



(b)

Figure 135 (a) Traditional VCO layout and (b) New VCO layout.

The novelty of this approach is that it is simple and requires no extra area and will not result in extra loading. Another feature of this approach is that the new layout is still layout friendly and very close to the conventional one, i.e., it will require less time to finish. Other approaches to match the signal paths have their problems. For example adding dummy lines to match signal path length will slow down the operation of the VCO because more capacitance is added, which will also require more area and power and will not match the resistance of the signal path. The second technique to enhance the matching mentioned above may result in a better signal path matching than the new technique, however, it was not considered in the actual implementation because it is more difficult to do the layout and will result in more space and mismatches in the layout of the VCO cells which may cause another source of mismatch.

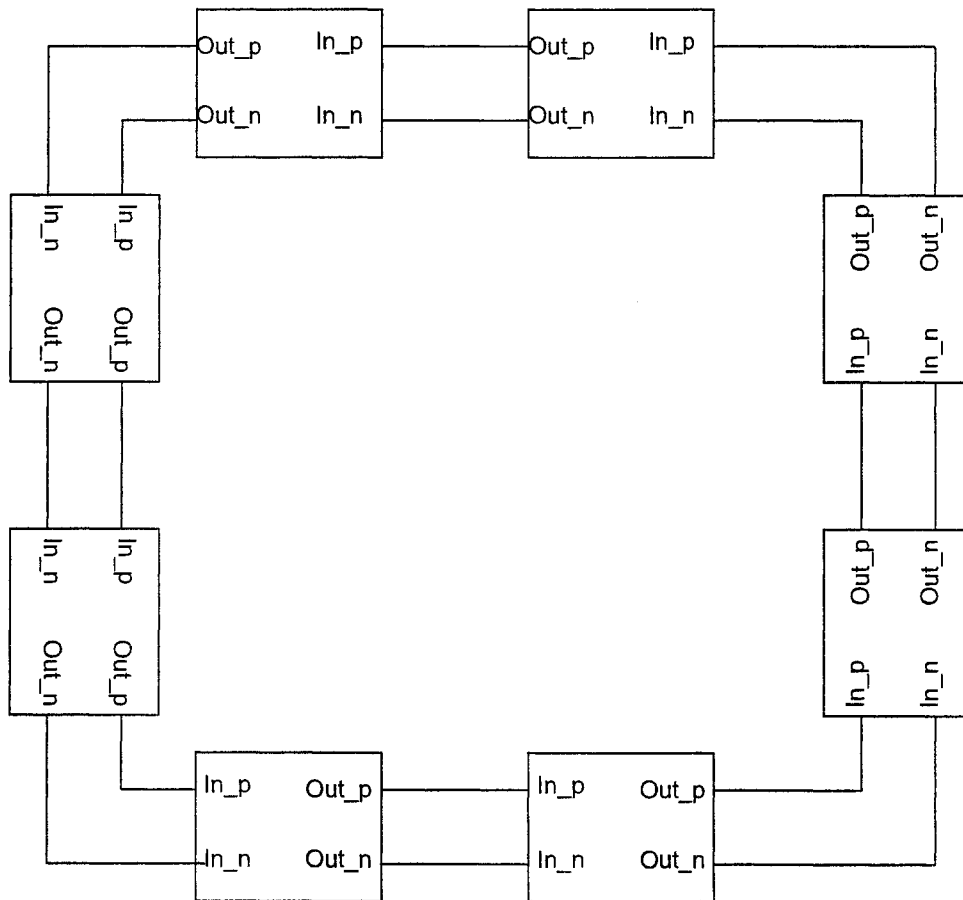


Figure 136 The Ring approach of the layout of the VCO

8.5.2. Parasitic insensitive clocking scheme

The current design of the fine loop consists of a phase detector cell (PD) and a VCO-plus-filter cell (VCOPF) as shown in Figure 137.

The VCOPF cell provides two sets of clocks; one is called CK1<0:19> that drives the PD cells and the second set is CK2<0:19> that drives the digital circuits.

The PD cell consists of 10 phase detectors that work in a time interleaved fashion using the 20 clocks that are coming out of the VCO, CK1<0:19> as shown in Figure 140.a).

Each one of the phase detectors inside the PD cell consists of 2 samplers, one LATCH one XOR and one analog MUX, and five clocks as inputs to it as shown in Figure 141. Those clocks are:

1. CK1<1> drives one of the samplers that is called data sampler.
2. CK1<2> drives the second sampler that is called phase sampler.
3. CK1<13> drives the LATCH.
4. CK1<4> and CK1<9> drive the XOR.

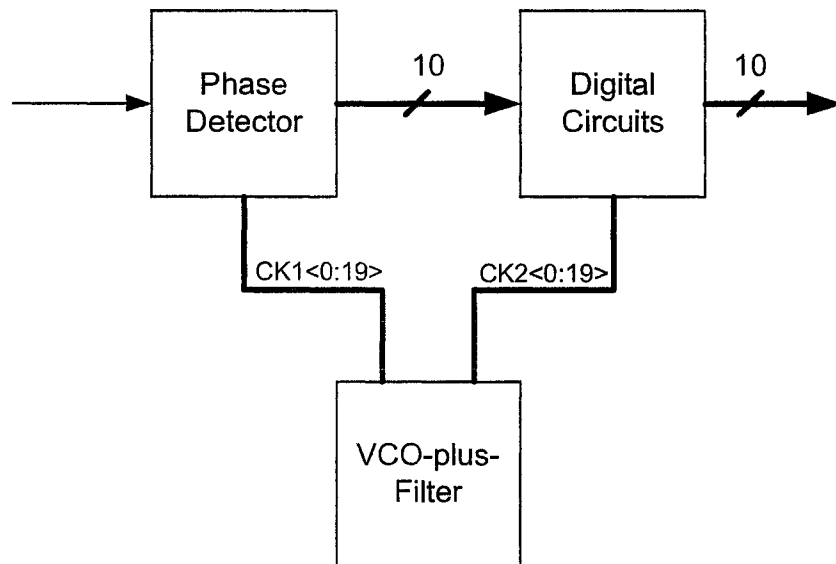


Figure 137 Original design of the fine loop.

The relationship among the clocks is shown in Figure 138, where each clock is shifted from its successor by $1/20^{\text{th}}$ of the clock period. Note that the period of all of clocks is the same.

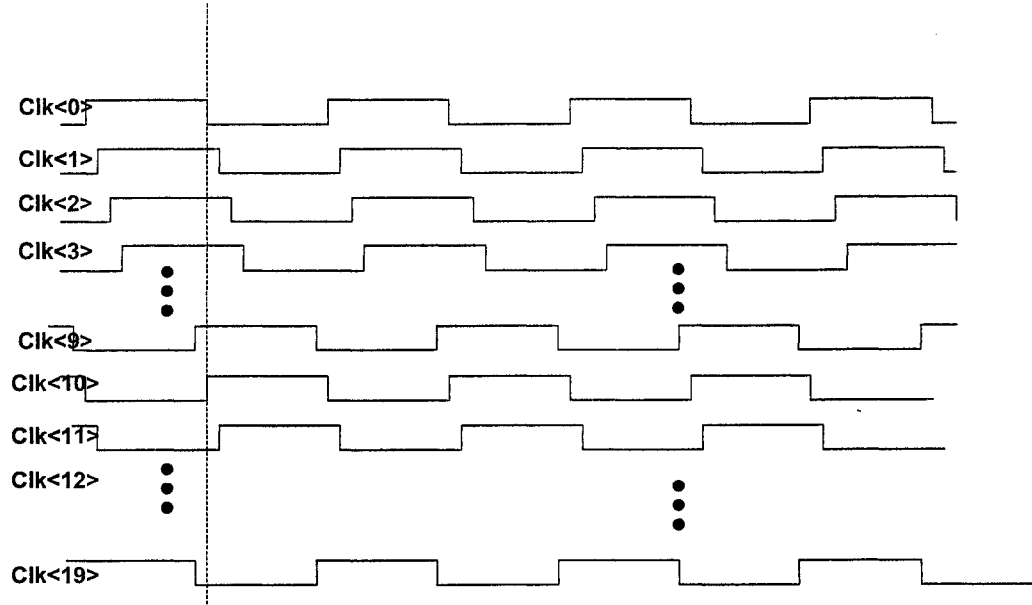


Figure 138 VCO Clocks.

Clocks CK1<1>, CK1<3>, CK1<5>, ..., CK1<19> are called *odd* clocks. Each one of these is an input to one sampler, LATCH and XOR as shown in Figure 141.

Clocks CK1<0>, CK1<2>, CK1<4>, ..., CK1<18> are called *even* clocks. Each one of these is an input to one sampler and XOR.

The difference between the loads of the *odd* and *even* clocks will result in a mismatch that will increase the BER of the receiver and reduce its jitter tolerance. This is mainly due to the fact that the samplers in each phase detector have two different types of clocks; one is *odd* and the second is *even*.

In order to enhance the jitter tolerance of the receiver, the two samplers have to have matched clocks, since they are the circuits that are most sensitive to mismatches in their clocks.

The matching between the two clocks that drive the samplers can be enhanced if the design can guarantee that the two clocks drive a matched load. This can be done by making the VCO provide another set of clocks specifically for the samplers of the phase detectors. This clock set is called CK3<0:19>. This means that the VCO will have the following sets of clocks:

1. CK1<0:19>: drives the LATCHes and XORs
2. CK2<0:19>: drives the digital section of the receiver
3. CK3<0:19>: drives the samplers in each phase detector

In order to differentiate between the new CK1 that drives the LATCHes and XORs only, and the old CK1 that drives the samplers, LATCHes and XORs, the new CK1 is called CK1n<0:19>.

The clocks that are coming out of the VCO are generated using a buffer. Instead of using another buffer to generate the new set of clocks; CK3<0:19>, the original buffer is split into two buffers, where each one of the new two buffers is sized according to its load. This is true only because the load is split into two parts as well.

Originally, the buffers in the VCO that generate CK1<0:19> are shown in Figure 139.a), where X denotes the strength of the buffer.

The new set of buffers that generates CK1n<0:19> and CK3<0:19> is shown Figure 139.b).

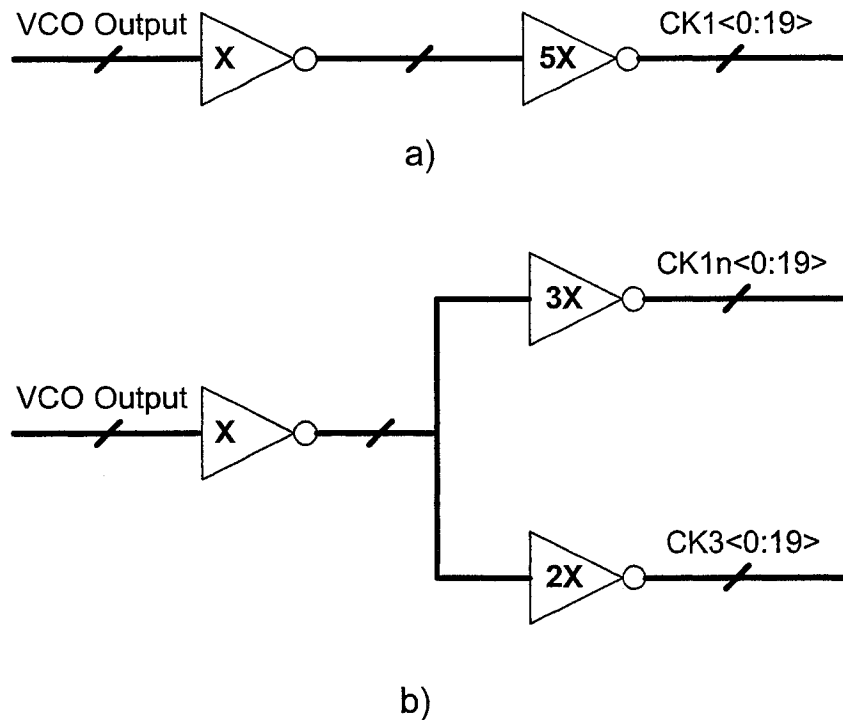


Figure 139 Buffer splitting to match the clocks that go to the samplers.

From Figure 139, the clocks that drive the samplers, CK1n<0:19>, are always matched. The clocks that drive the XOR and LATCHes have some mismatch among themselves because of a difference in their loads, however, those circuits are not sensitive to mismatches because their inputs are always stable around the clock transition.

This design is also explained in [6] where circuit implementation is also presented.

The novelty of this approach is that it presents an elegant method for matching the loads without any penalty. The loads of the sensitive circuits will always be matched with each other under all operating conditions. Usually, this problem is solved by adding dummy loads, which doesn't solve the problem completely because dummy loads will not result in a matched loads under all operating conditions and will result in more parasitics, area and power consumption.

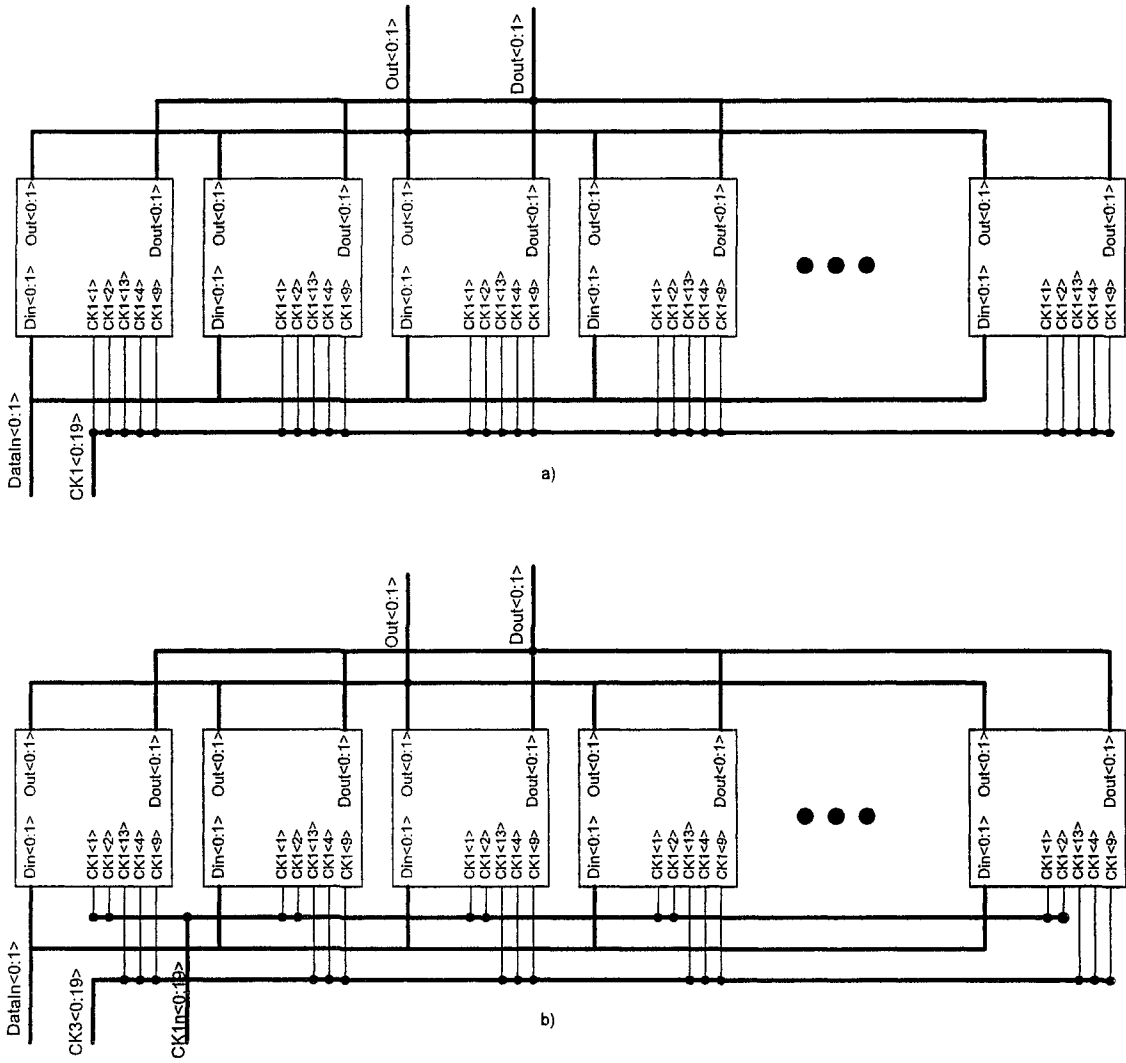


Figure 140 Original and new phase detector cell.

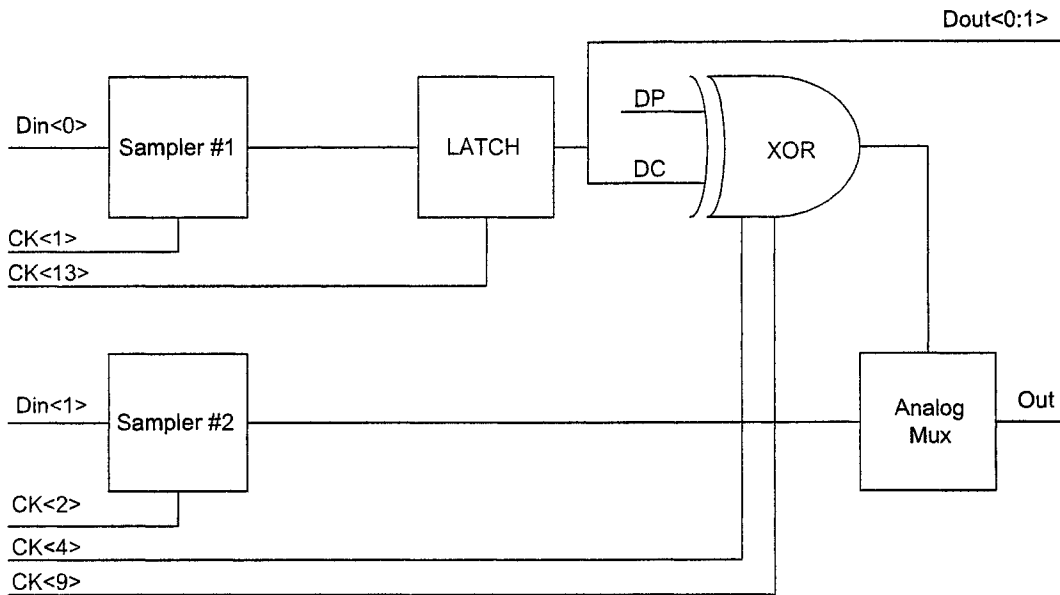


Figure 141 Phase detector implementation

8.6. Test Setup

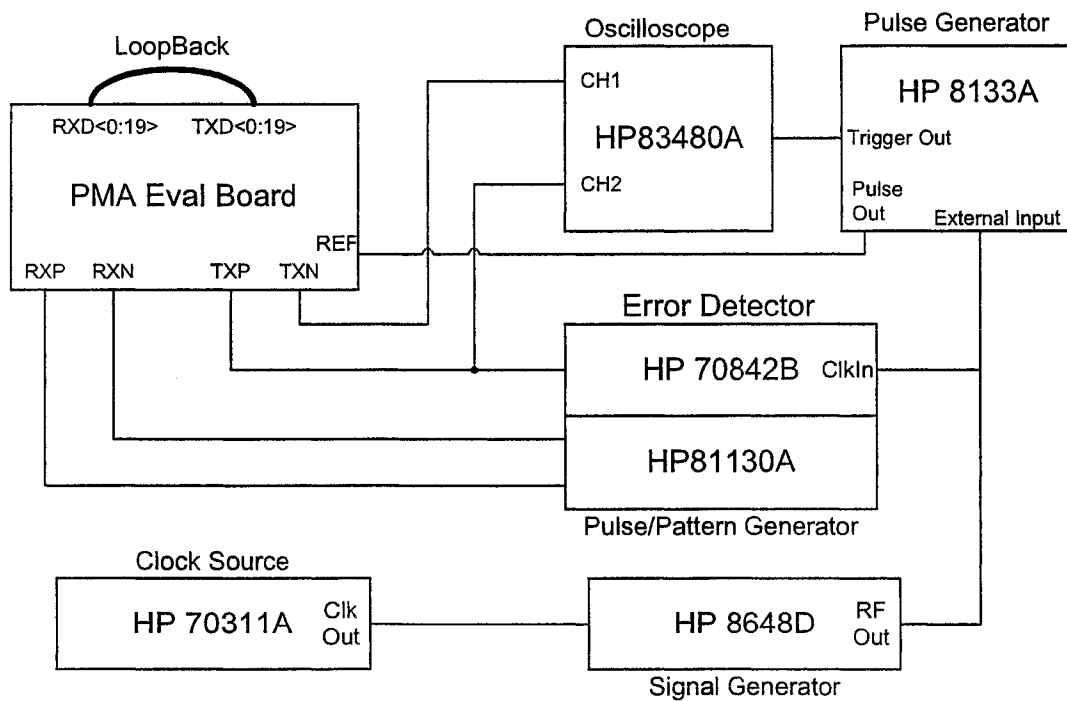


Figure 142 Test setup for the transceiver.

The test setup used to test the chip is shown in Figure 142. The setup is made of a board on which the chip exists, an oscilloscope to view the eye diagram of the transmitter, pattern generator that generates high speed differential data that are passed to the inputs of the receiver. There are some other clock sources that generate the reference clock and do some synchronization among all the machines.

8.7. Measured Results

Although full transceiver characterization has been done over all operating speeds, process corners, temperature and power supply variations, only those related to the receiver will be presented in this section. The chip showed reliable operation under all extremes.

8.7.1. Jitter Tolerance

Jitter tolerance is a key factor in determining the quality of the SERDES receiver. Jitter may be applied to the input of the receiver in three forms. Deterministic jitter may be applied through the use of long cables or additional FR4 trace length, random jitter may be added using a random noise, and periodic jitter may be included using a frequency synthesizer to modulate the input data.

In the testing, only periodic modulation was added to the input data to test jitter tolerance at specific frequencies. Modulation amplitude was increased until the number of bit errors increased above the 10^{-12} level. Figure 143 shows the jitter tolerance curves for a frequency modulation of up to 20MHz. This figure shows that for the 3.125Gb/s operation the SERDES is completely able to track out more than 1UI of jitter for modulation frequencies up to 4MHz. The tolerance drops off slightly after 4MHz and continues slightly lower for modulation frequencies up to 20MHz. For the 2.5Gb/s operation, the jitter tolerance stays at 1.0UI up to 8MHz, and drops to more than 0.7UI at 20MHz.

Most standards require a minimum 1UI of tolerance to periodic jitter at modulation frequencies of 1MHz or less. Above 1MHz, the tolerance requirements drop off quickly. For modulation frequencies of 2MHz or more, total jitter tolerance requirements can range from 0.5UI to 0.65UI depending on the application. Lab testing results show the jitter tolerance of this transceiver as meeting these requirements.

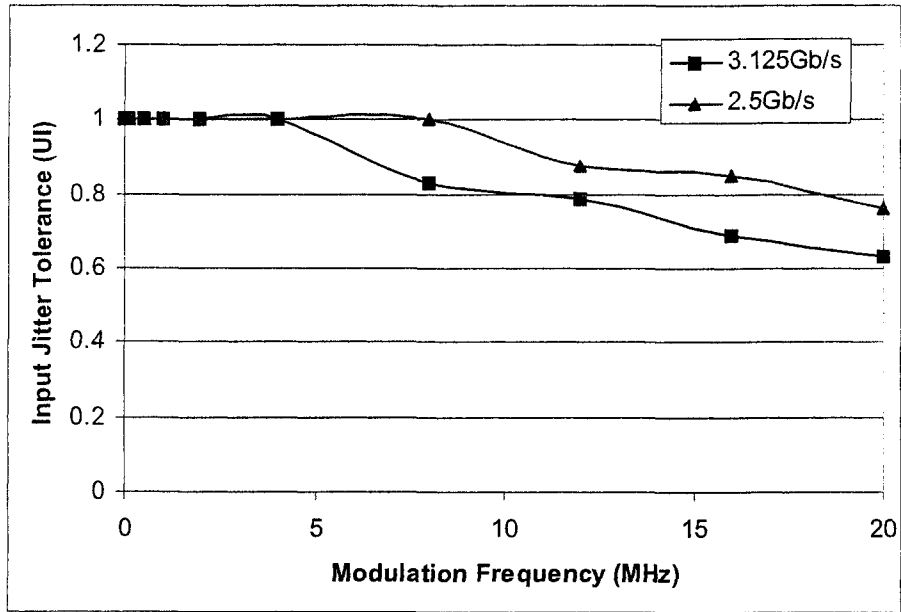


Figure 143 Receiver jitter tolerance for the 2.5Gb/s and 3.125Gb/s operations.

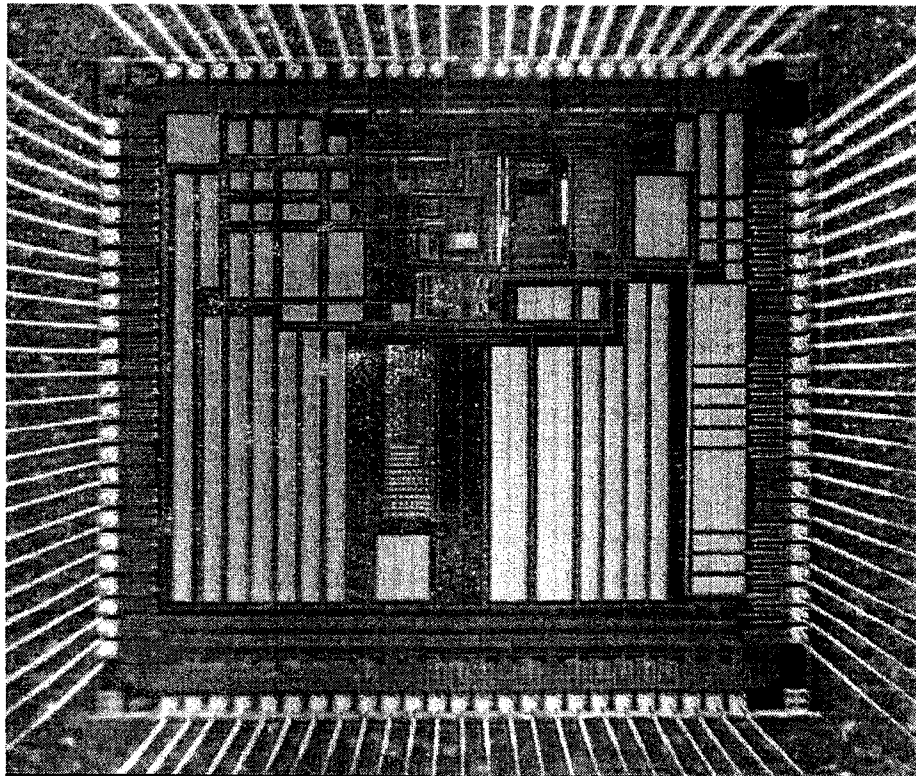


Figure 144 Die photo for the Transceiver.

Figure 144 shows a die photo of this transceiver while Table 8 presents a summary of the measurement results of the transceiver under the three speeds; 1.25, 2.5 and 3.125Gb/s.

Table 8 Performance Summary.

Test	Speed		
	1.25Gb/s	2.5Gb/s	3.125Gb/s
Random Jitter (rms)	7.6ps	3.3ps	2.9ps
Deterministic Jitter (pp)	27ps	24ps	32ps
1UI Jitter Tolerance Frequency	NA	8MHz	4MHz
20MHz Jitter Tolerance	NA	0.76UI	0.63UI
Max Transmission Distance over FR4 line w/o emphasis	40"	20"	10"
Max Transmission Distance over FR4 line w/ 30% emphasis	NA	60"	40"
Minimum Differential Eye Height for Error-Free Operation	210mV	180mV	280mV
Power Consumption (mW)	155	196	208
Core Area	1.2mm X 1.05mm		
Power Supply	1.8V		
Process	TSMC 0.18u 1P6M digital CMOS		

8.8. Summary And Conclusions

In this chapter, new concepts and design techniques that enhance the overall performance of high-speed CMOS transceivers have been presented and analyzed. The requirements of different circuits that make up the receiver have been carefully analyzed in order to build a robust receiver that is running at 3.125Gb/s from a 1.8V power supply. A systematic approach to implement the Gm circuit such that it will not interfere with the fine loop dynamics as well as not passing different sources of noise have been presented. This is particularly important because the Gm cell will be in the signal path when the receiver is operational, so, any impairment in its function will deteriorate the receiver performance.

The components of the phase detector; the samplers, the LATCH, the XOR and the analog mux have all been updated to run at a 1.8V power supply and a 3.125Gb/s data rate. This required that their bandwidth to be increased and to be redesigned such that they do not have power supply headroom problems.

The VCO is still running at $1/10^{\text{th}}$ of the data rate, which means that its nominal frequency has increased to 312.5MHz, so, it has to be reoptimized at this speed.

To enhance the jitter performance of the receiver, two patent pending techniques that match the VCO clocks as well as their loads have been presented. The effects of these techniques have been checked through extensive Spice and Matlab simulations. Measurements have shown that these techniques resulted in a very robust system that has low jitter and high jitter tolerance, which exceeded the requirements of many of the industry standards.

References

- [1] Behzad Razavi, *Monolithic Phase-Locked Loops and Clock Recovery Circuits: Theory and Design*, IEEE Inc., 1996.
- [2] A. Fiedler, R. Mactaggart, J. Welch, and S. Krishnan, "A 1.0625Gbps Transceiver with 2x-Oversampling and Transmit Signal Pre-Emphasis", *ISSCC Digest of Technical Papers*, FP15.1, pp. 238 -239, 464 , Feb. 1997.
- [3] R. Gu, J.M. Tran, H.-C. Lin, A.-L. Yee, and M. Izzard," A 0.5-3.5Gb/s Low-Power Low-Jitter Serial Data CMOS Transceiver", *ISSCC Digest of Technical Papers*, WA20.4, pp. 352 –353, Feb. 1999.
- [4] Moises E. Robinson and Bernie Grung, "Phase Lock loop and Transconductance Circuit for Clock Recovery," Patent pending 1999.
- [5] Ahmed A. Younis, Moises E. Robinson, Micheal Nix and Brian Brunn, "Ring Oscillator Layouts with Improved Signal-Path Matching for High-Speed Data Communications," Patent pending 2001.
- [6] Moises E. Robinson and Ahmed A. Younis, "Clock Distribution Scheme for Improved Jitter Performance," Patent pending 2001.
- [7] P. Larsson, "Parasitic Resistance in an MOS Transistor Used as On-Chip Decoupling Capacitance", *IEEE Journal of Solid-State Circuits*, Vol. 32, pp. 574 -576, April 1997.

CHAPTER 9. Design Techniques and Engineering Practice For High-Speed Analog ICs

9.1. Introduction

The purpose of this chapter is to collect the different design techniques and engineering practices that were used in this dissertation to analyze and design high speed analog circuits. These techniques can serve as a guideline for designing systems that need to exhibit aggressive noise and jitter performance.

In particular, section 9.2 will discuss issues and design considerations related to designing operational amplifiers from both theoretical and practical point of views. Issues related to designing and implementing comparators are presented in section 9.3. As a critical component in many analog circuits, section 9.4 describes techniques to implement matched capacitors in a pure digital process. It presents an approach to match two capacitors with each other as well as three capacitors with each other. Thermal noise, charge injection and clock feedthrough are all sources of errors in analog circuits and may deteriorate their performance if not considered carefully. Sections 9.5, 9.6, and 9.7 present detailed analysis of thermal noise, charge injection and clock feedthrough respectively, in order to enable the designer to carefully implement the analog circuit such that their effect is minimized.

As a major player in many analog systems, issues related to designing Gm cells for proper functionality are presented in section 9.8. These issues must be considered carefully and separately as they affect the system differently.

Section 9.9 presents a patent pending and elegant approach for clocking heterogeneous loads, which will greatly improve the performance of analog circuits

Different techniques that are commonly used in analog circuits to reduce the parasitics as well as to result in a better matching are presented in section 9.10. Although the techniques are presented for different ADC components, they are applicable to almost any analog circuit layout.

Many of these sections have already appeared earlier in different parts of this dissertation. This chapter's goal is to collect all these concepts in a single coherent design guide.

9.2. Design of the operational amplifier

There are many variables that enable the designer to decide on a specific architectures of the OA. Usually, an OA with the following characteristics is required:

- High SNR and SNDR.
- Large SFDR.
- Low power, small area and large accuracy.
- High speed (small slewing and settling times).
- Large Input/Output swing.
- Low voltage.
- Designed in an inexpensive process.
- High CMRR and PSRR.

In addition to other parameters that depend on the application and/or the process of design. The current design has the following parameters:

- CMRR and PSRR are not important.
- Large Input/Output swing.
- Open loop DC gain is greater than 8,000 at nominal conditions of operation.
- Gain-Bandwidth greater than 500MHz.
- High SNR and SNDR.
- Low power.

Single-stage operational amplifiers are preferred over two-stage operational amplifiers because they usually have higher bandwidth.

9.2.1. Operational Amplifier: Theoretical Analysis

Although there are many factors that affect its operation, the accuracy of the operational amplifier is measured mainly by its dc gain. An operational amplifier configured in a closed loop feedback configuration is shown in Figure 145.

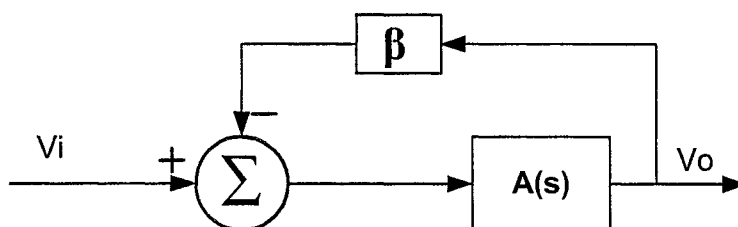


Figure 145 Feedback model of operational amplifier.

The transfer function of the closed loop gain is given by:

$$T(s) = A_{CL}(s) = \frac{A(s)}{1 + \beta \cdot A(s)} \quad (1)$$

Where β is the feedback factor and $A(s)$ is the open loop gain of the opamp.

In the design of this ADC, a single stage fully differential folded cascode was used. The open loop gain of the opamp can be given by:

$$A(s) = \frac{w_u}{s} \quad (2)$$

Where, w_u is the unity gain frequency in radians.

$$T(s) = A_{CL}(s) = \frac{w_u}{s + \beta w_u} = \frac{1}{\beta} \cdot \frac{1}{1 + \left(\frac{s}{\beta w_u} \right)} \quad (3)$$

Which means that the closed-loop opamp has a dc gain, at $s = 0$, equals to $1/\beta$ and it has a -3dB frequency given by:

$$w_{-3\text{dB}} = \frac{1}{\beta w_u} \quad (4)$$

The transfer function $T(s)$ relates the output to the input as:

$$T(s) = A_{CL}(s) = \frac{V_o(s)}{V_i(s)} \quad (5)$$

For a step input, $V_i(s) = V_s/s$, and thus,

$$V_o(s) = \frac{V_s}{s} \cdot \frac{1}{\beta} \cdot \frac{1}{1 + \left(\frac{s}{\beta w_u} \right)} = \frac{V_s}{\beta} \left[\frac{1}{s} - \frac{1}{s + \beta w_u} \right] \quad (6)$$

Where, V_s is the magnitude of the step input.

Taking the inverse Laplace transform to find the time domain response:

$$v_o(t) = \frac{V_s}{\beta} \left(1 - e^{-\frac{t}{\tau}} \right) \quad (7)$$

$$\text{Where, } \tau = \frac{1}{\beta w_u}$$

One can clearly see that for fast opamps, τ needs to be small which requires both large β , or feedback ratio, and large unity gain frequency.

Equation (7) also states that since the settling time is finite, there will be a settling error that is equal to $e^{-t/\tau}$.

For example, if a 1.0% accuracy is required, then one must allow $e^{-t/\tau}$ to reach 0.01, which is achieved at a time of 4.6τ . For settling within a 0.1 percent accuracy, the settling time needed becomes approximately 7τ .

The above analysis assumes that the opamp has an infinite dc gain, which is not the actual case. In practice, however, the opamp has a dc gain, A_0 , at $s = 0$, thus it can be modeled as:

$$A(s) = \frac{A_0}{1 + \frac{s \cdot A_0}{w_u}} \quad (8)$$

Substituting equation (8) in equation (1) will give the closed-loop gain, $A_{CL}(s)$ as:

$$A_{CL}(s) = \frac{1}{\beta'} \cdot \frac{1}{1 + \left(\frac{s}{\beta' w_u} \right)} \quad (9)$$

$$\text{Where, } \beta' = \frac{1}{\frac{1}{A_0} + \beta}$$

So, the error due to finite opamp gain can be approximated by:

$$err = \frac{1}{A_0 \beta} \quad (10)$$

So, for a 1 percent error gain, $err < .01$, or equivalently, for $\beta = 0.5$, the required dc gain, A_0 is > 200 .

Now, lets go back and derive the specifications from the above equations.

When excited with a step input, the opamp goes in two regions. In the first region, the opamp will slew if the output current is not enough to charge the output capacitors in exponential fashion as equation (7) states.

This is usually the case in most of the implementations, since the opamp will be designed with minimum power consumption.

The second region of operation will be the settling region where the opamp will settle to its final output. In a switch capacitor design, the opamp will be given a time to finish both regions of operations. This is usually going to be half a period when its output is valid. Assuming that the total time given to the opamp is t , then

$$t = t_{sl} + t_{ss} \quad (11)$$

Where, t_{sl} is the time needed for the opamp to finish slewing, and t_{ss} is the time needed by the opamp to finish settling.

As a rule of thumb, t_{sl} is usually set to 20% of t , and t_{ss} is set to 80% of t . So, for example, if we want to design an opamp that uses a resetting architecture and runs at 100MHz, then $t = 5ns$, $t_{sl} = 1ns$ and $t_{ss} = 4ns$. This means that the opamp should be able to finish slewing in 1ns and finish settling in 4ns.

If we assume we want a settling error to be 0.1 percent, then $7\tau = 4ns$ or equivalently, $\tau = 0.57143ns$. Using equation (8) with $\beta = 0.5$, $w_u = 3.5\text{Grad/s}$ which is approximately 560MHz.

Now let us consider a practical example. Suppose we want to design a 10-bit pipeline ADC that runs at 100MHz. The first thing to decide is how many bits each stage should resolve. If the 100MHz speed is too tough to achieve in a certain process, like CMOS, then we need to consider low number of bits per stage, since the higher the resolved bits in each stage the higher the required gain of that stage will be, which means the lower the feedback ratio. This will result in τ being large. The minimum number of bits per stage is 1, and let's say that we decided that we want 1 bit per stage ADC.

The second step is to determine β . This depends on the configuration of each stage that will achieve the required gain. For a 1-bit per stage, the required gain is 2. There are some configurations that achieve this gain with $\beta = 0.5$ and some others achieve it with $\beta = 0.33$. Clearly, for higher speed, we want to consider the one with $\beta = 0.5$. So, now, we have $\beta = 0.5$ determined. The next step is to find the required unity gain of the operational amplifier and its dc gain. The unity gain requirement can be derived from the speed of the ADC, which is 100MHz. As derived above, the opamp needs to have at least a unity gain of 560MHz.

There are many sources of errors in the ADC. For simplicity, let's consider that the finite dc gain of the opamp and not enough settling time are the only sources of errors. In general, all the sources of errors should contribute to less than half an LSB. Since we have only two sources, each should contribute at most one quarter of LSB. This, in turn, means that the dc gain of the opamp should be accurate to more than 12 bits for our 10-bit ADC and so as the settling error of the opamp. So, the required accuracy of the finite gain as well as the settling time should be less than $1/2^{12}$, which is 0.0244 percent.

To find the required unity gain of the opamp, the 4ns should be equal to 8.4τ , making τ equal to 0.48ns. Using equation (7), the unity gain frequency, $f_u = 670\text{MHz}$.

The dc gain of the operational amplifier is found using equation (10), where $err = 0.0244$ percent. Thus, A_0 should be at least 8192.

9.2.2. Operational Amplifier: Practical Design

The operational amplifier is a standard fully differential single stage folded cascode with boosting amplifiers as shown in Figure 146. The boosting amplifiers, shown in Figure 146 as BN and BP, are also fully differential folded cascode opamps. The operational amplifier that uses the boosting amplifiers is called the main opamp, while the boosting amplifiers are always referred to as boosting amplifiers. Single stage design has been considered to give better frequency response in addition to the fact that it is more stable over temperature, process and power supply variations. Although it has a worse frequency response than the regular differential opamp, the folded cascode boosting opamp was chosen over the regular differential opamp because it can be designed with higher gain. The two different boosting amplifiers were used instead of single ended design because they give higher gain, in addition to the fact that they are more area efficient since we need only two of them instead of four in the case of single ended design. The boosting amplifiers are of two types: the BN has an NMOS differential input stage, while the BP has a PMOS differential input stage. As shown in Figure 146, the

inputs of the BN boosting amplifier come from the drains of M10 and M11 transistors, which are supposed to be biased in the saturation region and have a drain-to-source voltage, $V_{ds} < -0.5V$. This means that the inputs to the differential pair of BN are going to be around 2V, hence an NMOS differential input stage is required. The bottom boosting amplifier, BP, has its inputs coming from the drains of M4 and M5 which are supposed to be biased at $V_{ds} < 0.5V$, hence a PMOS differential input stage is required.

The NMOS type boosting amplifier, BN, with its continuous time common mode feedback circuit, CMFB, is shown in Figure 147. It is very similar to the main opamp with the exception that it doesn't have boosting amplifiers and that the tail current source that consists of M1 and M1x transistors is cascoded so as to increase the source voltage of transistors M2 and M3. This is to decrease the excess bias voltage of those transistors in order to guarantee that they are in the saturation region of operation when they have a common mode voltage input applied to their gates. The CMFB circuit consists of all transistors Mc1-Mc9. The main function of the CMFB circuit is to set the common mode voltage of the output nodes, V_{op} and V_{on} , to the biasing voltage of transistors M8 and M9.

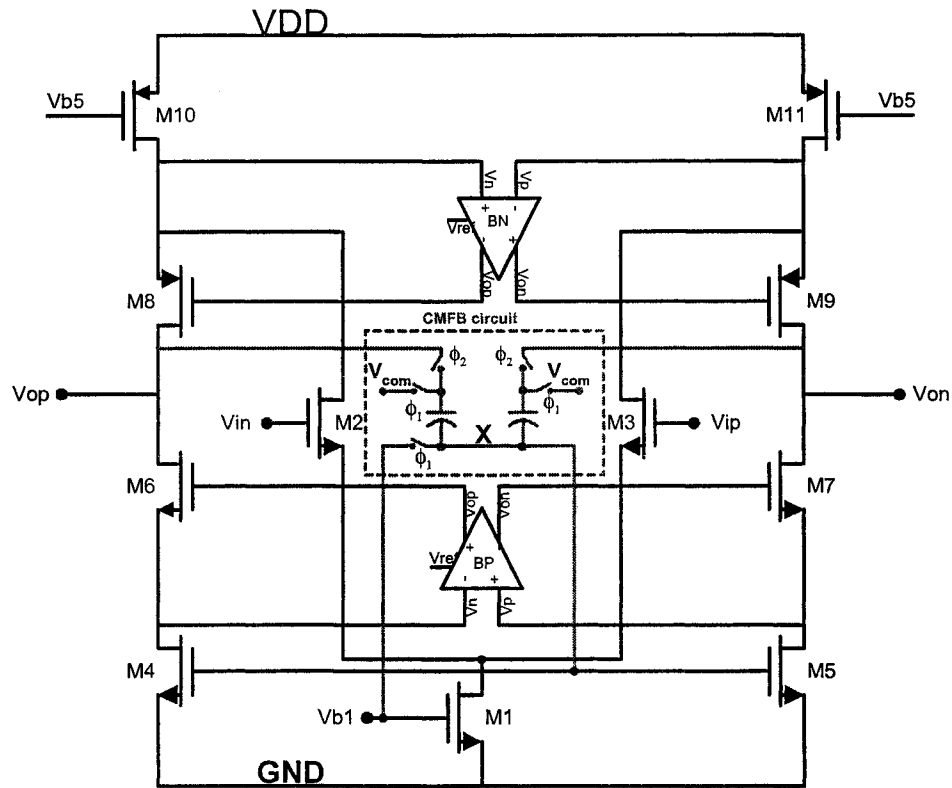


Figure 146 Main operational amplifier with boosting opamps and CMFB circuit.

9.2.3. CMFB circuit design of the main amplifier

The common mode voltage of the opamp can be controlled by many transistors. Using one side of the opamp, any one of transistors M1, M4 and M10 can be used to control the common mode voltage of the opamp. In this design, M4 was chosen. The relationship between the voltage at the gate of M4 and the common mode voltage is inverted. As the voltage at the gate of M4 increases, the common mode voltage drops and vice versa.

To maximize the output swing of the operational amplifier, a switched capacitor CMFB circuit is utilized to keep the common mode output voltage at the required level. The CMFB circuit is shown in red in Figure 146 and consists of 2 capacitors and couple of switches. The two capacitors have the same value which should be chosen such that it is not too large to load the main opamp or too small to be affected by the charge injection of the switches. The sizes of the switches should also be chosen carefully so that they won't have great effect on the capacitors. The operation of the CMFB circuit is as follows. The CMFB circuit works in two phases. In the sample phase of the opamp, the output of the opamp are disconnected from the CMFB circuit and V_{com} is connected instead, while, when being in the hold mode, the capacitors are disconnected from V_{com} and, then, connected to the output of the main opamp. V_{com} represents the required common mode voltage of the operational amplifier and it is set in this design to 1.25V. The second side of the capacitors are connected to the biasing voltage of the transistors used at nominal conditions. This will be illustrated soon.

At nominal conditions and without the CMFB being connected to the opamp, transistors M4 and M5 are designed to be biased with V_{bj} . With V_{bj} biasing M1, M4 and M5, the common mode of the output voltage of the main opamp is around V_{com} .

The two capacitors average the output of the opamp, with node X being set by V_{bj} in the sampling phase. If the common mode voltage of the output of the main amplifier comes to be similar to one set by design, then node X in the hold mode will also be similar to V_{bj} . If the common mode voltage of the outputs increases, the voltage at node X will increase to more than V_{bj} , which will increase the biasing voltage of the gates of M4 and M5 and thus, decreasing the common mode voltage of the output. If the common mode voltage of the outputs of the main opamp is less than that set by design, the voltage at node X will drop to below V_{bj} and thus increasing the common mode voltage of the outputs of the main opamp, and thus, the output of the main opamp will be kept close to the voltage set by design.

9.2.4. CMFB circuit design of the boosting amplifiers

Designing the CMFB circuit for the boosting amplifiers is a straightforward process. The output of the each boosting amplifier doesn't need to swing too much, thus, a continuous time CMFB circuit can be used. The first step is to design the boosting amplifier without the CMFB circuit such that the common mode output of the opamp is around $V_{dd}/2$. Once this is finished, part of the output current is generated by the CMFB circuit using transistors Mc8 and Mc9. For example, suppose that after designing the opamp without the CMFB circuit, the

W/L ratio of M4 and M5 comes to be 4. If we assume that one quarter of the output current will be provided by the CMFB circuit, then W/L of both M4 and M5 will be reduced to 3. With the CMFB circuit being not connected, half of the current in M11 will be in M5, so if Mc1 is made $1/4^{\text{th}}$ of M11, then $1/4^{\text{th}}$ of the current in M11 will be in Mc1. If V_{ref} equals the common mode voltage of V_{on} and V_{op} , then Mc2-Mc7 are designed such that its current in Mc4 is the same as the current through both of Mc2 and Mc3 together. This means that the current through Mc4 is $1/2$ of the current in Mc1, or equivalently, the current in Mc4 is $1/8^{\text{th}}$ of that of M11. Since the current in the path of M9 and M7 is $1/2$ of M11, then the current of Mc4 is $1/4^{\text{th}}$ of that in M9 or M7. So, $1/4^{\text{th}}$ of the current of M9 or M7 will be provided by the CMFB circuit, and the rest is provided by M5 which will be $3/4^{\text{th}}$ of the current. This is why the W/L ratio of M5 was reduced from 4 to 3 to represent the $3/4$ portion of the current.

V_{ref} in Figure 147 is set externally to the biasing voltage of M8 and M9 of the main opamp. This gives us the opportunity to test the main amplifier with or without the boosting circuits, since this voltage will be fed to the V_{ref} or to the transistors directly.

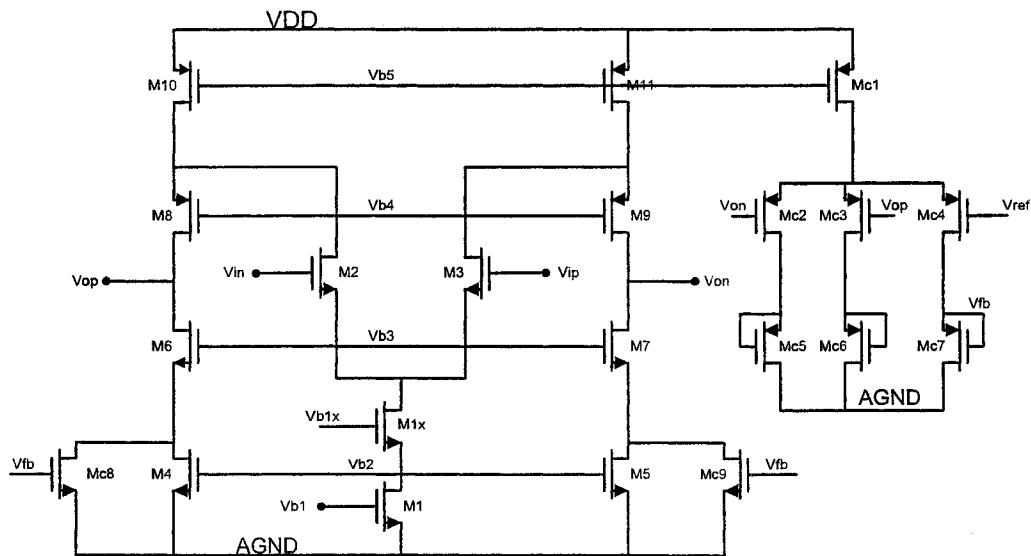


Figure 147 Boosting amplifier with NMOS differential input stage.

The BP boosting amplifier is the same as the NMOS type with the exception that a PMOS differential input stage is used in addition to an NMOS CMFB circuit instead of the PMOS one used above.

9.3. Comparator implementation

When designing a comparator, here are the main issues that need to be considered:

- Comparator offset voltage: This is defined as the voltage by which the input voltage needs to exceed the reference voltage of the comparator such that the output of the comparator changes its state.

- Comparator gain and metastability: If the difference between the input of the comparator and its reference input is below a certain value, called the *minimum resolvable signal* (V_{mrs}) value, the output of the comparator will not be able to reach the rails. This value will be determined by the comparator gain. The higher the gain, the smaller the value of V_{mrs} . If the difference between the input and the difference is less than V_{mrs} , the comparator is said to be in the metastable state.
- Kickback noise: It is the noise coupled from the output of the comparator to its inputs. This usually happens because the outputs of the comparator are high swing.
- Speed: Because of the speed requirement, a static comparator architecture has been chosen, although it will burn more power than dynamic comparators.

The comparator circuit that implements regeneration is shown in Figure 148. This comparator consists of two-stage preamp in order to decrease the minimum resolvable signal by increasing the gain of the preamp, which will increase the resolution of the overall comparator. The differential amplifier in the dashed box provides the difference circuit that amplifies the difference between V_{ip} and V_{rp} and also V_{rn} and V_{in} . Since this is a differential amplifier, then if V_{ip} is greater than V_{rp} , this should also guarantee that V_{rn} is greater than V_{in} . If V_{ip} is greater than V_{rp} , then Node B is at higher voltage than node A. This is because, if V_{ip} is higher than V_{rp} , the current in M3 transistor is larger than the current in M4 transistor due to the larger excess bias voltage on M3 than it is on M4. Those two currents in M3 and M4 will pass through the load; M0 and M1 respectively. Larger current in M0 than current in M1 means larger voltage drop on M0 than M1, which, in turn, means that the voltage at node A is less than the voltage at node B. Same thing applies to the bottom differential opamp in the box. If V_{in} is smaller than V_{rn} , Node B will be pushed further up and node A will be pushed further down, and hence this differential configuration will enhance both the speed by helping the upper differential amp to push the node voltages up and down, and the resolution by increasing the dynamic input range by two.

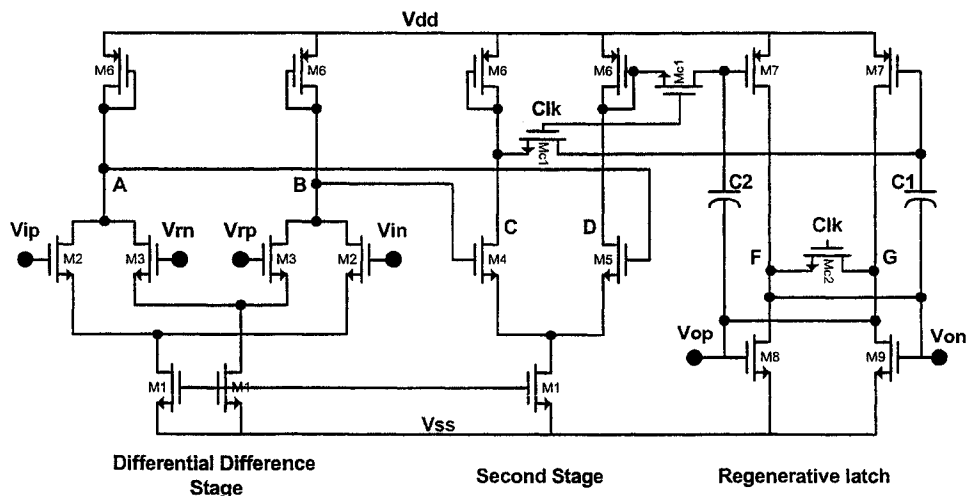


Figure 148 Static comparator with latch.

Same analysis applies to the second stage of the preamp. If node B is higher than Node A, this will cause the voltage at node C to be higher than that at node D. When the Clk is High, the bottom plates of the two boosting capacitors, C1 and C2, are connected together while the upper plates are connected to Nodes F and G, which are also connected to nodes C and D respectively. This due to the fact that the two transistors; M19 and M20, makes a short circuit and the regenerative latch is disabled.

So, when Clk is High, and V_{ip} is higher than V_{rp} , B will be higher than A, C will be higher than D, and so, the top plate of C1 will be at higher voltage than the top plate of C2.

When the clock is turned off, M19 and M20 transistors turn off and disconnect nodes C and D from nodes F and G respectively, while transistor M18 enables the regenerative latch. Since the top plate of C1 is at higher voltage than that of C2, then the excess bias of M12 transistor is larger than that of M13, which means that larger current will be going into M15 than that of M14. Since M14 and M15 behave as the loads of M13 and M12, respectively, V_{ds} of M15 will be larger than that of M14, and since the latch is in a positive feedback, that will push V_{ds} of M15 further to increase, while V_{ds} of M14 to decrease. V_{ds} of M15 will rail to V_{dd} while V_{ds} of M14 will rail to V_{ss} . One important issue in the design of this comparator that will affect the speed of the comparator is the size of the boosting capacitors. The size of the capacitors is chosen such that its kT/C effect is less than the accuracy required to be provided by the comparator, so, it should be larger than C_{min} , where C_{min} is determined from the kT/C requirement. The upper limit of the boosting capacitor, M_{ax} is determined from the speed of the comparator. The second stage of the preamp will source current to or sink current from any of the boosting caps. When the Clk switches from Low to High, and stays in the High period, the current sourced to or sinked from the caps should be able to reach its steady state before the Clk changes to Low, otherwise, the comparator might make wrong decision. So, the maximum output current of the second stage of the preamp as well as the time in which the voltage at the top plates of the capacitors will settle will determine the maximum size of the capacitors. Larger capacitor means it needs more time for the second stage of the preamp to be able to charge it, which means slower operation of the comparator.

One more thing regarding the operation of the comparator. The comparator should follow the output of the opamp of its stage. But, it should shut off just before the opamp does so, or putting it differently, it should shut off right when the next stage starts the holding mode. This results in that the clock of the comparator should follow Phi2 of the stage where the comparator belongs, but it should shut off a little earlier than Phi2. So, the clock generator will generate another signal like Phi2, but it shuts off earlier than Phi2.

9.4. Metal capacitor design

The TSMC process is a digital process that doesn't include a high precision capacitor. We implemented the capacitors in our design by using the four layers of metal; Metal2, Metal3, Metal4 and Metal5, as a sandwich capacitor. We depend on the parasitic capacitance between each two layers to make our

capacitors. A capacitor has Metal2 and Metal4 connected with each other, while Metal3 and Metal5 are also connected with each other to make the second plate of the capacitor.

Especial layout techniques have been taken into consideration to increase the matching of the capacitors. The input/output relationship of the operational amplifier circuit in each stage is given by:

$$v_{od} = \left(1 + \frac{C_s}{C_f}\right) v_{id} - \frac{C_s}{C_f} v_x \tag{12}$$

Each stage of the ADC has a gain of 2. The gain of each stage is represented by the first bracketed term in equation (26), which shows that C_s/C_f should have a value of 1. In order to get the required accuracy of a specific stage, C_s should be matched with C_f to the accuracy of the that or better. So, from gain accuracy stand point, absolute values of C_s and C_f are not as important as matching the capacitors with each others, which means that the two capacitors of every stage have to be matched to the accuracy of that stage or better.

To achieve this, especial layout techniques were followed such as common centroid and interdigitization. These two techniques were used together as shown in Figure 149, where one of the capacitors is called A, while the other is called B. Each capacitor is divided into eight smaller ones so that they can be interdigitized with those of the second capacitor. This procedure was followed in case there is a horizontal, vertical, or diagonal gradient in the process, its effect will be minimized.

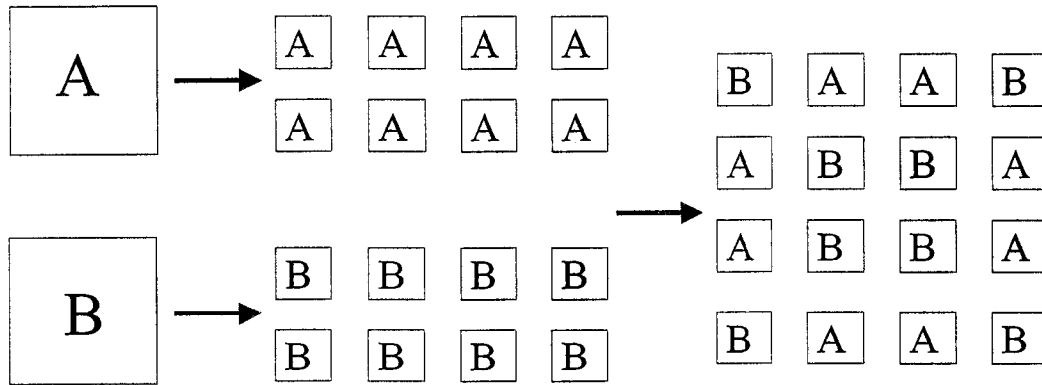


Figure 149 Common Centroid layout.

This approach is further expanded to match three capacitors with each other. Each capacitor is divided into 12 smaller units that are interdigitized as shown in Figure 150. Section 4.3.1 presents more information on capacitor mismatch effect on ADC performance.

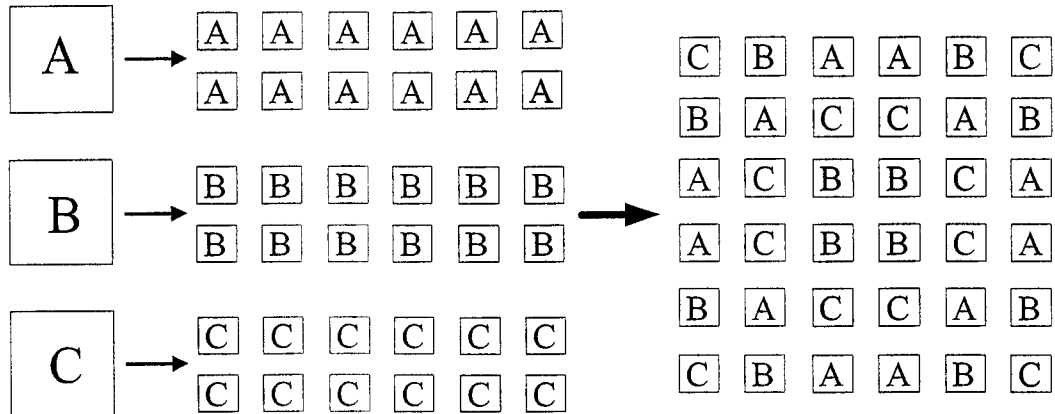


Figure 150 Common-centroid layout for 3 capacitors.

9.5. Thermal noise

Thermal noise is caused by the random motion of electrons. All particles at temperatures above absolute zero are in random motion. Since electrons carry charge, the thermal motion of electrons results in a random current that increases with temperature. This noise current is present in all circuits and corrupts any signals passing through. In a pipelined analog to digital converter, the first stage circuit is the most important source of noise. Two noise sources are significant: the sampling switches and the operational amplifier. The noise in the sampling switch comes from the fact that practically when it turns on to it has a finite resistance. The sampling switch is used to sample the input signal onto a sampling capacitor. As this happens, noise from the sampling switch is sampled with it onto the sampling capacitor. This operation is illustrated in Figure 151 where the noise *rms* value is:

$$\sqrt{V_{noise}^2} = \sqrt{\frac{kT}{C}} \quad (13)$$

Where k is the boltzman's constant = $1.38e-23$, T is the temperature in Kelvin and C is the sampling capacitor. As an example, if $C = C_s = 1pF$, then the *rms* kT/C noise is $64\mu V$.

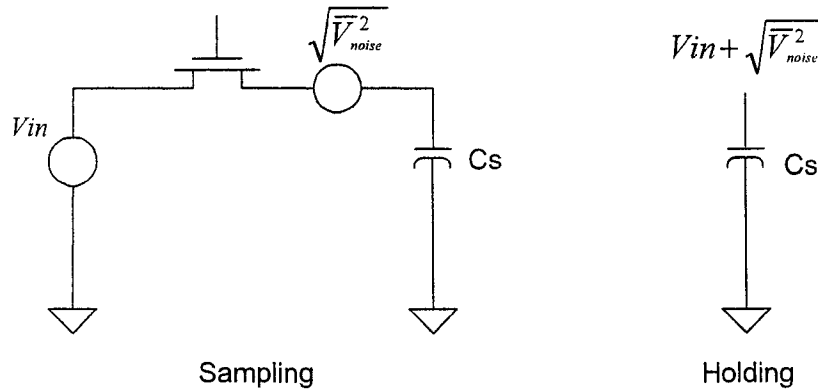


Figure 151 Thermal noise modeling.

This type of thermal noise is commonly referred to as kT/C noise because the noise power is proportional to kT/C where C is the size of the sampling capacitor. The operational amplifier also contributes thermal noise degradation to the signal being processed. The contribution of the sample and hold amplifier is also inversely proportional to a capacitance. In a single stage amplifier, it is inversely proportional to the load capacitance. In a Miller compensated amplifier it is inversely proportional to the compensation capacitance.

When designing an operational amplifier, usually minimum capacitor sizes are required for many reasons. The thermal noise puts lower limit on the size of the used capacitors. For example, for a 12-bit resolution ADC, the thermal noise of the overall ADC should be less than 1LSB. Since there are many sources of errors in the overall ADC, we might give the thermal noise a budget of 1/4LSB, which will be equivalent to 0.153mV. So, the *rms* thermal noise should be less than that, thus:

$$\sqrt{\frac{1.38e-23 * 300}{C}} \leq 1.53e-4 \quad (14)$$

Or, $C \geq 177 \text{ fF}$. This suggests that for a 12 bit ADC with 2.5V, the minimum size capacitor is 200fF so that the thermal noise is not the major contribution to the overall linearity.

Thermal noise is perhaps the most fundamental source of error in a pipelined ADC. Because it is random from one sample to the next, it is not easily corrected by calibration. Thermal noise can be alleviated by using large components or by oversampling. However, for a fixed input bandwidth specification, both of these remedies increase the power dissipation. Thus, a fundamental tradeoff exists between thermal noise, speed, and power dissipation.

9.6. Charge injection

Charge injection is the injection of charge from a transistor when it turns off into its nodes. Usually, this problem arises when a transistor is used as a switch. In this mode of operation, the transistor operates in the triode region, where V_{gs} usually goes to one of the rails depending on the transistor type. To understand this, we

need to analyze a transistor in its triode region of operation. Lets consider an NMOS transistor. When the transistor is turned ON, V_{gs} needs to be HIGH which means that $V_{gs} \gg V_{th}$. Since the transistor is working in its triode region, V_{ds} needs to be very small and ideally, it should be 0. For the purpose of this analysis, we will assume that V_{ds} is very small compared to $V_{gs} - V_{th}$. When the transistor operates in the triode region, an inverted channel occurs which behaves as a conductor. This will create a virtual capacitor that has the gate and the inverted channel as its two plates, and the gate oxide material that is under the gate as its insulator.

The amount of charge per unit area that can be stored in this capacitor can be approximated by:

$$Q'_{ch} = C_{ox} (V_{gs} - V_{TH}) \quad (15)$$

And the total charge stored in the channel will be:

$$Q_{ch} = C_{ox} (V_{gs} - V_{TH}) \times W \cdot L \quad (16)$$

When the transistor turns OFF, Q_{ch} will be dumped to the source and drain of the transistor as shown in Figure 152. Although the percentage of the total charge that is dumped to the drain is not exactly determined, many people assume that to be 50%. The charge that is dumped to v_{in} is not problematic, since v_{in} is a source-driven node, but the charge injected to the sampling capacitor will cause a voltage change on the capacitor. If we assume that the gate voltage rails to V_{dd} when the switch is ON, and that 50% of the total charge stored in the transistor will be dumped to the capacitor, the change in voltage on the capacitor due to charge injection is:

$$\Delta v_{load} = -\frac{C_{ox} (V_{DD} - v_{in} - V_{TH}) \times W \cdot L}{2 C_s} \quad (17)$$

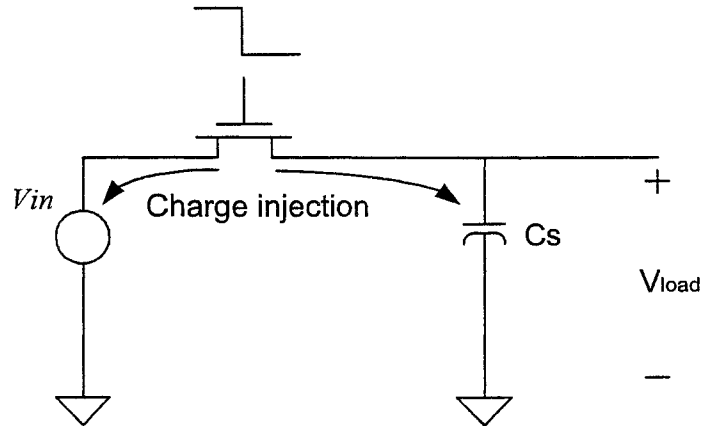


Figure 152 Charge injection for an NMOS switch transistor.

Equation (17) shows that the change in the voltage is signal-dependent which will result in signal dependant distortion of the signal. What makes things even worse is that the threshold voltage is also signal-dependant which will deteriorate the harmonic distortion of the circuit. The overall effect of charge injection on the system is that it adds to the nonlinearity of the system and causes the total harmonic distortion to drop.

9.7. Clock feedthrough

The clock feedthrough comes from the fact that a coupling exists between the gate of the transistor and its source and drain through two overlapping capacitors: C_{gs} and C_{gd} , where C_{gs} is the gate-to-source overlapping capacitor and C_{gd} is the gate-to-drain overlapping capacitor. As with the charge injection, when the transistor turns ON, the drain of the transistor is driven by the input signal and there is no clock feedthrough. When the clock signal that drives the gate of the switch turns OFF, a capacitive voltage divider exists between the gate-drain capacitance and the sampling capacitor as shown in Figure 153 where the overlapping capacitance is assumed to be half of the gate capacitance.

This will result in a voltage change on the sampling capacitor, C_s , according to the following equation:

$$\Delta v_{load} = \frac{C_{overlap} \cdot \Delta V_{gs}}{C_s + C_{overlap}} \quad (18)$$

Where $C_{overlap}$ is the overlapping capacitance value,

$$C_{overlap} = C_{ox} \cdot W \cdot LD \quad (19)$$

Where LD is the length of that overlaps the drain/source.

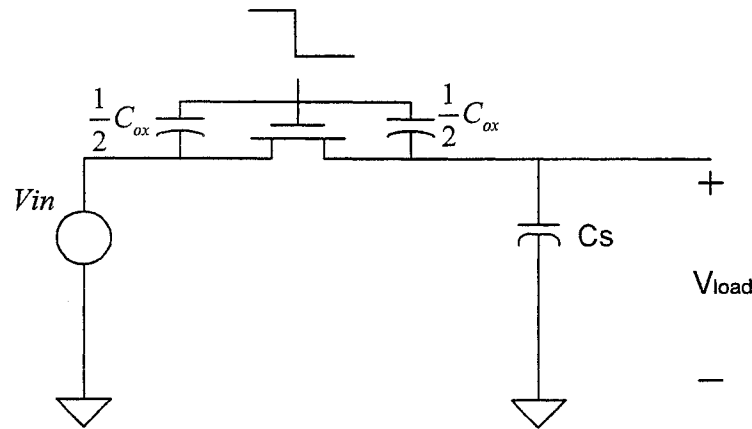


Figure 153 Clock feedthrough. modeling.

9.8. Gm cells

When designing the Gm circuit, the designer has to take care of some important characteristics:

- Input linear range: Input voltage range that the Gm circuit can handle without a significant amount of distortion due to non-linear effects.
- g_m value: The gain of the transconductance circuit. This is important because it will affect the loop dynamics.

- **Bandwidth:** This sets a limit on the output signal frequency. The bandwidth of the Gm cell has to be high enough so that it does not interfere with the fine loop dynamics and it should be low enough so that it will not pass the unnecessary noise to its output, which will deteriorate the receiver performance.
- **Input offset voltage:** The voltage required at the input to get an output current of 0A. Ideally, this should be 0V, however, due to mismatches in transistors, it will not be 0V. The offset of the Gm will appear as a phase error which will increase the BER.
- **PSRR:** Power Supply Rejection Ratio, since the Gm circuit is one of the most critical circuits on the fine loop signal path.

9.8.1. Designing for Figures of Merit and Simulation Results

The linear range is determined by the size of the differential pair, source degeneration resistor, bias current, and current ratio. The plot in Figure 154 shows the transconductance and linear operating range of the Gm circuit at 1.25Gb/s data rate. Each curve represents a different corner that was run on the circuit with the input being swept from -1.8V to 1.8V .

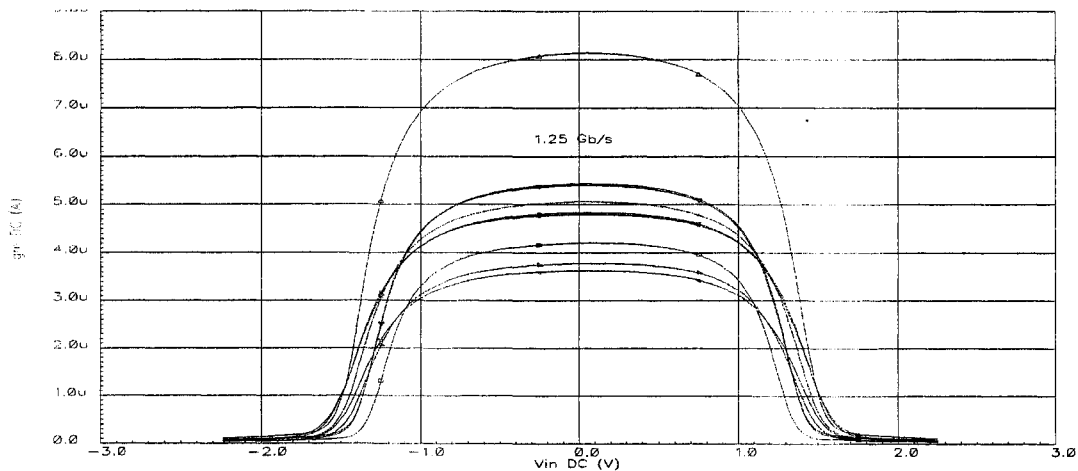


Figure 154 Linear range and g_m value of transconductance circuit for the 1.25Gb/s operation.

The Gm circuit bandwidth must be greater than the fine loop bandwidth, so that the loop dynamics do not change. Simulation results show that the bandwidth of the Gm circuit is in the range of 50MHz to 95MHz, depending on different simulation corners. Figure 155 shows the ac response of the Gm circuit, output voltage (Y-axis) over frequency (X-axis). Each curve represents a different corner that was run on the circuit. Differential ac signals with a magnitude of 0.5V were placed on the Gm inputs.

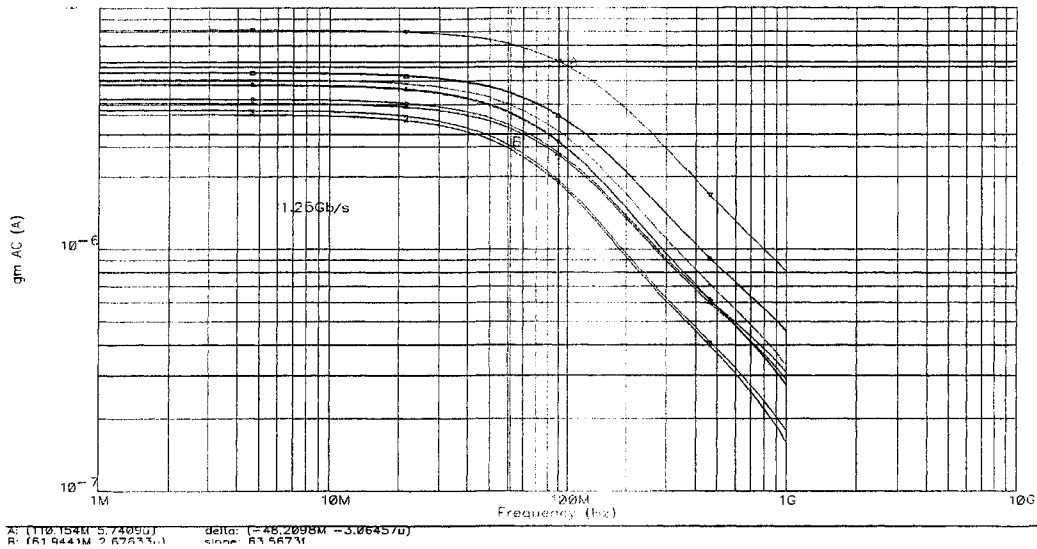


Figure 155 AC response of Gm circuit at different corners.

To determine the offset voltage, feedback was applied to the Gm input. This made the output current of the Gm approximately zero. Then the difference between the Gm inputs was measured over temperature. This measurement corresponds to the systematic input offset voltage. Simulations were run over different corners, and the results are shown in Figure 156. Additionally, there will be offset due primarily to input device mismatch.

Top-level simulations show that the Gm input offset voltage is less than 3 mV, which is negligible.

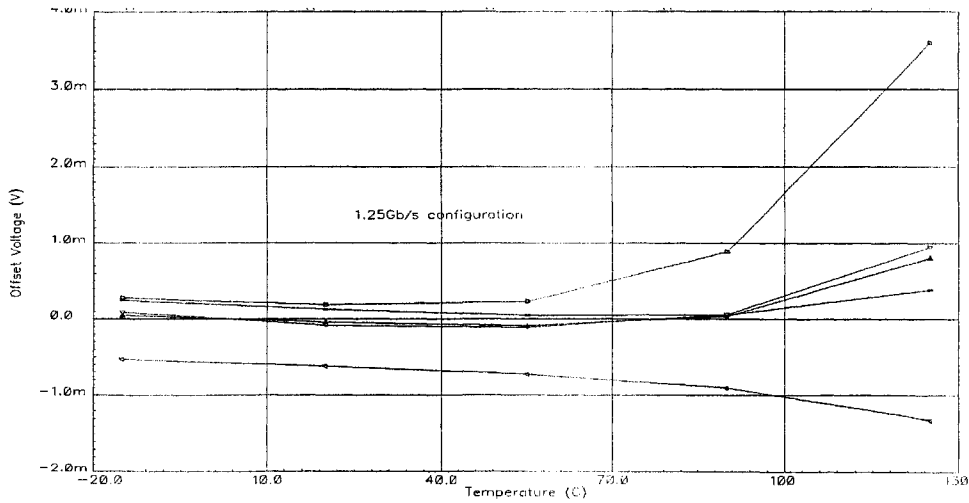


Figure 156 Gm systematic input offset voltage vs. temperature for the 1.25Gb/s operation.

9.9. Parasitic insensitive clocking Scheme

Many analog systems generate clocks that are intended to drive loads either inside the same system or in an outside system that might be either an analog or a digital one. In many cases, the circuits that generate the clocks are identical, however, the clocks drive different loads. This heterogeneous load will cause the clocks to mismatch from each other, which may affect the performance of the overall system. It is usually desirable to match the loads with each other, so that the generated clocks are matched as well. One way to do the matching is to add dummy cells. Although this approach does not result in a perfect matching because the dummy loads react different than the actual load when the circuit is operational, this approach is the easiest and simplest. The basic idea behind the new scheme is that the load is divided into two parts. The first part is sensitive to the mismatch in the clocks, while the second one is not sensitive to clock mismatch. Then, the load of the first part is matched with each other as much as possible. The second part need not to be matched. More information about this scheme can be found in section 8.5.2 [3].

9.10. Layout

The layout techniques will be illustrated via specific examples explained earlier in this dissertation.

9.10.1. Operational amplifier layout

The common centroid techniques were used wherever it was possible in order to have better matching, and metal overlapping was avoided as much as it could be in order to reduce the parasitics. For example, every two corresponding transistors in the main opamp have been laid out as common centroid so that they are matched together. Figure 157 shows the layout of the two transistors: M10 and M11 in the main opamp. Since the number of transistors doesn't completely agree with the common centroid requirement, dummy transistors were added as shown in the same figure.

The sampling as well as the integrating capacitors are also laid out in common centroid fashion to increase the matching. Special attention was paid to the overlapping of the metal wires connecting the layers that make the capacitors so that they contribute of equivalent parasitic capacitance. The reason for this is that overlapping was not completely avoided.

Figure 149 shows the way the capacitors are laid out. This layout is less sensitive to the gradient effects of the process in the horizontal direction, vertical direction or diagonal direction.

The complete layout of the operational amplifier is shown in Figure 158.

9.10.2. ADC Stage layout

Each stage consists of a one opamp, comparators clock generator, CMFB and some switches. The layout of each stage was constructed such that the analog part that is made of the opamp, comparators and CMFB circuit is separated from the digital part that is made of the clock generator circuit by more than 100 μ .

This is shown in Figure 159. As mentioned above, especial attention was made to avoid any overlapping of wires as much as possible. All the signals that are passed from the digital portion to the analog portion are shielded by surrounding them with ground lines.

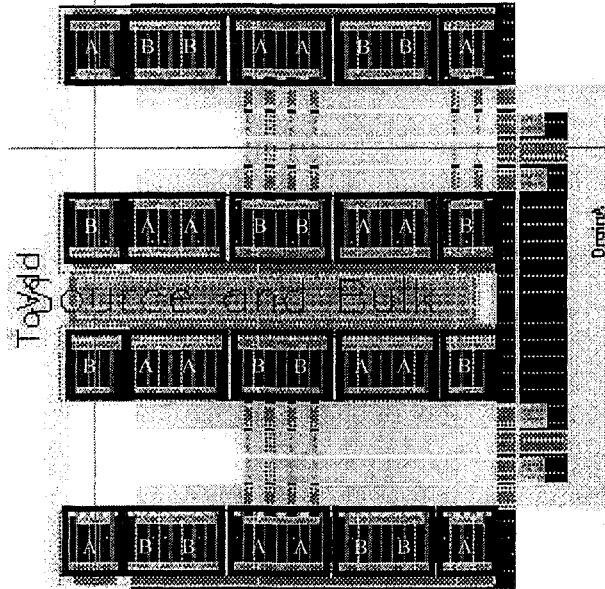


Figure 157 PMOS transistor laid out in CC fashion.

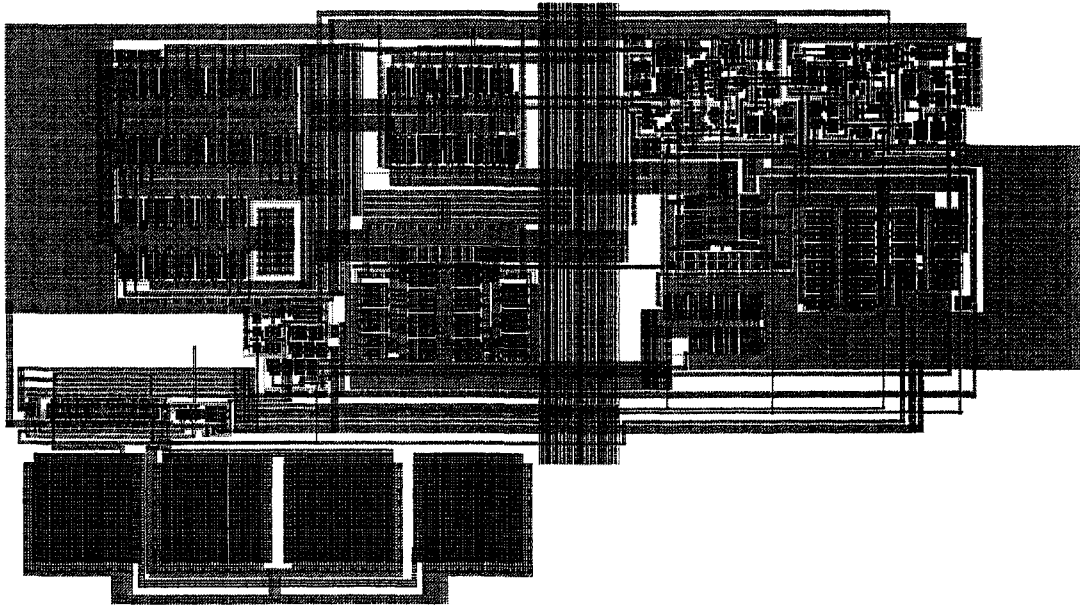


Figure 158 layout of the operational amplifier.

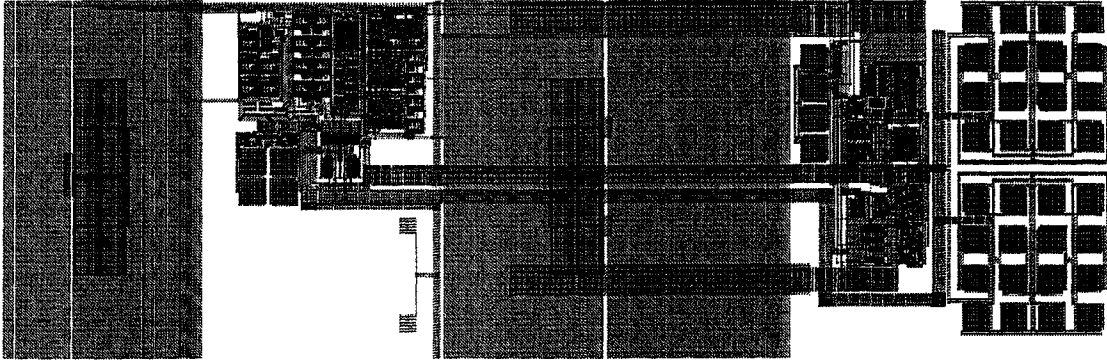


Figure 159 layout of a single stage.

9.10.3. Overall Layout.

The ADC has been carefully laid out in order to enhance matching, reduce parasitic and reduce the coupling between the digital and the analog parts.

To enhance the matching, symmetry of differential signal paths was followed wherever possible. For example, instead of connecting two differential signals as shown in Figure 160.a), the actual layout was done as shown in Figure 160.b).

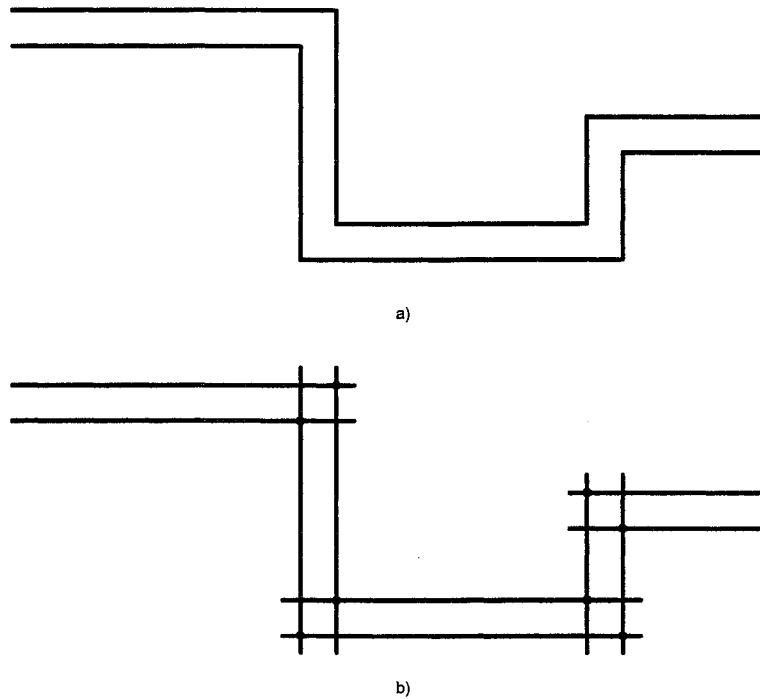


Figure 160 Layout matching technique.

Parasitics have been reduced in the layout by making sure of using common centroid as well as digitization techniques whenever possible. This is particularly important in this design because there are many

big transistors especially those in the main operational amplifier that need to be matched with each other. The capacitors of each stage are also laid out in a common centroid fashion to enhance their matching. In particular, the first stage of the ADC which is a single-ended-to-differential (STD) circuit has 3 capacitors that needed to be matched with each other. This new technique is illustrated in Figure 150 where the three capacitors are A, B and C. Each one is divided into 12 unit capacitors. These unit capacitors are distributed as shown in Figure 150 to enhance the matching. This way, the parasitics will be common to the three capacitors. The overall layout of the STD circuit is shown in Figure 161.

Critical differential signals are also separated from each other by ground lines in order to reduce the mutual coupling between them.

In order to decouple the analog portion from the digital ones, the power supplies of the two parts are separated from each other by a distance of more than 100um as shown in Figure 162. This is believed to be the best way to keep the digital noise away from affecting the analog circuits. Additional decoupling capacitors are also added underneath the power supply rails as shown in Figure 163.

The overall layout of the chip has been done in a straight line as shown in Figure 163 in order to reduce the effect of process gradient.

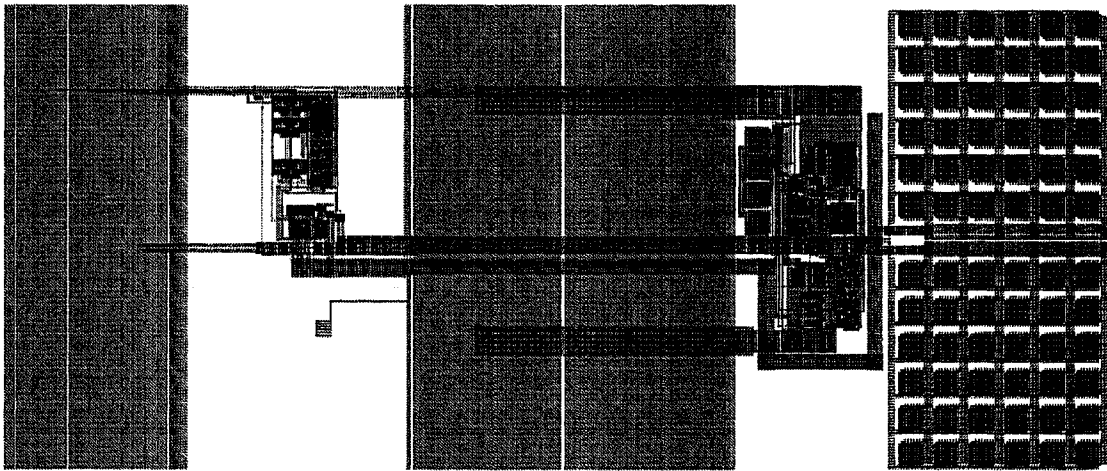


Figure 161 Single-ended-to-differential (STD) stage.

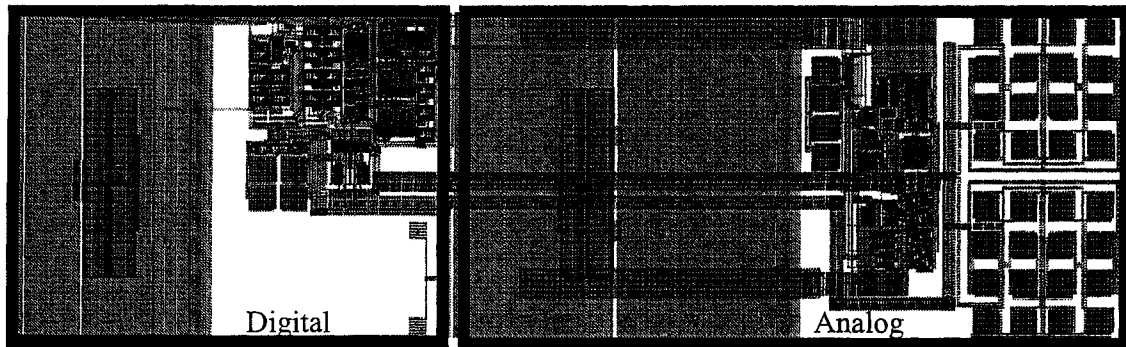


Figure 162 Layout of the MSB stage in the ADC. This figure shows the separation between the analog circuits and the digital ones.

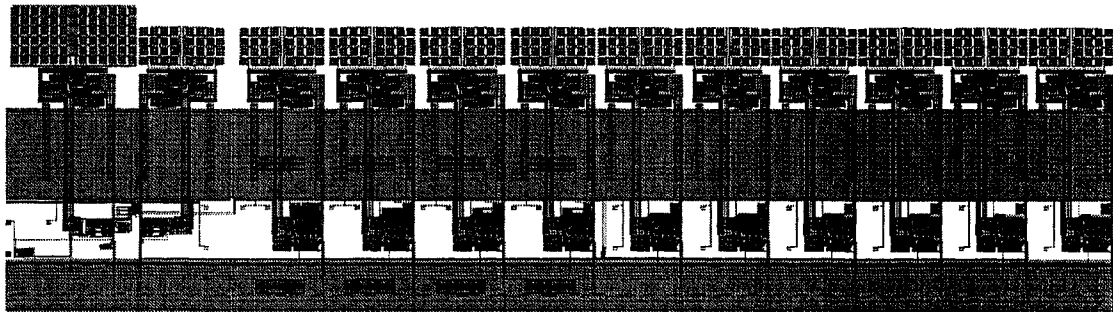


Figure 163 ADC top level layout.

Further layout techniques that are specific to VCOs can be found in section.

9.11. Conclusions

This chapter presented a collection of design techniques and engineering practices that were used in this Ph.D. work and are applicable to almost all high speed analog IC design.

In particular, this chapter presented issues and design considerations related to designing operational amplifiers and comparators. Techniques to match two capacitors with each other as well as three capacitors with each other have been discussed in this chapter. The effects of thermal noise, charge injection and clock feedthrough were also analyzed in order to help the designer to minimize their effect on the overall system.

Issues related to designing Gm cells are also presented in this chapter. Those issues must be considered carefully during the design, so that optimum performance is achieved.

Different techniques that are commonly used in analog circuits to reduce the parasitics as well as to result in a better matching are also discussed.

References

- [1] T. B. Cho and P. R. Gray, "A 10 b, 20 Msample/s, 35 mW pipeline A/D converter," *IEEE J. Solid-State Circuits*, vol. 30, pp. 166-172, Mar.1995.
- [2] Ahmed A. Younis, Moises E. Robinson, Micheal Nix and Brian Brunn, "Ring Oscillator Layouts with Improved Signal-Path Matching for High-Speed Data Communications," Patent pending 2001.
- [3] Moises E. Robinson and Ahmed A. Younis, "Clock Distribution Scheme for Improved Jitter Performance," Patent pending 2001.

CHAPTER 10. Conclusions

This dissertation presents analog building blocks that are used to produce high performance analog systems. In particular, it addressed two systems: analog-to-digital converters (ADCs) and multi-gigabit per second transceivers. Design techniques and engineering practices that were used throughout the implementation of these two systems were clearly emphasized and analyzed.

Two ADC architectures were considered in this dissertation. The first one was a 10-bit pipeline ADC with a 100MS/s conversion rate. In this system, gain boosting techniques as well as methods to analyze the frequency behavior of operational amplifiers were presented and explained. Implementation of comparators as well as operational amplifiers with different common mode feedback circuits were explained in details.

Practical layout methods to enhance component matching and reduce the parasitics were also presented. The common centroid technique has been extended from matching two capacitors with each other that is commonly used in most of ADC designs, to matching three capacitors with each other. This technique was not only applied to capacitors, but also to transistors that must be matched.

This dissertation also included layout techniques for signal routing and shielding as well as noise decoupling by separating the digital circuits from the analog ones and using decoupling capacitors connected to the power supplies underneath the power buses.

No matter how good the layout is, there will still be some mismatches among the components of the ADC. In this dissertation, two calibration algorithms that reduce the effect of gain errors and remove the DAC errors were presented. The ultimate result of these two algorithms is a reduction in the overall DNL of the ADC.

Not only techniques and circuit designs have been presented in this dissertation, but also new architectures as well. A new ADC architecture has been introduced that has better performance than the current ones. Power simulation of the new ADC revealed that a 500MHz 10-bit ADC is achievable with 270mW in a 0.18 μ m and 1.8V digital CMOS process and it will have small die area compared with other architectures that can achieve the same number of bits at the same speed. The high speed characteristic of this ADC is mainly due to a novel frequency difference detector circuit whose performance by far exceeds the current designs available in the literature. This dissertation presents a complete analysis of the new frequency difference detector circuit and many of its circuit variations.

Circuit implementation and design techniques that were used to design and enhance the performance of the receiver block of a 3.125Gb/s transceiver in a 0.18 μ m and 1.8V digital CMOS process were presented and fully explained in this dissertation. In particular, issues related to the high speed and low power supply operation of the circuits of the receiver were analyzed. Design and layout techniques to enhance the jitter tolerance of the receiver have also been explained. The effects of these techniques have been checked through extensive Matlab and Spice simulations of the receiver top level. Silicon results have shown that these techniques resulted in an outstanding and very robust receiver that has high jitter tolerance, which exceeded the requirements of many of the industry standards.

In order to reduce the design cycle time of any analog circuit, all of the design techniques and engineering practices for high speed analog ICs that were used throughout the dissertation were collected in one chapter. Although these techniques have been mainly used in the design of a pipeline ADC and a high speed receiver, they can be applied to almost any analog circuit.

A list of publications and patents by the author related to this work are:

1. **Ahmed A. Younis and M. M. Hassoun**, "A High Speed Fully differential CMOS Opamp," Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, Lansing MI, Aug 8-11, 2000.
2. **Ahmed A. Younis, M. M. Hassoun, and V. Navin**, "A Calibration Algorithm for a 16-bit Multi-path Pipeline ADC," Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, Lansing MI, Aug 8-11, 2000.
3. **Ahmed A. Younis and M. M. Hassoun**, "A Calibration Algorithm for DNL Reduction in a 1.5-bit per stage Pipeline Analog-to-Digital Converter," submitted to *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*.
4. **Ahmed A. Younis, Marwan M. Hassoun and Moises E. Robinson**, "VCO-based ADCs," Disclosed for patenting, 2001.
5. **Ahmed A. Younis and Micheal Nix**, "Frequency and Time Detection Apparatus," Disclosed for patenting, 2001.
6. **Ahmed A. Younis, C. Boecker, B. Das, K. Hossain, F. Abughazaleh, Y. Chen, M. Gaboury, M. Shafer, S. Irwin, M. Robinson and B. Grung**, "A Programmable, Low-Power and Low-Jitter 3.125Gb/s CMOS Transceiver," submitted the *IEEE Journal of Solid-State Circuits*. (Invited paper)
7. **Ahmed A. Younis, C. Boecker, K. Hossain, F. Abughazaleh, B. Das, Y. Chen, M. Robinson, S. Irwin and B. Grung**, "A Low Jitter, Low Power, CMOS 1.25-3.125Gbps Transceiver," Proceedings of the European Solid-State Circuits Conference, 2001.

8. **Ahmed A. Younis, Moises E. Robinson, Micheal Nix and Brian Brunn**, "Ring Oscillator Layouts with Improved Signal-Path Matching for High-Speed Data Communications," Patent pending 2001.
9. **Moises E. Robinson and Ahmed A. Younis**, "Clock Distribution Scheme for Improved Jitter Performance," Patent pending 2001.

Other related publications and patents by the author:

1. **Shah, J.C.; Younis, A.A.; Sapatnekar, S.S.; Hassoun, M. M.**, "An Algorithm for Simulating Power/Ground Networks using Padè Approximants and its Symbolic Implementation," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Volume: 45 Issue: 10, Oct. 1998, Page(s): 1372 -1382. (Invited paper)
2. **S. Bataineh and A. Younis**, "An Optimal Manhattan Street Network Routing Algorithm (OMRA)," to appear in the International Journal of Computers and Applications, Vol. 24, No. 1, 2002.
3. **J. C. Shah, A. A. Younis, S. S. Sapatnekar and M. M. Hassoun**, "Symbolic Analysis of Power/Ground Networks using Moment-matching Methods," Proceedings of the European Conference on Circuit Theory and Design, pp. 1292-1297, 1997.
4. **A. A. Younis, M. M. Amourah and R. L. Geiger**, "A high frequency CMOS 4th Order programmable Filter," Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, Lansing MI, Aug 8-11, 2000.
5. **Ahmed Younis**, "A New Delay-Locked Loop (DLL) Architecture," Disclosed for patenting 2001.
6. **Ahmed Younis**, "A New High-Speed Synchronous Counter Architecture," Disclosed for patenting 2001.