

2001

A re-configurable pipeline ADC architecture with built-in self-test techniques

Hui Liu

Iowa State University

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

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

Recommended Citation

Liu, Hui, "A re-configurable pipeline ADC architecture with built-in self-test techniques " (2001). *Retrospective Theses and Dissertations*. 658.

<https://lib.dr.iastate.edu/rtd/658>

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.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

A re-configurable pipeline ADC architecture with built-in self-test techniques

by

Hui Liu

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

Major: Electrical Engineering (Microelectronics)

Program of Study Committee:
Marwan M. Hassoun, Co-major Professor
William C. Black, Co-major Professor
Robert J. Weber
Chris Chong-Nuen Chu
Yuhong Yang

Iowa State University

Ames, Iowa

2001

Copyright © Hui Liu 2001. All rights reserved.

UMI Number: 3034203

UMI[®]

UMI Microform 3034203

Copyright 2002 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 doctor's dissertation of
Hui Liu
has met the dissertation requirements of Iowa State University**

Signature was redacted for privacy.

Co-Major Professor

Signature was redacted for privacy.

Co-Major Professor

Signature was redacted for privacy.

For the Major Program

TABLE OF CONTENTS

LIST OF FIGURES	vi
LIST OF TABLES	ix
ABSTRACT	x
1 INTRODUCTION	1
1.1 Motivation.....	1
1.2 Dissertation Organization	3
2 ON-CHIP ANALOG SIGNAL GENERATION	5
2.1 Introduction.....	5
2.2 Bitstream Generation	5
2.2.1 Difference between Software and Hardware Generation Methods.....	5
2.2.2 Basic Block Diagram of Generating Optimized Bitstream	6
2.2.3 Bitstream Generator: $\Sigma\Delta$ Modulator	7
2.2.3.1 Oversampling without Noise Shaping.....	8
2.2.3.2 Oversampling with Noise Shaping	11
2.2.3.3 First Order Sigma-delta Modulator	12
2.2.3.4 High Order $\Sigma\Delta$ Modulator	14
2.2.4 Parameter for the Performance of Bitstream: Length N.....	18
2.2.5 Bitstream Optimization	22
2.3 Shift Register Ring	23
2.4 1-bit DAC	24
2.5 Analog Low Pass Filter.....	24
2.5.1 Specification of LPF	24
2.5.2 Power Consideration of the Generated Analog Signal.....	29
2.6 Multi-phase Signal Generation	31
2.7 Chapter Conclusion.....	33
3 ADC TESTING	34
3.1 Introduction.....	34
3.2 Error Measurements.....	34
3.2.1 Quantization Errors	35

3.3 Coherent Sampling [15].....	37
3.4 Linear Histogram Testing	39
3.5 Dynamic Histogram Testing.....	41
3.6 Chapter Conclusion.....	42
4 ON-CHIP DYNAMIC TESTING OF ADCs	43
4.1 Introduction.....	43
4.2 SNR.....	43
4.2.1 FFT Method	43
4.2.2 Narrow Band Filter Method.....	45
4.2.3 Bias of SNR Estimate via Digital Filter.....	47
4.3 Chapter Conclusion.....	51
5 DESIGN OF RE-CONFIGURABLE PIPELINE ADC	52
5.1 Introduction.....	52
5.2 9-bit 40MHz Pipeline ADC Implementation	52
5.2.1 System Specification.....	52
5.2.2 Multiplying Digital-to-Analog Converter	54
5.2.3 Operational Amplifier Design.....	58
5.2.3.1 DC Gain.....	58
5.2.3.2 Bandwidth	59
5.2.3.3 Load Capacitance	60
5.2.3.4 Gain-boosting OPAMP Design	60
5.2.3.4.1 Principle of Gain-boosting.....	60
5.2.3.4.2 Fully Differential Folded Cascode OPAMP Design.....	63
5.2.3.4.3 Common-mode Feedback (CMFB) Circuit of Main OPAMP.....	64
5.2.3.4.4 Boosting Amplifier Design	65
5.2.3.4.5 CMFB for Boosting Amplifier	65
5.2.3.4.6 OPAMP Bias Circuit and Clock Generation Circuit	69
5.2.4 Latched Comparator Implementation.....	71
5.3 Re-configurable Pipeline Structure.....	75
5.3.1 Error Sources in 1-bit-per-stage Pipeline ADC.....	75
5.3.1.1 Gain Error.....	75
5.3.1.2 Comparator Offset	76

5.3.1.3 Charge Injection and Amplifier Offset	76
5.3.2 Variation between Pipeline Stages	77
5.3.3 Re-configurable Pipeline ADC	77
5.3.3.1 4-Switch Re-configuration	78
5.3.3.2 8-Switch Re-configuration	80
5.3.3.3 Simplified Re-configuration Algorithm	80
5.3.3.3.1 Grouping Algorithm for 9-bit Pipeline ADC	80
5.3.3.3.2 Proof of Grouping Algorithm	82
5.3.4 Design Consideration of Re-configurable ADC Cell [38]	85
5.4 Comparison with other Pipeline ADC Calibration Techniques	87
5.5 Layout Considerations	89
5.6 Chapter Conclusion.....	92
6 EXPERIMENTAL RESULTS	93
6.1 Introduction.....	93
6.2 Experimental Results of Re-configurable Pipeline ADC	93
6.2.1 PCB Design of Testing Chip.....	93
6.2.2 Testing Setup.....	96
6.2.3 Experimental Results	97
6.2.3.1 Dynamic Testing	97
6.2.3.2 Static Testing	98
6.2.3.3 Temperature Testing.....	101
6.3 Experimental Results of the on-chip Analog Signal Generation	103
6.4 Chapter Conclusion.....	107
7 CONCLUSION	110
7.1 Conclusions.....	110
7.2 Recommended Future Work.....	112
REFERENCES.....	114
ACKNOWLEDGEMENTS	119

LIST OF FIGURES

Figure 1.1 Diagram of BIST for pipeline ADC.....	2
Figure 2.1 Typical bitstream circuit diagram	7
Figure 2.2 $\Sigma\Delta$ -based digital encoding of a sinusoid	8
Figure 2.3 An oversampling system without noise shaping [32]	10
Figure 2.4 The linear model of the modulator.....	11
Figure 2.5 First order sigma-delta modulator.....	12
Figure 2.6 Second-order $\Sigma\Delta$ modulator.....	14
Figure 2.7 Third-order $\Sigma\Delta$ modulator.....	15
Figure 2.8 Cascade-of-integrators structure used to realize the 5 th -order modulator	16
Figure 2.9 Frequency response of a long bitstream: length= 2^{16} (Matlab™ simulation result).....	19
Figure 2.10 Frequency response of a short bitstream, length= 2^{10} (Matlab™ simulation result).....	19
Figure 2.11 SNR vs. bitstream position.....	23
Figure 2.12 7 th order sigma-delta modulator frequency response	25
Figure 2.13 7 th order modulator followed by 7 th order Butterworth LPF, SNR=98.1132dB, ENOB=16.0055	26
Figure 2.14 7 th order modulator plus 4 th order LPF, SNR=75.8411dB, ENOB=12.3058	26
Figure 2.15 The same modulator followed by a 2 nd order modulator, SNR=43.0123, ENOB=6.8525	27
Figure 2.16 7 th order vs. 2 nd order modulator frequency response	28
Figure 2.17 Filter off-chip scheme	29
Figure 2.18 Power for the input and output of the $\Sigma\Delta$ modulator (ω_x is the analog testing signal generated by the $\Sigma\Delta$ modulator).....	30
Figure 2.19 Diagram of multi-phase signal generation.	32
Figure 2.20 Generated multi-phase signals	32
Figure 3.1 ADC transfer curve	34
Figure 3.2 Spectrum of ideal 6bit ADC with N=1024, M=24, where noise floor is not correct.....	38
Figure 3.3 Spectrum of ideal 6bit ADC with N=1024, M=23.....	38
Figure 3.4 The INL of an 8-bit ADC.....	40
Figure 3.5 The DNL of a 8-bit ADC	40

Figure 3.6 Spectrum of 6-bit ADC obtained with relatively prime M	41
Figure 4.1 Digital narrow band filter (NBF) [18].....	45
Figure 4.2 The frequency response $ H_{bp}(f) $ of the bandpass output of the.....	48
Figure 4.3 The FFT of the bandpass output of the narrow-band digital filter	49
Figure 4.4 The FFT of the notch output of the narrow-band digital filter.....	49
Figure 4.5 The frequency response $ H_n(f) $ of the notch output of the narrow-band digital filter...	50
Figure 4.6 Bias of estimated SNR vs. pole radius	50
Figure 5.1 The general architecture of pipeline ADC	53
Figure 5.2 Fully Differential Multiplying Digital-to-Analog Converter	55
Figure 5.3 Timing Diagram	55
Figure 5.4 MDAC during different clock phase.....	56
Figure 5.5 Gain circuit using an opamp	59
Figure 5.6 Main operational opamp with boosting amplifier	62
Figure 5.7 Cascode gain stage with gain enhancement	62
Figure 5.8 Bode plots of the original, additional and improved gain stage	63
Figure 5.9 Common mode feedback circuit	65
Figure 5.10 Boosting-n amplifier with CMFB circuit.....	66
Figure 5.11 Boosting-p amplifier with CMFB circuit.....	66
Figure 5.12 Opamp bias circuit	69
Figure 5.13 Clock generation circuit.....	70
Figure 5.14 Schematic of the CMOS comparator	71
Figure 5.15 Timing diagram.....	72
Figure 5.16 The transient analysis of comparator	74
Figure 5.17 4-tages of pipeline with re-configuration ability using 4-switch network	79
Figure 5.18 4-Switch network	79
Figure 5.19 The grouping method of a 9-stage ADC	81
Figure 5.20 The first six measurements	81
Figure 5.21 The second six measurements.....	82
Figure 5.22 The last six measurements	82
Figure 5.23 Gain-of-1/gain-of-2/gain-of-1.9 Multiplying Digital-to-Analog Converter	86
Figure 5.24 The layout of the chip	89
Figure 5.25 Die photo.....	90

Figure 5.26	The floor plan of clock tree for stage 1 through stage 9	90
Figure 5.27	The layout of one stage of ADC	91
Figure 6.1	Cross-section of 4-layer PCB.....	93
Figure 6.2	The layout of PCB.....	94
Figure 6.3	Photo of PCB	95
Figure 6.4	The single-to-differential circuit for LF testing	95
Figure 6.5	Setup of testing.....	96
Figure 6.6	The testing results of chip #1 at normal temperature and Fclk=40MHz.....	98
Figure 6.7	The testing results of chip #2 at normal temperature and Fclk=40MHz.....	99
Figure 6.8	The testing results of chip #3 at normal temperature and Fclk=40MHz.....	99
Figure 6.9	Typical plot of FFT of output of ADC (Chip #2, Configuration No.9)	100
Figure 6.10	The static test of re-configurable ADC (Chip #2, Configuration No.9).....	100
Figure 6.11	The temperature testing results of chip #2 with Fin=0.8MHz and Fclk=40MHz	101
Figure 6.12	The temperature testing results of chip #2 with Fin=1.6MHz and Fclk=40MHz	102
Figure 6.13	The temperature testing results of chip #2 with Fin=4.0MHz and Fclk=40MHz	102
Figure 6.14	The setup of testing on-chip analog signal generator.....	103
Figure 6.15	The frequency response of the output of the shift register ring	104
Figure 6.16	The frequency response of the output of the LPF	104
Figure 6.17	The schematic of DFF	105
Figure 6.18	The clock tree buffer used for 1024-shift register.....	106
Figure 6.19	The floor plan of 1024-shift register	106
Figure 6.20	The measured output of the shift register ring	107
Figure 6.21	The zoomed in layout of the error.....	108
Figure 6.22	The zoomed in micrograph of the error	109

LIST OF TABLES

Table 5.1 The specification of main opamp with 2pF load plus switch	64
Table 5.2 The specification of boost_p with 1pF load	67
Table 5.3 The specification of boost_n with 1pF load	67
Table 5.4 The specification of gain-boosted opamp with 2pF load	68
Table 5.5 Specification of the latched comparator	73
Table 5.6 Proof of the grouping algorithm	83
Table 5.7 Comparison with the other calibration techniques	88
Table 6.1 Configurations of re-configurable pipeline ADC	97

ABSTRACT

High-performance analog and mixed-signal integrated circuits are integral parts of today's and future networking and communication systems. The main challenge facing the semiconductor industry is the ability to economically produce these analog ICs. This translates, in part, into the need to efficiently evaluate the performance of such ICs during manufacturing (production testing) and to come up with dynamic architectures that enable the performance of these ICs to be maximized during manufacturing and later when they're operating in the field. On the performance evaluation side, this dissertation deals with the concept of Built-In-Self-Test (BIST) to allow the efficient and economical evaluation of certain classes of high-performance analog circuits. On the dynamic architecture side, this dissertation deals with pipeline ADCs and the use of BIST to dynamically, during production testing or in the field, re-configure them to produce better performing ICs.

In the BIST system proposed, the analog test signal is generated on-chip by sigma-delta modulation techniques. The performance of the ADC is measured on-chip by a digital narrow-band filter. When this system is used on the wafer level, significant testing time and thus testing cost can be saved.

A re-configurable pipeline ADC architecture to improve the dynamic performance is proposed. Based on dynamic performance measurements, the best performance configuration is chosen from a collection of possible pipeline configurations. This basic algorithm can be applied to many pipeline analog systems. The proposed grouping algorithm cuts down the number of evaluation permutation from thousands to 18 for a 9-bit ADC thus allowing the method to be used in "real" applications.

To validate the developments of this dissertation, a 40MS/s 9-bit re-configurable pipeline ADC was designed and implemented in TSMC's 0.25 μ m single-poly CMOS digital process. This includes a fully differential folded-cascode gain-boosting operational amplifier with high gain and high unity-

gain bandwidth. The experimental results strongly support the effectiveness of re-configuration algorithm, which provides an average of 0.5bit ENOB improvement among the set of configurations. For many applications, this is a very significant performance improvement.

The BIST and re-configurability techniques proposed are not limited to pipeline ADCs only. The BIST methodology is applicable to many analog systems and the re-configurability is applicable to any analog pipeline system.

1 INTRODUCTION

1.1 Motivation

Rapid growth in the area of wireless communication has increased the demand for high performance integrated circuits. Testing of these circuits represents a challenge. Yet built-in self-test (BIST) is an efficient way to save testing time and cut testing cost. While the concept of BIST has been in use for more than half a century, BIST implemented in pipeline Analog-to-Digital Converter (ADC) has become popular only during the past decade. Today, there are still many interesting topics that deal with BIST applications.

High-performance digital communication requires ADCs with resolutions of at least 7-bits and sampling rates higher than 40Msample/s. In the past decade, the pipeline ADC architecture has become suitable for such applications because it can be insensitive to offsets in comparators and operational amplifiers by using redundancy and digital correction. However, traditional designs of pipeline ADCs have relied on high-gain operational amplifiers and excellent capacitor matching to produce high-performance converters. Such analog components are becoming more difficult to design in scaled technologies because of the reduced power-supply voltages. How to improve the ADC's performance is the major task of designers in this area.

The motivation of this Ph.D work is to explore system level as well as circuit level design techniques for improving the performance of pipeline ADCs and to explore the BIST application in pipeline ADCs. A prototype chip was implemented and tested to demonstrate proposed architecture.

Figure 1.1 shows the proposed diagram of a re-configurable pipeline ADC architecture with built-in self-test technique. It is composed of two major parts: a software part and a hardware part. The software uses a $\Sigma\Delta$ modulator to generate on-chip analog signal, which appears as a bitstream. This

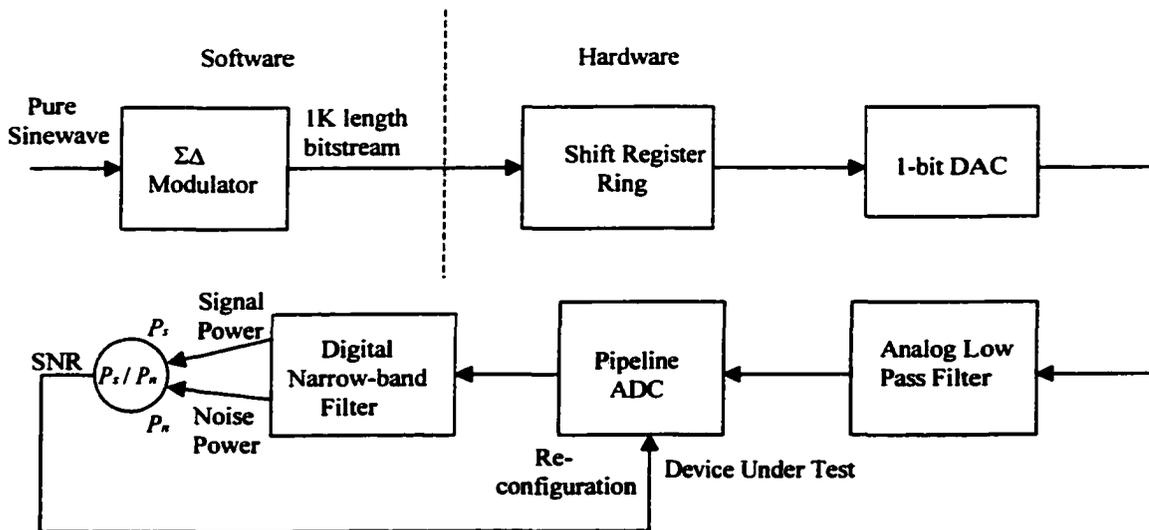


Figure 1.1 Diagram of BIST for pipeline ADC

bitstream is stored using a shift register. By repeating the bitstream, a signal source is emulated on-chip. This signal source is always ON during the test procedure. As an example, consider a bitstream of length 1K. This 1K bitstream goes through a 1-bit DAC to generate an analog signal, which is in 1bit format. However this 1bit bitstream is noise-shaped: noise is pushed to high frequency range. By using an analog low-pass filter (LPF), the high frequency range noise is filtered out. The output of the analog LPF is the real analog testing signal that is needed for testing the pipeline ADC, which is the device under test (DUT). The performance of the ADC will be measured by the digital narrow band filter (NBF), which will measure the signal to noise ratio (SNR). According to the measurement results, a re-configuration algorithm is applied on the re-configurable pipeline ADC to improve the ADC's performance.

The key contributions of this Ph.D work are:

- 1) A BIST system for re-configurable pipeline ADCs is proposed. It is a unique system with analog input testing signal generated on-chip as well as the dynamic performance of the ADC

measured on-chip. When this system is used in wafer test, it will save testing time and thus testing costs.

- 2) A re-configurable pipeline ADC architecture to improve the dynamic performance is proposed. Based on the dynamic performance measurements, the best performance configuration is chosen from a collection of possible pipeline configurations. This basic algorithm can be applied to many pipeline analog systems.
- 3) The grouping algorithm for re-configurable pipeline ADCs is proposed. It can cut down the number of evaluation permutation from thousands to 18 for a 9-bit ADC thus allowing the method to be used in “real” applications.
- 4) A 40Msample/s 9-bit re-configurable pipeline ADC is designed and implemented in TSMC’s 0.25 μ m single-poly CMOS digital process. It includes a fully differential folded-cascode gain-boosting operational amplifier with high gain and high unity gain bandwidth.
- 5) Verification of the prototype under different temperature conditions, with the experimental results strongly supporting the effectiveness of the re-configuration algorithm. It provides an average of 0.5bit ENOB improvement among the set of configurations.

1.2 Dissertation Organization

In chapter 2, the principles of on-chip analog signal generation are introduced. The underlying principles of $\Sigma\Delta$ modulators are described, with the simulation results shown. This part is the software part as shown in Figure 1.1. Matlab™ is used for simulation.

Chapter 3 concentrates on the basic principles of ADC testing, including static testing and dynamic testing. Testing principles such as Coherent sampling and histogram testing methods are described. Those testing methods will be used on the prototype chip testing.

In chapter 4, the on-chip dynamic performance measurement scheme is described. The focus is on the principle of Narrow-Band Filter (NBF). This method is used in BIST instead of traditional industry Fast Fourier Transform (FFT) method, which is difficult to be implemented on-chip.

Chapter 5 focuses on the design of the re-configurable pipeline ADC. Design principles of each component in pipeline ADC are presented. The re-configuration grouping algorithm is proposed, and the re-configurable pipeline ADC architecture is described.

The experimental results of the prototype chip are given in chapter 6. The testing procedures are described. Testing results in different testing temperatures are presented. Experimental results validate that the proposed re-configuration algorithm provides an average of 0.5bit ENOB improvement among the set of configurations.

Finally the conclusion of this work is presented in chapter 7.

2 ON-CHIP ANALOG SIGNAL GENERATION

2.1 Introduction

An important component of a mixed-signal BIST is a precision analog signal generator. In this chapter, the concept of on-chip analog signal generation is introduced. The basic diagram of a bitstream generator is presented, and the parameters for the performance of the bitstream are discussed. Finally the optimization of the bitstream is covered. Simulation results are shown for each part. In this dissertation, a technique to generate sinewaves with small die area and simple circuits is carried out. By using some memory on the IC, the output of a sigma-delta ($\Sigma\Delta$) modulator can be periodically reproduced [1]. First, the difference between software and hardware generation methods is discussed. Then, the basic block diagram to generate an optimized bitstream is introduced. Finally, the method to encode the analog signals into a 1-bit pattern for digital storage is explained.

2.2 Bitstream Generation

2.2.1 Difference between Software and Hardware Generation Methods

Bitstream generation can be done in two ways. The first approach uses analog circuits, such as Colpitts or Wien-Bridge oscillators [1]. However, in a mixed-signal test strategy, the test circuits should themselves be testable. This makes testing these circuits difficult. The second approach is to use a digital signal generator and convert the signal with a digital-to-analog converter (DAC) [2]. Digital generators do not require calibration and are easily testable using standard digital test techniques. But, traditional memory-based generators, such as direct digital frequency synthesis

(DDFS), require large areas. What is significant is that these schemes require the use of a multi-bit DAC, which is highly susceptible to process variations and requires rigorous testing.

In this dissertation, a different method to generate the bitstream: recording a short portion of the output of a sigma-delta oscillator and reproducing it periodically [5]. This is a software generation method. Compare with the two traditional hardware generation methods discussed above, there are two main differences between them: complexity and stability.

First, let us think about the order of modulator. Increasing the order of the modulator means increased complexity when the modulator is realized in hardware. However, if realized via software, an increase in order does not increase the complexity of the generation and high-order modulators can be implemented efficiently. In this dissertation, the analog signal is generated for testing the pipeline ADC. If the ADC has N -bit resolution, then the analog input signal for on-chip simulation should be at least $N+2$ -bit accuracy. The high accuracy of the analog input signal can be guaranteed fulfilled using software without increasing any on-chip hardware.

Second, let us consider stability. The software generation implementation does not necessarily have to be stable in the classical sense, it only needs to be stable long enough to acquire a sufficient number of points to create a bitstream.

2.2.2 Basic Block Diagram of Generating Optimized Bitstream

The basic circuit to generate the bitstream is very simple: a 1-bit shift register with the output fed back to the input, a 1-bit DAC, and an analog filter. A block diagram of a typical circuit is shown in Figure 2.1 [4] [5] [7].

Now, let us see how to encode an analog signal into a 1-bit bitstream pattern. The pattern is labeled “data” in Figure 2.1.

2.2.3 Bitstream Generator: $\Sigma\Delta$ Modulator

A $\Sigma\Delta$ modulator is used to transfer an analog signal into a 1-bit bitstream. For a $\Sigma\Delta$ DAC, the signal conversion is performed by encoding a multi-bit digital input signal into a single-bit stream with a peak-to-peak amplitude of Δ using digital signal processing and over-sampling techniques [5]. Figure 2.2 shows this process. It applies to a sinusoidal digital input signal of amplitude A and frequency f_i , that has been greatly over-sampled (i.e., $f_s \gg f_i$). We can see from Figure 2.2, the output toggles between the high and low state in such a way that the input signal is encoded into the density of the output waveform. Such a signal is known as a pulse-density modulated (PDM) signal. The difference between the output bit pattern and the input signal is denoted as quantization error. It is noted that the input multi-bit digital signal can be recovered quite simply by digitally filtering the 1-bit output. This is because the $\Sigma\Delta$ encoding process ensures that the input signal and the quantization error, occupy different frequency regions.

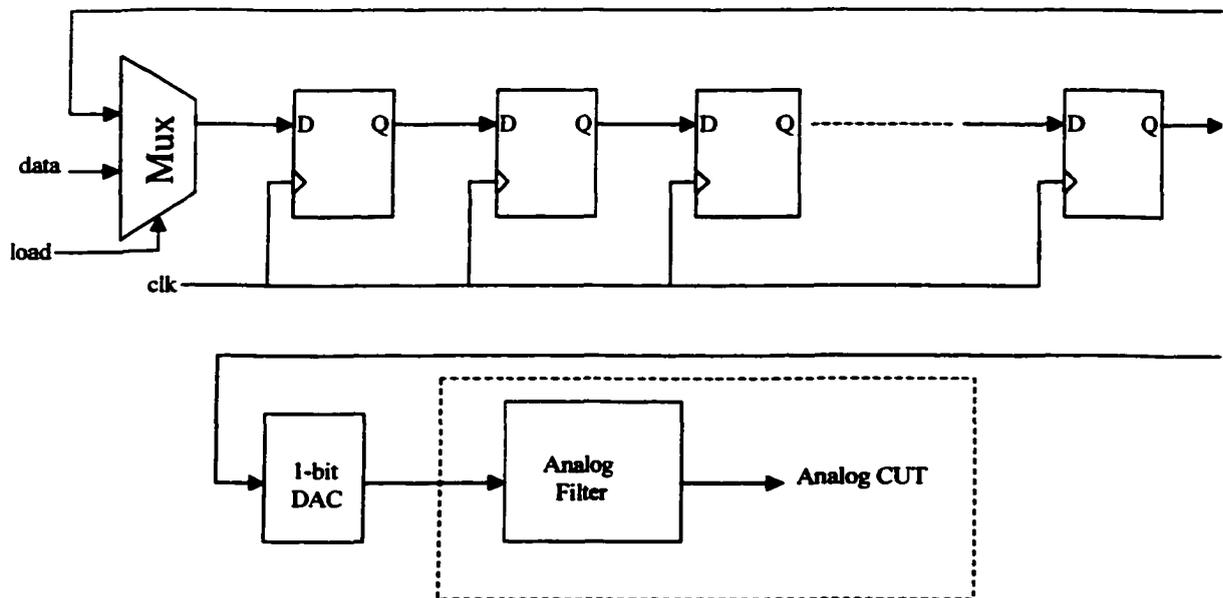


Figure 2.1 Typical bitstream circuit diagram

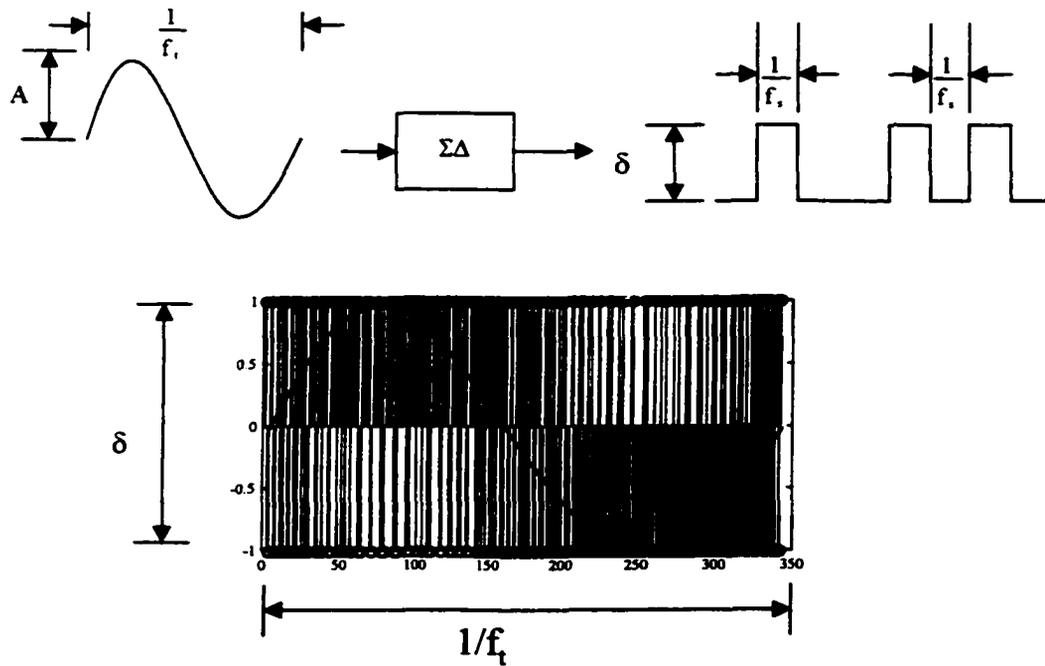


Figure 2.2 $\Sigma\Delta$ -based digital encoding of a sinusoid

To transform the digital input signal into analog form, the serial stream is simply filtered using an analog filter with bandwidth f_B , which is greater than f_t (as shown in Figure 2.2). While the filtering operation will eliminate most of the quantization noise, it will not eliminate all of it; some noise lies in-band and cannot be separated from the signal by filtering. It will be shown later in this chapter with simulation results.

There are different kinds of $\Sigma\Delta$ modulators, which mean different PDMs or noise shaping for the bitstream. The next section will explore some of them.

2.2.3.1 Oversampling without Noise Shaping

Quantization introduces distortion. If we assume the error has statistical properties that are independent of the signal, then the error can be represented as noise. Let Δ be the quantization level

spacing. When we treat the quantization error e as having equal probability of lying anywhere in the range $\pm\Delta/2$, its mean square value is given by [8]

$$e_{rms}^2 = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} e^2 de = \frac{\Delta^2}{12} \quad (2.1)$$

Oversampling occurs when the signals of interest are bandlimited to f_B while the sampling rate is at f_s , where $f_s > 2f_B$. The oversampling ratio (OSR) is defined as

$$OSR = \frac{f_s}{2f_B} \quad (2.2)$$

Assuming the quantization noise is white, and noting that the total noise power is e_{rms}^2 in the range of $\pm \frac{f_s}{2}$, the spectral density of the quantization noise is

$$E(f) = e_{rms} \sqrt{\frac{1}{f_s}} = \left(\frac{\Delta}{\sqrt{12}} \right) \sqrt{\frac{1}{f_s}} \quad (2.3)$$

As shown in Figure 2.3, after quantization, since the signals of interest are all below f_B , $y_1(n)$ is filtered by $H(f)$ to create the signal $y_2(n)$. This filter eliminates quantization noise (together with any other signals) greater than f_B . Then the noise power P_n that falls into the signal band will be given by

$$P_n = \int_{-f_B}^{f_B} E^2(f) df = e_{rms}^2 (2f_B T) = \frac{e_{rms}^2}{OSR} = \frac{\Delta^2}{12} \frac{1}{OSR} \quad (2.4)$$

Assuming the input signal is a sinusoidal wave, and N -bit quantizer is used, its maximum peak value without clipping is $2^N(\Delta/2)$. For this maximum sinusoidal wave, the signal power, P_s , is given by

$$P_s = \left(\frac{\Delta 2^N}{2\sqrt{2}} \right)^2 = \frac{\Delta^2 2^{2N}}{8} \quad (2.5)$$

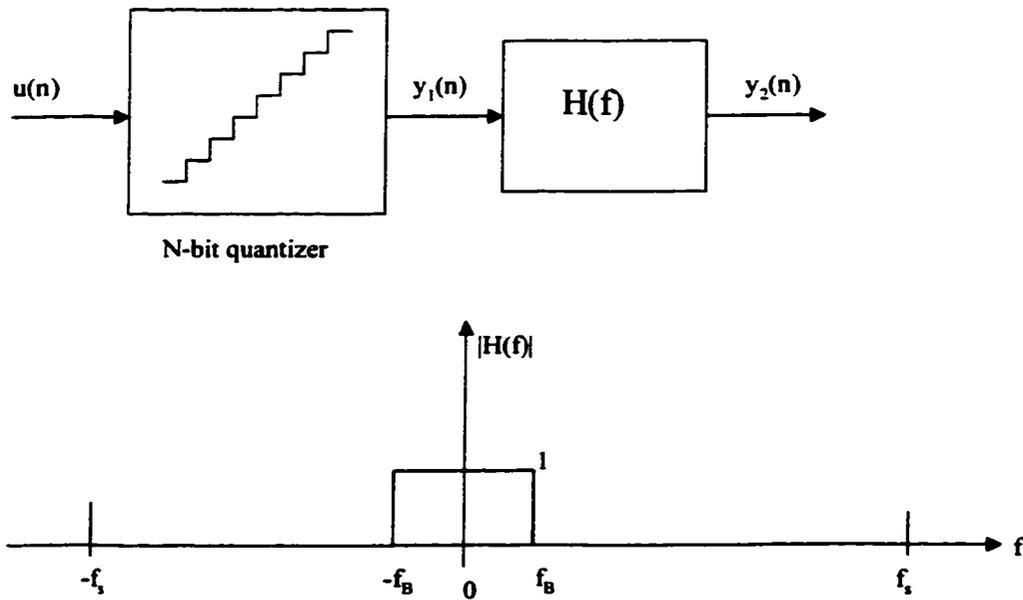


Figure 2.3 An oversampling system without noise shaping [32]

The signal power will not change after oversampling and low pass filtering $H(f)$ because the assumption is that the signal's frequency content is below f_B .

From equation (2.4) and (2.5), we can calculate the maximum SNR (in dB) to be the ratio of the maximum sinusoidal power to the quantization noise:

$$SNR_{\max} = 10 \log \left(\frac{P_s}{P_n} \right) = 10 \log \left(\frac{3}{2} 2^{2N} \right) + 10 \log(OSR) \quad (2.6)$$

which is also equal to

$$SNR_{\max} = 6.02N + 1.76 + 10 \log(OSR) \quad (2.7)$$

The first term is the SNR due to the N-bit quantizer while the OSR term is the SNR enhancement obtained from oversampling. Each doubling of the sampling frequency will decrease the in-band noise by 3dB, increasing the resolution by half a bit.

2.2.3.2 Oversampling with Noise Shaping

A general noise-shaped sigma-delta ($\Sigma\Delta$) modulator is shown in Figure 2.4 [32]. This structure is known as an interpolative structure and is analogous to an amplifier realized using an opamp and feedback. In this structure, the feedback reduces the effect of the noise of the output stage of the opamp in the closed-loop amplifier's output signal at low frequencies when the opamp gain is high. At high frequencies, when the opamp's gain is low, the noise is not reduced.

The signal transfer function $STF(z)$ and the noise transfer function $NTF(z)$ are:

$$STF(z) = \frac{Y(z)}{U(z)} = \frac{H(z)}{1 + H(z)} \quad (2.8)$$

$$NTF(z) = \frac{Y(z)}{E(z)} = \frac{1}{1 + H(z)} \quad (2.9)$$

The zeros of the noise transfer function will be equal to the poles of $H(z)$. When $H(z)$ goes to infinity, $NTF(z)$ will go to zero.

If we choose $H(z)$ such that its magnitude is large from 0 to f_B (i.e., over the frequency band of interest), we can noise-shape the quantization noise in a useful manner. With such a choice, the signal transfer function will approximate unity over the frequency band of interest very similarly to an opamp in a unity-gain feedback configuration [32]. Furthermore, the noise transfer function $NTF(z)$

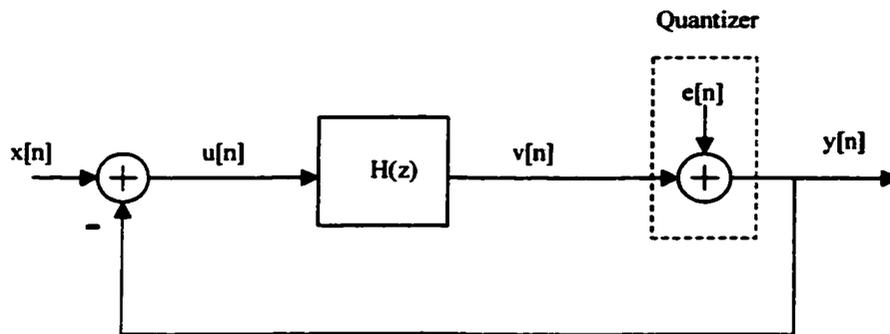


Figure 2.4 The linear model of the modulator

will approximate zero over the same band. Thus the quantization noise is reduced over the frequency band of interest while the signal itself is largely unaffected.

2.2.3.3 First Order Sigma-delta Modulator

The diagram of the first-order $\Sigma\Delta$ modulator is shown in Figure 2.5.

The modulator consists of an integrator, an internal ADC or quantizer, and a DAC used in the feedback path. It is easily shown that the output of the integrator is

$$v[i] = x[i - 1] - e[i - 1] \quad (2.10)$$

and the quantized signal is

$$y[i] = x[i - 1] + (e[i] - e[i - 1]) \quad (2.11)$$

Thus this circuit differentiates the quantization error, making the modulation error the first difference of the quantization error while leaving the signal unchanged, except for a delay.

The DAC in the modulator is required to be nearly as linear as the overall conversion resolution. Any DAC nonlinearity can be modeled as an error source that adds directly to the input. This error source benefits from the oversampling but unlike $e[n]$, which models the ADC quantization error, is

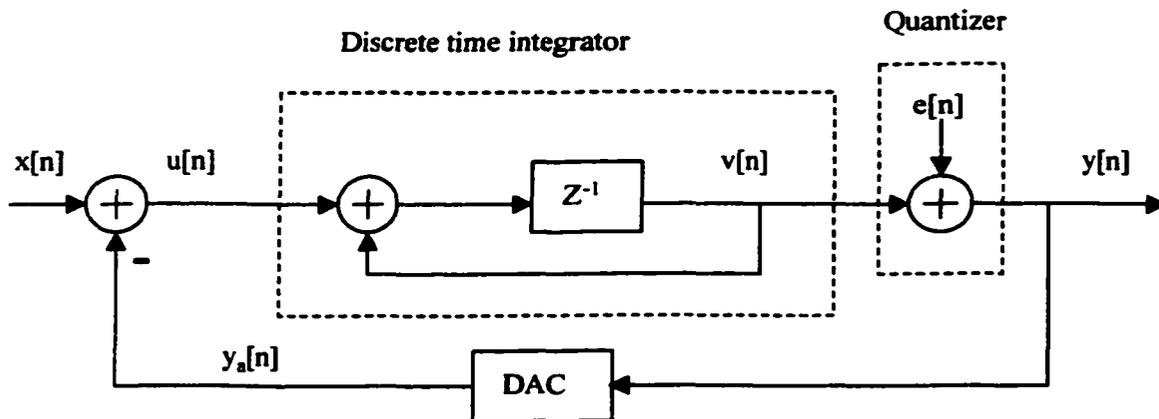


Figure 2.5 First order sigma-delta modulator

not subject to the noise shaping. 1-bit DAC is frequently used in oversampling techniques. The advantage of 1-bit DAC is that it is inherently linear [32], it has only two output values, and since two points define a straight line, no trimming or calibration is required. This inherent linearity is one of the major motivations for making use of oversampling techniques with 1-bit converters. It is common to use a 1-bit DAC and a corresponding 1-bit quantizer, which is simply a comparator. Consequently, if the sampling frequency is high enough, the sigma-delta ADC allows the use of a 1-bit quantizer to achieve high overall resolution.

To calculate the effective resolution of the $\Sigma\Delta$ modulator, we assume that the error e behaves as white noise that is uncorrelated with the input signal, the spectral density of the modulation noise

$$n_i = e_i - e_{i-1} \quad (2.12)$$

may then be expressed as

$$N(f) = E(f) |1 - e^{-j\omega T}| = 2e_{rms} \sqrt{T} \sin\left(\frac{\omega T}{2}\right) \quad (2.13)$$

where $\omega = 2\pi f$, $T = 1/f_s$, f_s is oversampling frequency.

In Figure 2.5, the feedback around the quantizer reduces the noise at low frequencies but increases it at high frequencies. The total noise power in the signal band is

$$P_n = \int_{-f_B}^{f_B} |N(f)|^2 df \approx e_{rms}^2 \frac{\pi^2}{3} (2f_B T)^3 = e_{rms}^2 \frac{\pi^2}{3} \frac{1}{OSR^3} = \frac{\Delta^2 \pi^2}{36} \left(\frac{1}{OSR}\right)^3 \quad (2.14)$$

where f_B is the signal band, and OSR is the oversampling ratio.

From equation.(2.5) and (2.14), we can calculate the maximum SNR as

$$SNR_{max} = 10 \log\left(\frac{P_s}{P_n}\right) = 10 \log\left(\frac{3}{2} 2^{2N}\right) + 10 \log\left(\frac{3}{\pi^2} (OSR)^3\right) \quad (2.15)$$

which is also equal to

$$SNR_{max} = 6.02N + 1.76 - 5.17 + 30 \log(OSR) \quad (2.16)$$

$$P_n = \int_{f_a}^{f_b} |N(f)|^2 df \approx e_{rms}^2 \frac{\pi^4}{5} (2f_B T)^5 = e_{rms}^2 \frac{\pi^2}{5} \frac{1}{OSR^5} = \frac{\Delta^2 \pi^4}{60} \left(\frac{1}{OSR} \right)^4 \quad (2.19)$$

From equation (2.5) and (2.19), we can calculate the maximum SNR as

$$SNR_{\max} = 10 \log \left(\frac{P_s}{P_n} \right) = 10 \log \left(\frac{3}{2} 2^{2N} \right) + 10 \log \left(\frac{5}{\pi^4} (OSR)^5 \right) \quad (2.20)$$

which is also equal to

$$SNR_{\max} = 6.02N + 1.76 - 12.9 + 50 \log(OSR) \quad (2.21)$$

We can see that doubling the OSR improves the SNR for a second-order modulator by 15dB, or equivalently, a gain of 2.5 bits/octave.

If we use A as the amplitude of the encoded test signal, Δ is the modulator output level. Then, the signal-to-noise ratio can also given by [9]

$$SNR = 15 \log_2 OSR + 6 \log_2 \left(\frac{A}{\Delta} \right) - 11.14, dB \quad (2.22)$$

We will use this equation later.

The technique can be extended to provide higher order predictions by adding more feedback loops to the circuit. Figure 2.7 shows the third-order $\Sigma\Delta$ modulator, where DAC is not shown because 1-bit DAC is used.

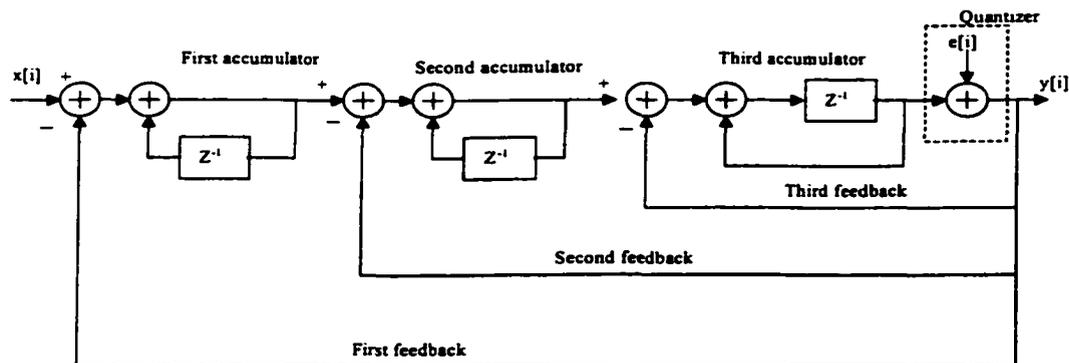


Figure 2.7 Third-order $\Sigma\Delta$ modulator

In this dissertation, the higher order $\Sigma\Delta$ modulator is used for generating the high-resolution analog signal. Here is an example of the design of a 5th order sigma-delta modulator (Figure 2.8). In this example, all the zeros of the noise transfer function (NTF) are placed at $z=1$ (i.e. dc) so that the modulator could be used for various oversampling ratios. In other words, the zeros are not spread over the frequency band of interest, as that would restrict the modulator's usefulness to a particular oversampling ratio [32].

Since all the zeros are assumed to be at $z=1$, the NTF has the following form:

$$NTF = \frac{(z-1)^5}{D(z)} = \frac{(z-1)^5}{\prod_{i=1}^5 (z-p_i)} \quad (2.23)$$

Thus for a modulator of order 5, we have 5 degrees of freedom. To find the pole of the system, we need the modulator stability while shaping as much quantization noise away from dc as possible. Therefore a high-pass filter is obtained when poles are placed in a Butterworth configuration, so that the cutoff frequency of the Butterworth filter becomes the single degree of freedom [10]. Specifically, with a passband edge at $f_s/25$, a 5th-order Butterworth high-pass filter has a peak gain equal to 1.34 (≤ 1.5 [12] [10] [13]), and the $NTF(z)$ is given by

$$NTF = \frac{(z-1)^5}{z^5 - 4.4189z^4 + 7.8412z^3 - 6.9818z^2 + 3.1186z - 0.5589} \quad (2.24)$$

and

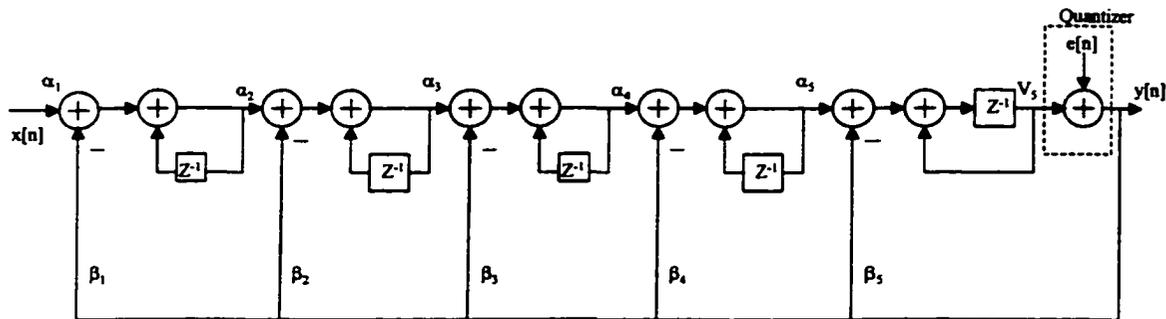


Figure 2.8 Cascade-of-integrators structure used to realize the 5th-order modulator

$$NTF = \frac{1}{1 + H(z)} \quad (2.25)$$

thus,

$$H(z) = \frac{1 - NTF(z)}{NTF(z)} = \frac{0.5808z^4 - 2.1577z^3 + 3.0170z^2 - 1.8812z + 0.4412}{(z - 1)^5} \quad (2.26)$$

Next, a suitable implementation structure is chosen. In this example, a cascade-of-integrators structure is used, as shown in Figure 2.8.

The α_i coefficients are set as $\alpha_1 = \beta_1$, $\alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = 1$. By initially setting $\alpha_1 = \beta_1$, we are allowing the input signal to have a power level similar to that of the feedback signal, $y(n)$. In other words, if α_i were initially set equal to one and β_1 were quite small, then the circuit would initially be stable for only small input signal levels [32].

Coefficients β_i are found by deriving the transfer function from the 1-bit DAC output to the V_5 and equating that function to $-H(z)$ in equation(2.26).

$$H(z) = \frac{z^4(\beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5) - z^3(\beta_2 + 2\beta_3 + 3\beta_4 + 4\beta_5) + z^2(\beta_3 + 4\beta_4 + 6\beta_5) - z(\beta_4 + 4\beta_5) + \beta_5}{(z - 1)^5} \quad (2.27)$$

Equating (2.27) with (2.26), the coefficients are found to be:

$$\begin{aligned} \alpha_1 &= 0.0001, & \alpha_2 &= 1.0, & \alpha_3 &= 1.0, & \alpha_4 &= 1.0, & \alpha_5 &= 1.0 \\ \beta_1 &= 0.0001, & \beta_2 &= 0.0025, & \beta_3 &= 0.0206, & \beta_4 &= 0.1164, & \beta_5 &= 0.4412 \end{aligned}$$

As mentioned before, in the design of high order $\Sigma\Delta$ modulator, stability needs to be considered. The choice of a Butterworth high pass configuration for the noise transfer function is due to the fact that the cutoff frequency of the Butterworth has some relation with the order of the modulator. In order to make the modulator stable: the higher the order of modulator, the lower the cutoff frequency. This means when we use a high order modulator to get higher in-band SNR, the cutoff frequency of Butterworth shifts left towards the origin point. Thus a higher order analog low pass filter (LPF) will

be needed to follow the modulator in order to get a higher SNR. This will be detail discussed in detail in section 2.5.

2.2.4 Parameter for the Performance of Bitstream: Length N

A $\Sigma\Delta$ modulator is based on an infinite-impulse response system whose output maps the input signal into an infinite-long sequence of bits. As a result, with a periodic input such as a sinusoidal signal, the output signal does not repeat itself. This implies that there isn't a single sequence that we can extract from the modulator output and claim that it represents the input periodic signal. Fortunately, we can come close [6]. By ensuring that the input signal completes an integer number of cycles inside the PDM stream, the output bitstream when repeated will be a close approximation to the original PDM signal. This is accomplished by following the rules of Coherent sampling [15]. For the best approximation, one should also ensure that the $N+1^{\text{th}}$ bit of the sequence from which the N bits are being extracted is the same as the first bit of the pattern to avoid the most obvious discontinuity. Figure 2.9 shows the frequency response of a long bitstream, length is 2^{16} . It is the output of $\Sigma\Delta$ modulator. Figure 2.10 shows the frequency response of a short bitstream, length equals 2^{10} . This short bitstream is part of the original long 2^{16} one. From these two plots, we can see the short bitstream keeps the property of the long one: same noise shaping, but with decreased in-band SNR. The simulation shows the deterioration is about 1 to 2 bits by measuring in-band SNR and ENOB. As long as the resolution of the short bitstream is acceptable for the DUT, this method is fine.

According to the rules of Coherent sampling [15], an integer number of cycles of the test signal should be embedded in the pulse-density modulated (PDM) bitstream. This way, the test frequency f_i (analog input frequency for testing DUT) should be chosen as a submultiples of the sampling frequency f_s , which is

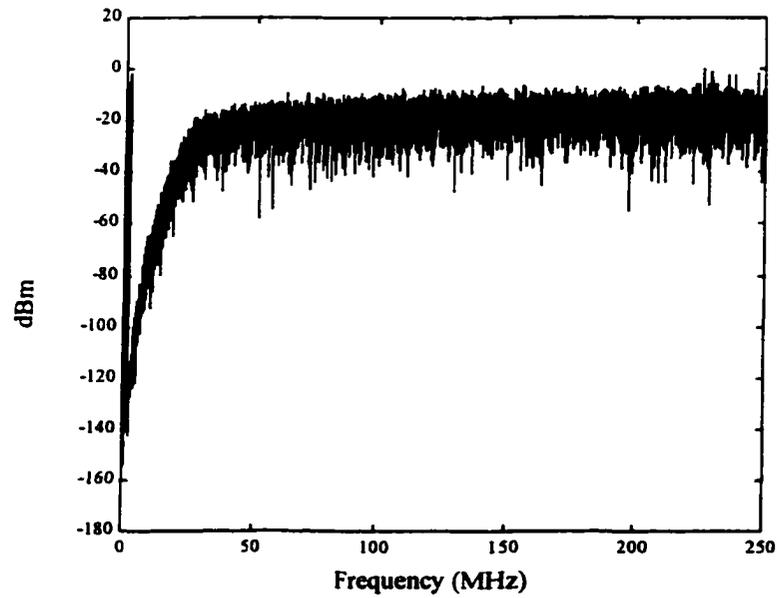


Figure 2.9 Frequency response of a long bitstream: length= 2^{16} (MatlabTM simulation result)

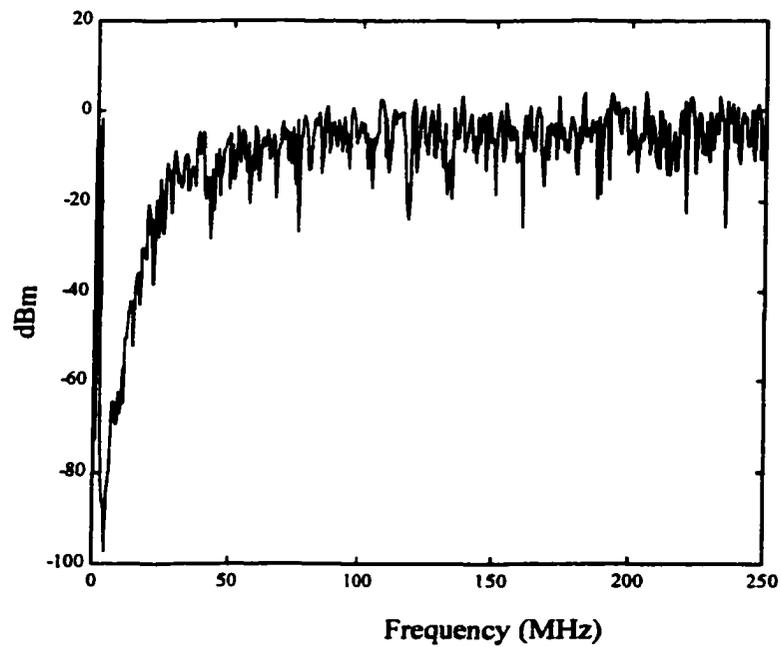


Figure 2.10 Frequency response of a short bitstream, length= 2^{10} (MatlabTM simulation result)

$$f_t = \frac{M}{N} f_s, \quad M = 1, 2, 3, \dots, \frac{N}{2} \quad (2.28)$$

where N is the length of the bitstream.

From equation (2.28), we can see the test frequency f_t is an integer multiple of the primitive frequency, f_s/N . In other words, the test frequency should consist of only those frequencies that are harmonically related to the primitive frequency. This also suggests that the primitive frequency limits the frequency resolution of the signal generation scheme [6]. For a fixed sampling frequency f_s , the resolution can only be improved by increasing the sequence length N . Also, it is noted, to encode the test signal into a PDM stream, it is essential that the signal lie within the bandwidth of the $\Sigma\Delta$ modulator, i.e.,

$$f_t \leq f_B \quad (2.29)$$

According to equation (2.22), for the 2nd order $\Sigma\Delta$ modulator, for a desired SNR, the bandwidth f_B of the $\Sigma\Delta$ modulator can be expressed as

$$f_B = f_s \times 2^{\frac{1}{15} \left(SNR + 26 - 6 \log_2 \left(\frac{A}{\Delta} \right) \right)} \quad (2.30)$$

Substituting equation (2.28), and equation (2.30) into equation (2.29), we got,

$$\frac{M}{N} f_s \leq f_s \times 2^{\frac{1}{15} \left(SNR + 26 - 6 \log_2 \left(\frac{A}{\Delta} \right) \right)} \quad (2.31)$$

or, eliminating f_s , it can be written as

$$\frac{M}{N} \leq 2^{\frac{1}{15} \left(SNR + 26 - 6 \log_2 \left(\frac{A}{\Delta} \right) \right)} \quad (2.32)$$

From equation (2.32), we can see the basic relationship between the sequence length N , the test tone frequency index M , the signal quality denoted by SNR over the modulator bandwidth of f_B , the

amplitude of the encoded test signal A , and the modulator output level Δ . Here we need to mention that the SNR value is over the modulator bandwidth f_B , so it is an ideal SNR value.

As an example, if a design has a shift register ring with length $N=1024$, running at a frequency of 500MHz, assuming the signal amplitude A is 1, the bitstream level is ± 1 , ($\Delta=2$), then according to equation (2.32), we got,

$$M \leq 1024 * 2^{\frac{1}{15}(SNR+32)} \quad (2.33)$$

There is a trade-off between the number of available frequencies and their signal quality if the bitstream length is fixed. For instance, if 80dB is desired, then test frequencies corresponding to M less than 5.8 are possible. So, M can take on integer values between 1 and 5. The primitive frequency in this case is $500\text{MHz}/1024=0.488\text{MHz}$. The corresponding test frequencies that can be generated using a 500MHz clock are 0.488MHz, 0.976MHz, 1.464MHz, 1.952MHz, 2.44MHz.

Also, from equation (2.33), we can find the minimum length of sequence that can meet the SNR quality requirement. By setting $M=1$ in equation (2.32), we got

$$N_{\min} = 2^{\frac{1}{15}\left(SNR+26-6\log_2\left(\frac{A}{\Delta}\right)\right)} \quad (2.34)$$

Here we need to note that N_{\min} is the ideal value, because the SNR value only counts the in-band noise power.

In this dissertation, we use this bitstream generation method to generate an on-chip analog signal. This analog signal will feed in the ADC, which is the device under test (DUT) (Figure 1.1). This DUT does not have frequency selectivity, and it will accept a much higher frequency range. Thus, the noise power needs to be counted to a much higher frequency range after the LPF to attenuate the high frequency noise. Consequently, the N_{\min} will be higher than equation (2.34).

2.2.5 Bitstream Optimization

Once a long sequence has been obtained, the best bitstream of length N in the sequence can be chosen. This can be accomplished through the use of different selection criteria [1]. These criteria include maximum spurious-free dynamic range (SFDR), amplitude precision and maximum SNR. The optimization is based on a search, where different N -length bitstreams selected from $y(n)$ are evaluated. This can be performed sequentially, moving the selection window bit by bit, or randomly, choosing a different bitstream every time. Both methods have similar results, but the random method usually converges to better results faster since closely spaced bitstreams in the sequence $y(n)$ tend to have similar characteristics. The random method thus finds clusters of better bitstreams faster.

Figure 2.11 shows a plot of SNR versus the bitstream position. From the plot, we can see the SNR varies from 56dB to 81dB. The small variance of amplitude will have a similar effect, which may cause the SNR to have about a 20dB variance.

Simulation shows for a fixed length bitstream, the best bitstream will change the position if the amplitude is changing. But what will be kept is the SNR of the best bitstream. For amplitude A_{mp1} , there is a best bitstream in position P_1 , which will generate SNR_1 . For another amplitude A_{mp2} , there is another best bitstream in position P_2 , which will generate SNR_2 . Simulations show as long as the original long bitstream length is fixed, and using the same order sigma-delta modulator, these two SNR values: SNR_1 and SNR_2 will have a similar value. This means, in the best bitstream search, we do not need to perform double loop search: position and amplitude. One loop search is enough.

After bitstream optimization, the accuracy of analog signal encoded in the bitstream can achieve the best result.

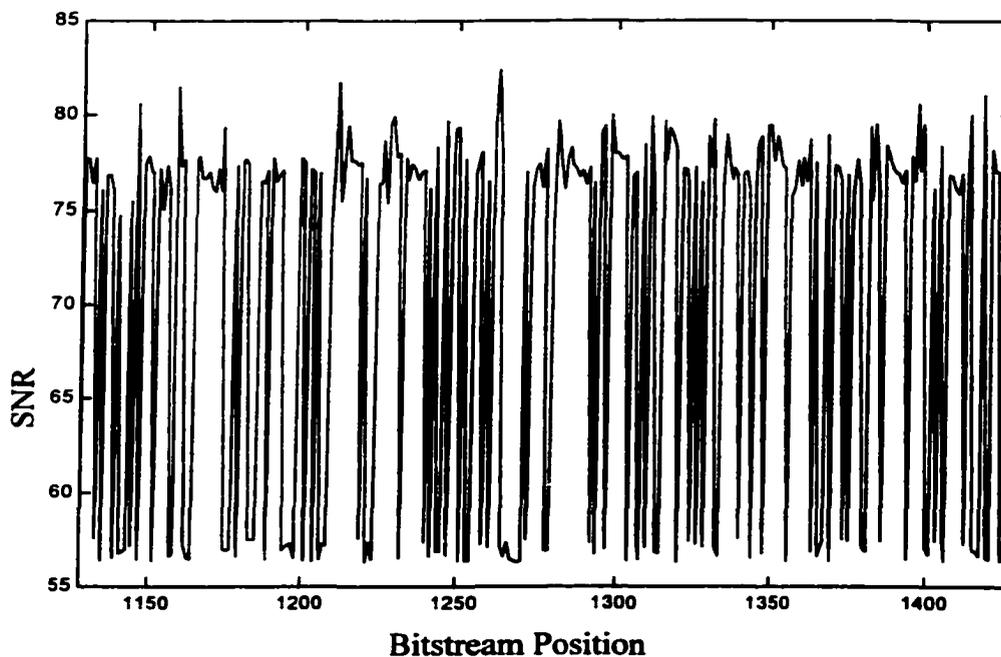


Figure 2.11 SNR vs. bitstream position

2.3 Shift Register Ring

Now we have the bitstream which is generated from the software, i.e. Matlab™ (Figure 1.1). We need to store this bitstream. The shift register is used here to store the bitstream and also to repeat the bitstream. If we use a signal source to provide a sinewave for the tester, we would have the signal source always ON during the test. Similarly here, we want the bitstream always provided for the DUT, so we use the shift register to repeat the bitstream. It works like a signal source which is always running. The length of the shift register will be N , which is also the length of bitstream. The operating frequency of the shift register ring will be the frequency of the oversampling $\Sigma\Delta$ modulator. For example, if the analog input signal for testing is 1.46434875MHz (this frequency will be explained

soon), the oversampling frequency f_s is 500MHz, which means the oversampling ratio is 1024/3, then the shift register ring should operate at 500MHz. This also sets a limit on the software, which means even in the software, the oversampling ratio has a limit.

The bitstream length N will be decided by the size of the shift register on the chip. Within N bitstream, there will include M complete cycles, as we discussed before, to follow the Coherent sampling rule, and this N and M need to be relatively prime numbers. In the example above, the N is 1024, M is 3, (We want 3 cycles in the bitstream stored on chip). According to equation (2.28),

$$f_i = \frac{3}{1024} \times 500MHz = 1.46434875MHz .$$

2.4 1-bit DAC

The output of shift register ring (Figure 1.1) is a repeated bitstream. The bitstream then goes through a 1-bit DAC to generate 1-bit analog signal. This 1-bit DAC makes the transition from the digital codes (one or zero) to the analog domain (V_{DD} or V_{SS} , V_{DAC}^+ or V_{DAC}^- , etc.). It acts as a buffer between the digital logic and the analog filter. This 1-bit DAC also need to work at the same speed as the shift register.

2.5 Analog Low Pass Filter

2.5.1 Specification of LPF

As mentioned earlier, the bitstream is noise shaped, with noise at the high frequency range. The output of the 1-bit DAC still has the same noise shaping: noise at the high frequency range. Therefore an analog low pass filter is needed to filter out the high frequency noise in order to get the high quality analog signal that is encoded in the bitstream by the $\Sigma\Delta$ modulator.

The choice of analog low pass filter is guided by the chosen NTF (noise transfer function) of the sigma-delta modulator and the attenuation of the out-of-band noise needed. Since we have chosen the high pass Butterworth configuration for the noise transfer function, a low pass Butterworth filter is chosen for filtering out the high frequency noise.

As mentioned earlier, for a high order $\Sigma\Delta$ modulator, its high SNR depends on the following LPF to attenuate the out-of-band noise (Figure 1.1).

Figure 2.12 shows a 7th order sigma-delta modulator frequency response. Figure 2.13 shows the frequency response of this 7th order modulator followed by a 7th order LPF. The resulting SNR is 98.1132dB. Figure 2.14 shows the same modulator followed by a 4th order LPF. The resulting SNR is 75.8411dB. Figure 2.15 shows the modulator followed by a 2nd order LPF, SNR=43.0123dB.

From the above three comparisons, we can see clearly that the performance of the analog signal generated on chip depends on the analog LPF chosen following the modulator. Generally speaking,

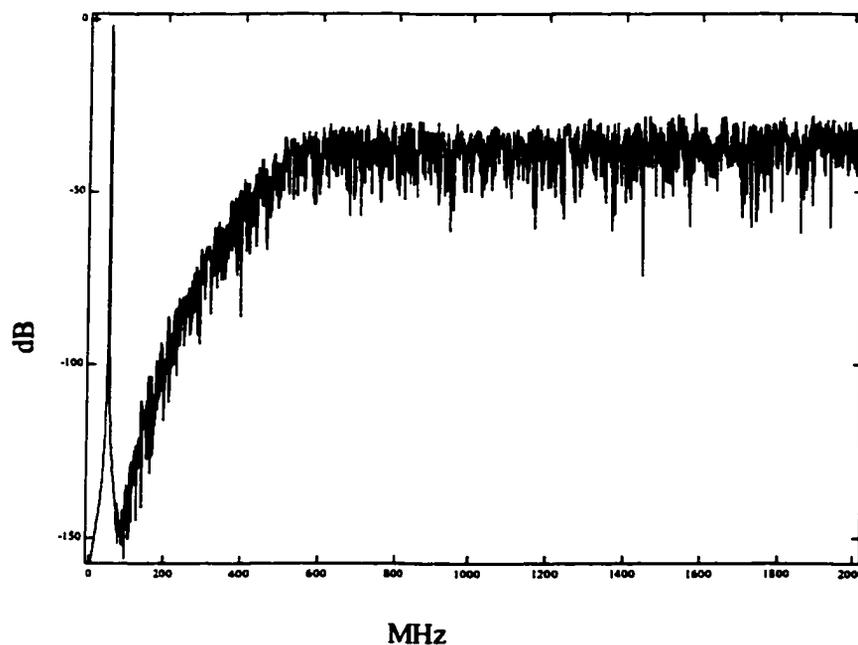


Figure 2.12 7th order sigma-delta modulator frequency response

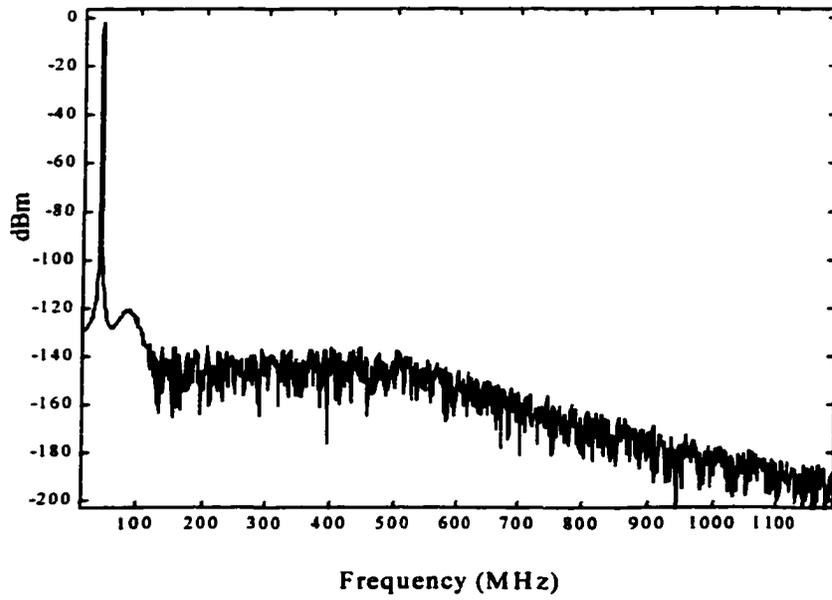


Figure 2.13 7th order modulator followed by 7th order Butterworth LPF, SNR=98.1132dB,
ENOB=16.0055

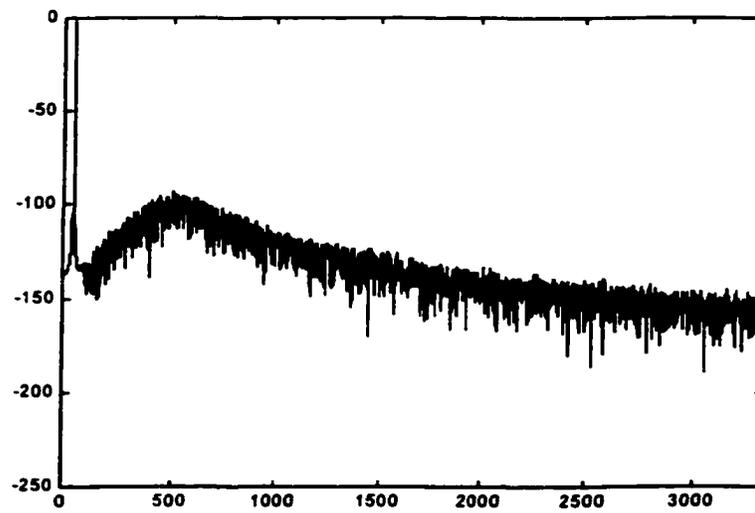


Figure 2.14 7th order modulator plus 4th order LPF, SNR=75.8411dB, ENOB=12.3058

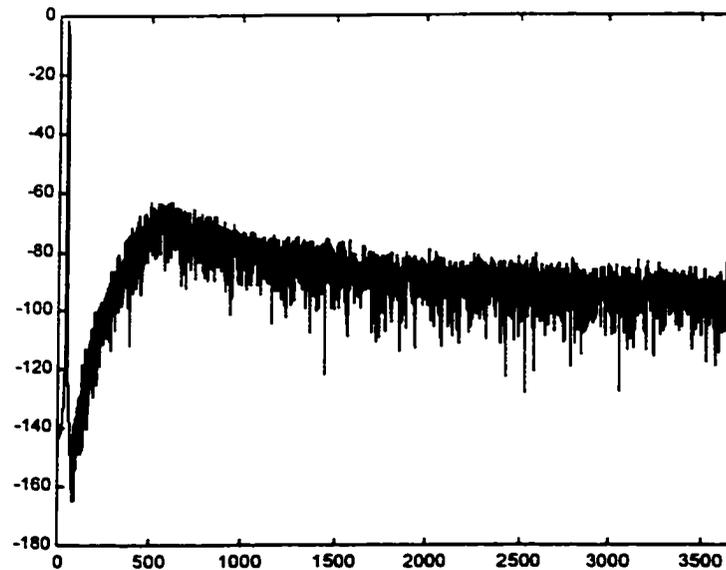


Figure 2.15 The same modulator followed by a 2nd order modulator, SNR=43.0123,
ENOB=6.8525

From the above three comparisons, we can see clearly that the performance of the analog signal generated on chip depends on the analog LPF chosen following the modulator. Generally speaking, the order of the analog LPF should preferably be one order higher than the loop-filter order of the sigma-delta modulator, to suppress the high-frequency noise [16]. Otherwise, the analog signal generated will be affected by the nonlinearity of the LPF.

What we need to note is that the above SNR calculation counts the noise power through the entire Nyquist frequency band, not only the in-band SNR as calculated in equations (2.16) and (2.21). The SNR here is lower than in those equations. Since at this point, we do not know what the device under test (DUT) is, we do not know if it has frequency selection or saying limited bandwidth, so we need to account for all the Nyquist frequency range noise power.

Figure 2.16 shows the 7th order vs. 2nd order modulators. From this figure, we can see that higher order modulators need high order LPFs to attenuate out-of-band noise. The noise floor of the lower order modulator is relatively flat in low the frequency range compared with the higher order modulator. For our design, the goal is to test a 9-bit ADC. An analog signal with 11bit accuracy is needed for the test. Considering the random noise in the 1-bit DAC and the analog LPF that further deteriorate the signal's quality, a 12-bit accuracy signal is needed from the modulator that is software generated. Thus, a 4th order Butterworth LPF is needed (Figure 2.14)

For a practical system, the LPF can reside off or on chip. As shown in Figure 2.17, BG is the bitstream generator, F is the filter, and CUT is the circuit under test. While an off-chip LPF has some limitations as a BIST method, it has merits for calibration and characterization [1]. First, it gives full flexibility in the passband choice of for the filter, extending the ranges of signals that can be generated. Second, the filter can be tuned to optimize its performance. The drawback is the use of two pins for the filter and the deteriorating performance and design complexity due to the pin and package parasitics for those pins.

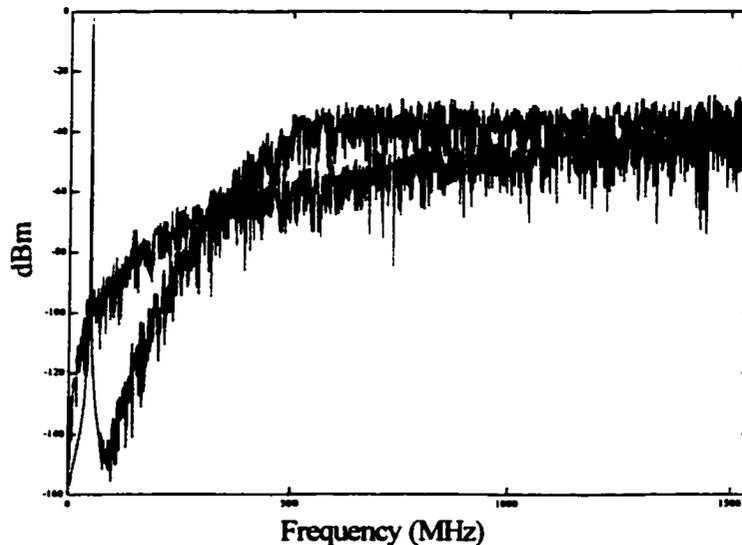


Figure 2.16 7th order vs. 2nd order modulator frequency response

2.5.2 Power Consideration of the Generated Analog Signal

If the LPF is ideal, the generated analog signal power at its output will be the same as the input signal power of the modulator, which is implemented by software (Figure 1.1).

Let the amplitude of analog input signal of the modulator be A_m , the digital levels for the modulator output code $y[n]$ be 1 and 0. The 1-bit DAC, which follows the LPF, has two voltage levels: V_{DAC}^+ and V_{DAC}^- . In this design, using TSMC's $0.25\mu\text{m}$ process, V_{DAC}^+ will be V_{ref}^+ , and V_{DAC}^- will be V_{ref}^- , which will be the input range of the analog LPF. The power of the analog input signal $A_m \sin \omega t$ of the modulator is then

$$P_{S_IN} = \frac{1}{2} A_m^2 \quad (2.35)$$

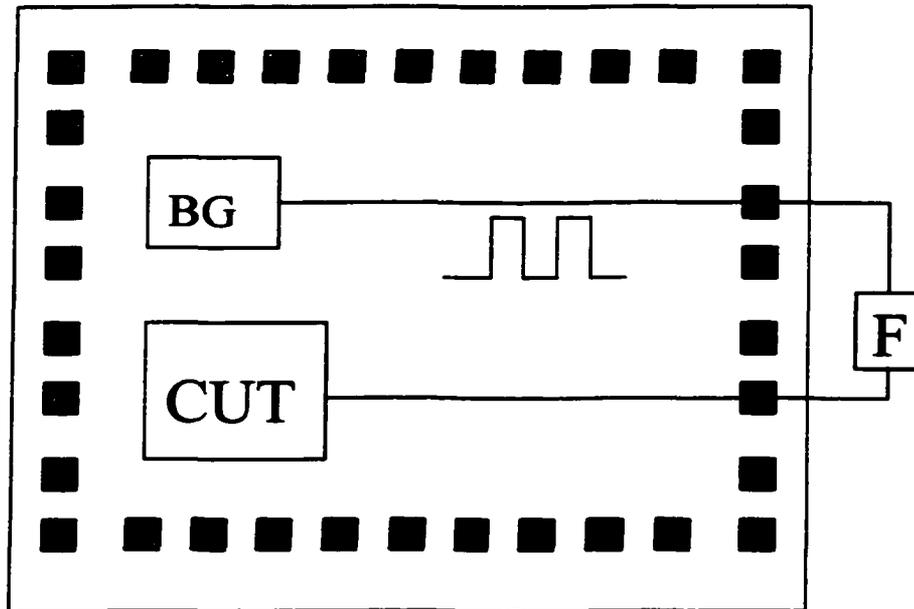


Figure 2.17 Filter off-chip scheme

and the average power for the output code $y[n]$ (for code level 0 or 1) is

$$P_{TOT} = \frac{1}{M} \sum_{i=1}^{i=M} y^2[i] = \frac{1}{2}. \quad (2.36)$$

The average power for the output of the 1-bit DAC is

$$P_{TOT,DAC} = \frac{1}{M} \sum_{i=1}^{i=M} y_{DAC}^2[i] = \frac{1}{2} (\Delta V_{DAC})^2 = \frac{1}{2} \Delta V_{DAC}^2 \quad (2.37)$$

where $\Delta V_{DAC} = V_{DAC}^+ - V_{DAC}^-$. This is shown in Figure 2.18. And we also can see

$$P_{TOT,DAC} = P_{dc} + P_S + P_N \quad (2.38)$$

Where P_{dc} is dc power, P_S is the signal power at the output of the LPF, and P_N is the output all-band noise power. They are:

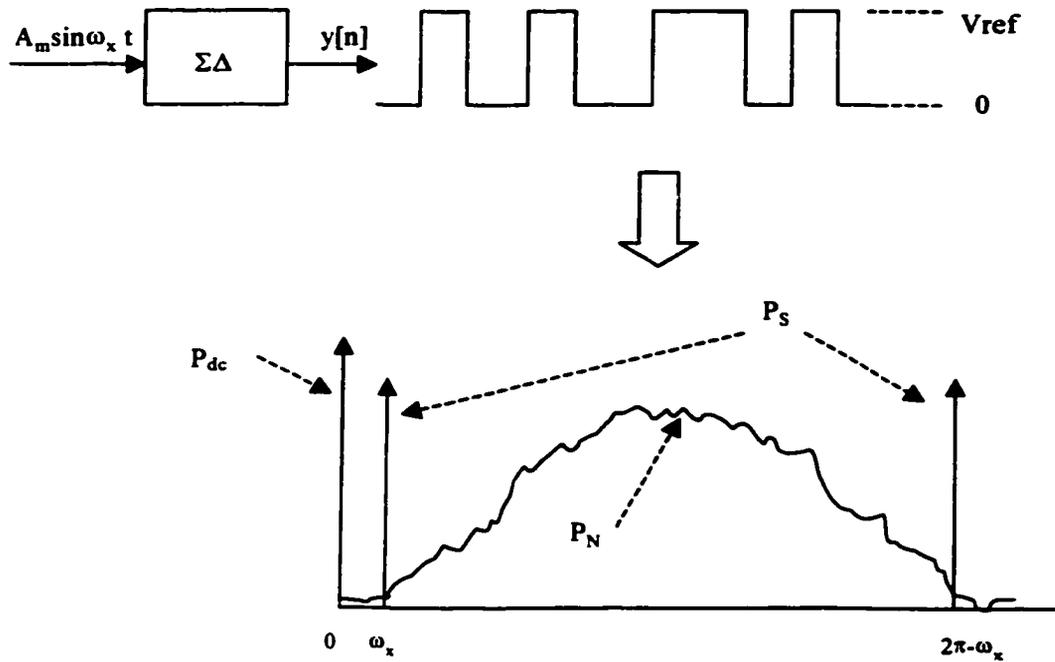


Figure 2.18 Power for the input and output of the $\Sigma\Delta$ modulator (ω_x is the analog testing signal generated by the $\Sigma\Delta$ modulator)

$$P_{dc} = \left(\frac{1}{2}\Delta V_{DAC}\right)^2 = \frac{1}{4}\Delta V_{DAC}^2 \quad (2.39)$$

$$P_S = \left(\frac{1}{2}A_m\right)^2 = \frac{1}{4}A_m^2 \quad (2.40)$$

$$P_N = \frac{1}{4}\Delta V_{ref}^2 - \frac{1}{4}A_m^2 \quad (2.41)$$

After the analog LPF, if it is ideal, we will get the signal power, which is P_S in equation (2.40).

The reason that $P_S = \frac{1}{2}P_{S_IN}$ is the output code of the modulator is two levels: 1 and 0. If the code levels are +1 and -1, we will get $P_S = P_{S_IN}$. In this dissertation, we use the code levels +1 and -1.

2.6 Multi-phase Signal Generation

The shift register ring also has the capability of providing a multi-phase signal [6]. By tapping off selected locations in the chain, one can obtain signals phase-shifted from the original to a resolution of $360/K$ degrees, where K is the shift register length N divided by the number of cycles in the bitstream,

$$K = \frac{\text{bitstream length } N}{\text{\#of cycles}} \quad (2.42)$$

For example, if the bitstream length N is 1024, and it contains a 2-cycle sinewave, by driving outputs with the 256th and 512th flip-flops, we can get two output sinusoidal signals with π phase shift. These kinds of signals will be needed in the design if the DUT is fully differential. The diagram of multi-phase signal generation is shown in Figure 2.19, and generated two-phase signals are shown in Figure 2.20.

The schematics of the D-flip flops and a discussion of their operations can be found in section 6.3.

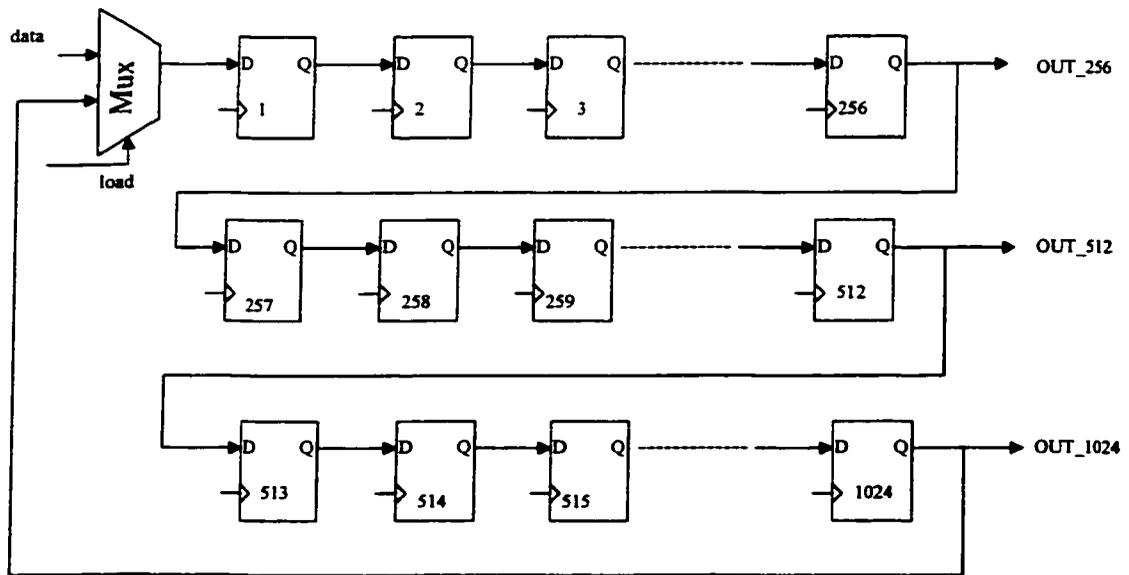


Figure 2.19 Diagram of multi-phase signal generation.

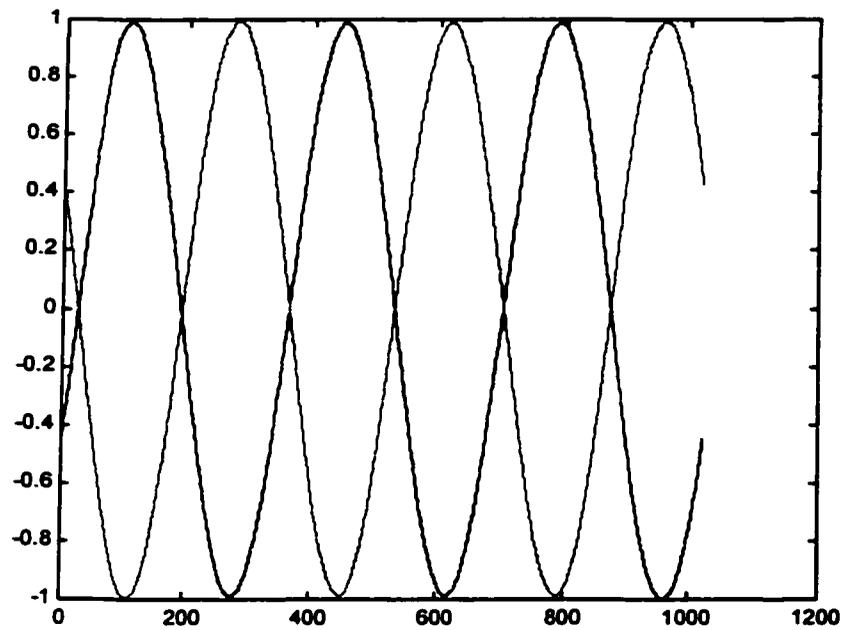


Figure 2.20 Generated multi-phase signals

2.7 Chapter Conclusion

In this chapter, the on-chip analog signal generation was discussed. It has been shown how a periodic signal that is encoded in an infinitely long PDM (pulse density modulated) bitstream can be very well approximated by a short-length periodic sequence of bits. It was also shown how to use memory on-chip to store the analog signal on chip. By using $\Sigma\Delta$ modulator, a PDM bitstream is generated and stored and repeated by shift register ring. An analog LPF following shift register ring attenuates the high frequency range noise. Thus the on-chip analog signal is generated. With the on-chip analog signal generated, we can use it as an exciting signal for the device we want to test, which is the pipeline ADC in this dissertation.

3 ADC TESTING

3.1 Introduction

ADC testing is an important topic in this dissertation. As will be shown later, the re-configurable pipeline ADC architecture is based on its performance, in other words, the ADC testing results. In this chapter, the basic techniques of ADC testing will be carried out. It is composed of two parts: static testing and dynamic testing. The goal is to provide theoretical background to ADC testing.

First the quantization error in the ADC is analyzed. Then the techniques of Coherent sampling and histogram testing method are covered. Some examples are given for static and dynamic testings.

3.2 Error Measurements

A key consideration in converter testing is that an ADC is not the mathematical inverse of a DAC [15]. An ideal DAC has no transfer ambiguity, but an ADC cannot be described by a point-to-point

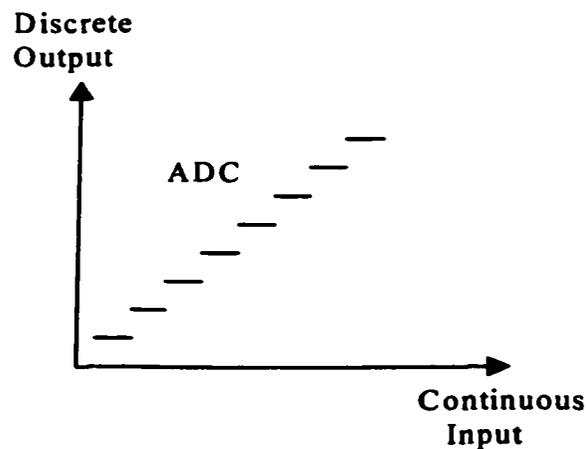


Figure 3.1 ADC transfer curve

map. An ideal ADC has a discrete output set, for each code out, there is a fuzzy input: a continuum of input voltage. This means an ADC has one-way uncertainty. If a specified analog level is applied to the input of a perfect ADC, we know precisely what its output code response will be. But if we are told, instead, only what the output code state is, we cannot tell the exact input voltage, only its range, as shown in Figure 3.1. The uncertainty is uniformly distributed over the width of the step, or the least significant bit (LSB), or the quantum size. It is a tenet of ADC testing that one does not test an ADC by applying DC voltages (or steps) and looking to see if the ADC responds with the right or wrong codes [15].

It is clear that we can identify and correct for all of the contributing factors: gain, offset, phase, and noise, by looking at the full collection of ADC codes in context (i.e., by analyzing the set as a vector not as isolated code words). ADC testing involves much statistical work.

3.2.1 Quantization Errors

The quantization process introduces an irreversible error. The quantization step is determined by the number of steps a signal is quantized into. This number of quantization steps is expressed in a number of (binary-weighted) bits N . A signal $A_j + \varepsilon$ is ideally quantized into level A_j as long as the value of ε is between $-\frac{q_s}{2} < \varepsilon < \frac{q_s}{2}$, where q_s is the quantization level spacing, i.e. LSB (least significant bit). Quantization error basically never exceeds an amplitude level equal to $\pm \frac{q_s}{2}$. Signals that are somewhat larger than $A_j + \frac{q_s}{2}$ are quantized to the next quantization level A_{j+1} .

The mean-square-error due to quantization can be calculated. Assume over a long period of time all levels of uncertainty within the quantizing region $A_j + \frac{q_s}{2}$ appear the same numbers of times. A

uniform probability density function over the interval $-\frac{q_s}{2}$ to $+\frac{q_s}{2}$ is defined as follows. The mean-squared value of ε will be

$$E(\varepsilon^2) = \frac{1}{q_s} \int_{-\frac{q_s}{2}}^{\frac{q_s}{2}} \varepsilon^2 d\varepsilon \quad (3.1)$$

The symbol $E(\cdot)$ represents the statistical expectation. The rms quantization error voltage can be represented by:

$$e_{rms}^2 = E(\varepsilon^2) = \frac{1}{12} q_s^2 \quad (3.2)$$

or the RMS quantization distortion voltage is

$$D = \frac{q_s}{\sqrt{12}} \text{ volts, RMS} \quad (3.3)$$

In an N-bit linear binary ADC, there are 2^N code levels in the full-scale range (FSR). The two end steps have no outer bounds, and so do not actually have statistical centers.

The RMS amplitude is

$$\text{FS Sine Amplitude} = \frac{q_s \times 2^N}{\sqrt{8}} \text{ volts, RMS} \quad (3.4)$$

The signal-to-noise ratio (SNR) can be calculated using equation (3.4) divided by the equation (3.3), resulting in:

$$SNR = 2^N \sqrt{1.5} \quad (3.5)$$

and expressing this in dB, results in:

$$SNR = N \times 6.02 + 1.76 \quad dB \quad (3.6)$$

It is sometimes convenient to compare the actual, in-circuit performance of converters by equating the distortion and/or noise with an ideal converter with fewer bits. The equivalent number of bits based on distortion or noise is given by solving the above equation for N:

$$ENOB = \frac{SNR - 1.76}{6.02} \quad (3.7)$$

3.3 Coherent Sampling [15]

The trivial case of coherence is the condition where everything runs at the same rate (i.e., where all ratios are 1:1). As the term implies, coherence means the condition in which every element is allowed to run at a different rate, if required, yet be completely time coordinated in any way the programmer chooses.

Here is an example, $F1$ is the sampling rate of an ADC under test. Say, 50Ms/s. We want to test this converter near Nyquist frequency. Now we want to find the test frequency $F2$. Coherent sampling determines the relation between $F1$ and $F2$ in this way:

$$F2 = F1 * M/N \quad (3.8)$$

where M and N are two relatively prime integers. If we choose $N=1024$, then a value of 511 is suitable for M , giving a test frequency $F1=50*511/1024=24.951172\text{MHz}$.

In digital signal processing (DSP)-based testing, N equals to the number of samples, and M equals to the number of signal cycles. The amount of information available from a sampled waveform is maximized when M and N are relatively prime. With relatively prime ratios, the amount of information is proportional to N and is independent of M . A prime M/N ratio ensures that each cycle contributes unique, independent information.

Figure 3.2 and Figure 3.3 show the spectrum of a 6-bit ADC by choosing different M/N ratios, where in Figure 3.2 M and N are not relatively prime numbers and in Figure 3.3 M and N are relatively prime numbers. We can see the noise floor is different between these two plots. In this dissertation, we always follow the rule of Coherent sampling.

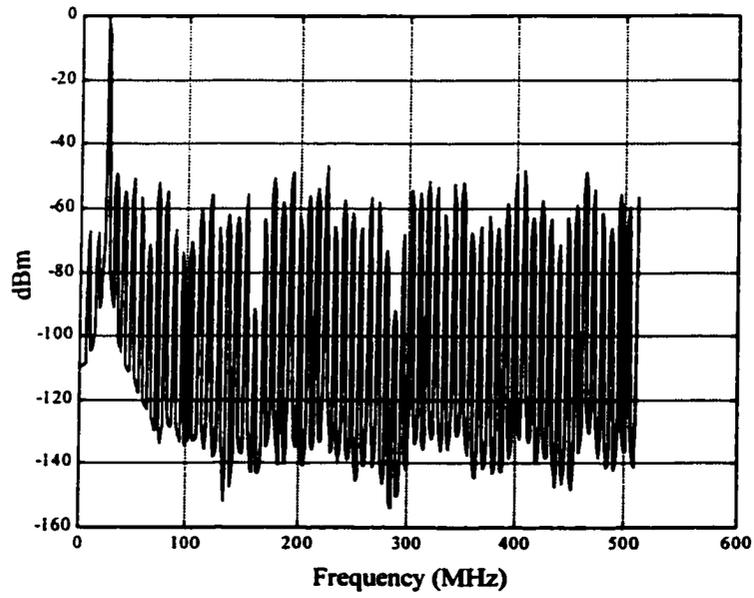


Figure 3.2 Spectrum of ideal 6bit ADC with $N=1024$, $M=24$, where noise floor is not correct

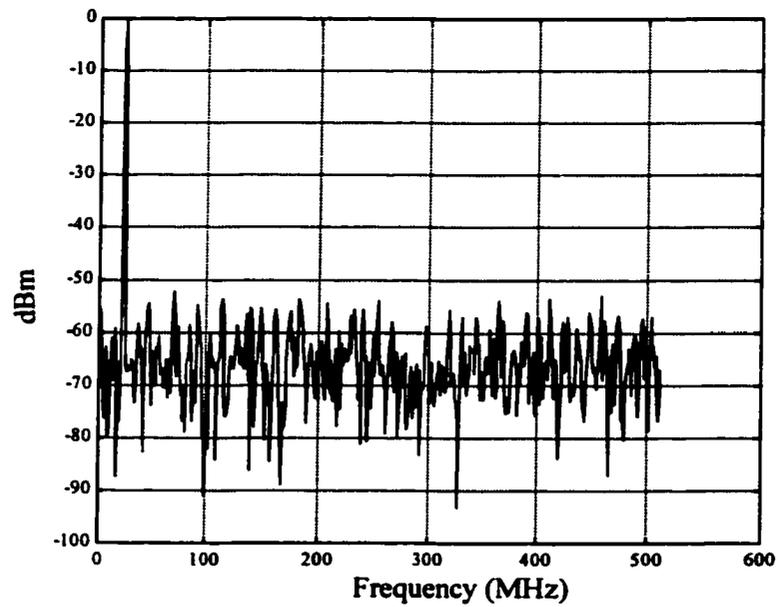


Figure 3.3 Spectrum of ideal 6bit ADC with $N=1024$, $M=23$

3.4 Linear Histogram Testing

The word “histogram” means a drawing or record of the past. In ADC testing, a histogram shows how many times each different output code word appears in the response vector, without regard to the location. The complete list is derived from the ADC output vector by a single TALLY instruction [15].

The analog input can be any wave whose amplitude distribution is known. However a linear ramp simplifies the computation because step width is directly proportional to the tally of each code. DNL is quickly obtained by subtracting the average step size. Ramp amplitude is not critical but is usually made a little larger than the nominal ADC range to allow for unit-to-unit variation in analog gain.

If there is an 8-bit ADC, containing 2048 data, there is totally 256 unique codes. We need to count the hits in each code or say step, like hit (1) counting how many times code level 1 appears, and hit(2) counting how many times code level 2 appears. The hit(0) and hit(255) are discarded, because these two steps have no outer bound. Also we need to count the average hit, which is (total hits)/(256-2). DNL is calculated by

$$DNL(i) = \frac{hit(i) - hit_avg}{hit_avg} \quad (3.9)$$

where *hit_avg* is average hit.

And INL is calculated by

$$INL(i) = INL(i-1) + \frac{DNL(i) + DNL(i-1)}{2} \quad (3.10)$$

It is noted that the histogram method is blind to nonmonotonicity. It is also noted that the histogram method does pick up missing code, wherever the DNL = -1LSB.

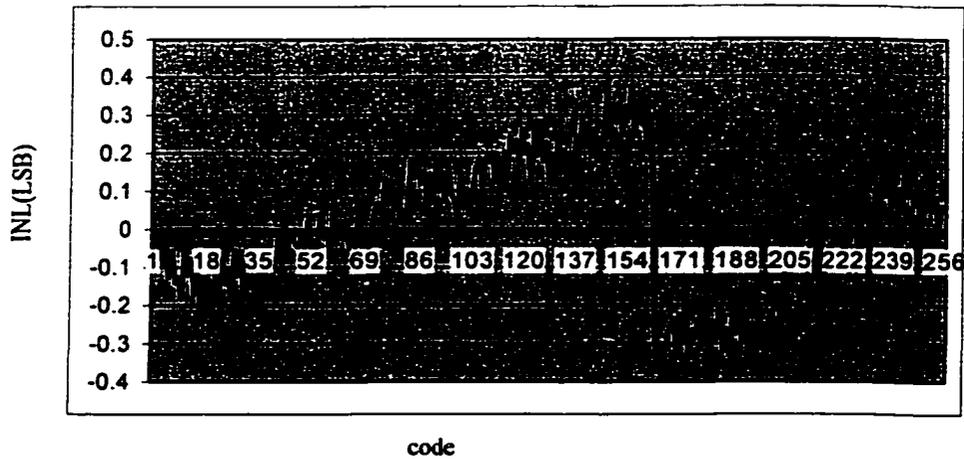


Figure 3.4 The INL of an 8-bit ADC

Even in a so-called static test, linearity is a function of more than the DC decision levels. Inequality of step widths is partly caused by comparator noise, which randomly displaces the instantaneous decision levels. Use of a slower ramp can increase the tally resolution.

Figure 3.4 and Figure 3.5 show the typical INL and DNL plot of an 8-bit pipeline ADC. Since no DNL is equal to -1LSB , there are no missing codes.

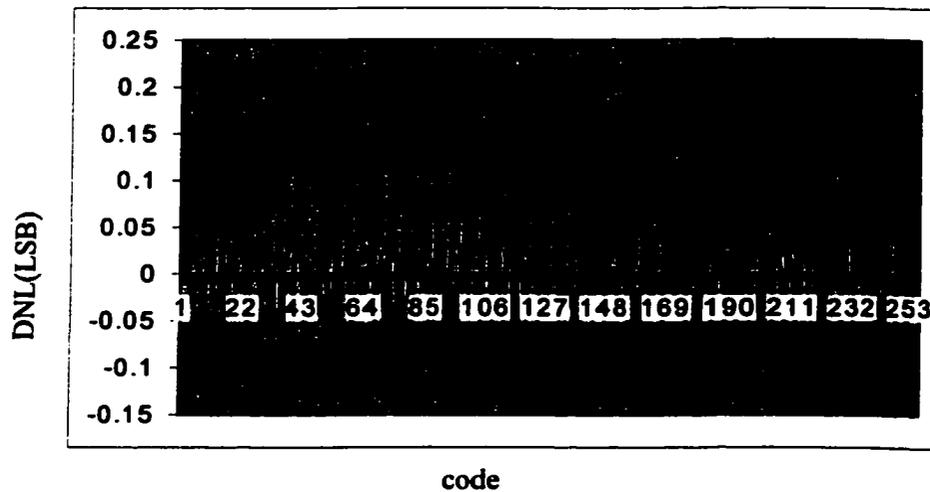


Figure 3.5 The DNL of a 8-bit ADC

3.5 Dynamic Histogram Testing

The ADC response spectrum is easily obtained from the ADC output vector by the Fast Fourier transform (FFT). Figure 3.6 shows the spectrum from a single “tone”, or pure sinusoid, at the frequency $F_t = 43/1024 * F_s = 43/1024 * 50MHz = 2.099609375MHz$. The physical frequency is usually the sampling rate, F_s , which is 50MHz. The absolute frequency is not critical in most cases, but the ratio is. If the input has to be exact in absolute frequency, then it would be F_t , not F_s . That becomes the physical standard. The critical thing is not to end up with a fraction of a test cycle left over. This is ideal for Coherent DSP techniques. For example, the F_s we chose here is 49.999872MHz, and F_t is 2.099604MHz.

From the dynamic test, especially when an FFT plot is available, the signal to noise ratio (SNR) and the Spurious-free dynamic range (SFDR) can be calculated. SFDR is defined to be the difference between the fundamental signal power and the highest harmonic power, which can be observed from the FFT plot. The SNR is defined to be the signal to noise power ratio, which can also be calculated from the FFT result.

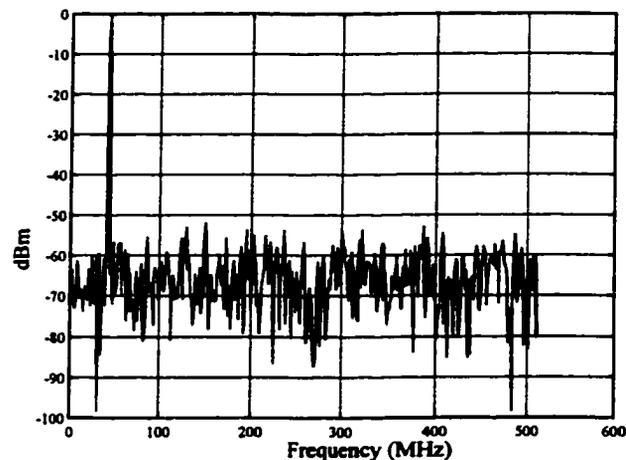


Figure 3.6 Spectrum of 6-bit ADC obtained with relatively prime M

3.6 Chapter Conclusion

In this chapter, the background knowledge of ADC testing is addressed. The Coherent sampling concept is introduced. The histogram testing methods for both static testing and dynamic testing are covered. The goal is to provide theoretical background to ADC testing, which is important in this dissertation because the re-configuration pipeline ADC architecture is based on its dynamic performance measurement results. This will be shown in chapter 5.

4 ON-CHIP DYNAMIC TESTING OF ADCs

4.1 Introduction

As we know, the key problem to be solved in BIST for pipeline ADCs is the on-chip measurement of SNR performance of the ADCs. The standard method in industry is the Fast Fourier Transform (FFT) method. Another alternative method, which will be used in this dissertation, is the Narrow-band filter method [18]. In this chapter, these two methods of dynamic testing of pipeline ADCs are discussed. The emphasis is on the second method: a digital narrow band filter (NBF) method. The comparison of the two methods is covered based on the measuring accuracy and the complexity of the hardware implementation.

4.2 SNR

4.2.1 FFT Method

Assume that in the frequency band of interest, the on-chip analog test stimulus is of the form

$$x_m(t) = A_x \cos(\omega_x t + \phi_x). \quad (4.1)$$

Let the ADC output be

$$y_{out}(n) = s(n) + \eta(n) \quad (4.2)$$

where $s(n)$ is the signal and $\eta(n)$ is the noise.

For optimum accuracy, a sample record $y_{out}(n)$ consisting of M samples must contain an integer number of whole cycles of the sinewave [15]. To compute the SNR, first calculate $Y_{out}(k)$, which is

the M-point DFT (Discrete Fourier Transform) of $y_{out}(n)$ and is given by

$$Y_{out}(k) = \sum_{n=0}^{M-1} y_{out}(n) e^{-j(2\pi/M)kn} \quad (4.3)$$

Let the desired frequency component ω_x be the j -th element of $Y_{out}(k)$. Based on Parseval's relation for the FFT, together with classical statistical theory [24], an estimate of the variance of the signal $s(n)$, which is also the signal power \hat{P}_s , is given by

$$\hat{\sigma}_s^2 = \hat{P}_s = \frac{2}{(M-1)M} |Y(j)|^2 \quad (4.4)$$

An unbiased estimate of the noise power is given

$$\hat{\sigma}_\eta^2 = \hat{P}_\eta = \frac{2}{(M-1)M} \sum_{k=1}^{(M-1)/2} |Y(k)|^2, k \neq j \quad (4.5)$$

From equation (4.4) and (4.5), we can get SNR for the test frequency ω_x as

$$SNR = 10 \log_{10} \left\{ \frac{|Y(j)|^2}{\sum_{k=1}^{(M-1)/2} |Y(k)|^2, k \neq j} \right\} \quad (4.6)$$

Since the SNR is defined in the frequency domain and no approximations are made, the SNR as computed by the FFT method in equation (4.6) is completely unbiased.

The FFT method is the current industry standard. The resources required to compute an M-point FFT include M memory locations to store the samples. In addition, sine and cosine functions must be available. These may be either computed directly, or stored in a look-up table, which would require M/4 entries for M samples [18]. Unfortunately, most mixed-signal IC's do not have this kind of computational power, so alternate method must be used.

4.2.2 Narrow Band Filter Method

The main idea in this method is using a digital narrow band filter (NBF) to separate the signal and noise. It is based on a method for on-chip time-recursive implementation of an arbitrary transform (such as FFT), which is described in [20]. Figure 4.1 shows a block diagram of the filter.

NBF has two outputs, one is a notch output, and another is a band pass output, both of them are tuned to the frequency of the input test tone. This filter is a biquad with very good sensitivity and noise properties. $X_m(z)$ is the filter input, while $X_{bp}(z)$ is the band pass output and $X_{nr}(z)$ is the notch output. The transfer function $H_{bp}(z)$ from input $X_m(z)$ to the band pass output $X_{bp}(z)$ is

$$H_{bp}(z) = \frac{X_{bp}(z)}{X_m(z)} = -\frac{k_{bw}}{2} \frac{(1+z^{-1})(1-z^{-1})}{1-(2-k_{bw}-k_w^2)z^{-1}+(1-k_{bw})z^{-2}} \quad (4.7)$$

The bandwidth of the band pass output of the filter is set by choosing the appropriate PR (pole radius). It is shown that the parameter k_{bw} in Figure 4.1 primarily determines the pole radius of the

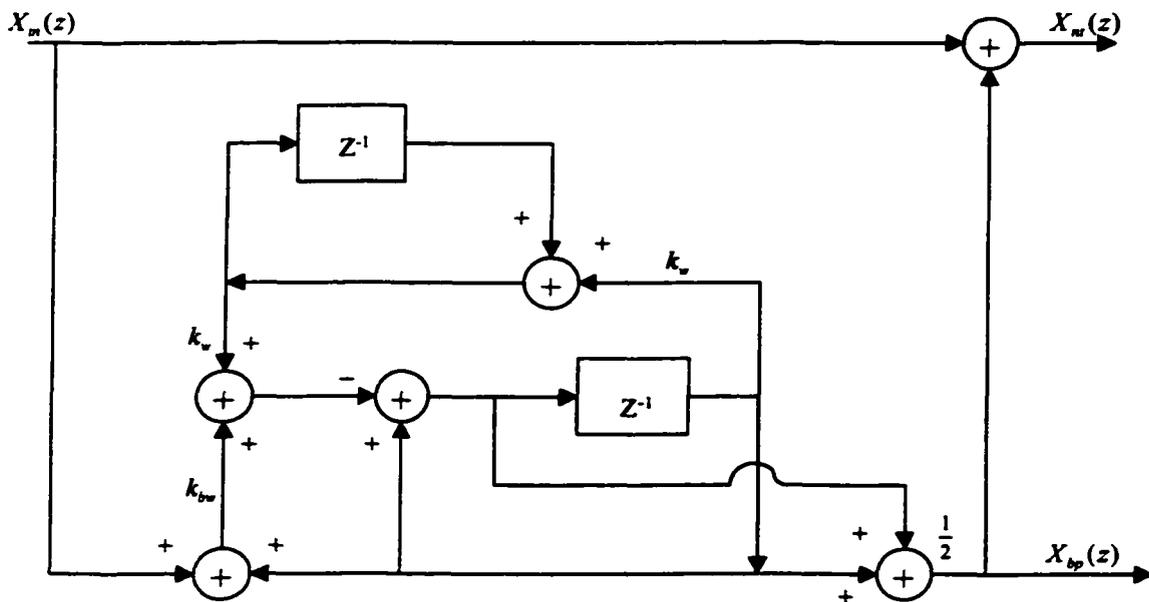


Figure 4.1 Digital narrow band filter (NBF) [18]

filter. It is given approximately by $k_{bw} \cong 1 - PR^2$. The center frequency f_o for the band pass output is set by selecting the parameter k_w in Figure 4.1 to be

$$k_w = 2\sqrt{1 - \frac{K_{bw}}{2} \sin\left(\frac{2\pi f_o T}{2}\right)} \quad (4.8)$$

where $T = \frac{1}{f_s}$, f_s being the sampling clock frequency of the filter. In this design, which uses NBF for on-chip measurement of pipeline ADC, this f_s will be the ADC's sampling frequency, and f_o will be the frequency of analog input frequency fed into the ADC under test.

The transfer function of the notch output is

$$H_{nr}(z) = \frac{X_{nr}(z)}{X_{in}(z)} = -\frac{2 - k_{bw}}{2} \frac{1 - \frac{2(2 - k_{bw} - k_w^2)}{2 - k_{bw}} z^{-1} + z^{-2}}{1 - (2 - k_{bw} - k_w^2)z^{-1} + (1 - k_{bw})z^{-2}} \quad (4.9)$$

Since the notch output is obtained by merely subtracting the band pass output $X_{bp}(z)$ from the input $X_m(z)$, all of the desirable properties concerning the sensitivity and noise apply to it as well.

The narrow band digital filter is connected to the output of the pipeline ADC.

Let $y(n)$ be the digital output ADC, it will also be the digital sequence applied to the filter input X_m . The signals \hat{s} and $\hat{\eta}$ denote the digital code emerging from the band pass output $X_{bp}(z)$ and the notch output $X_{nr}(z)$, respectively. The sequence $\hat{s}(n)$ emerging from the band pass output $X_{bp}(z)$ is zero mean. The estimated signal power \hat{P}_s is computed as the sum of squares, given by

$$\hat{P}_s = \hat{\sigma}_s^2 = \frac{1}{M-1} \sum_{n=1}^M (\hat{s}(n))^2 \quad (4.10)$$

The sequence $\hat{\eta}$ emerging from the notch output x_{nr} is not zero-mean, so its variance, which is the estimated noise power \hat{P}_n , is calculated as [24]

$$\hat{P}_n = \hat{\sigma}_\eta^2 = \frac{1}{M-1} \sum_{n=1}^M (\hat{\eta}(n))^2 - \frac{(\sum_{n=1}^M \hat{\eta}(n))^2}{M(M-1)} \quad (4.11)$$

The estimated SNR is then calculated from the quotient as follows,

$$SNR_{est} = 10 \log_{10} \left(\frac{\hat{\sigma}_s^2}{\hat{\sigma}_\eta^2} \right) = 10 \log_{10} \left(\frac{\hat{P}_s}{\hat{P}_n} \right) \quad (4.12)$$

As shown, we can obtain an estimate of the SNR without having to compute an FFT. One sum and two squared-sums are needed to accumulate the signal and noise power.

4.2.3 Bias of SNR Estimate via Digital Filter

It may be seen that if a noisy sine-wave signal consisting of white noise and a single tone is passed through the band pass filter of Figure 4.2, some of the noise will pass through due to the finite bandwidth of the filter. It will then be tallied with the signal power, resulting in an overestimation. Figure 4.3 and Figure 4.4 show the FFT of bandpass and notch outputs of the narrow-band digital filter. From Figure 4.3, we can see the over-estimation effect apparently where there is a clearly visible “skirt” associated with the FFT of the bandpass output. This skirt is the noise energy which has passed through the bandpass filter to be integrated along with the signal power. In addition, the notch filter of Figure 4.5 removes some of the noise power with the result that noise power is underestimated. The net result is that the narrow band filter based estimation SNR_{est} is indeed slightly higher than the true SNR [18][19]. Figure 4.6 shows the relation between pole radius of the narrow band filter and the bias of the estimated SNR. As the pole radius approaches unity, the bias decreases until the estimated SNR converges on the correct value. It is stated that simulations show the narrow-band digital filtering method can yield accurate measurement results for 16-bit ADC’s [18].

This method yields a biased estimate of the signal power, and hence the SNR test result is biased. In addition, it is necessary to wait for the filter to reach steady state before initiating the measurement. We also notice that, when the desired precision is increased, the necessary time to wait for the narrow-band digital filter to reach steady state also increases. However, if the pole locations of the filter are chosen properly, the bias can be made very small and the settling time can be made tolerable. The amount of precision required for the coefficients will affect the choice of the number of bits used for the arithmetic in the filter, which will in turn affect the silicon area needed to build in hardware.

This method can give results that approach the accuracy of the FFT method. The resources required to implement the narrow-band digital-filtering method are less than what is required to

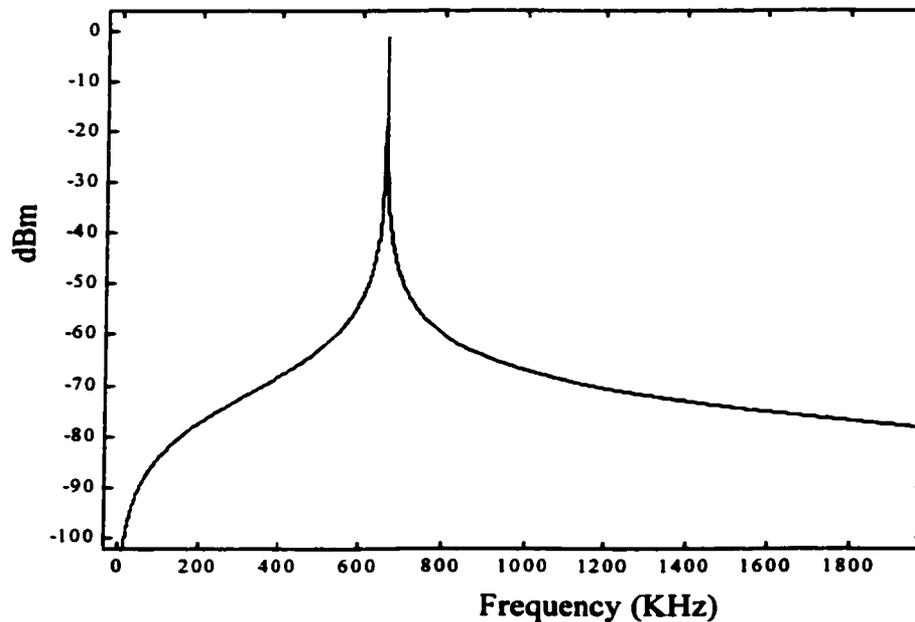


Figure 4.2 The frequency response $|H_{bp}(f)|$ of the bandpass output of the narrow-band digital filter

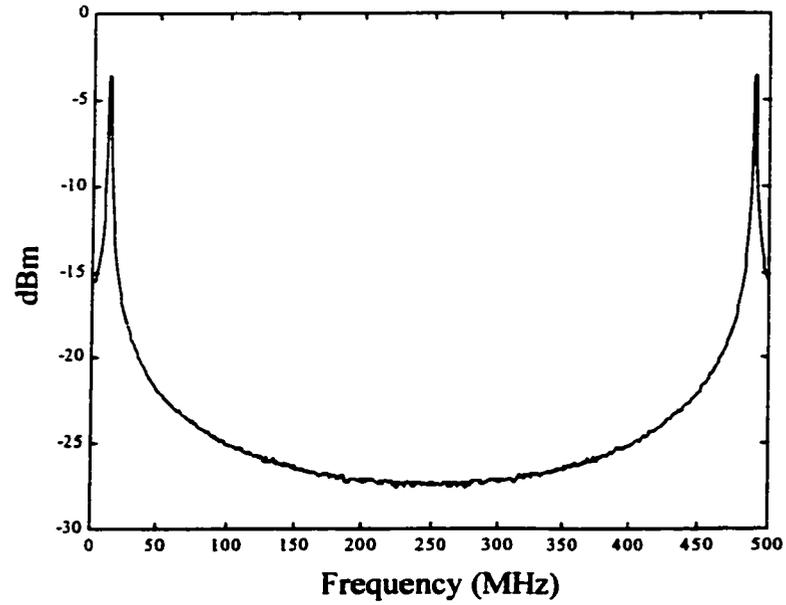


Figure 4.3 The FFT of the bandpass output of the narrow-band digital filter

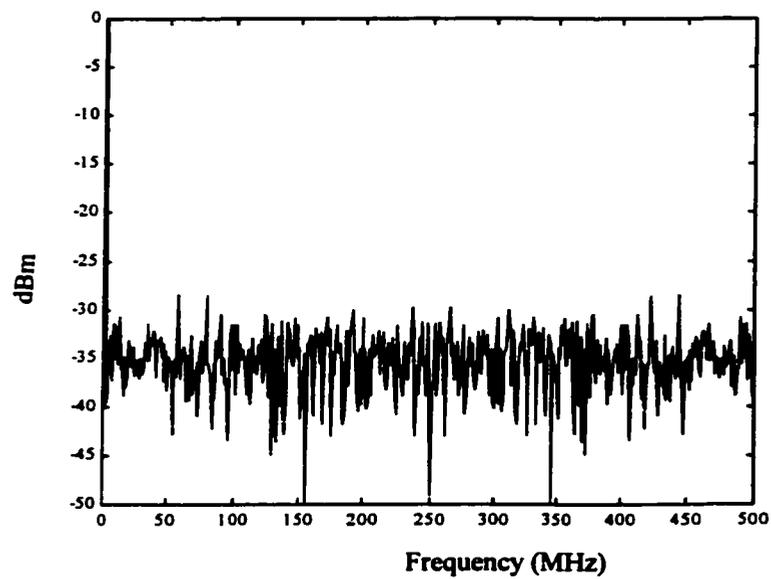


Figure 4.4 The FFT of the notch output of the narrow-band digital filter

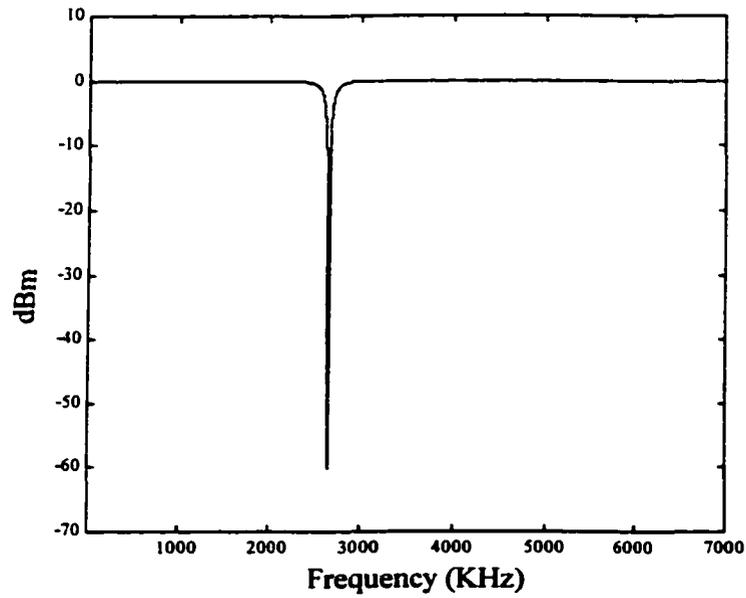


Figure 4.5 The frequency response $|H_n(f)|$ of the notch output of the narrow-band digital filter

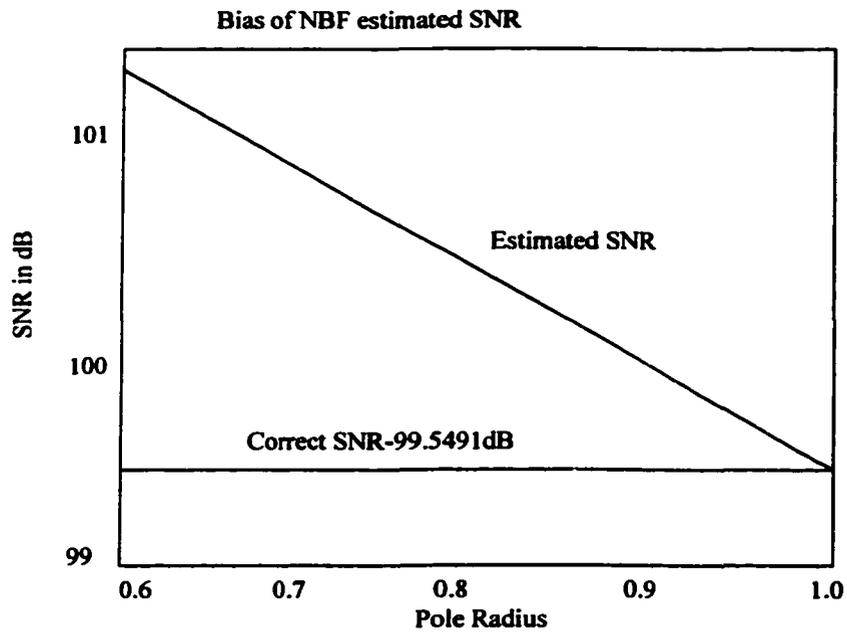


Figure 4.6 Bias of estimated SNR vs. pole radius

implement the FFT method, because sine and cosine values need not be made available, and the samples do not all need to be captured in memory. From equation (4.10) and (4.11), we can see, in addition to the narrow-band digital filter itself, only one sum and two squared-sums need be accumulated to obtain the signal and noise powers [18].

4.3 Chapter Conclusion

In this chapter, the on-chip dynamic performance testing method for ADCs is covered. Compared with the standard *Fast Fourier Transform (FFT)* method in industry, the *Narrow-band filter (NBF)* method requires less hardware and less computational power. It is a preferred method for most IC's on-chip dynamic performance measurement. The structure and theoretical analysis of the NBF are carried out in this chapter, with simulation results shown. The goal is to provide theoretical and technical knowledge on how to fulfill on-chip dynamic performance measurement of ADCs.

5 DESIGN OF RE-CONFIGURABLE PIPELINE ADC

5.1 Introduction

In this chapter, the basic architecture of pipeline ADC is described and the design procedure of pipeline ADC is carried out. A re-configurable pipeline ADC architecture is proposed. The grouping algorithm that can find the best performing configuration quickly is presented. Based on dynamic performance measurements, the best performing configuration is chosen from a collection of possible pipeline configurations. To validate the algorithm, a 9-bit 40MHz re-configurable pipeline ADC is designed and implemented in TSMC's 0.25 μ m digital CMOS process.

5.2 9-bit 40MHz Pipeline ADC Implementation

Figure 5.1 shows the general architecture of a pipeline ADC. In this figure, inter-stage gain is set to the normal value of 2 for a 1-bit per stage architecture. In this chapter, the focus is on re-configurability of the pipeline ADC. For 9-bit, 1-bit per stage, 9 stages are shown in the structure. In this figure, each stage is identical, except the first S&H stage, and consists of a fully differential sample and hold amplifier (SHA), a 1-bit sub-ADC and a 1-bit sub-DAC. The sub-DAC functionality is rolled in as part of the SHA switched-capacitor circuit, which is referred to as the multiplying DAC (MDAC). The 1-bit sub-ADC is simply a comparator.

5.2.1 System Specification

This design is targeting at a 9-bit 40Msamples/s pipelined ADC using 0.25 μ m TSMC digital process with power supply 2.5V. As shown in Figure 5.1, this ADC is a radix 2, 1-bit-per-stage

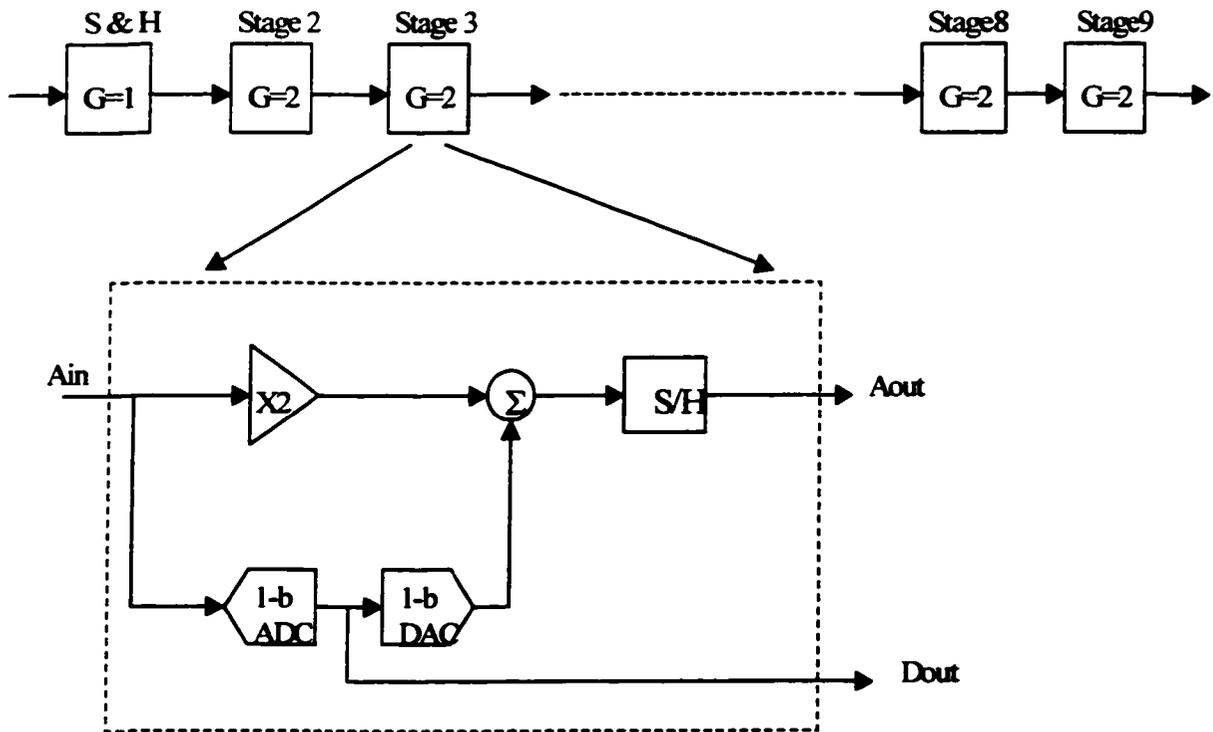


Figure 5.1 The general architecture of pipeline ADC

pipelined ADC. The pipeline starts with a front-end sample-and-hold (S/H) stage. The analog output of the S/H stage is fed into the first multiply-by-two (MX2) stage, stage 2, which is followed by 7 identical MX2 stages, stage 3 to stage 9. The first MX2 stage will give the digital output, which is the most significant bit (MSB) of the digital output word. Stage 9 is followed by another 1-bit sub-ADC, which will give the least significant bit (LSB). This is not shown in the Figure 5.1. In the pipeline ADC, the various stages of the pipeline operate concurrently. At any instant, while the first stage processes the current input sample, the second stage processes the amplified residue of the previous input sample from the first stage. The basic timing of a pipeline ADC is using two non-overlapping clocks.

5.2.2 Multiplying Digital-to-Analog Converter

Figure 5.2 shows the schematic of the fully differential Multiplying Digital-to-Analog converter (MDAC) [29]. It consists of an opamp, 4 capacitors and 13 switches including a center sampling switch. Figure 5.3 shows the timing diagram of the clocks. The two main clocks Φ_1 and Φ_2 are non-overlapping. To reduce the sample-to-hold transition error, two extra clocks, Φ_{1-} and Φ_{1+} are also used.

The operation of this MDAC can be shown clearly in Figure 5.4. When Φ_1 , Φ_{1-} and Φ_{1+} are high, switches M1 ~ M9 are on, the opamp inputs are connected to each other and also connect to the common-mode voltage, at the same time, the opamp outputs are also reset to the common-mode voltage. The voltage on the C_s sampling capacitors and C_f integrating capacitors track the input. When Φ_{1-} goes low, the inputs of the opamp are released from the common-mode voltage but remain connected to each other through M1. M2 and M3 are chosen to be much smaller than M1 so that the charge injection error is small and decreases exponentially until M1 is turned off. Then Φ_1 goes low and M1 is turned off.

The signal is sampled on the sampling capacitors as well as the integrating capacitors. The charge injection from switch M1 is signal independent. This is further removed by the differential configuration, assuming the charge from M1 is evenly distributed into the two input nodes [35]. Thus the size of M1 is chosen to be large. Then Φ_{1+} goes low, and switches M4, M5, M6, M7, M8, M9 are turned off. Since the charge on the bottom plates of the capacitors has no place to go, the charge injection from these switches does not affect the charge on the capacitors. Hence, it has no effect on the sampled signal. This is called bottom plate sampling. M4, M5, M6, M7 are chosen to be large to reduce the effects of nonlinear resistance and impedance mismatch. During Φ_1 , neglecting the opamp offset, the charge stored on the capacitors is:

$$Q_1 = V_{in}(C_s + C_f) = 2V_{in}C \quad (5.1)$$

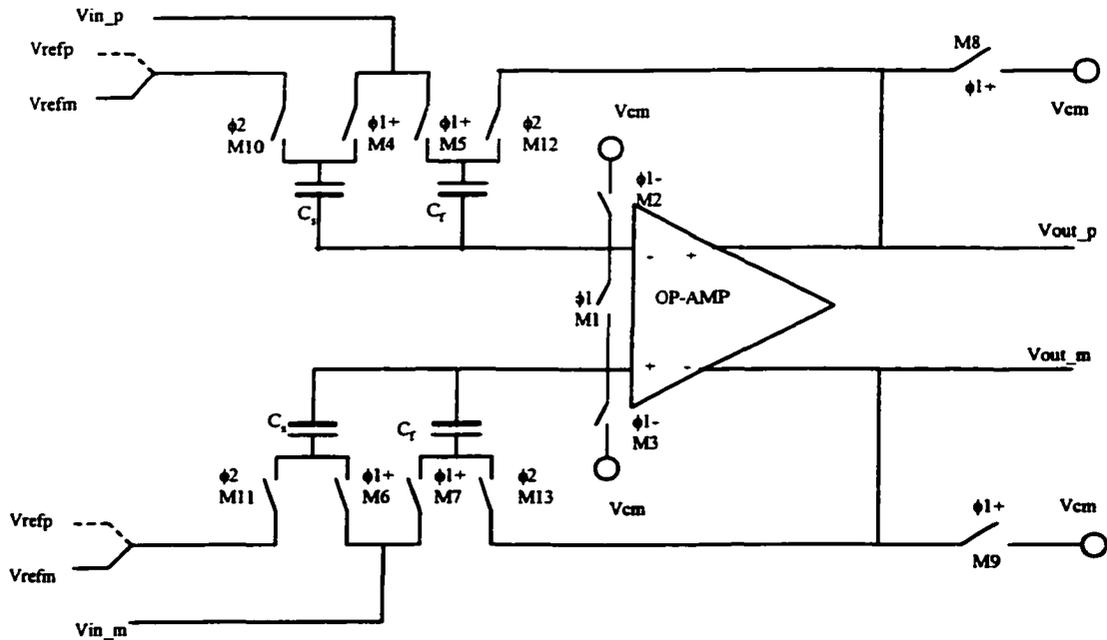


Figure 5.2 Fully Differential Multiplying Digital-to-Analog Converter

in this structure, $C_s = C_f = C$.

During Φ_2 , the integrating capacitors C_f are connected to the opamp outputs, and the sampling capacitors C_s are connected to either V_{refp} or V_{refm} depending upon the digital output of the

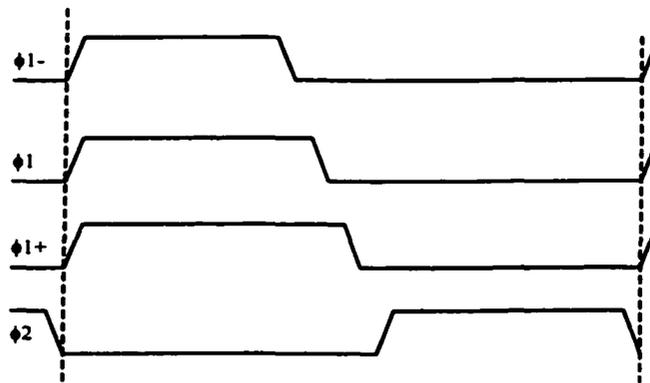
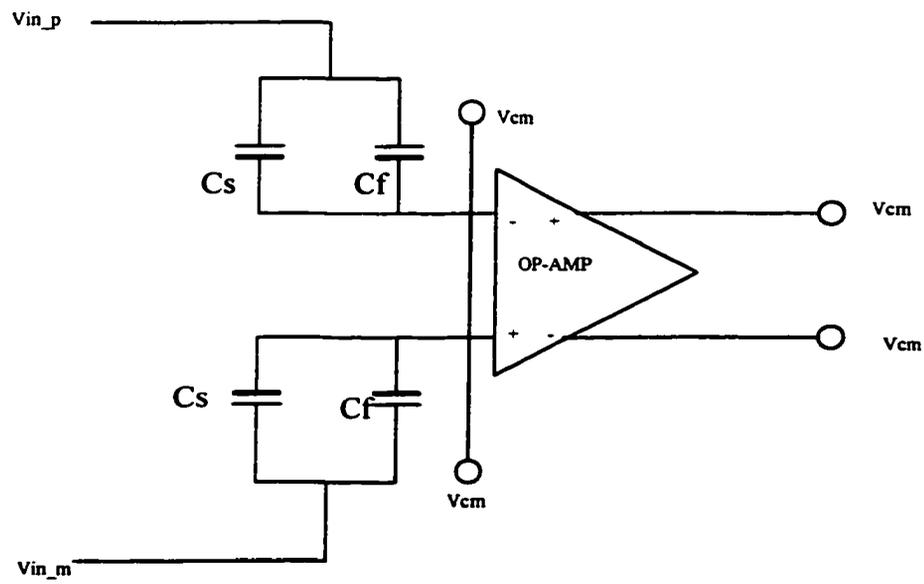
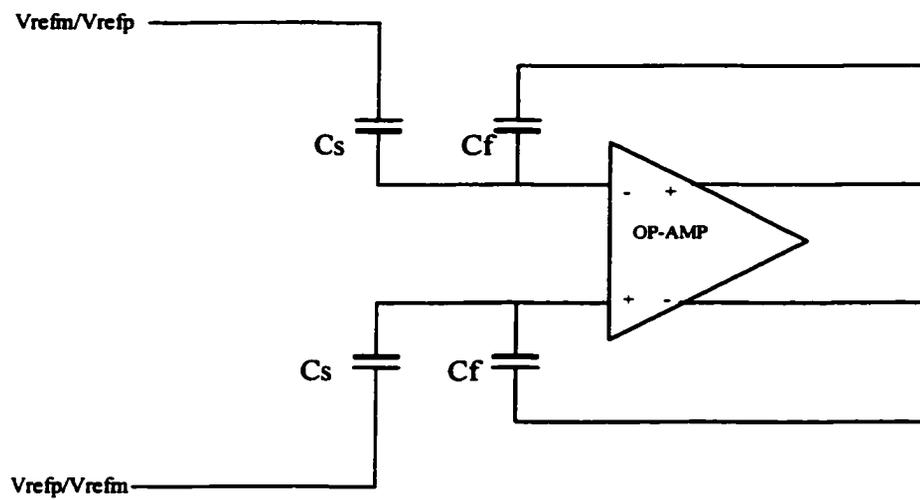


Figure 5.3 Timing Diagram



(a) MDAC during sampling phase



(b) MDAC during amplifying phase

Figure 5.4 MDAC during different clock phase

previous stage. The total charge during this phase is:

$$Q_2 = V_{out}C_f - V_{refp}(V_{refm})C_s = V_{out}C - V_{refp}(V_{refm})C \quad (5.2)$$

By the principle of charge conservation, the charge is the same in both clock phases. By equating the two charges, $Q_1 = Q_2$, the transfer function of the MDAC is obtained as

$$V_{out} = 2V_{in} - V_{refp}(V_{refm}) \quad (5.3)$$

The resulting output consists of two parts: one arising from the feedforward of the integrating capacitors and the other from the charge transfer from the sampling to the integrating capacitors. Since only the second part is dependent on the ratios of the capacitors, the feedforward technique reduces the effect of the capacitor mismatch on the inter-stage gain. This is important because the accuracy of the inter-stage gain of 2 determines the linearity of the ADC. Beside feedforward, there are two other motivations for sharing the integrating capacitors (used also for sampling the input during the sampling mode). Firstly, the speed of the pipeline ADC is always limited by the opamp settling time in the S/H “hold” phase. To optimize the speed, we want to maximize the loop transmission bandwidth, which is determined by the following equation:

$$BandWidth_{closedloop} = \beta \cdot \omega_t \quad (5.4)$$

where ω_t is the unity gain bandwidth, and $\beta = \frac{C_f}{C_s + C_f} = \frac{1}{2}$ in this case. Without sharing the

integrating capacitor, the feedback factor β will be 1/3 since $C_s = 2C_f$. Therefore, the speed is about 50% higher compared to the conventional scheme. Secondly, the sampling capacitor size is reduced by about 50%, which results in considerable die-area saving considering the total number of stages.

However, the size of the capacitors should be decided carefully taking $\frac{KT}{C}$ noise into consideration.

For this $\frac{KT}{C}$ value should be calculated and it should be at least less than 0.1LSB.

5.2.3 Operational Amplifier Design

The first step in the design of ADC is to derive the opamp specifications from the given ADC requirements. The following section will quantize the opamp specifications for the ADC requirements. The prototype was targeted at 40Msamples/s and 9-bit accuracy.

5.2.3.1 DC Gain

The finite open-loop DC gain of the opamp gives a fixed gain error at the output. The DC gain should be such that this error is much less than $\frac{1}{2}$ LSB. The effect of the opamp's finite open loop gain can be understood by Figure 5.5.

The transfer function of Figure 5.5 is given by

$$\frac{V_{out}}{V_{in}} = \frac{A_0}{1 + A_0\beta} \quad (5.5)$$

where A_0 is opamp's open loop gain. Due to the finite DC gain, the closed loop gain will be different from $\frac{1}{\beta}$. The relative error should be much less than $\frac{1}{2}$ LSB. The relative error is given by

$$Error = \left| \frac{\frac{A_0}{1 + A_0\beta} - \frac{1}{\beta}}{\frac{1}{\beta}} \right| < \frac{1}{2} LSB \quad (5.6)$$

Solving the equation for A_0 with $\beta = 1/2$, the open-loop DC gain requirement of the opamp can be obtained. In this design, for a 9-bit ADC with input of range 1V, $\frac{1}{2} LSB = \frac{1}{2^{10}}$, and we can get

$$A_0 > 2^{11}.$$

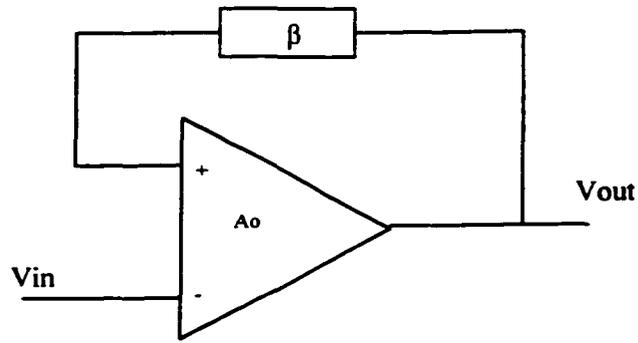


Figure 5.5 Gain circuit using an opamp

5.2.3.2 Bandwidth

The bandwidth of the opamp limits the speed of the operation of the ADC. For a 40MHz application, assuming 50% duty cycle, the settling time is less than 12.5ns. With the assumption that the opamp is compensated such that the higher-order poles play a minor role in settling, at the midband frequencies, the transfer function is

$$A(s) = \frac{\omega_{ia}}{s} \quad (5.7)$$

With the feedback setting like Figure 5.5, the close loop gain A_{CL} is

$$A_{CL}(s) = \frac{A(s)}{1 + \beta A(s)} \cong \frac{1}{\beta} \left[\frac{1}{1 + \frac{s}{\omega_{ia}\beta}} \right] \quad (5.8)$$

Thus, the closed-loop amplifier has a -3dB frequency given by $\omega_{-3\text{dB}} \cong \beta\omega_{ia}$. The time constant is

$$\tau = \frac{1}{\omega_{-3\text{dB}}} = \frac{1}{\beta\omega_{ia}}, \text{ where } \omega_{ia} \text{ is unity gain bandwidth.}$$

Hence the unit step response for the system is

$$V_{out} = \frac{1}{\beta}(1 - e^{-t/\tau}) \quad (5.9)$$

In general, all the sources of errors should contribute to less than $\frac{1}{2}$ LSB. Since we have two sources, each should contribute at most $\frac{1}{4}$ LSB. This means that the accuracy of the finite gain as well as the settling time should be accurate to more than 11 bit for a 9-bit ADC. Therefore, the required accuracy of settling should be within $1/2^{11}$, which is 0.05%, and the settling time is approximately 7.6τ .

From the above analysis, for a 9-bit 40MHz ADC, $7.6\tau < 12.5\text{ns}$. Let us assume $7.6\tau = 7.6\text{ns}$, then $\tau = 1\text{ns}$, and a unity gain bandwidth of 320MHz is required.

5.2.3.3 Load Capacitance

The load capacitance is an important consideration as it affects the bandwidth of the opamp. The load capacitance would include $C_S + C_f$ of the next stage and also the capacitance in the switched-capacitor common-feedback circuit.

5.2.3.4 Gain-boosting OPAMP Design

The operational amplifier is a fully differential folded cascode with boosting amplifiers as shown in Figure 5.6.

5.2.3.4.1 Principle of Gain-boosting

The gain-boost technique [36] is based on increasing the cascoding effect of M2 by adding an additional gain stage as shown in Figure 5.7. This stage reduces the feedback from the output to the drain of the input transistor. The addition of the amplifier ideally increases the output impedance by a

factor equal to one plus the loop gain (A_{add}) over that which would occur for a classical cascode current mirror.

$$R_{out} = (g_{m2}r_{ds2}(A_{add} + 1) + 1)r_{ds1} \quad (5.10)$$

Therefore, the DC gain can be increased by several orders of magnitude:

$$A_{o,tot} = g_{m1}r_{ds1}(g_{m2}r_{ds2}(A_{add} + 1) + 1) \quad (5.11)$$

Figure 5.8 shows the original gain of the folded cascode amplifier without gain boosting (A_{orig}), the gain of the additional gain stage (A_{add}), and the improved cascoded gain stage with gain boosting (A_{tot}). Where the DC value of A_{tot} would be approximately equal to $A_{orig} * (1 + A_{add})$. However, for $\omega > \omega_1$ (the 3dB frequency of the final amplifier), the output impedance is dominated by C_L which results in a first order roll-off of $A_{tot}(\omega)$. Moreover, this implies that ω_2 (3dB frequency of the boosting amplifier) should be greater than ω_1 so as not to limit the speed of the final opamp. This is equivalent to the condition that the unity gain frequency ω_4 of the boosting gain stage has to be large than the 3dB bandwidth ω_3 of the original stage, but can be much lower than the unity gain frequency ω_5 of the original stage. The additional boosting stage introduces both poles and zeroes and hence they get cancelled. However, the location of the poles and zeroes can affect the transient response a lot. It will introduce a slow settling component in the transient response if the pole and the zero are not close enough. Hence, the farther they are (can be seen from the dip in the phase response), the slower is the settling in the transient response. The improved gain stage has the same unity-gain frequency as the original stage. The additional stage needs not to be fast with respect to the unity-gain frequency of the original design.

A safe range of locating the unity gain frequency ω_4 of the boosting stage is given by the following condition

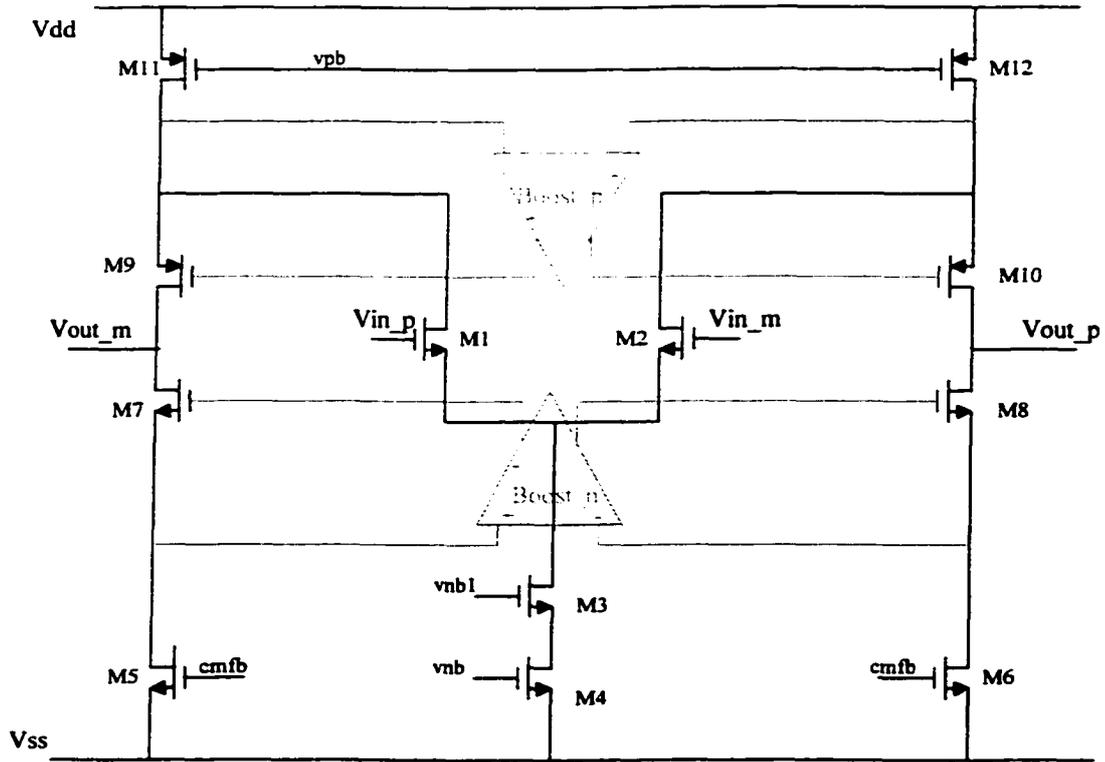


Figure 5.6 Main operational opamp with boosting amplifier

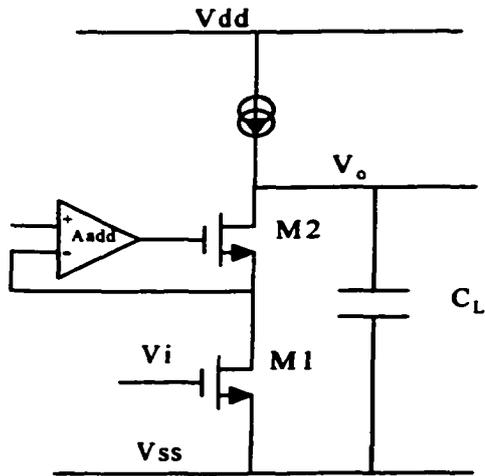


Figure 5.7 Cascode gain stage with gain enhancement

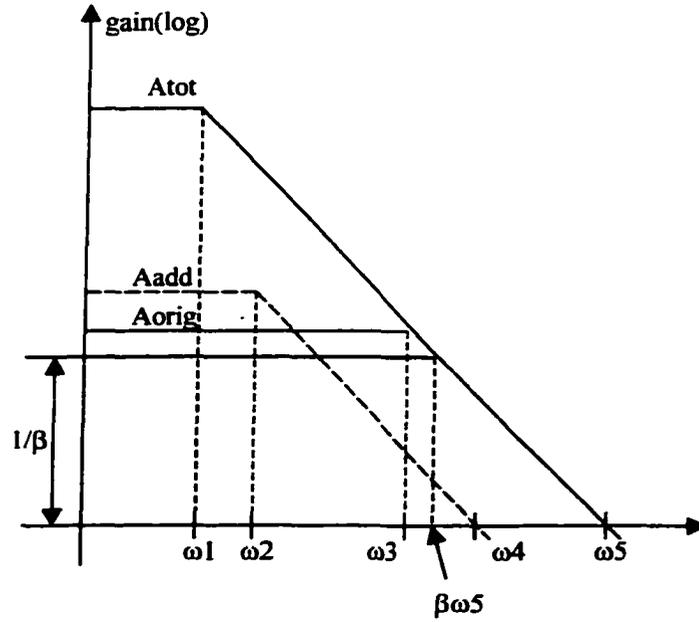


Figure 5.8 Bode plots of the original, additional and improved gain stage

$$\beta\omega_5 < \omega_4 < \omega_6 \quad (5.12)$$

where ω_6 is the second pole location of the final amplifier and β is feedback factor.

5.2.3.4.2 Fully Differential Folded Cascode OPAMP Design

The main opamp (which is the amplifier that uses the boosting amplifiers) is a fully differential folded cascode opamp, as shown in Figure 5.6. The DC gain is

$$A_v = \frac{g_{m1}}{g_{ds6} \frac{g_{ds8}}{g_{m8}} + (g_{ds2} + g_{ds12}) \frac{g_{ds10}}{g_{m10}}} \quad (5.13)$$

Transistors M5 and M6 are also used to control common mode voltage. This will be discussed in the next section. The specifications of the main opamp (without boosting) is shown in Table 5.1. These

are based on typical simulation results. However, all process corners and temperatures were simulated to verify operation under these conditions.

5.2.3.4.3 Common-mode Feedback (CMFB) Circuit of Main OPAMP

A common mode feedback circuit is necessary for fully differential opamp to establish the common mode output voltage. Two typical types of CMFB circuits are continuous time and switched-capacitor types. Since the opamp is used in switched capacitor circuit, the CMFB circuit is generally preferred to be of switched-capacitor type, since they allow a larger output swing.

Figure 5.9 shows the CMFB circuit [32], where “cmfb” can also be found in Figure 5.6. The CMFB circuit consists of four capacitors and six switches. The four 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 will not have great effect on the capacitors. Capacitors C_C generate the average of the output voltages, which is used to create control voltages for the opamp current sources. The DC voltage across C_C is determined by capacitors C_S , which are switched between bias voltages and between being in parallel with C_C . V_{cm} is the required common mode voltage of the opamp and it is set to be

Table 5.1 The specification of main opamp with 2pF load plus switch

DC Gain	660
Unity-Gain Bandwidth	730MHz
Phase Margin	64°
Input Range	0.75V ~ 1.75V
Power Dissipation	24.6mW

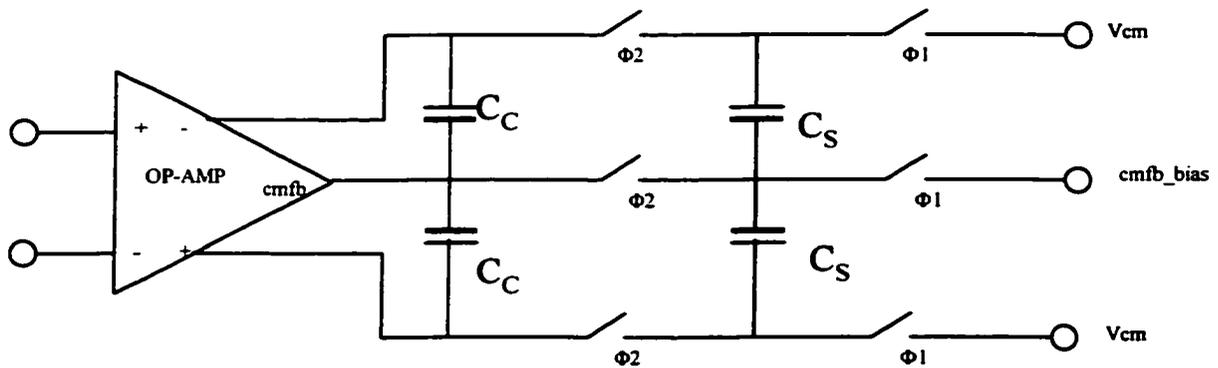


Figure 5.9 Common mode feedback circuit

1.25V in this design. At nominal condition and without the CMFB being connected to the opamp, transistors M5, M6 in Figure 5.6 are biased with “cmfb_bias”, and the output voltage of the opamp is around V_{cm} .

5.2.3.4.4 Boosting Amplifier Design

The boosting amplifiers are shown in Figure 5.10 and Figure 5.11. The boosting amplifiers are all fully differential folded-cascode amplifiers, with continuous time common-mode feedback circuits. Boosting-n has PMOS differential input stage, while boosting-p has NMOS differential input stage.

The specifications of the boosting amplifiers are shown in Table 5.2 and 5.3. These are based on typical simulation results. However, all process corners and temperatures were simulated to verify operation under these conditions.

5.2.3.4.5 CMFB for Boosting Amplifier

A continuous-time CMFB circuit [32] is used for both boosting amplifiers, as shown in Figure 5.10 and Figure 5.11. Let us take CMFB in Figure 5.11 for example to explain its operation. In this

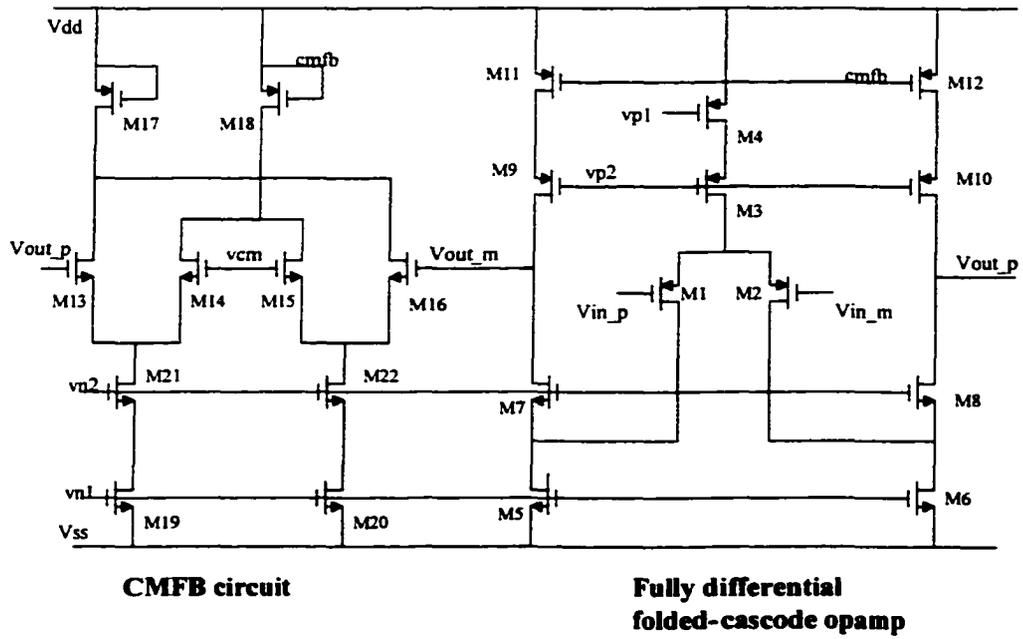


Figure 5.10 Boosting-n amplifier with CMFB circuit

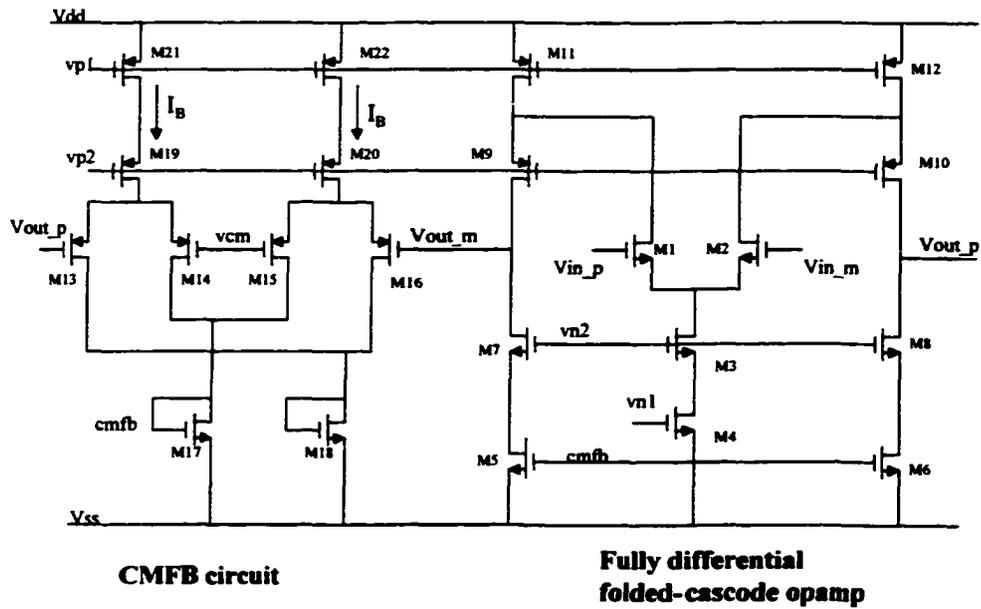


Figure 5.11 Boosting-p amplifier with CMFB circuit

Table 5.2 The specification of boost_p with 1pF load

DC Gain	480
Unity-Gain Bandwidth	182MHz
Phase Margin	74°
Input Range	0.75V ~ 1.75V
Power Dissipation	9.6mW

design, common-mode voltage v_{cm} is 1.25V. Assume V_{out_p} and V_{out_m} are equal in magnitude, but opposite in sign, compare to v_{cm} , and assume the two differential pairs M13 ~ M16 have infinite common-mode input rejection, which implies that the large-signal output currents of the differential pairs depend only on their input differential voltages. Since the two pairs have the same differential voltages being applied, the current in M13 will be equal to the current in M15, while the current in M14 will be equal to the current in M16. Now, letting the current in M14 be denoted as $I_{D14} = I_B / 2 + \Delta I$, where I_B is the bias current of the differential pair and ΔI is the large-signal current change in I_{D14} . The current in M15 is given by $I_{D15} = I_B / 2 - \Delta I$, and current in M17 is given by

Table 5.3 The specification of boost_n with 1pF load

DC Gain	320
Unity-Gain Bandwidth	520MHz
Phase Margin	70°
Input Range	0.75V ~ 1.75V
Power Dissipation	9.6mW

$$I_{D17} = I_{M14} + I_{M15} = (I_B / 2 + \Delta I) + (I_B / 2 - \Delta I) = I_B \quad (5.14)$$

Thus, as long as the voltage V_{out_p} is equal to the negative value of V_{out_m} compare with v_{cm} , the current through diode-connected M17 will not change even when large differential signal voltages are present. Since the voltage of M17 is used to control the bias voltage of the output stage of the opamps, this means that when no common-mode voltage is present, the bias currents in the output stage will be the same regardless of whether a signal is present or not.

Next consider what happens when a common-mode voltage other than v_{cm} is present. For example, assume the common-mode signal is higher than 1.25V. This will cause the current in both M14 and M15 to increase, which causes the current in M17 to increase, which in turn causes its voltage to increase. This voltage is the bias voltage that sets the current levels in the n-channel current sources M5 and M6 at the output of the opamp. Thus, both current sources will have large currents pulling down to the negative rail, which will cause the common-mode voltage to decrease, bringing the common-mode voltage back to 1.25V. Thus, as long as the common-mode loop gain is large enough, and the differential signals are not so large as to cause transistors in the differential pairs to turn off, the common-mode output voltage will be kept close to v_{cm} .

The current sources I_B in CMFB are the high-output impedance cascode current sources to ensure

Table 5.4 The specification of gain-booster opamp with 2pF load

DC Gain	105dB
Unity-Gain Bandwidth	630MHz
Phase Margin	62°
Input Range	0.75V ~ 1.75V
Power Dissipation	44mW

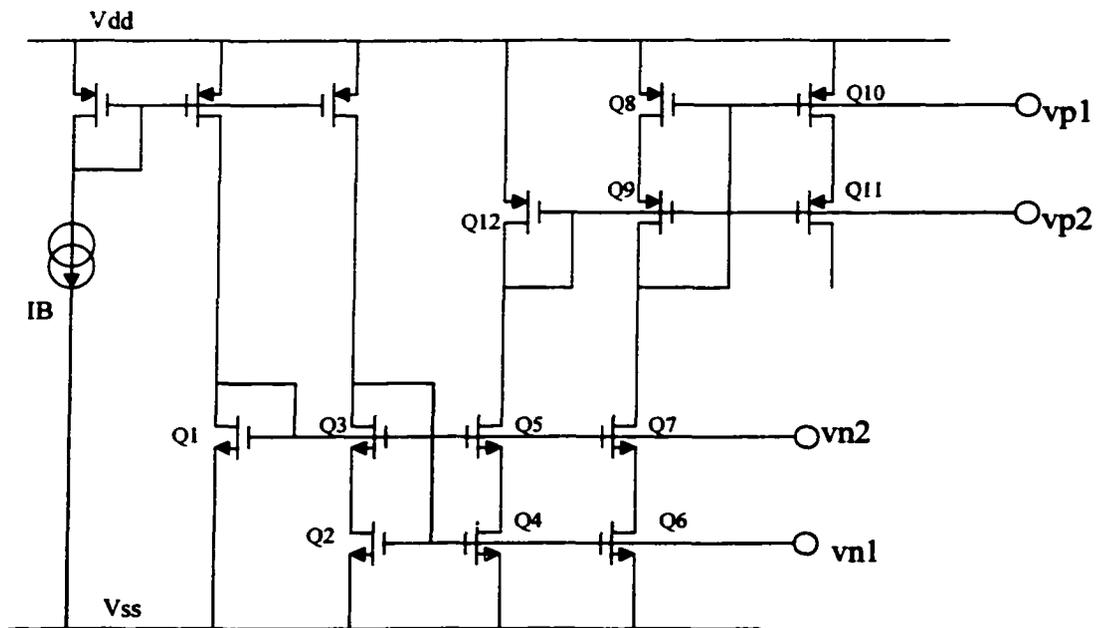


Figure 5.12 Opamp bias circuit

good common-mode rejection of the two differential pairs.

Table 5.4 shows the specifications of the gain-boosted opamp at 2pF capacitor load. These are based on typical simulation results. However, all process corners and temperatures were simulated to verify operation under these conditions.

5.2.3.4.6 OPAMP Bias Circuit and Clock Generation Circuit

Figure 5.12 shows the bias circuit used for opamps. The wide-swing cascode current mirror [32] is used as biasing circuit. The basic idea of this current mirror is to bias the drain-source voltages of transistors Q2, Q4, and Q8, Q10 to be close to the minimum possible without them going into the triode region.

The n-channel wide-swing cascode current mirror consists of transistors Q2 ~ Q5, along with the diode-connected biasing transistor Q1. The pair Q2, Q3 acts like a diode-connected transistor at the

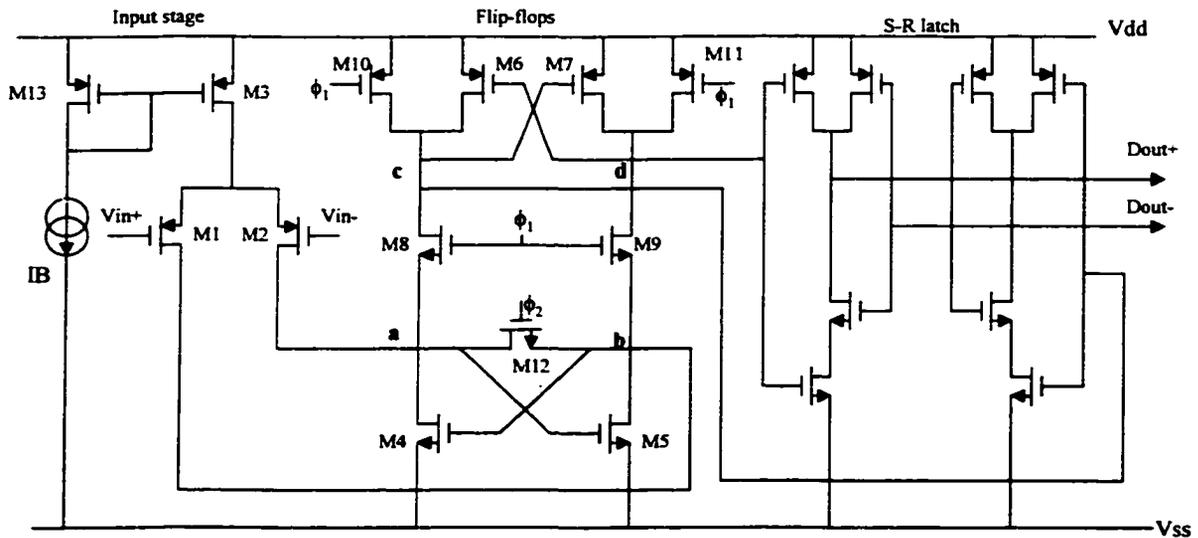


Figure 5.14 Schematic of the CMOS comparator

together at the rising edge. This is done by bypassing the delay chains with transistors indicated with arrows to reset the internal node and in turn to line up the rising edges of the clocks. Falling edges are then created by the delay inverter chain action. The clock ϕ_{1-} , ϕ_{1+} in Fig.5.3 can be generated in the similar way, and the same for clock ϕ_{2-} , ϕ_{2+} .

5.2.4 Latched Comparator Implementation

The comparator starts to compare the output of the opamp after the output has settled to its final value, mandating the requirement of a very high speed comparator. A high-speed CMOS comparator [25] with low-power, small area was implemented. The comparator circuit is depicted in Fig.5.14. It consists of a differential input pair (M1, M2), which acts as a preamplifier, a CMOS latch circuit, and an S-R latch. The CMOS latch is composed of a n-channel flip-flop (M4, M5) with a pair of n-channel transfer gates (M8, M9) for strobing and an n-channel switch (M12) for resetting, and a p-

channel flip-flop (M6, M7) with a pair of p- channel precharge transistors (M10, M11). $\phi 1$ and $\phi 2$ are the two nonoverlapping clocks to reduce comparator offset [37].

The dynamic operation of this circuit is divided into a reset time interval and a regeneration time interval. The two intervals are approximately between t_1 and t_2 and between t_2 and t_4 , respectively, as shown in Figure 5.15.

During $\phi 2$, the comparator is in the reset mode. M12 is closed, and the current flows through it. Thus forces the previous two logic state voltages to be equalized. After the input stage settles on its decision, a voltage difference proportional to the input voltage difference is established between nodes a and b in the end. This voltage will act as the initial imbalance for the following regeneration time interval. At the same time, as the n-channel flip-flop is reset by M12, the p-channel flip-flop is also reset by the two closed precharge transistors M10 and M11, which charge nodes c and d to the positive power supply voltage. This way, the CMOS latch is set to the astable high-gain mode.

The regeneration is initialized by the opening of switch M12. When $\phi 1$ is low, the strobing transistors M8 and M9 isolate the n-channel flip-flop from the p-channel flip-flop. By using two nonoverlapping clocks, the regenerative process can be done in two steps. The first step of regeneration is within the short time slot between $\phi 2$ getting low and $\phi 1$ getting high, which is

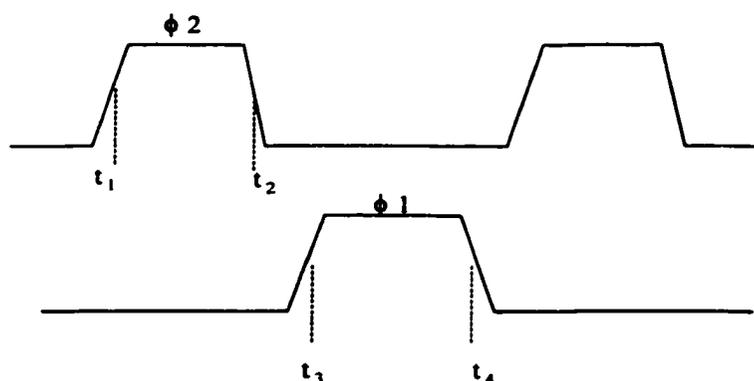


Figure 5.15 Timing diagram

$t_2 \sim t_3$ as shown in Figure 5.15. The second regeneration step starts when ϕ_1 gets high and M8 and M9 are closed, which is $t_3 \sim t_4$. The n-channel flip-flop, together with the p-channel flip-flop, regenerates the voltage differences between nodes a and b and between nodes c and d. The voltage difference between node c and node d is soon amplified to a voltage swing nearly equal to the power supply voltages. The following S-R latch is driven to full complementary digital output levels at the end of the regenerative mode and remains in the previous state in the reset mode. There is no slew-rate problem in the regeneration period because the p-channel flip-flop is used instead of two class-A current sources.

The first regeneration step is very important, not only in raising the regeneration speed but in reducing the total input offset voltage. The differential errors caused by the charge injection from M8 and M9, the mismatches in the p-channel flip-flop, the two precharge transistors, and the S-R latch are divided by the amplification gain in the first regeneration step, when referred to the input as an equivalent offset voltage. Therefore, their contribution to the total equivalent input offset voltage can be neglected if the gain is large enough.

The transient analysis of the comparator is shown in Figure 5.16. The minimum resolvable voltage is 0.2mV, which means 12bit resolution for 1V input range, and clock is 50MHz. The temperature range is 0 ~ 100 °C, with all the typical and corner models.

The specification of comparator is shown in Table 5.5.

Table 5.5 Specification of the latched comparator

Minimum Resolvable Voltage	0.2mV
Power Dissipation	1mW

BIST_and Comp_v1_test schematic : May 14 16:33:49 2000

Transient Response

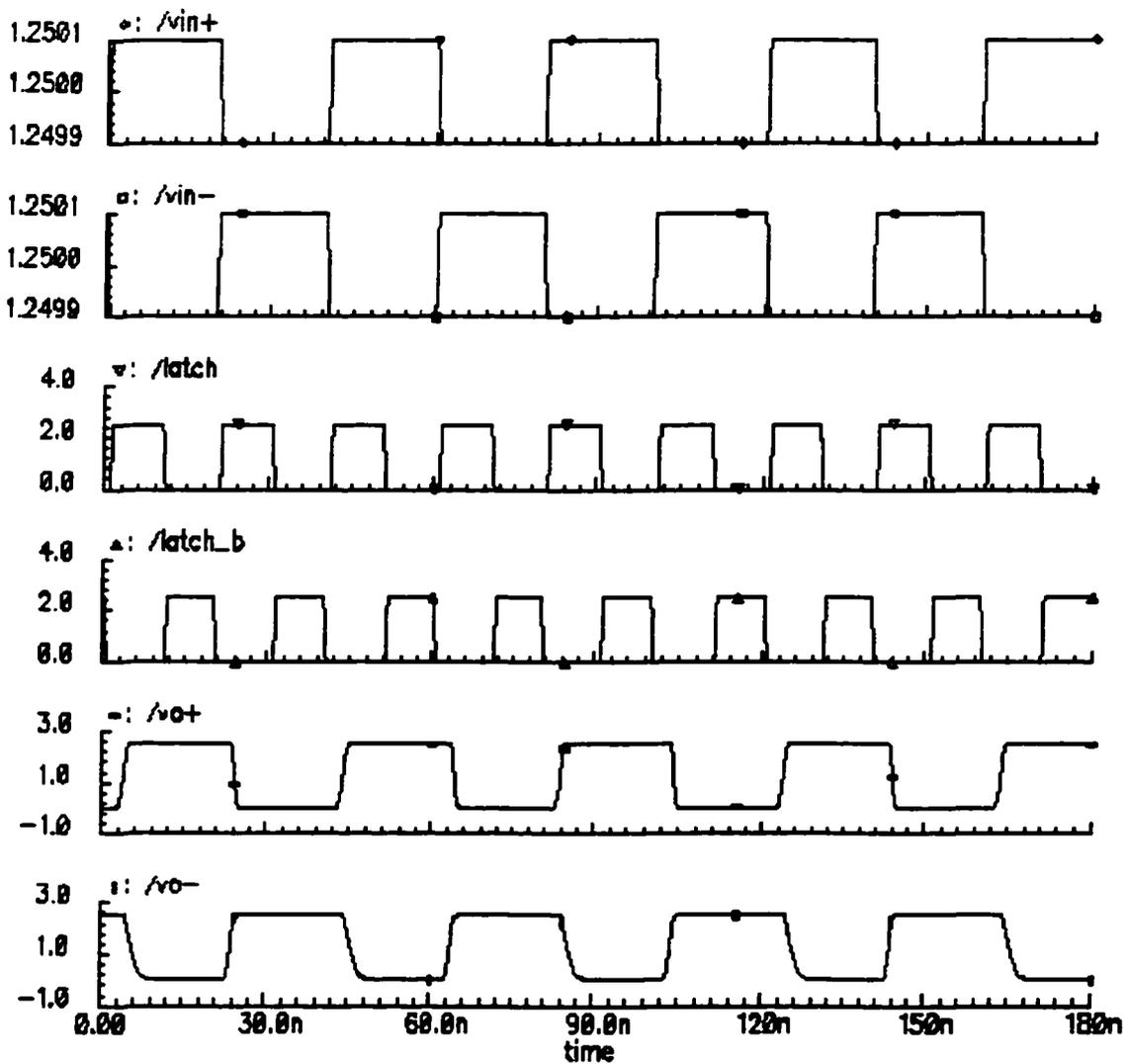


Figure 5.16 The transient analysis of comparator

5.3 Re-configurable Pipeline Structure

5.3.1 Error Sources in 1-bit-per-stage Pipeline ADC

Before we start the re-configurable pipeline structure, some error sources affecting typical implementations of pipelined ADC need to be discussed. The errors may be divided into two categories [31]: noise, which varies from sample to sample, and mismatches, which do not vary from sample to sample. Mismatch related errors could be corrected by calibration, while noise related errors cannot be easily corrected by calibration. In general, the major sources of error in pipelined, switched-capacitor ADCs are comparator offset, charge injection from the sampling switches, finite opamp gain, capacitor mismatch and noise. The effects of these errors except noise can be summed up to be one of Gain error, Comparator offset or DAC error.

5.3.1.1 Gain Error

The first part of the transfer characteristic of the single stage of the ADC (Eq.5.3) is $2V_{in}$. The limitation in achieving this is the finite opamp gain and the capacitor mismatch. Ideally, the open loop gain of the opamp is infinite to get the accurate value in switched-capacitor circuits. The effect of opamp's finite open loop gain can be understood by equation (5.5) and Figure 5.5. Here we show equation (5.5) again for convenience.

$$\frac{V_{out}}{V_{in}} = \frac{A_0}{1 + A_0\beta} \quad (5.15)$$

where A_0 is the open loop gain of the opamp and β is the feedback factor. The close loop gain is usually approximated to $1/\beta$, assuming an infinite open loop gain. Equation (5.15) can be modified as

$$\frac{V_{out}}{V_{in}} = \left[\frac{1}{\beta} \right] \left[\frac{A_0}{\frac{1}{\beta} + A_0} \right] \quad (5.16)$$

We can define K as the gain error coefficient that results from finite opamp gain and given by

$$K = \frac{A_0}{\frac{1}{\beta} + A_0} \quad (5.17)$$

Due to the gain error coefficient K, the final gain can be either greater than or less than 2, which is the desired value for a 1-bit-per-stage pipeline ADC. Thus, missing codes or missing decision levels can happen.

5.3.1.2 Comparator Offset

Comparator offset is another important source of error in pipeline data converter, since it may result in missing code or missing decision level. Practically it is produced by the mismatch between the two transistors in the differential pair that constitutes the input stage of the comparator. Offset cancellation techniques using multi-stage and switched capacitor configurations can be used [32].

5.3.1.3 Charge Injection and Amplifier Offset

The charge injection error and the opamp offset can be dealt with similarly, since the effect of these errors on the transfer characteristic would be the same. The opamp offset can be modeled as an output offset voltage, V_{off} and can be introduced in the transfer characteristic as

$$V_{out} = 2V_{in} \pm V_{ref} + V_{off} \quad (5.18)$$

This means the transfer characteristic will shift by the amount of the offset V_{off} [33].

Charge injection from the switches can be made input voltage-independent through proper sequencing the switching and having a differential switched capacitor configuration. Thus, voltage

independent charge injection can be modeled as an offset at the output, similar to the amplifier output offset voltage. Both the amplifier offset and charge injection can cause missing decision levels or missing codes.

5.3.2 Variation between Pipeline Stages

As we mentioned, in the 9bit 1-bit-per-stage pipeline structure, there are 9 stages, each stage is identical, except the first sample and hold stage. The errors in the pipeline structure, which were discussed in the previous section, will show in every stage. Ideally, those errors will be identical since every stage is identical. But in practical, due to process variations, gradients on the wafer, the error effects will be different in each stage. This way, if we measure the performance of each stage alone, we will get different results: stage i achieves the performance of 9-bit accuracy, but stage j may only have only performance of 8-bit accuracy, for example. If the design has enough margins to guarantee every stage has a performance of at least 9-bit accuracy, then the ADC has, no doubt, 9-bit accuracy. The design goal is met. But what if the uncontrollable reason happens that makes some stage's performance cannot achieve the 9-bit performance? If this "bad" performance stage happens to be the most significant bit (MSB), the result is that the whole ADC will not be able to achieve 9-bit accuracy. The design goal will not be met.

Fortunately, this case can be avoided by using re-configurable structure, which will be discussed in the next section.

5.3.3 Re-configurable Pipeline ADC

The most promising topology for high-resolution high-speed ADCs in CMOS is the pipeline architecture [26][27][28][29]. In pipeline ADC architectures, the accuracy requirement for each stage is different [30]. For example, for a 9-bit ADC, the first stage (MSB) requires 9-bit accuracy; the 2nd

stage requires 8-bit accuracy, and so on; the last stage (LSB) only requires 1-bit of accuracy. In the design, if all the stages are identical replicas of each other, ideally, the stages will have identical performance. But in silicon, the stages will vary in their performance due to the gradients on the wafer, process variance, etc. If the performance of each stage can be measured, the ideal ADC configuration would be to make the most accurate stage the MSB stage, the 2nd most accurate stage the 2nd stage, and so on, until the LSB stage. This way the pipeline ADC will achieve the best possible performance. Since the performance of a single stage cannot be measured alone, a re-configurable pipeline ADC architecture is proposed. By measuring the performance of different configurations, the one with the best performance will be chosen as the final configuration. The architecture lends itself very well to on-chip testing in addition to production testing application when the yield for a particular ADC product can be increased by the re-configuration process.

In order to make the whole pipeline ADC the re-configurable, some control logic is needed. There are several structures that can fulfill re-configuration requirements.

5.3.3.1 4-Switch Re-configuration

In this section, for simplicity, we will use a 4-bit 1-bit-per-stage ADC as an example. Figure 5.17 shows the 4-switch network of re-configurable pipeline ADC. Figure 5.18 shows the switch configuration. S0, S1 in Figure 5.18 are control signals. A 4-stage pipeline ADC will need 8 control signals. These control signals will be generated off-chip, and shift in by shift registers. They are just used for controlling the configuration of the pipeline stages. The final values of these control signals are based on the performance of these 4 stages.

From Figure 5.17, we can see each cell can have 4 possible inputs, depending on the control logic. A 4-stage ADC can have as many as $4!=24$ configurations and therefore 24 configurations of the pipeline ADC need to be measured. From the measurement results, the best configuration is found.

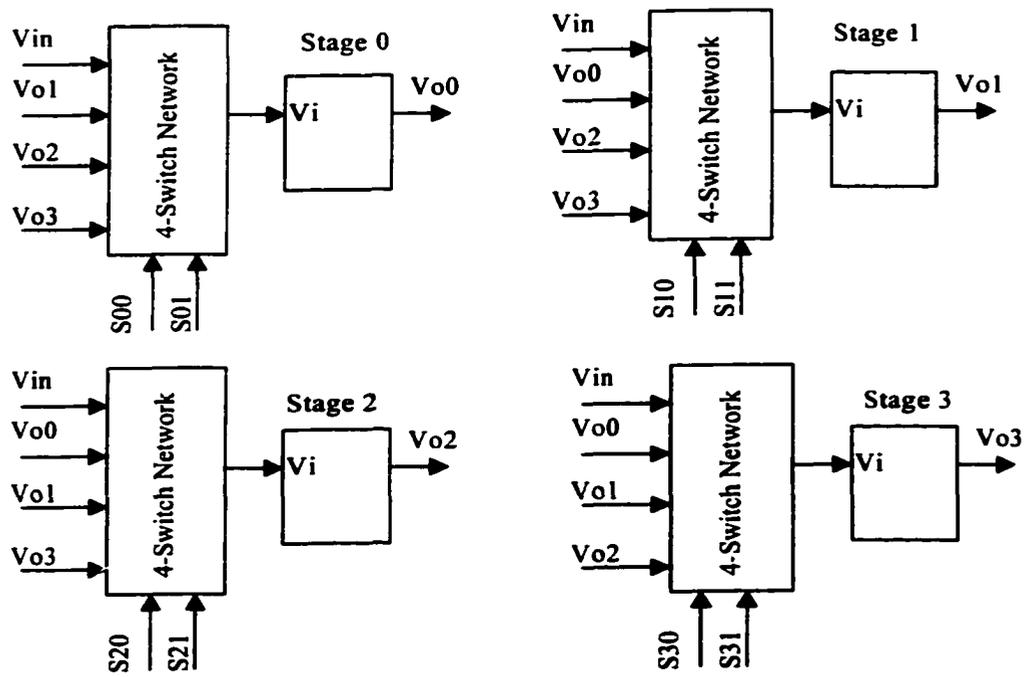


Figure 5.17 4-tages of pipeline with re-configuration ability using 4-switch network

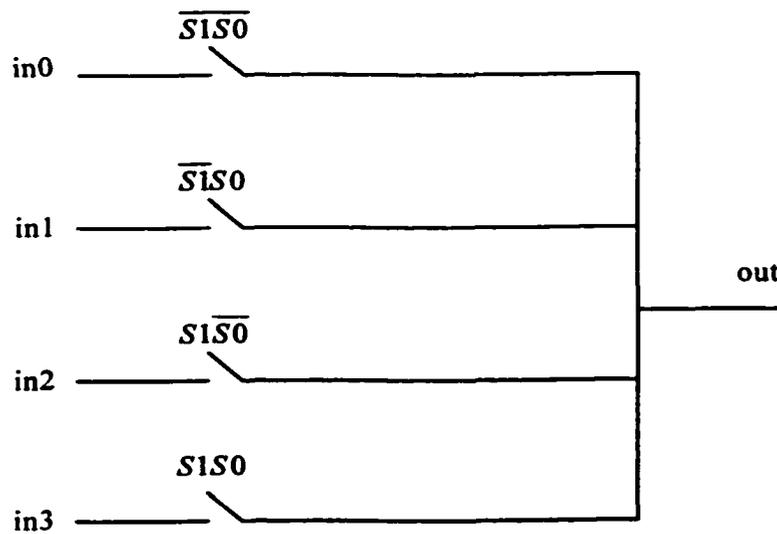


Figure 5.18 4-Switch network

This is a lot of measurement work needs to be done in order to find the best performing ADC configuration.

5.3.3.2 8-Switch Re-configuration

If the pipeline ADC is an 8-bit with 1-bit-per-stage, an 8-switch configuration can be used. It will be similar to the 4-switch configuration, but each stage will have 8 possible inputs, and needs 3 control signals for each stage. In total 8 stages will need 24 control signals. An 8-stage ADC will have as many as $8!=40320$ different configurations. This is a lot of measurements needs to be done in order to find the best performing ADC configuration. In the most cases, it is not a practical idea. An alternative method has to be used, which is discussed next.

5.3.3.3 Simplified Re-configuration Algorithm

5.3.3.3.1 Grouping Algorithm for 9-bit Pipeline ADC

The example shown here is a 9-bit 1-bit-per-stage pipeline ADC. The grouping method is shown in Figure 5.19. The measurement procedure is as follows:

Step 1: Randomly partition the whole ADC (stages 1 through 9) into 3 partitions, partitions I, J and K. Each partition includes 3 stages. Actually this partitioning is not totally random, spatially (layout-wise) close neighbors will be grouped together considering that the process difference or wafer gradients will have smaller difference between two neighboring stages than two stages which are layout far apart.

Step 2: Re-configure partition I, as shown in Figure 5.20, where partition I is the MSB partition. Note that the sequence of J and K is not important. Find the best configuration of part I by re-configuring each of the stages in partition I to be the first cell and taking a dynamic measurement of

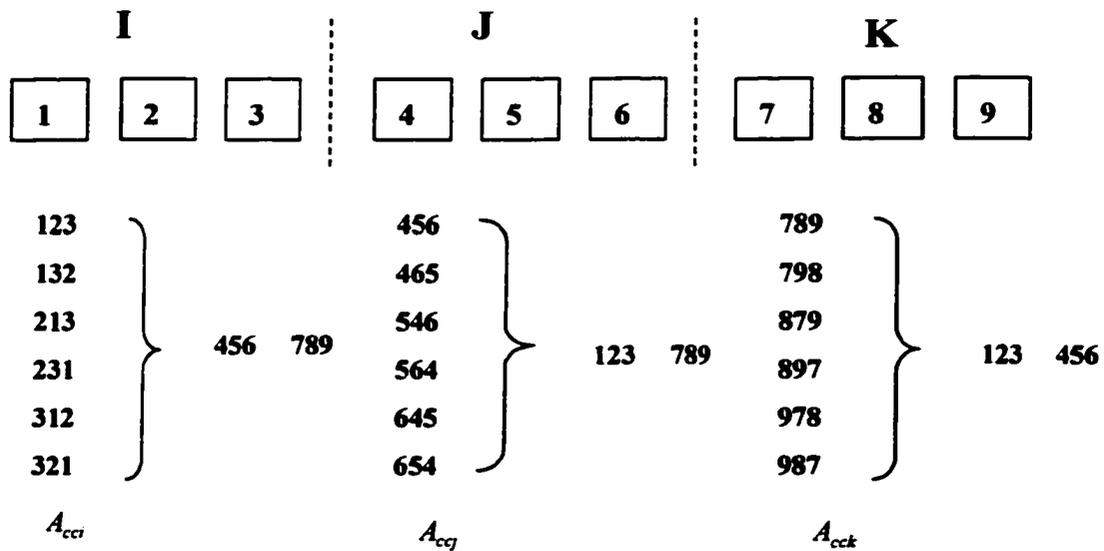


Figure 5.19 The grouping method of a 9-stage ADC

the performance. In the process record the best accuracy that partition I can achieve: A_{cci} . Six measurements need to be done for this part.

Step 3: Repeat step 2 for partition J while making partition J the MSB partition, as shown in Figure 5.21. The sequence of I, K is not important on the assumption that each stage in the structure will have at least 6bits accuracy, which is reasonable for a 9bit design.. Find the best configuration within J and record the best accuracy that partition J can achieve: A_{ccj} . Six measurements needed to be done for this part.

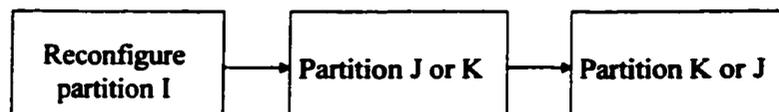


Figure 5.20 The first six measurements

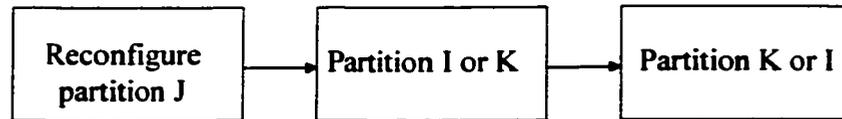


Figure 5.21 The second six measurements

Step 4: Repeat step 2 for partition K while making partition K the MSB partition, as shown in Figure 5.22. The sequence of I, J is not important as step 3. Find the best configuration within K and record the best accuracy that partition K can achieve: A_{ck} . Six measurements are needed.

Step 5: Decide the ideal configuration of I, J and K. If $A_{ci} > A_{cj} > A_{ck}$, the configuration will be: $I \rightarrow J \rightarrow K$.

The conclusion is that a total of 18 dynamic performance measurements need to be done for a 9-bit re-configurable ADC. The above method is accurate on the assumption that each stage in the structure has at least an inherent 6-bit accuracy. For a 9-bit design, this is a reasonable assumption. A more relaxed version of the algorithm would take only 3 measurements under the assumption that each grouping of three has the same accuracy due to the spatial relationships.

5.3.3.3.2 Proof of Grouping Algorithm

With the assumption that each stage has at least 6-bit accuracy, the grouping method described above will have the following possible cases to be tested, shown in Table 5.6, where the accuracy of

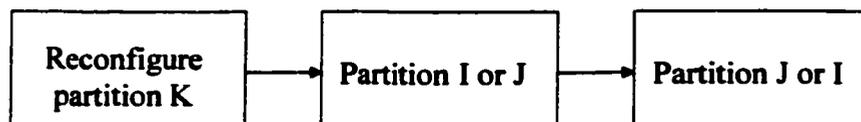


Figure 5.22 The last six measurements

Table 5.6 Proof of the grouping algorithm

Accuracy of group I (bit)			Accuracy of group J (bit)			Accuracy of group K(bit)			Accuracy of ADC(bit)
Stage1	Stage2	Stage3	Stage4	Stage5	Stage6	Stage7	Stage8	Stage9	
9	9	9	X	X	X	X	X	X	9
9	9	8	X	X	X	X	X	X	9
9	9	7	X	X	X	X	X	X	9
9	9	6	X	X	X	X	X	X	8*
9	8	8	X	X	X	X	X	X	9
9	8	7	X	X	X	X	X	X	9
9	8	6	X	X	X	X	X	X	8*
9	7	7	X	X	X	X	X	X	8*
9	7	6	X	X	X	X	X	X	8*
9	6	6	X	X	X	X	X	X	7*
8	8	8	X	X	X	X	X	X	8
8	8	7	X	X	X	X	X	X	8
8	8	6	X	X	X	X	X	X	8
8	7	7	X	X	X	X	X	X	8
8	7	6	X	X	X	X	X	X	8
8	6	6	X	X	X	X	X	X	7*
7	7	7	X	X	X	X	X	X	7
7	7	6	X	X	X	X	X	X	7
7	6	6	X	X	X	X	X	X	7
6	6	6	X	X	X	X	X	X	6

each stage in group I is shown, and the accuracy of each stage in group J and K are shown as “X”, which means doesn’t matter. Group I is under the measurement.

The six cases with a symbol * are the cases needed to be paid attention. For example, let’s study the case with accuracy of each stage as $9 \rightarrow 9 \rightarrow 6 \rightarrow X \rightarrow X \rightarrow X \rightarrow X \rightarrow X \rightarrow X$. If each group is the same, such as $9 \rightarrow 9 \rightarrow 6 \rightarrow 9 \rightarrow 9 \rightarrow 6 \rightarrow 9 \rightarrow 9 \rightarrow 6$, then the testing will give only 8bit accuracy. So the total ADC will be 8bit. But by reorganize this 9 stages, 9bit accuracy can be achieved as $9 \rightarrow 9 \rightarrow 9 \rightarrow 9 \rightarrow 9 \rightarrow 9 \rightarrow 6 \rightarrow 6 \rightarrow 6$. The reason that this problem arises is the initial partition is inaccurate. Here we need to mention the initial partition is not totally random. The whole ADC is partitioned into three groups according to their spatial position. As we know, the goal of IC design is to make each stage identical, but due to process variations, wafer gradients, the errors in each stage will differ. However, these effects are not random. Spatially (layout-wise) close neighbors will have the close performance than two stages that are layout far away, because the matching is better between two neighbor stages than two far away stages. So are the process difference and wafer gradients. The initial partition is pseudo-random. Close neighbors are partitioned into the same group as described in the grouping algorithm.

With the assumption that the proper initial partition is done, the grouping algorithm described above will find the best configuration for the pipeline ADC under test.

Grouping algorithms for 8-bit or 10-bit, etc. ADCs can be developed in a similar way. The algorithm and groupings are functions of the inherent accuracy of the process. The grouping and inherent accuracy together determine the optimal number of measurements to obtain optimal performance.

5.3.4 Design Consideration of Re-configurable ADC Cell [38]

In this project, a 9-bit 40MS/s re-configurable pipeline ADC is implemented with the grouping algorithm discussed above. The only difference in this re-configurable ADC is that its front-end S/H stage will also take part in the re-configuration. Each cell in the re-configurable ADC will be exactly same, including the front-end S/H stage. The front-end S/H stage is the most important stage in the whole ADC, so the best performing cell needs to be chosen for this stage. Because of this, each stage is designed such that it can be configured as a gain-of-1 or a gain-of-2 stage in the 9-bit re-configurable ADC structure. In order to avoid the analog output of each stage exceeding the reference boundary error, which will cause missing decision levels and could not be eliminated by digital calibration alone [33], a gain-of-1.9 instead of gain-of-2 is used for stages 2 through 9. Thus, each stage will provide less than 1-bit resolution. An extra stage, stage 10 will be added following stage 9. A total of 10 stages will produce 9-bit resolution. But as mentioned earlier, stage 10 will not play in the re-configuration algorithm. Figure 5.23 shows the schematic of the gain-of-1/gain-of-2/gain-of-1.9 multiplying digital-to-analog converter (MDAC) [38]. The single-ended structure is shown for simplicity. During ϕ_1 , capacitors C_s and C_f both sample the analog input, and the small capacitor C_c is connected to ground or vcm (common-mode voltage for differential circuit) through M6. During ϕ_2 , when the gain is set to 1, to make this stage a S/H stage, C_s is connected to the output of the opamp through M5, C_f is also connected to the output of the opamp through M4, while M3 is open, and C_c is still connected to ground/vcm through M8. This way, during ϕ_1 , the charge stored on the capacitors is

$$Q_1 = V_{in}(C_s + C_f) \quad (5.19)$$

and during ϕ_2 ,

$$Q_2 = V_{out}(C_s + C_f) \quad (5.20)$$

where $C_s = C_f$. By equating $Q_1 = Q_2$, we have

$$V_{out} = V_{in} \quad (5.21)$$

When the gain is set to 2, during ϕ_2 C_s is connected to V_{ref} instead of the output of the opamp through M3, while M5 is open: C_f is connected to the output of opamp; C_c is connected to ground/vcm through M8. The charge stored on the capacitors will be

$$Q_2 = V_{out}C_f + V_{ref}C_s \quad (5.22)$$

By equating the two charges, $Q_1 = Q_2$, the transfer function becomes

$$V_{out} = 2V_{in} - V_{ref} \quad (5.23)$$

When the gain is set to 1.9, instead of 2, the capacitor C_c will have some function. For the S/H stage, the gain will be 0.975. During ϕ_2 , C_c will connect to the output of the opamp through M7. The charge stored on the capacitors will be

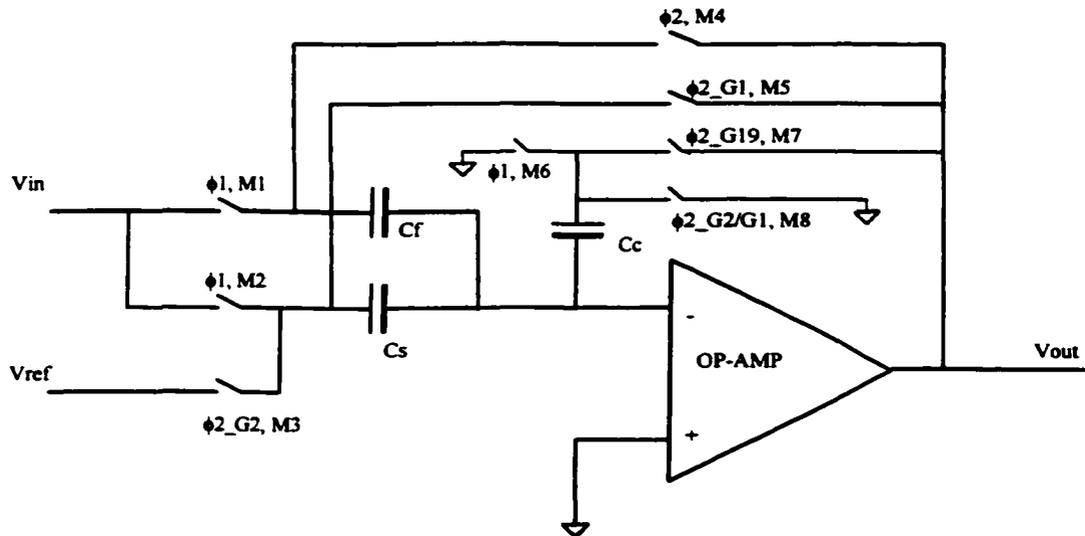


Figure 5.23 Gain-of-1/gain-of-2/gain-of-1.9 Multiplying Digital-to-Analog Converter

$$Q_2 = V_{out}(C_f + C_s + C_c) \quad (5.24)$$

Let $Q_1 = Q_2$, we get

$$V_{out} = \frac{C_s + C_f}{C_s + C_f + C_c} = \frac{2}{2.05} = 0.9756 \quad (5.25)$$

where $C_s = C_f$, and $C_c = 0.05C_s$.

For stage 2 through 10, the charge transfer will be as follows: during ϕ_2 , charge Q_2 will be

$$Q_2 = V_{out}(C_f + C_c) + V_{ref}C_s \quad (5.26)$$

By letting $Q_1 = Q_2$,

$$V_{out} = \frac{C_s + C_f}{C_f + C_c} V_{in} - \frac{C_s}{C_f + C_c} V_{ref} \quad (5.27)$$

By letting $C_s = C_f$, and $C_c = 0.05C_s$, equation (5.27) will be

$$V_{out} = \frac{2}{1.05} V_{in} - \frac{1}{1.05} V_{ref} = 0.95(2V_{in} - V_{ref}) \quad (5.28)$$

Thus the gain of 1.9 is achieved.

5.4 Comparison with other Pipeline ADC Calibration Techniques

In the past decade, there are several different kinds of calibration techniques proposed for pipeline ADCs. These techniques can be classified into two categories: analog calibration and digital calibration. Furthermore, within each classification, these calibration techniques can function while the ADC is in normal operation mode (background calibration) or requires special execution time at ADC startup or during normal conversion stoppages (foreground calibration). Table 5.7 shows the comparison of the re-configuration algorithm proposed in this thesis with some calibration techniques. Due to their similarities in terms of the need for measuring the performance of each stage

(or a subset thereof), the on-chip precision trimming method proposed in [39] is explicitly compared with the re-configuration algorithm proposed in this thesis. In [39], a differential polysilicon fuse amplifier is used as a pull-up or pull-down to a logic supply to trim the capacitor ratios and offset. In the re-configuration algorithm proposed herein, no on-chip trimming is used and no memory cell is needed. Each ADC stage cell is exactly the same as the normal pipeline ADC stage.

The other classes of algorithms are lumped together in the comparison in Table 5.7 due to their general similarities and methods of operations.

Similar to the on-chip precision trimming method [39], the re-configuration algorithm can improve the yield on the production level. The re-configuration algorithm can also be used together with the above mentioned calibration techniques, e.g. performing re-configuration first, then followed

Table 5.7 Comparison with the other calibration techniques

	Re-configuration algorithm	On-chip precision trimming [39]	Digital Self-calibration [33]	Background digital calibration [40]	Analog foreground calibration [41]	Analog background calibration [42]
Area	Low	Medium	Medium	High	High	High
Power	Low	Low	Medium	High	High	High-to-Medium
Number of Measurements	Medium	Low	High	Low	Medium	Medium
Cost	Medium-to-low	Medium	Medium	Medium	High-to-Medium	High-to-Medium
Performance Improvement	Medium	High	Medium	Medium	Medium	Medium

by any of the digital or analog calibration algorithm methods; or performing digital or analog calibration first, then followed by the re-configuration algorithm. The re-configuration algorithm proposed herein has very low hardware and time overhead and combining it with some of the other techniques can provide better results for the ADC's performance, and improve overall yield.

5.5 Layout Considerations

Figure 5.24 shows the layout of the chip, and the die photo is shown in Figure 5.25. It has 80 pins. On the bottom is the 10-stage re-configurable pipeline ADC. The layout of each stage is identical. Each stage is not physically connected with each other in the pipeline. Their connections will be controlled by the configuration codes. There are a total 40 control codes, and they all share one pin. There is a 40-bit shift register to shift in these codes. Above the 10-stage ADC is the clock tree that is

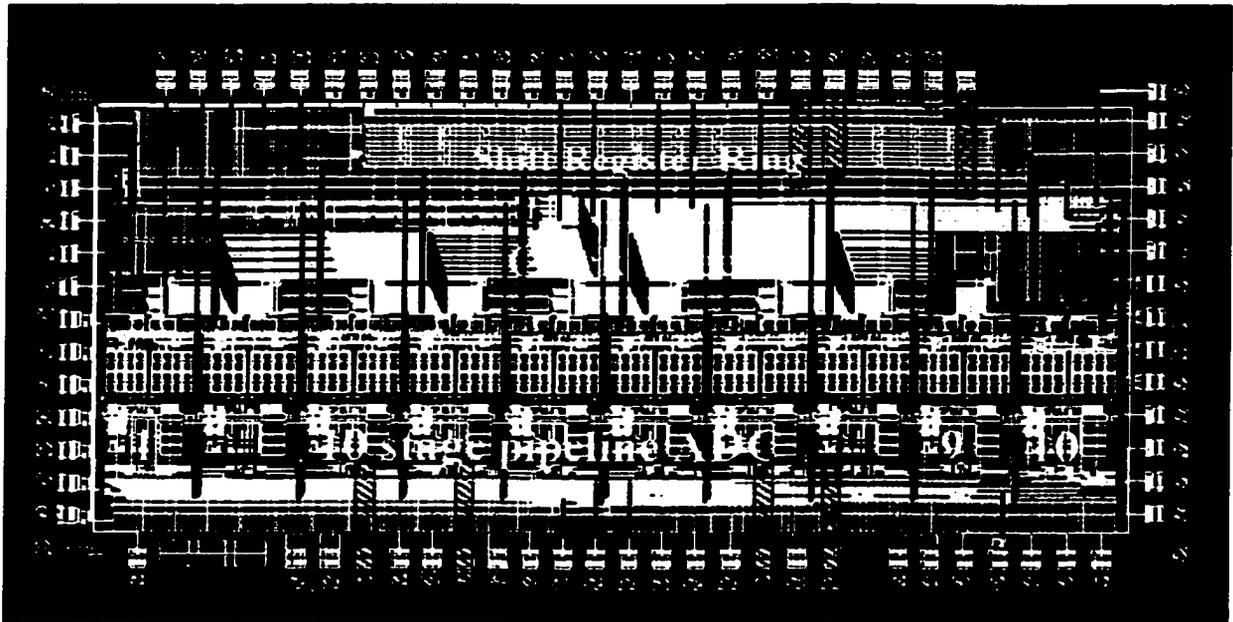


Figure 5.24 The layout of the chip

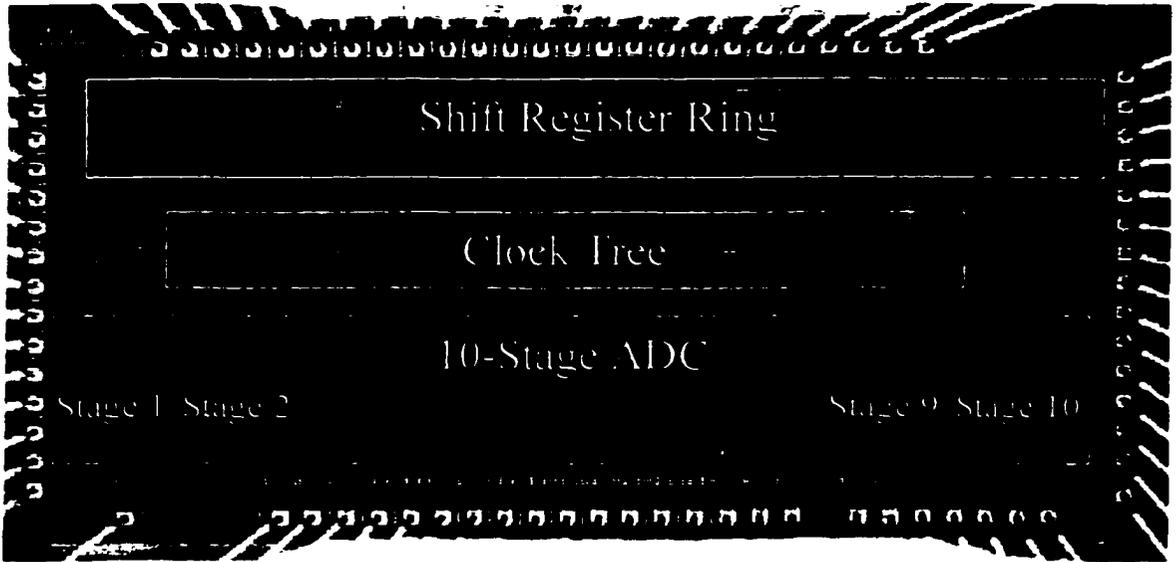


Figure 5.25 Die photo

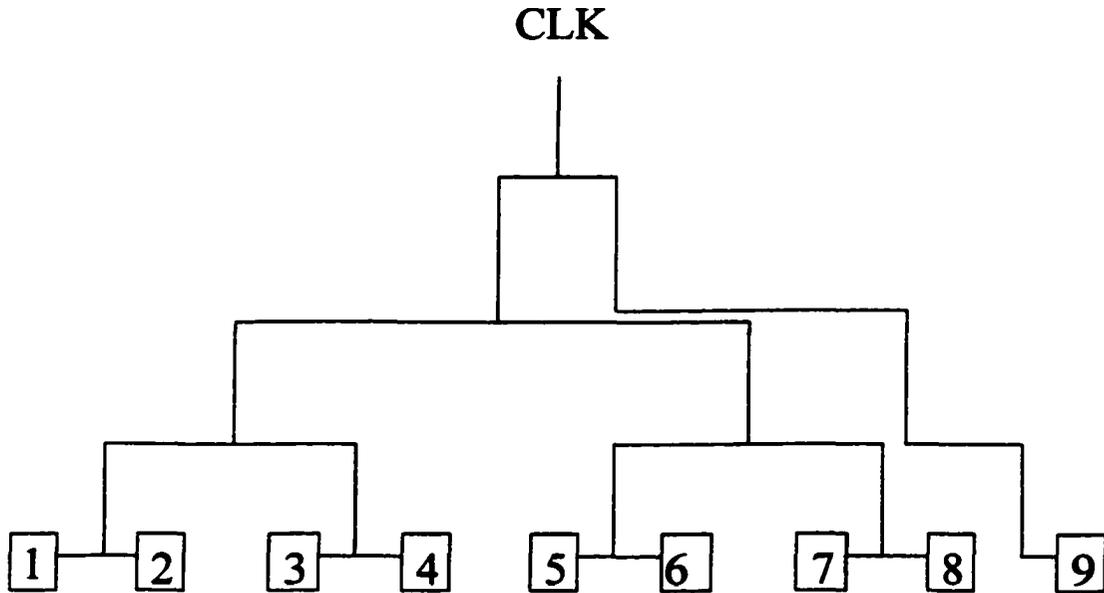


Figure 5.26 The floor plan of clock tree for stage 1 through stage 9

used to even the clock delay between each stage, instead of laying out clock lines from the last stage to the first stage. The clock tree for stage 1 through stage 9 is shown in Figure 5.26, where stage 10 is not in the clock tree, because it always stays as the last stage. The top part of the overall chip is a 1024 shift register, which will shift in the PDM bitstream generated by the software. The output of the bitstream will go through an off-chip low-pass filter, to generate the analog signal. Since this part will

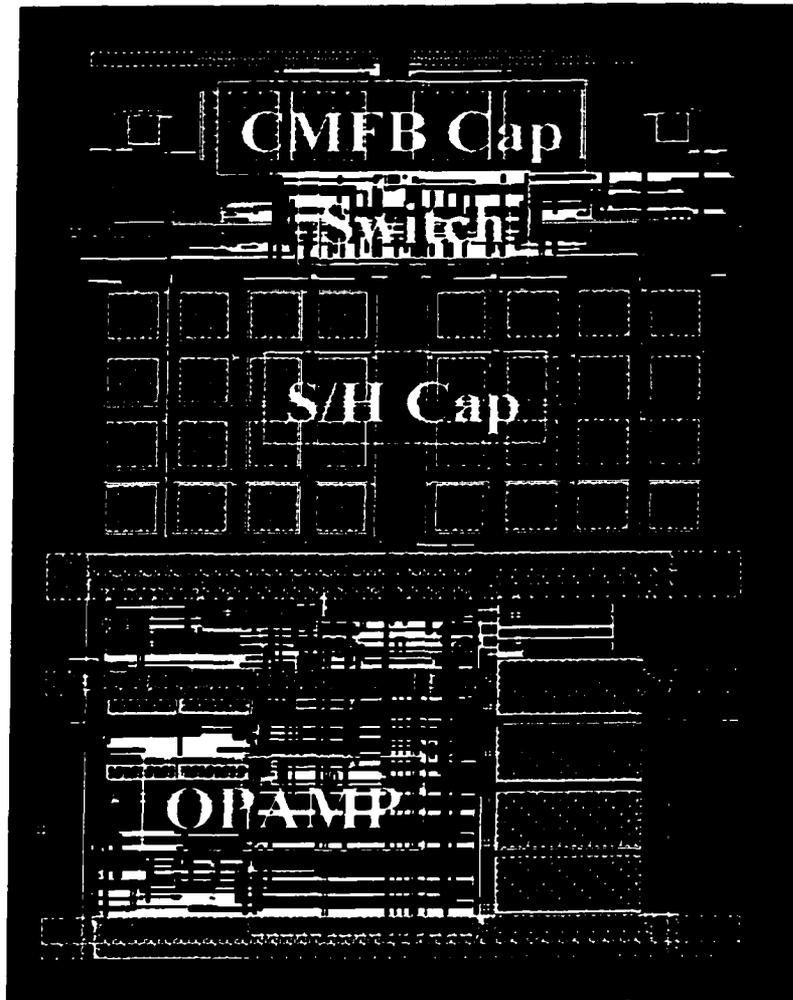


Figure 5.27 The layout of one stage of ADC

be noisy due to its high operating frequency, it uses separate power lines from the re-configurable ADC. The on-chip de-coupling capacitors are used to minimize the noise.

Figure 5.27 shows the layout of one stage alone of the ADC. The top is the digital part, mainly switches. The bottom is the analog part, mainly the OPAMP, including compensation capacitors. The S/H capacitors are between the digital and analog parts. They occupy large areas, since metal capacitors are used.

5.6 Chapter Conclusion

In this chapter, the basic pipeline ADC architecture was described. The design procedure of the ADC components: the opamp and the comparator were covered. A 9-bit 40MHz pipeline ADC was implemented in TSMC's 0.25 μ m digital CMOS process. Based on the basic pipeline ADC structure, a re-configurable pipeline ADC architecture was proposed and implemented in the design. A grouping method that can find the best performing configuration quickly was also proposed. By measuring the performance of different configurations, the one with the best performance will be chosen as the final configuration. The architecture lends itself very well to on-chip testing in addition to production testing application when the yield for a particular ADC product can be increased by the re-configuration process.

6 EXPERIMENTAL RESULTS

6.1 Introduction

In this chapter, the experimental results of the designed chip are presented. First, the printed circuit board design is presented, then the testing of the re-configurable pipeline ADC is shown, including dynamic tests and static tests. The experimental results strongly verify the advantage of re-configuration. At the end of this chapter, the on-chip analog signal generator is tested and the testing issues are addressed.

6.2 Experimental Results of Re-configurable Pipeline ADC

6.2.1 PCB Design of Testing Chip

A 4-layer printed circuit board (PCB) is designed using EGALE™ and fabricated to verify the design.

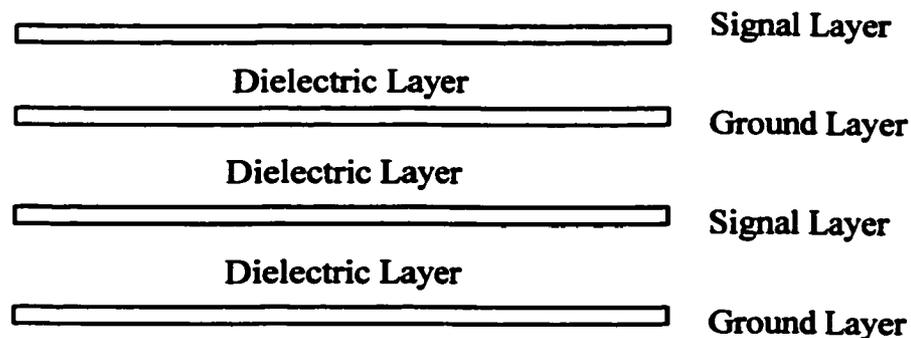


Figure 6.1 Cross-section of 4-layer PCB

Figure 6.1 shows the cross-section of a 4-layer PCB. The first and 3rd layers are signal layers, and the 2nd and 4th layers are ground layers. Most signals will use the first layer. Signals will use the 3rd layer only because of routing necessarily. Figure 6.2 shows the layout of the PCB. The upper-left corner is the digital signal part, and the rest is the analog signal part. The chip uses TQFP 80-pin package, and sits in the middle between the analog and the digital parts. Figure 6.3 shows the photo of the PCB. The size of the board is 8.5'' x 9.75''.

The pipeline ADC is a fully differential circuit, so on the board, there is a circuit to generate a differential signal from a single-ended signal. This single-to-differential circuit has two parts, one is for high-frequency (HF) testing, and the other is for low-frequency (LF) testing. For HF testing, e.g.

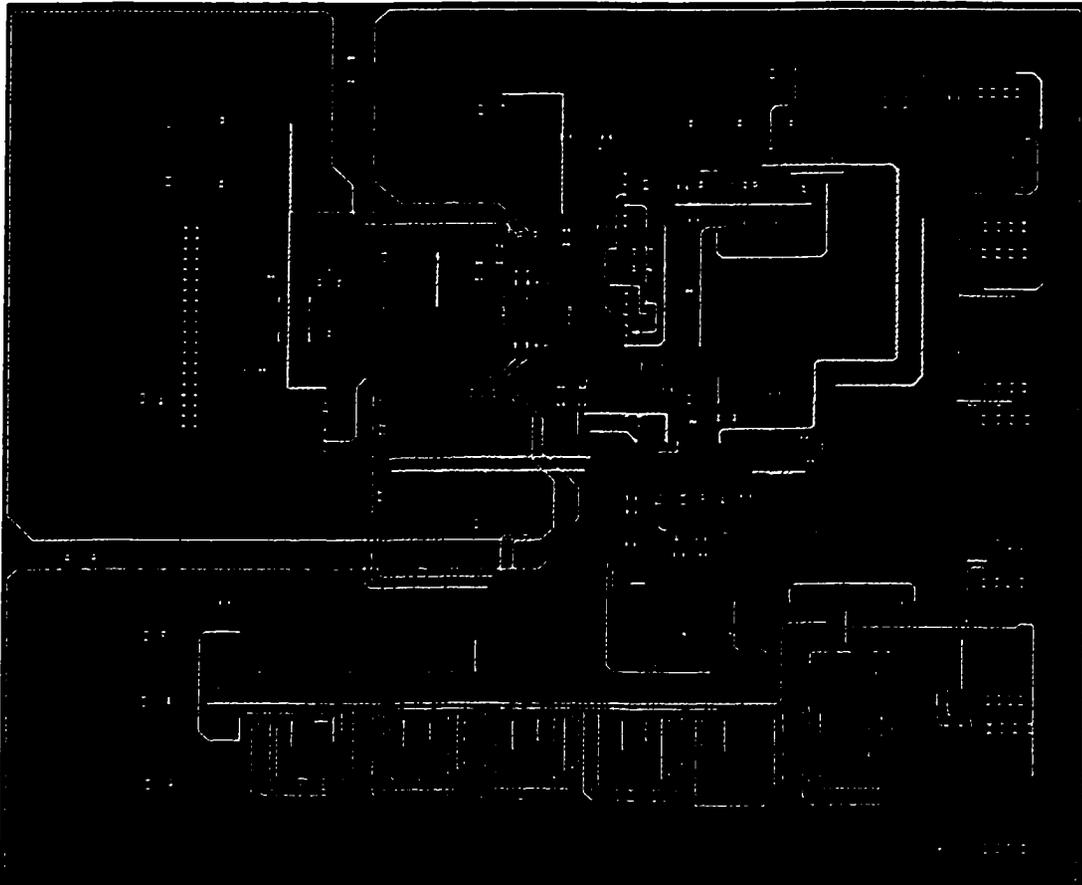


Figure 6.2 The layout of PCB

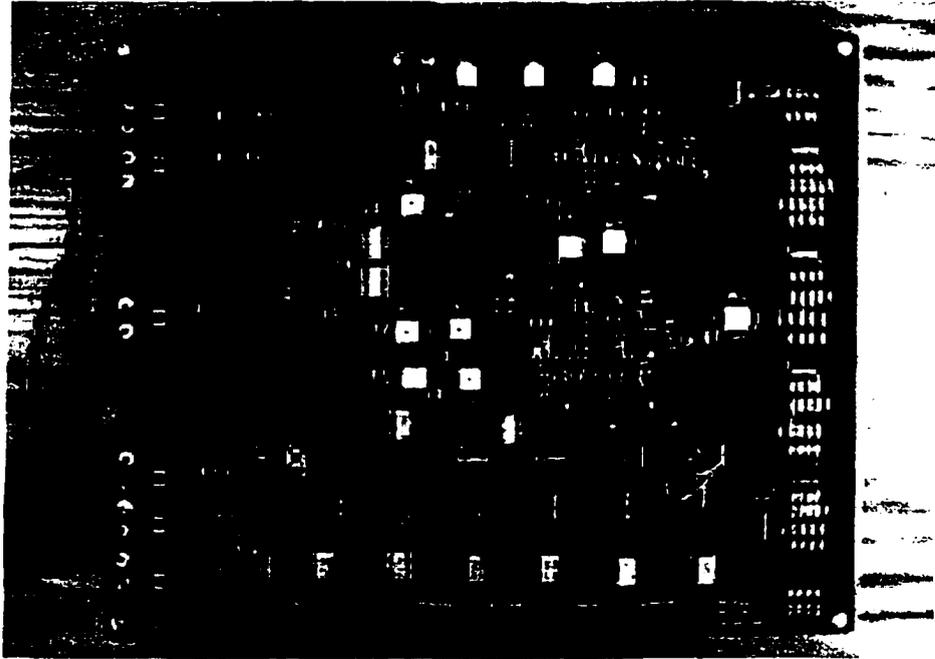


Figure 6.3 Photo of PCB

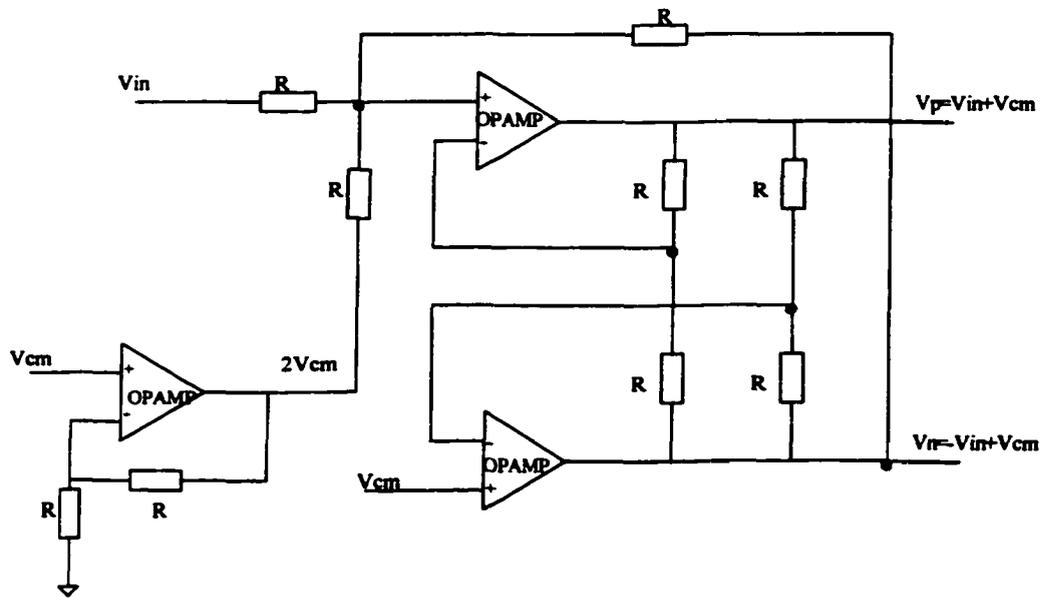


Figure 6.4 The single-to-differential circuit for LF testing

dynamic testing, a transformer is used to generate a differential signal from a single ended input analog signal, and the center tap of the transformer is connected to the common-mode voltage of the signal. For LF testing, e.g. static testing, since the LF analog signal will not be able to pass through the transformer, another circuit is designed to generate the differential signal. The circuit is shown in Figure 6.4, where V_{in} is the analog input, V_p and V_n are the differential analog outputs, and V_{cm} is the common-mode voltage. All resistors R have the same values. The OPAMP used is a THS3001 from Texas Instruments.

6.2.2 Testing Setup

The testing setup is shown in Figure 6.5. The HP3325A provides an analog input signal: a sinewave for dynamic testing, and ramp for static testing. HP8133A provides the clock signal. An HP3312A provides a low-frequency clock input for shifting in the control codes for the re-configurable ADC. A TLA704 logic analyzer from Tektronix captures the digital output data, then send it to PC for computation. A bunch of DC power sources are used.

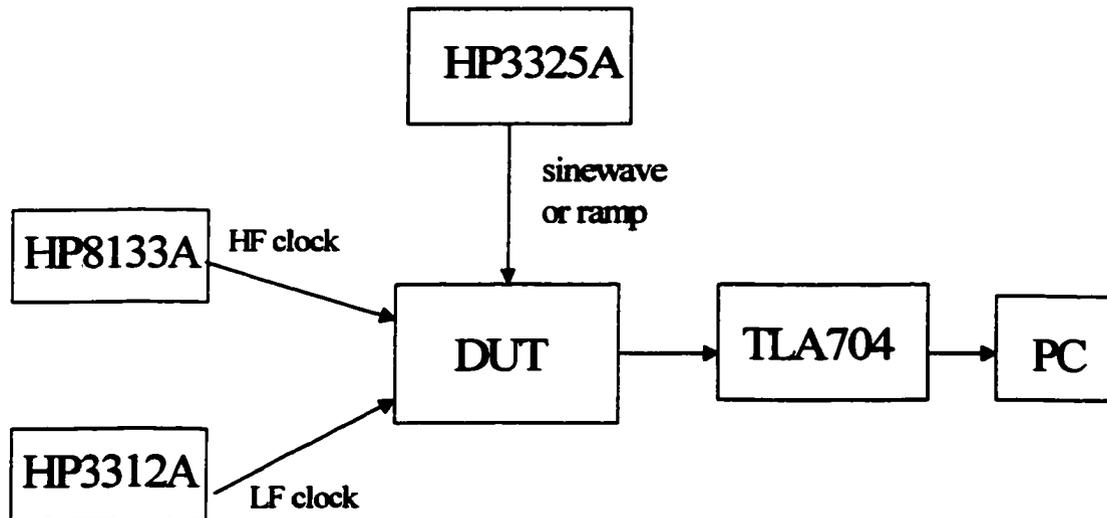


Figure 6.5 Setup of testing

6.2.3 Experimental Results

As mentioned in chapter 5, there are a total of 3 groups resulting in 18 configurations for the re-configurable pipeline ADC. The configurations are shown in Table 6.1.

6.2.3.1 Dynamic Testing

The dynamic testing is performed as shown in the setup in Figure 6.5. Five analog input frequencies F_{in} are tested: 0.8MHz, 1.6MHz, and 4MHz, 10MHz, 17.6MHz and 20MHz. The clock frequency F_{clk} is 40MHz, and the 4096 output points are captured and an FFT is performed on these output codes. SNRD and ENOB are calculated. There are three chips under test. Chip #1's testing results are shown in Figure 6.6, chip #2's testing results are shown in Figure 6.7 and chip #3's testing results are shown in Figure 6.8.

Figure 6.9 shows the typical FFT plot of the digital output of the ADC tested.

From the testing results, we can see at normal temperature, chip #1 has configuration No.9 as the best configuration, with the ENOB variance range of about 0.25bit. Chip #2 has configuration No.13

Table 6.1 Configurations of re-configurable pipeline ADC

Group #1	Group #2	Group #3
#1: 123-456-789	#7: 456-123-789	#13: 789-123-456
#2: 132-456-789	#8: 465-123-789	#14: 798-123-456
#3: 213-456-789	#9: 546-123-789	#15: 879-123-456
#4: 231-456-789	#10: 564-123-789	#16: 897-123-456
#5: 312-456-789	#11: 645-123-789	#17: 978-123-456
#6: 321-456-789	#12: 654-123-789	#18: 987-123-456

as the best configuration, with the ENOB variance range of about 0.45bit. Chip #3 also has configuration No.13 as the best configuration, with ENOB variance range about of 0.6bit. But at high input frequencies, for example, when the input frequency is close to Nyquist frequency, the ENOB variance between different configurations decreases. It is thought that this is due to the fact that the performance of the ADC at Nyquist is deteriorated by the front-end S/H stage, which is same for all the stages.

6.2.3.2 Static Testing

The static test is performed on two chips. The input analog signal is a ramp with frequency 5KHz and clock frequency of 40MHz. Total 16384 output data are captured by logic analyzer. Figure 6.10 shows the INL/DNL plots ($INL < 0.95LSB$, $DNL < 0.25LSB$).

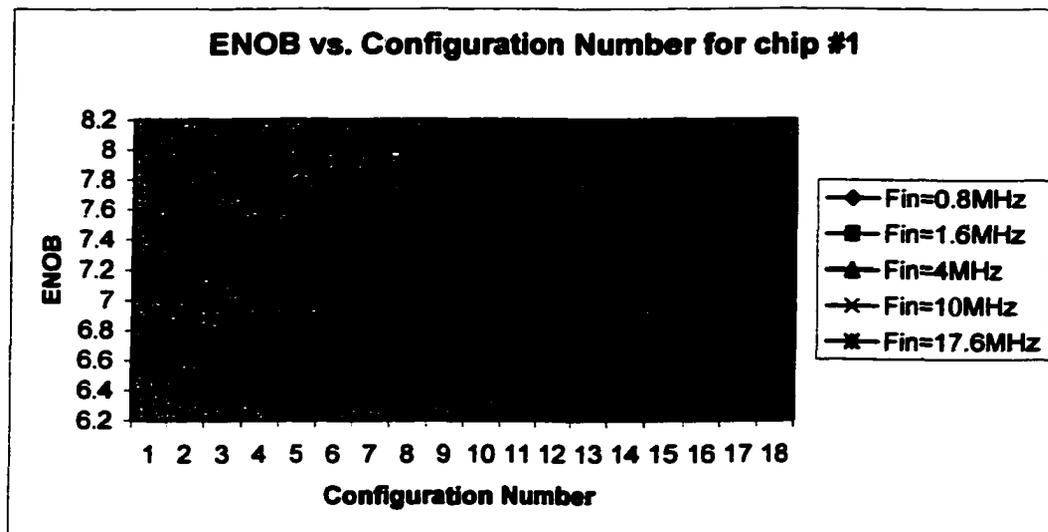


Figure 6.6 The testing results of chip #1 at normal temperature and $F_{clk}=40MHz$

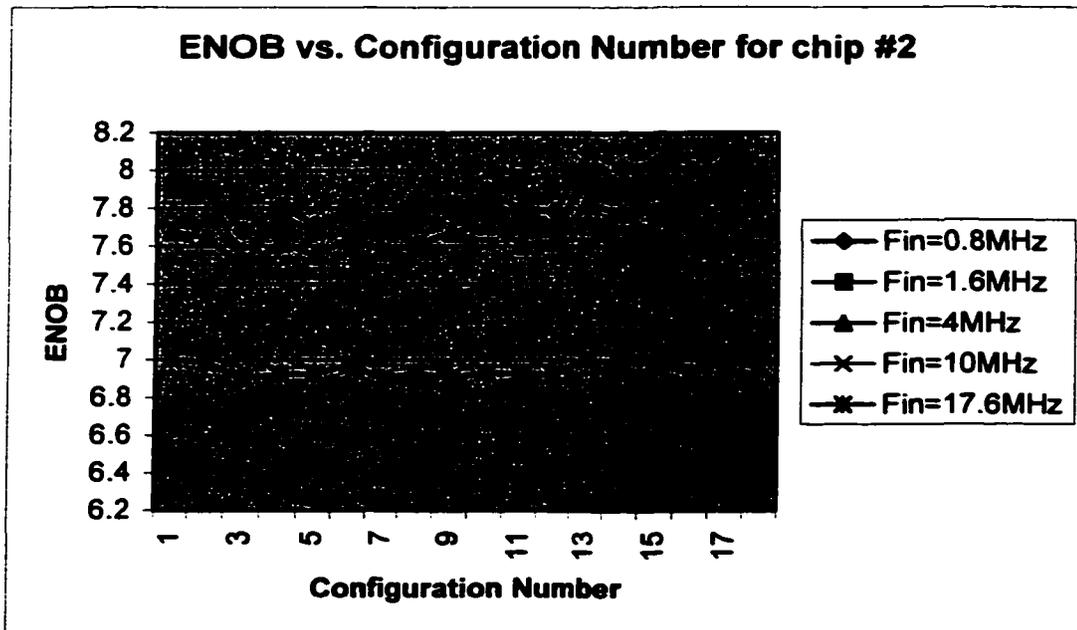


Figure 6.7 The testing results of chip #2 at normal temperature and Fclk=40MHz

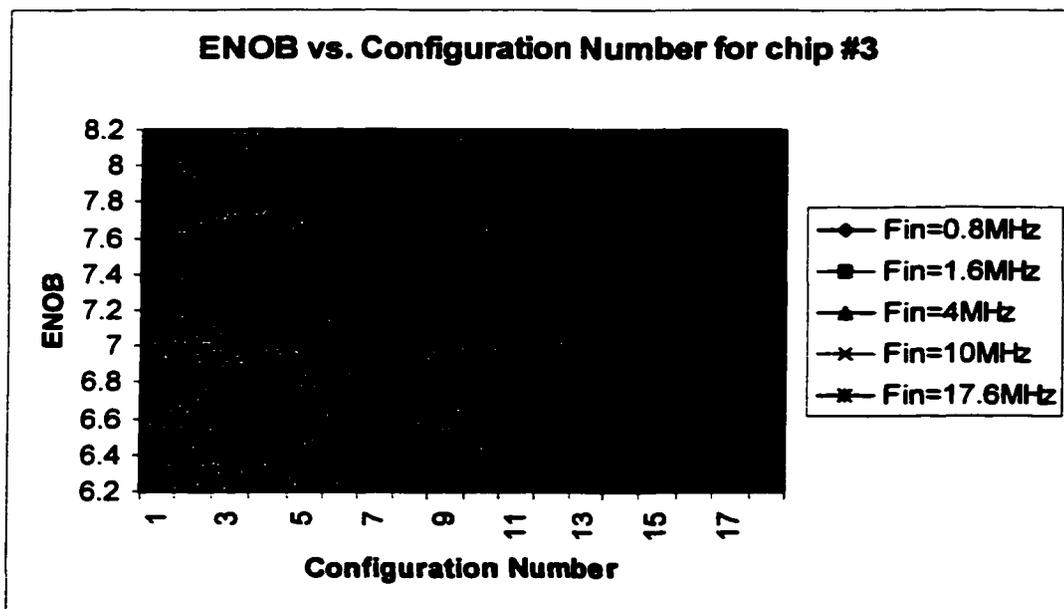


Figure 6.8 The testing results of chip #3 at normal temperature and Fclk=40MHz

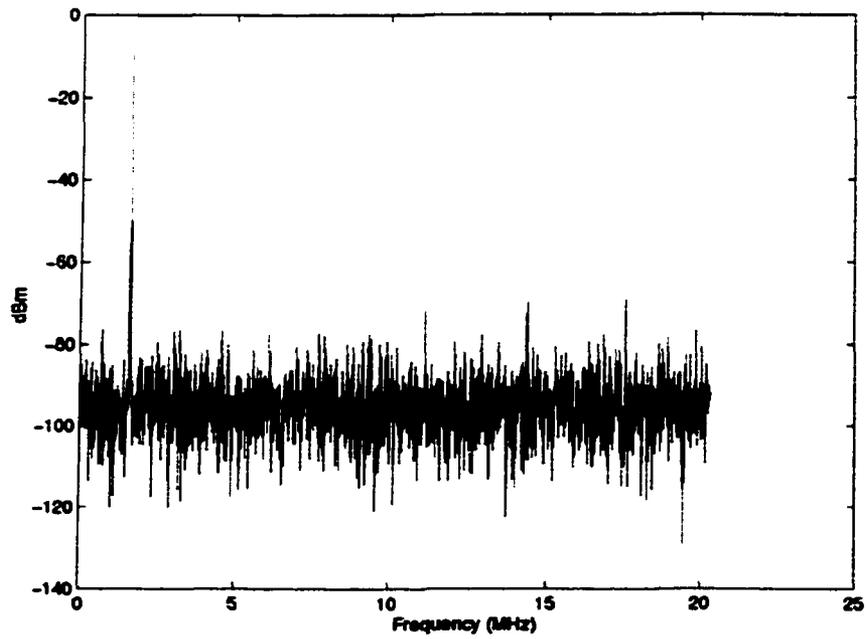


Figure 6.9 Typical plot of FFT of output of ADC (Chip #2, Configuration No.9)

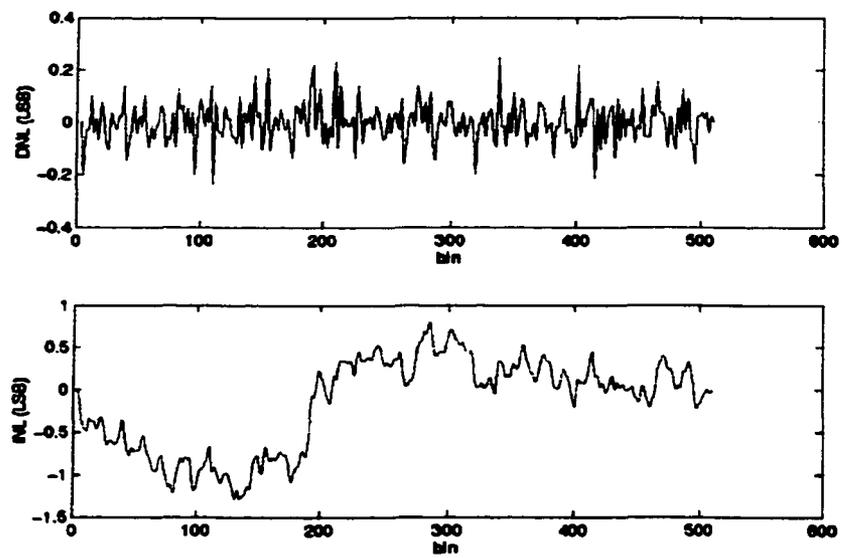


Figure 6.10 The static test of re-configurable ADC (Chip #2, Configuration No.9)

6.2.3.3 Temperature Testing

In order to prove the idea of re-configuration, a temperature test on chip #2 is done. Thermonics' Precision Temperature Forcing System T-2500 is used for temperature testing. The temperature is set to be -25°C , 0°C , 75°C for cold and hot environment test and 27°C for room temperature. Temperature test at 100°C was also attempted, but found the socket was not connecting well with the chip at 100°C , so the configurations could not be adjusted. Thus there is no 100°C testing result. Figure 6.11 to 6.13 show the temperature testing results on chip #2.

From the above testing results, we can see, at low temperature, -25°C , the best configuration is No.1 for chip #2, while at temperature 0°C , 27°C and 70°C , configuration No.13 is the best configuration. And we also observed with temperature increasing, the ENOB is increasing for all configurations, which means performance is getting better at high temperature.

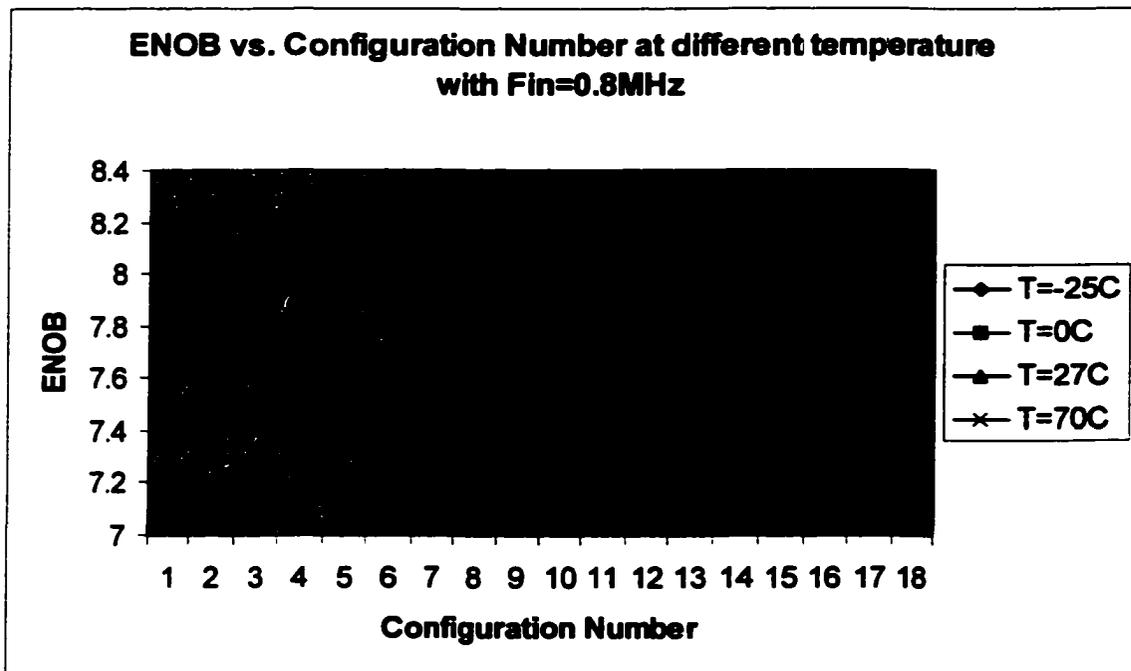


Figure 6.11 The temperature testing results of chip #2 with $F_{in}=0.8\text{MHz}$ and $F_{clk}=40\text{MHz}$

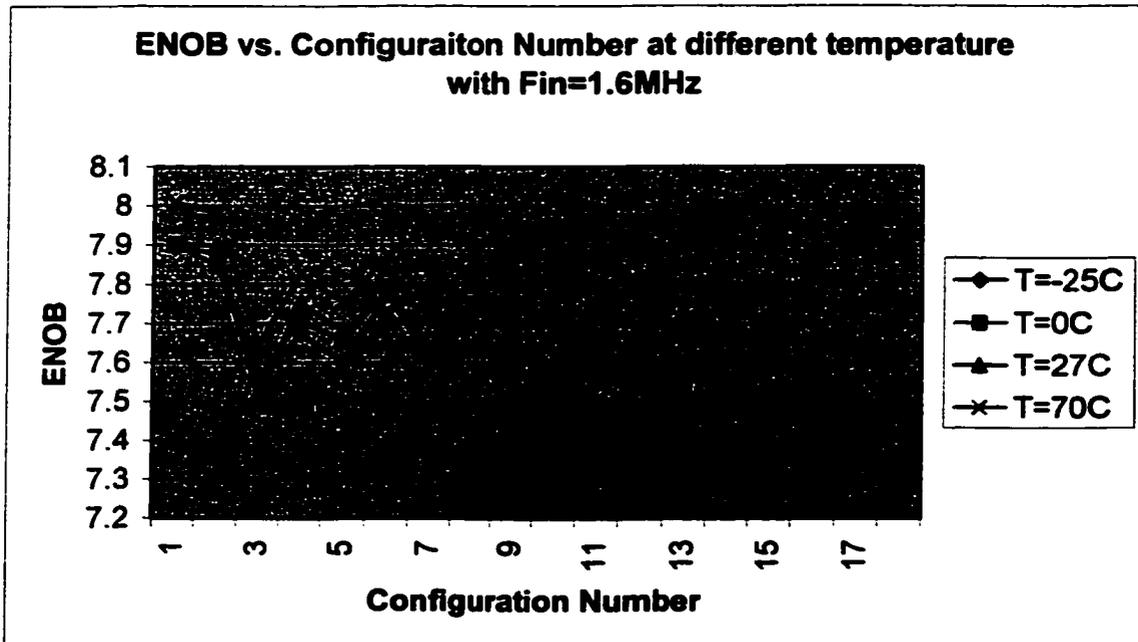


Figure 6.12 The temperature testing results of chip #2 with Fin=1.6MHz and Fclk=40MHz

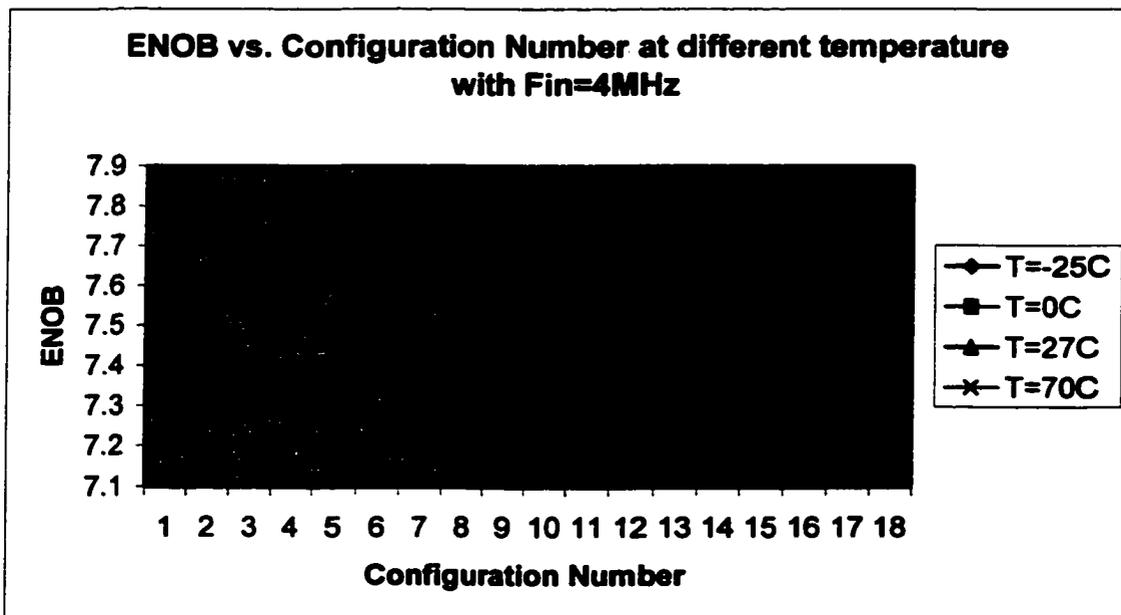


Figure 6.13 The temperature testing results of chip #2 with Fin=4.0MHz and Fclk=40MHz

6.3 Experimental Results of the on-chip Analog Signal Generation

The on-chip analog signal generation was tested in this project. As shown in Figure 5.25, the upper part of the die photo is the shift register ring. The 1024-bit shift register is implemented and according to simulations can run at speeds up to 500MHz. Matlab™ is used to generate the 1024-length PDM bitstream required. The 5th order $\Sigma\Delta$ modulator is used in Matlab™ to generate the PDM bitstream. The testing setup is shown in Figure 6.14.

An Interface RS-690 Digital Word Generator is used to generate the digital data, which is the 1024 PDM bitstream. An HP8130A is used for clock input. The bitstream is generated repeatedly. The 1024-bit shift register stores the bitstream and repeats it periodically. The output of the shift register is observed by a Tektronix TDS 754D oscilloscope. This output is also fed into the LPF, and observed by the oscilloscope at the output of the LPF. The simulation results are shown in Figure 6.15 for the output of the shift register ring and Figure 6.16 for the output of the LPF.

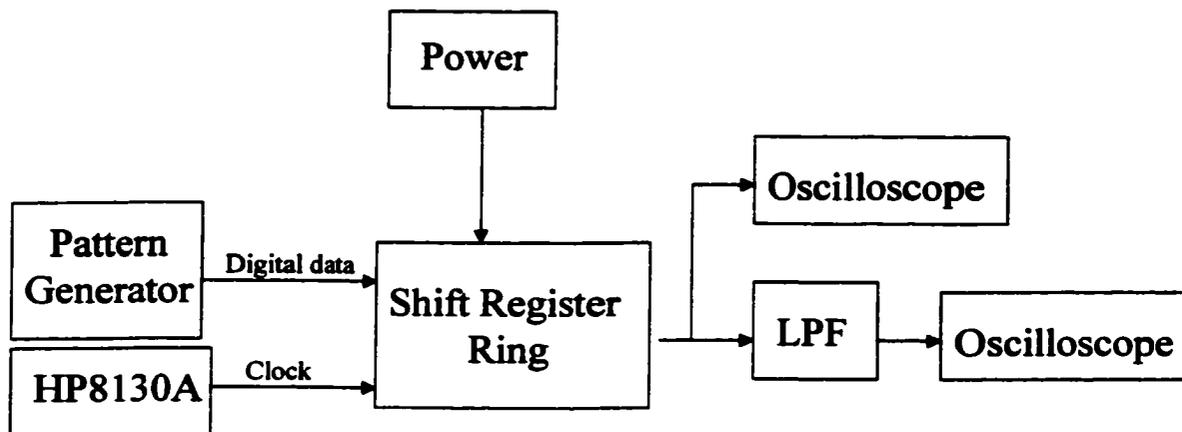


Figure 6.14 The setup of testing on-chip analog signal generator

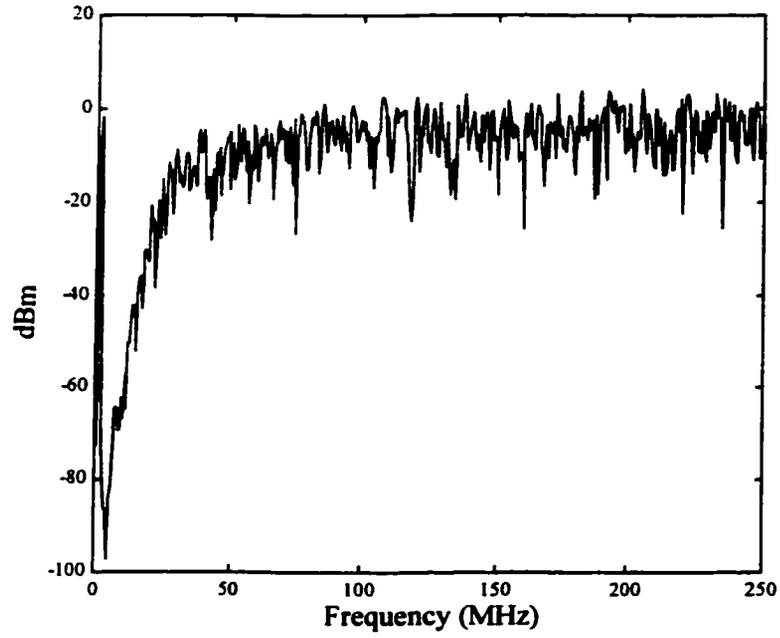


Figure 6.15 The frequency response of the output of the shift register ring

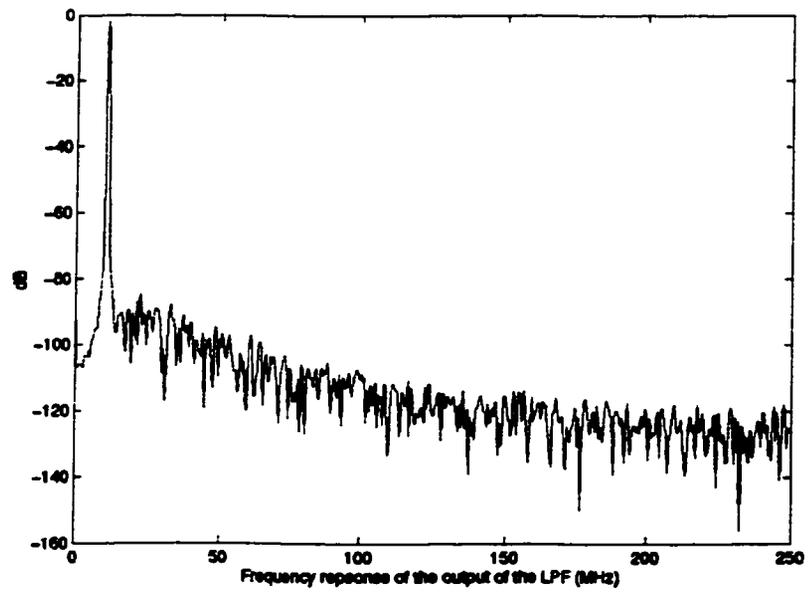


Figure 6.16 The frequency response of the output of the LPF

The schematic of the D-flip flop used in the 1024-bit shift register is shown in Figure 6.17, and the clock buffer for it is shown in Figure 6.18. The floor plan of the shift register is shown in Figure 6.19. The clock line is laid out from the last stage to the first stage.

In Figure 6.17, when clk is low, the data D will go through the N-latch and stay at the output of its inverter without going to P-latch, because P-latch is off now. The data will be kept there by the N-latch's second stage feeding back to its inverter input. When clk is high, the data at the output of the N-latch will go through the P-latch and Q is the positive edge triggered D register output. The rise time and fall time are both 0.2ns each.

In the testing of the on-chip analog signal generation, the results were not as expected. Only noise was observed at the output of the shift register and the output of the LPF. It is shown in Figure 6.20. In the testing, it was noticed that the power supply for the shift register consumes no current at any frequencies, i.e. current equals zero. But in the simulation, the shift register ring consumes 63mA current at 250MHz. By investigating the layout, it was found the power supply pad is not connected

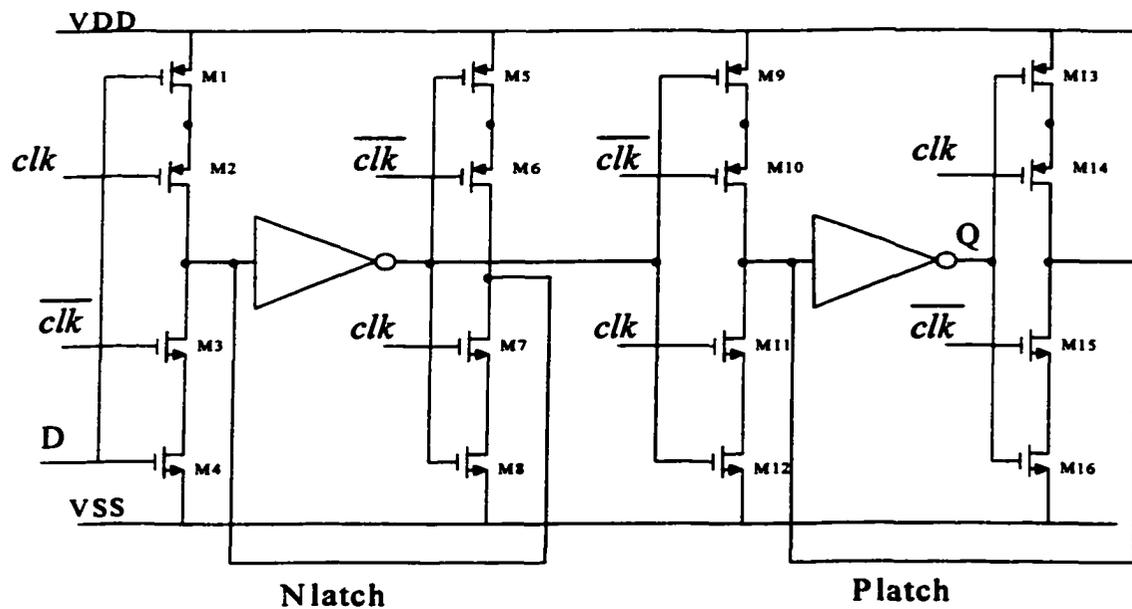


Figure 6.17 The schematic of DFF

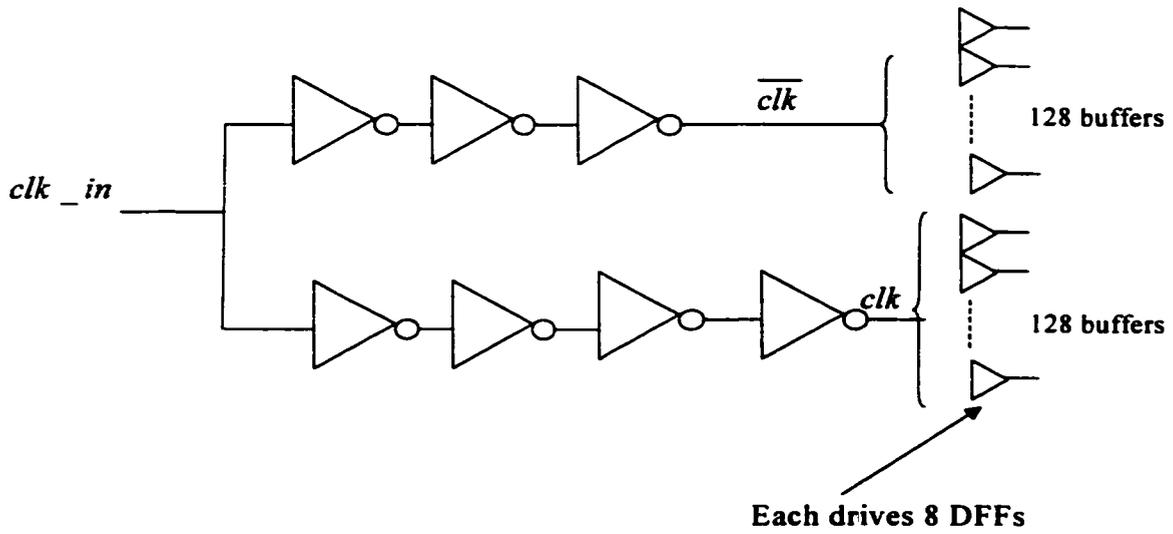


Figure 6.18 The clock tree buffer used for 1024-shift register

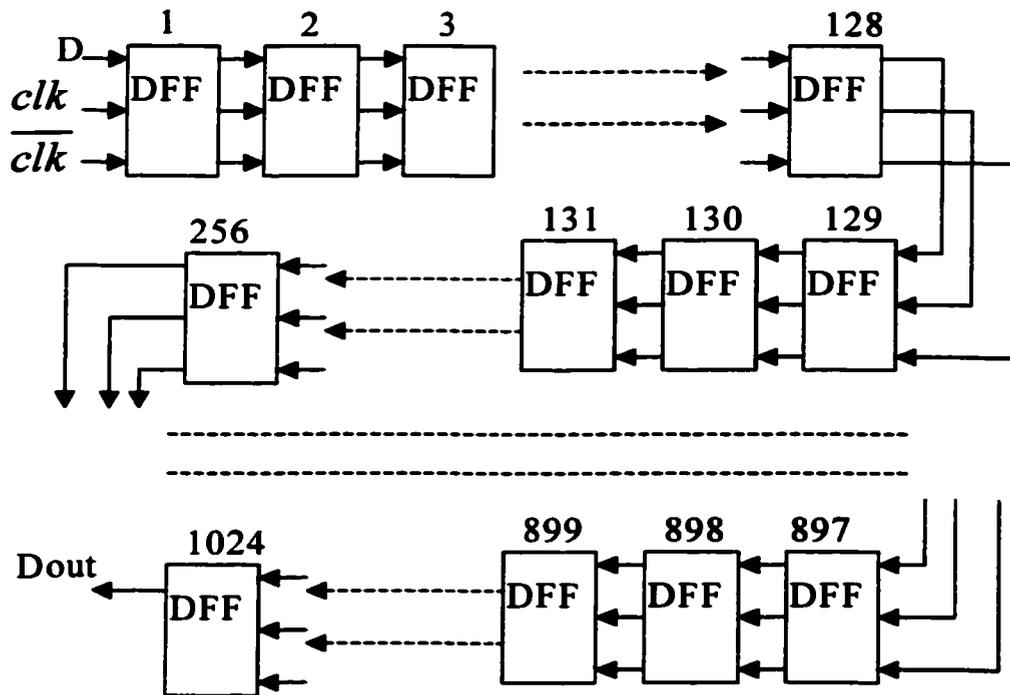


Figure 6.19 The floor plan of 1024-shift register

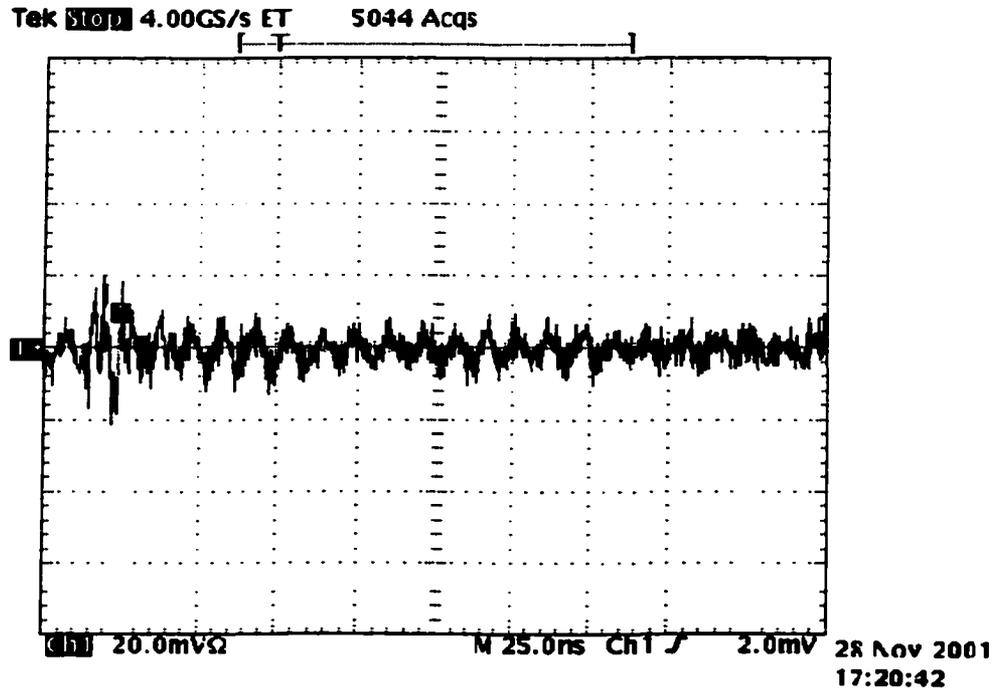


Figure 6.20 The measured output of the shift register ring

to the power supply line used for the shift register ring. Via3 was missing between metal3 and metal4. The shift register ring and re-configurable pipeline ADC use separate power supplies. That is why the re-configurable pipeline ADC is working fine. The zoomed in plot of layout error is shown in Figure 6.21, and the zoomed in plot of the micrograph of the error is shown in Figure 6.22.

6.4 Chapter Conclusion

In this chapter the experimental results of the designed chip are presented. The chip is measured for all the configurations under different temperatures. The measurement results strongly support the effectiveness of the re-configuration algorithm. It can provide significant ENOB improvement among

the set of the configurations. It is believed that a more careful design of the switching network for the re-configuration would cause the algorithm to provide even better performance than reported for this prototype. At the end of this chapter, the testing issues of the on-chip analog signal generation were addressed.

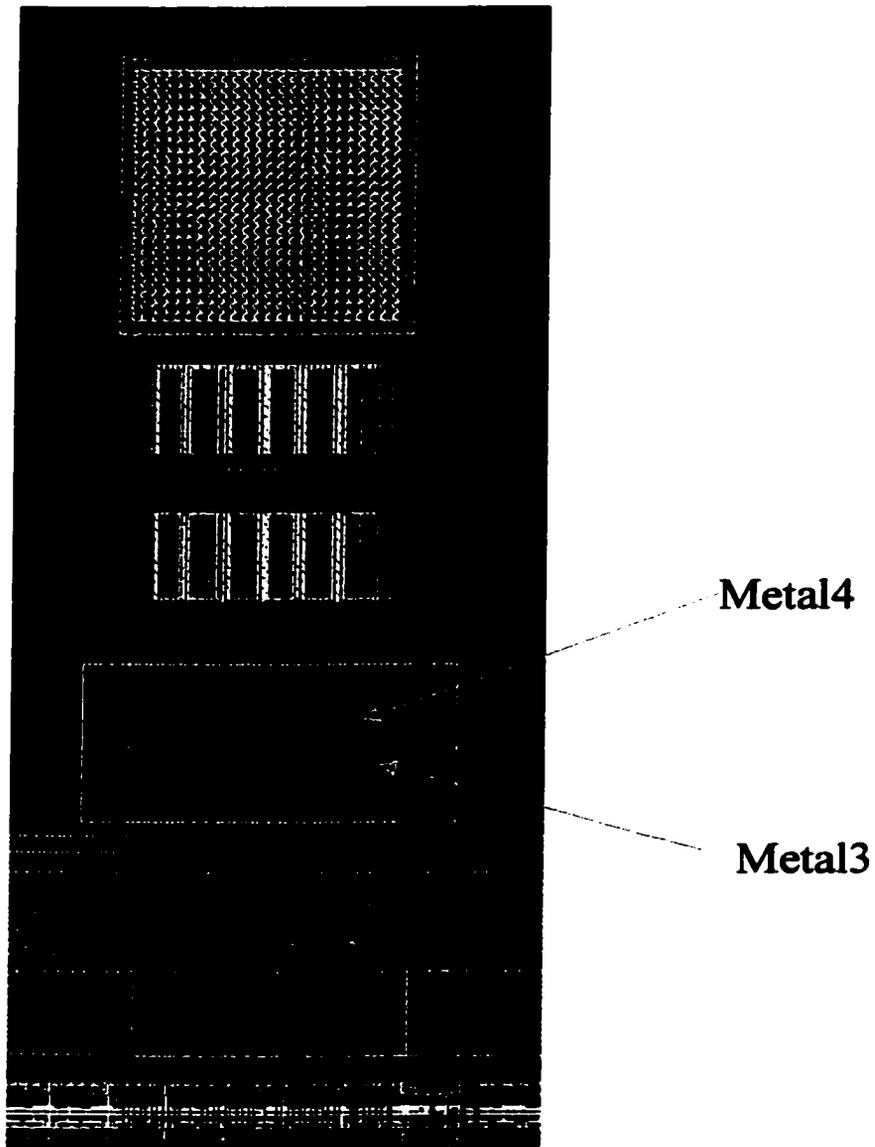


Figure 6.21 The zoomed in layout of the error

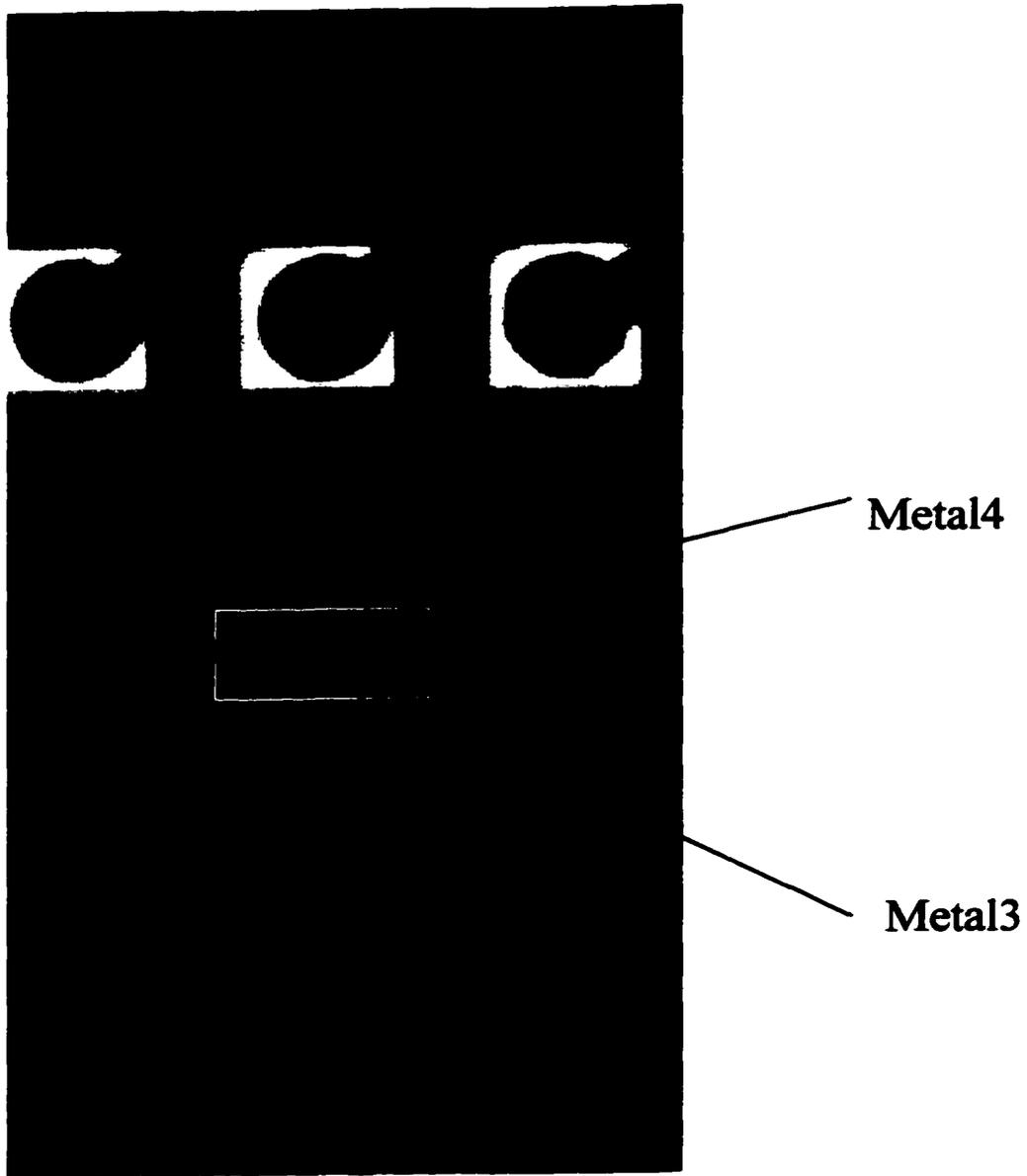


Figure 6.22 The zoomed in micrograph of the error

7 CONCLUSION

7.1 Conclusions

High-performance analog and mixed-signal integrated circuits are integral parts of today's and future networking and communication systems. The main challenge facing the semiconductor industry is the ability to economically produce these analog ICs. This translates, in part, into the need to efficiently evaluate the performance of such ICs during manufacturing (production testing) and to come up with dynamic architectures that enable the performance of these ICs to be maximized during manufacturing and later when they're operating in the field. On the performance evaluation side, this dissertation has dealt with the concept of Built-In-Self-Test (BIST) to allow the efficient and economical evaluation of certain classes of high-performance analog circuits. On the dynamic architecture side, this dissertation has dealt with pipeline ADCs and the use of BIST to dynamically, during production testing or in the field, re-configure them to produce better performing ICs. Specifically:

In chapter 2, the principles of on-chip analog signal generation were introduced. The underlying principles of $\Sigma\Delta$ modulator were described, and the simulation results were shown. Compared with the traditional memory-based on-chip analog signal generator method, such as DDFS, this $\Sigma\Delta$ method doesn't require large area or a multi-bit DAC. It has less complexity and is easily handles stability issues.

As an important topic for re-configurable ADCs, the testing fundamentals of such devices were discussed in chapter 3. Both static testing and dynamic testing were described. Coherent sampling and the histogram testing method are introduced in this chapter.

In chapter 4, the on-chip dynamic performance measurement scheme was introduced. The focus was on the principle of Narrow-band Filtering. This method is used in BIST instead of traditional industry Fast Fourier Transform (FFT) method because it requires less hardware and doesn't require that the ICs have much computational power.

Chapter 5 focused on the design of a re-configurable pipeline ADC. An algorithm for efficiently re-configuring and grouping pipeline ADC stages was introduced. The algorithm cuts down the number of evaluation permutations from $9!=362880$ to 18 thus allowing the method to be used in "real" applications. The design principles for each component in pipeline ADC were also introduced. Specifically, a fully differential folded-cascode gain-boosting OPAMP architecture that can achieve a gain of 105dB and a unity-gain bandwidth of 630MHz was implemented.

Based on the proposed BIST and the re-configurable ADC architecture, a prototype chip was implemented in TSMC's 0.25 μ m single-poly CMOS digital process and tested.

In Chapter 6, the experimental results of the prototype chip under different temperatures conditions were given. The experimental results validated that the proposed re-configuration algorithm provides an average of 0.5bit ENOB improvement among the set of configurations. For many applications, this is a very significant performance improvement.

The BIST and re-configurability techniques proposed are not limited to pipeline ADCs only. The BIST methodology is applicable to many analog systems and the re-configurability is applicable to any analog pipeline systems.

The specific contributions of this dissertation are:

- 1) A BIST system for re-configurable pipeline ADCs is proposed. It is a unique system with analog input testing signal generated on-chip as well as the dynamic performance of the ADC measured on-chip. When this system is used in wafer test, it will save testing time and thus testing costs.

- 2) A re-configurable pipeline ADC architecture to improve the dynamic performance is proposed. Based on the dynamic performance measurements, the best performance configuration is chosen from a collection of possible pipeline configurations. This basic algorithm can be applied to many pipeline analog systems.
- 3) The grouping algorithm for re-configurable pipeline ADCs is proposed. It can cut down the number of evaluation permutation from thousands to 18 for a 9-bit ADC thus allowing the method to be used in “real” applications.
- 4) A 40Msample/s 9-bit re-configurable pipeline ADC is designed and implemented in TSMC’s 0.25 μ m single-poly CMOS digital process. It includes a fully differential folded-cascode gain-boosting operational amplifier with high gain and high unity gain bandwidth.
- 5) Verification of the prototype under different temperature conditions, with the experimental results strongly supporting the effectiveness of the re-configuration algorithm. It provides an average of 0.5bit ENOB improvement among the set of configurations.

7.2 Recommended Future Work

Built-in self-test and high-performance analog-to-digital converter design will continue to be the important topics in the area of wireless communication. The techniques proposed in this dissertation work are conceptual and therefore have endless potential of being further improved or expanded to meet future demands.

- 1) For the on-chip analog signal generator, the on-chip analog low-pass filter is important in the design. Although an off-chip analog LPF can be used, this will limit the BIST implementation.
- 2) Although the re-configurable grouping algorithm was proposed for this re-configurable pipeline ADC structure, it really is not limited to pipeline ADCs only. In fact, it can be used for many analog pipeline systems. More research work can be done in this area.

3) Due to time and cost limits, the whole BIST system for re-configurable pipeline ADC was not wholly implemented. Future work can be done to implement the rest of the system.

Any silicon implementation of the improvement on the above aspects in the future is a worthwhile goal.

REFERENCES

- [1] Benoit Dufort, Gordon W. Roberts, "On-Chip Analog Signal Generation for Mixed-Signal Built-In Self-Test", *IEEE J. Solid-State-Circuits*, vol.34, pp318-330, Mar.1999
- [2] H. T. Nicholas and H. Samueli, "A 150 MHz direct digital frequency synthesizer in 1.25 μm CMOS with -90 dBc spurious performance", *IEEE J. Solid-State-Circuits*, vol.26, pp1959-1969, Dec.1991
- [3] Rudy Van De Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994
- [4] Benoit Dufort, Gordon W. Roberts, "Signal Generation Using Periodic Signal and Multi-Bit Sigma-Delta Modulated Streams", *IEEE Internal Test Conference*, pp396-405, 1997
- [5] Evan M. Hawrysh, Gordon W. Roberts, "An Integration of Memory-Based Analog Signal Generation into Current DFT Architectures", *IEEE International Test Conference*, pp528-537, Oct.1996
- [6] Evan M. Hawrysh, Gordon W. Roberts, "An Integration of Memory-Based Analog Signal Generation into Current DFT architectures", *IEEE Trans. On Instrumentation and Measurement*, vol.47, pp. 748-759, June1998
- [7] Benoit Dufort, Gordon W. Roberts, "Optimized periodic Sigma-Delta bitstreams for analog signal generation", in *Proc. 1997 Midwest Symp. Circuits and Systems*, Sacramento, CA, pp. 289-292, Aug. 3-6, 1997,
- [8] Steven R. Norsworthy, *Delta-Sigma Data Converter*, IEEE Press, 1997

- [9] G. W. Roberts and A. K. Lu, *Analog Signal Generation For Built-In Self-Test of Mixed-Signal Integrated Circuits*, Kluwer Academic Publishers, Norwell, Ma, USA, 1994
- [10] Richard Schreier, "An Empirical Study of High-Order Single-Bit Delta-Sigma Modulators", *IEEE Trans. On Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, No.8, pp. 461-466, Aug. 1993
- [11] S. Hein and A. Zakhor, "On the stability of interpolative sigma delta modulators", in *Proc. 1991 IEEE Int. Symp. Circuits and Systems*, vol. 3, pp. 1621-1624, June 1991
- [12] T. Ritoniemi, T. Karema, and H. Tenhunen, "Design of stable high order 1-bit sigma-delta modulators", in *Proc. 1990 IEEE Int. Symp. Circuits and Systems*, vol. 4, pp.3267-3270, May 1990
- [13] R. Schreier and Y. Yang, "Stability tests for single-bit sigma-delta modulators with second-order FIR noise transfer functions", in *1992 Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1316-1319
- [14] A. K. Lu and G. W. Roberts, "An Analog Multi-Tone Signal Generator for Built-in Self-Test Applications", *IEEE International Test Conference*, pp. 650-659, 1994
- [15] Matthew Mahoney, "DSP-Based Testing of Analog and Mixed-Signal Circuits", Washington, DC: Computer Society Press of the IEEE, 1987
- [16] Ben M. J. Kup, Eise Carel Dijkmans, "A Bitstream Digital-to-Analog Converter with 18-b Resolution", *IEEE J. Solid-State-Circuits*, vol.26, pp. 1757-1763, Dec.1991
- [17] Standard for Terminology and Test Methods for Analog-to-Digital Converters, *IEEE Std 1241 Draft*, May, 1997

- [18] Michael F. Toner, Gordon W. Roberts, "A BIST Scheme for a SNR, Gain Tracking, and Frequency Response Test of a Sigma-Delta ADC", *IEEE Trans. On Circuits and Systems II, Analog and Digital Signal Processing*, vol.42, pp1-15, Jan.1995
- [19] Michael F. Toner, Gordon W. Roberts, "Towards Built-In-Self-Test for SNR Testing of a Mixed-Signal IC", in *1993 IEEE Int. Symp. Circuits Syst.* Chicago, IL, pp. 1599-1602, May 1993
- [20] M. Padmanabhan and K. Martin, "Filter banks for time-recursive implementations of transforms", *IEEE Trans. Circuits Syst.*, vol. 40, No.1, pp. 41-50, Jan. 1993
- [21] M. F. Toner and G. W. Roberts, " A BIST scheme for an SNR test of a Sigma-Delta ADC", in *1993 Proc. Int. Test Conf.*, Baltimore, MD, pp. 805-814, Oct. 1993
- [22] Neil H. E. Weste and Kamran Eshrachian, *Principles of CMOS VLSI Design*, Addison-Wesley Publishing Company, 1993
- [23] Kai Hwang, *Computer Arithmetic*, John Wiley & Sons Inc. 1979
- [24] Irwin Miller and Marylees Miller, *John E. Freund's Mathematical Statistics*, Prentice-Hall Inc. 1999
- [25] G. M. Yin, F. Op't Eynde, W. Sansen, "A High-Speed CMOS Comparator with 8-bit resolution", *IEEE Journal of Solid-State Circuits*, vol.27, No.2, pp.208-211, Feb.1992.
- [26] Y. T. Wang and B. Razavi, "An 8-b 150-MHz COMS A/D converter", *IEEE Journal of Solid-State Circuits*, vol.35, pp.308-317, Mar.2000.
- [27] W. Bright, "8-b 75-MS/s 70-mW parallel pipelined ADC incorporating double sampling", in *ISSCC Dig. Tech. Papers*, Feb. 1998, pp.146-147

- [28] K. Y. Kim, N. Kusayanagi, and A. Abidi, "A 10-b, 100-MS/s COMS A/D converter," *IEEE Journal of Solid-State Circuits*, vol.32, pp.302-311, Mar.1997.
- [29] S. H. Lewis, H. Scott Fatterman, George F. Gross, Jr., R. Ramchandran and T. R. Viswanathan, "A 10-b 20-Msamples/s Analog-to-Digital Converter", *IEEE Journal of Solid-State Circuits*, vol.27, pp.351-357, Mar.1992.
- [30] S. H. Lewis, "Optimizing the stage resolution in pipelined, multistage, analog-to-digital converters for video-rate applications." *IEEE Trans. Circuits Syst II.*, vol.39, pp.515-523, Aug.1992.
- [31] David William Cline, "Noise, Speed, and Power Trade-offs in Pipelined Analog-to-Digital Converters", University of California, Berkeley, Ph.D Thesis, 1995
- [32] David A. Johns and Ken Martin, *Analog Integrated Circuit Design*, John Wiley, New York, 1997
- [33] Andrew N. Karanicolas, Hae-seung Lee and Kantilal L. Bacrania, "A 15-b 1Msamples/s Digital Self-Calibrated Pipeline ADC", *IEEE Journal of Solid-State Circuits*, vol.28, pp.1207-1215, Dec.1993
- [34] T. Cho, "Low-power Low-voltage Analog-to-Digital Conversion Techniques using pipelined Architectures", University of California, Berkeley, Ph.D Thesis, 1995
- [35] Yuh-Min Lin, B. Kim and Paul R. Gray, "A 13-b 2.5MHz Self-Calibrating Pipelined A.D converter in 3- μ m CMOS", *IEEE Journal of Solid-State Circuits*, vol. 26, No.4, pp. 628-635, Apr. 1991

- [36] Klass Bult and Govert J. G. M. Geelen, "A Fast-Settling CMOS OpAmp for SC Circuits with 90-dB DC Gain". *IEEE Journal of Solid-State Circuits*, vol. 25, No.6, pp. 1379-1384, Dec. 1990
- [37] Clemens M. Hammerschmied and Qiuting Huang, " Design and Implementation of an Untrimmed MOSFET-Only 10-Bit A/D Converter with -79-dB THD", *IEEE Journal of Solid-State Circuits*, vol. 33, No.8, pp. 1148-1157, Aug. 1998
- [38] H. Liu and M. Hassoun, "High Speed Re-configurable Pipeline ADC Cell Design", *Mixed-Signal Design, 2001. SSMSD. 2001 Southwest Symposium on* , 2001 pp. 158 -161
- [39] M. Wit, K. Tan, and R. K. Hester, "A Low-Power 12-b Analog-to-Digital Converter with On-Chip Precision Trimming", *IEEE Journal of Solid-State Circuits*, vol.28, pp.455-461, Apr.1993.
- [40] Un-Ku Moon and Bang-Sup Song, "Background Digital Calibration Techniques for Pipelined ADC's", *IEEE Trans. Circuits Syst II.*, vol.44, No. 2, pp.102-109, Feb.1997.
- [41] K. Dyer, D. Fu, P. Hurst and S. Lewis, "A Comparison of Monolithic Background Calibration in Tow Time-interleaved Analog-to-Digital Converters", *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE, International Symposium on, Volume: 1*, 1998, pp. 13 -16
- [42] S. H. Lewis and P. J. Hurst, "An Analog Background Calibration Technique for Time-Interleaved Analog-to-Digital Converters", *IEEE Journal of Solid-State Circuits*, vol.33, pp.1912-1919, Dec.1998.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my major advisor Prof. Marwan Hassoun for his support and guidance though all my PhD study procedure. His optimism and self-confidence, together with his wide knowledge in the area of electrical engineering, gave me a great guidance and inspiration in my study for the past four years.

I would like to thank my co-major advisor Prof. William Black, Jr. for his great instruction and help in my PhD study.

I would like to thank all my committee members: Prof. Robert Weber, Prof. Chris Chong-Nuen Chu, and Prof. Yuhong Yong for being in my committee.

I would like to thank all my colleagues for making my stay enjoyable.

I would like to thank David Folberg for the help in testing and Richard Henry for the great support on testing. I would like to thank Thanh-Hai Huynh for bounding the chips.

I would like to thank Steve Gao for accompanying me though all the testing procedures and his great support in my life.

Finally, I would like to express my deepest appreciation to my family: my parents and sisters, who always encourage me to do my best and who have always been there to share my difficulties and time of joy.

This research is sponsored by Texas Instruments, Inc. and the testing is done in Conexant Systems, Inc. I would like to express my appreciation to them.