

# Circuits and Algorithms for Pipelined ADCs in Scaled CMOS Technologies

by

Lane Gearle Brooks

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

© Massachusetts Institute of Technology 2008. All rights reserved.

Author .....  
Department of Electrical Engineering and Computer Science  
May 6, 2008

Certified by .....  
Hae-Seung Lee  
Professor  
Thesis Supervisor

Certified by .....  
Gregory Wornell  
Professor  
Thesis Supervisor

Accepted by .....  
Terry P. Orlando  
Chairman, Department Committee on Graduate Students



# Circuits and Algorithms for Pipelined ADCs in Scaled CMOS Technologies

by

Lane Gearle Brooks

Submitted to the Department of Electrical Engineering and Computer Science  
on May 6, 2008, in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy in Computer Science and Engineering

## **Abstract**

CMOS technology scaling is creating significant issues for analog circuit design. For example, reduced signal swing and device gain make it increasingly difficult to realize high-speed, high-gain feedback loops traditionally used in switched capacitor circuits. This research involves two complementary methods for addressing scaling issues. First is the development of two blind digital calibration techniques. Decision Boundary Gap Estimation (DBGE) removes static non-linearities and Chopper Offset Estimation (COE) nulls offsets in pipelined ADCs. Second is the development of circuits for a new architecture called zero-crossing based circuits (ZCBC) that is more amenable to scaling trends. To demonstrate these circuits and algorithms, two different ADCs were designed: an 8 bit, 200MS/s in TSMC 180nm technology, and a 12 bit, 50 MS/s in IBM 90nm technology. Together these techniques can be enabling technologies for both pipelined ADCs and general mixed signal design in deep sub-micron technologies.

Thesis Supervisor: Hae-Seung Lee

Title: Professor

Thesis Supervisor: Gregory Wornell

Title: Professor



## Acknowledgments

I would like to thank my advisers, family, and friends for helping and supporting me with this work.

I would like to thank NDSEG, CICS, and DARPA for funding my research.



# Contents

<b>1</b>	<b>Introduction</b>	<b>19</b>
1.1	Pipelined ADC Overview . . . . .	23
1.2	Comparator Based Switched Capacitor Circuits . . . . .	26
1.3	Pipelined ADC Error Models . . . . .	27
1.3.1	Finite Opamp Gain . . . . .	27
1.3.2	Finite Current Source Output Impedance . . . . .	28
1.3.3	Capacitor Mismatch . . . . .	31
1.3.4	Charge Injection and Stage Offset . . . . .	32
1.3.5	Bit Decision Comparator Offset . . . . .	32
1.3.6	Errors from Multiple Stages . . . . .	33
1.4	Redundancy . . . . .	34
<b>2</b>	<b>Decision Boundary Gap Estimation</b>	<b>39</b>
2.1	Gap Correction . . . . .	41
2.2	Gap Estimation . . . . .	44
2.2.1	Max-Min Gap Estimator . . . . .	46
2.2.2	Bin-Reshaping Gap Estimator . . . . .	48
2.2.3	Cost-Minimizing Estimator . . . . .	51
2.2.4	Estimator Discussion . . . . .	55
2.3	Simulation Results . . . . .	58
2.4	Conclusion . . . . .	60

<b>3</b>	<b>Zero-Crossing Based Circuits</b>	<b>61</b>
3.1	Background . . . . .	61
3.1.1	Opamp-Based Switch Capacitor Circuits . . . . .	61
3.1.2	Comparator-Based Switched Capacitor Circuits . . . . .	62
3.2	Zero-Crossing Based Circuits . . . . .	64
3.3	ZCBC Pipelined ADC Implementation . . . . .	67
3.3.1	DZCD Design . . . . .	67
3.3.2	Current Source Splitting . . . . .	68
3.3.3	Shorting Switches . . . . .	68
3.3.4	Reference Voltage Switches . . . . .	70
3.3.5	Current Source Implementation . . . . .	71
3.3.6	Bit Decision Flip-Flops . . . . .	72
3.3.7	First Stage Considerations . . . . .	73
3.4	Experimental Results . . . . .	74
3.5	Power Efficiency Analysis . . . . .	76
3.5.1	DZCD Noise Analysis . . . . .	76
3.5.2	Comparison to Original CBSC Implementation . . . . .	84
3.5.3	FOM Discussion . . . . .	85
3.6	Conclusion . . . . .	86
<b>4</b>	<b>Chopper Offset Estimation</b>	<b>89</b>
4.1	Chopper Offset Estimation . . . . .	92
4.1.1	Traditional Chopper Stabilization . . . . .	92
4.1.2	Chopper Offset Estimation (COE) . . . . .	93
4.1.3	COE Decimation . . . . .	95
4.2	Random Chopping Vector . . . . .	96
4.2.1	Minimum Variance Linear Unbiased Estimator . . . . .	97
4.2.2	MVLU Performance . . . . .	99
4.2.3	MVLU Example . . . . .	99
4.2.4	Distortion Performance . . . . .	101



4.2.5	Random vs. Deterministic Chopping . . . . .	102
4.3	Additional COE Architectures . . . . .	103
4.3.1	Input Referred Offset Compensation with COE . . . . .	104
4.3.2	COE for Pipelined ADCs . . . . .	105
4.3.3	Per-Stage COE for Pipelined ADCs . . . . .	106
4.3.4	Multistage Chopping . . . . .	108
4.4	Conclusion . . . . .	109
<b>5</b>	<b>ZCBC Revisited</b>	<b>111</b>
5.1	System Level Improvements . . . . .	111
5.1.1	Embedded SRAM and Programmable Output Drivers . . . . .	112
5.1.2	Triple Well for Improved Substrate Isolation . . . . .	112
5.1.3	On-chip Bias and Voltage Generation . . . . .	112
5.1.4	Single Ground . . . . .	114
5.1.5	Packaging Considerations . . . . .	115
5.2	Fully Differential ZCBC . . . . .	115
5.2.1	Common Mode Control . . . . .	118
5.2.2	Symmetry for Improved Power Supply Noise Rejection . . . . .	119
5.2.3	Differential Zero-Crossing Detector . . . . .	122
5.2.4	Chopper Offset Estimation . . . . .	124
5.3	Voltage References . . . . .	125
5.3.1	Off-chip Reference Voltage Issues . . . . .	126
5.3.2	Voltage Reference Switching via Capacitor Splitting . . . . .	128
5.3.3	Capacitor Splitting with Fully Differential Designs . . . . .	132
5.4	Redundancy For Increased Signal Range . . . . .	135
5.5	Complete ZCBC Pipeline Stage . . . . .	139
5.6	Sub-ADC Design . . . . .	142
5.6.1	Bit Decision Comparator Design . . . . .	145
5.7	Noise Analysis . . . . .	149
5.7.1	Dynamics . . . . .	149

5.7.2	Input Referred Noise Derivation . . . . .	150
5.7.3	Substituting Real Circuit Parameters . . . . .	154
5.7.4	Linearity from Finite Current Source Impedance . . . . .	156
5.7.5	Differential ZCD Design Methodology . . . . .	157
5.7.6	Number of Ramps Analysis . . . . .	161
5.8	Experimental Results . . . . .	163
5.8.1	Overall Performance . . . . .	163
5.8.2	ZCD Offset Performance . . . . .	164
5.8.3	I/O Noise Coupling . . . . .	167
5.8.4	BDC Offset . . . . .	168
5.9	Conclusion . . . . .	170
<b>6</b>	<b>Conclusion</b>	<b>171</b>
6.1	ZCBC Future Work . . . . .	171
6.1.1	Reference Voltages . . . . .	171
6.1.2	PVT Hardening . . . . .	173
6.1.3	Common Mode Feedback . . . . .	175
6.2	Conclusions . . . . .	176

# List of Figures

1-1	Trend analysis for published pipelined ADCs. . . . .	20
1-2	Block diagram of an $N_j$ bit/stage pipeline stage. . . . .	24
1-3	Typical circuit implementation of 1 bit/stage pipeline stage. Single-ended version shown for simplicity. . . . .	25
1-4	Ideal stage voltage transfer function (left) and ADC transfer function (right). . . . .	26
1-5	Implementation of a 1 bit/stage CBSC pipeline stage. . . . .	27
1-6	Single stage and ADC transfer function from finite op-amp gain or finite current source output impedance. . . . .	28
1-7	Single stage and ADC transfer function from capacitor mismatch when $\epsilon < 0$ . . . . .	31
1-8	Single stage and ADC transfer function from capacitor mismatch when $\epsilon > 0$ . . . . .	32
1-9	Single stage and ADC transfer function from positive charge injection or stage transfer offset. . . . .	33
1-10	Single stage and ADC transfer function from negative charge injection or stage transfer offset. . . . .	33
1-11	Single stage and ADC transfer function from a positive bit decision comparator offset. . . . .	34
1-12	Single stage and ADC transfer function from a negative bit decision comparator offset. . . . .	34
1-13	ADC transfer function when first 2 stages have finite opamp gain. . .	35

1-14	Block diagram of an $M_j$ bit/stage pipeline stage. Over-range protection is offered when $M_j > N_j$ . . . . .	35
1-15	Ideal stage voltage transfer function (left) for a 1.5 bit/stage pipeline stage and resulting ADC transfer function (right). . . . .	36
1-16	Single stage and ADC transfer function from positive charge injection or stage transfer offset. . . . .	36
1-17	Single stage and ADC transfer function from a positive bit decision comparator offset. . . . .	37
1-18	Single stage and ADC transfer function from capacitor mismatch when $\epsilon > 0$ . . . . .	37
2-1	Block diagram of correction scheme for a single stage. . . . .	42
2-2	Transfer function of raw and corrected samples. . . . .	43
2-3	Block diagram of concatenated stages utilizing DBGE. . . . .	43
2-4	Signal flow graph modelling the code gap of stage $k$ of a 1 bit/stage pipelined ADC. . . . .	44
2-5	Histogram of an example data set (in the absence of noise) corrupted by unknown offsets. . . . .	45
2-6	Signal flow graph of error model including circuit noise $w_k$ . . . . .	47
2-7	Histogram of an example data set corrupted by a code gap and additive circuit noise. . . . .	48
2-8	Example of a histogram resulting from a uniformly distributed input when gap size is not an integer. . . . .	49
2-9	Histogram showing geometric interpretation of the Bin-Reshaping estimation method. . . . .	50
2-10	Histograms under various $\hat{g}$ estimates. Actual $g = 9$ . . . . .	53
2-11	DNL vs $\hat{g}$ . . . . .	54
2-12	Raw and calibrated INL of 13 stage 1.5 bit/stage ADC with mismatch parameters specified in Table 2.1. . . . .	56

2-13	Raw and Calibrated DFT response of 13 stage 1.5 bit/stage ADC with mismatch parameters specified in Table 2.1. . . . .	57
3-1	Sample transient response of (a) an opamp-based and (b) a CBSC switched capacitor gain stage. . . . .	63
3-2	Sample input waveforms into a CBSC comparator. . . . .	64
3-3	Zero-crossing based switched capacitor pipelined ADC stage. . . . .	65
3-4	Sample transient response of a ZCBC switched capacitor gain stage. . . . .	66
3-5	Two stages of the 1.5 bit/stage zero-crossing based pipelined ADC. . . . .	67
3-6	Shoring switch implementation. . . . .	70
3-7	Shorting switch timing diagram. . . . .	70
3-8	Current source implementation. . . . .	71
3-9	The bit decision flip-flop phase generation circuit, including the voltage-control delay line implementation. . . . .	73
3-10	Die photo of 0.05mm <sup>2</sup> ADC in 0.18 $\mu$ m CMOS. . . . .	75
3-11	DNL and INL plots for 100MS/s and 200MS/s operation. . . . .	76
3-12	Measure frequency response to near Nyquist rate input tone. . . . .	77
3-13	Measured power consumption versus sampling frequency. . . . .	78
3-14	Simulated transient response used for noise analysis verification. . . . .	79
4-1	Traditional Chopper Stabilization for offset compensation . . . . .	92
4-2	Frequency domain view of Chopper Stabilization . . . . .	93
4-3	Block diagram manipulations with corresponding filter responses that all yield the same overall response. . . . .	94
4-4	Alternate chopping technique utilizing a Chopper Offset Estimation (COE) block. . . . .	94
4-5	Sample probability density functions of signals when chopping vector $p$ is a random Bernoulli vector. . . . .	97
4-6	Simulated offset estimate. . . . .	100
4-7	Frequency response of an ADC with second and third order distortion. Chopping disabled. . . . .	102

4-8	Frequency response of an ADC with second and third order distortion. Random chopping enabled. . . . .	102
4-9	Block diagram using Chopper Offset Estimation (COE) and Offset Controller (OC) blocks to null the ADC offset in the analog domain. .	104
4-10	Example charge-pump based input referred COE offset compensation implementation. . . . .	105
4-11	Block diagram of an $m$ stage pipelined ADC with identical COE offset correction distributed to each pipeline stage. . . . .	105
4-12	Block diagram of an $m$ stage pipelined ADC utilizing individual Chop- per Offset Estimate (COE) and Offset Control (OC) blocks for each stage for per-stage offset compensation. . . . .	106
4-13	Block diagram of an $m$ stage pipelined ADC utilizing multistage chop- ping vectors to estimate and null the offset of each stage individually.	108
5-1	Bonding Diagram of Second ZCBC Chip. . . . .	114
5-2	Fully differential implementation . . . . .	116
5-3	Fully differential timing diagram . . . . .	117
5-4	Large Signal Current Source . . . . .	120
5-5	Small Signal Current Source . . . . .	120
5-6	Power supply to output voltage transfer function from parameters ex- tracted via simulation. . . . .	121
5-7	Permanently disabled dummy current sources ( $I_{\text{dum}\pm}$ ) are added to provide symmetric parasitic capacitance for improved power supply noise rejection. . . . .	123
5-8	Differential zero crossing detector . . . . .	123
5-9	On-Chip Transient Reference Voltage Simulation Results . . . . .	127
5-10	Traditional implementation of voltage references for a 1.5 bit/stage pipeline stage. . . . .	128
5-11	Ideal stage voltage transfer function (left) and ADC transfer function (right) for a 1.5 bit/stage ADC. . . . .	130

5-12	Voltage transfer function (left) and ADC transfer function (right) for a 1.5 bit/stage ADC including series resistance mismatch for the voltage reference switches. . . . .	130
5-13	Alternative 1.5 bit/stage ZCBC implementation where $C_1$ has been split to eliminate the $V_{\text{refc}}$ voltage reference. . . . .	131
5-14	Analog multiplexer implementation . . . . .	132
5-15	Schematic of a $\log_2(n + 1)$ bit/stage ZCBC pipeline stage using capacitor splitting (only circuits active during the transfer phase have been included). Capacitor $C_1$ is split into $n$ equal parts. . . . .	133
5-16	Differential ZCBC showing series on-resistance of reference switches .	133
5-17	Typical residue plots without redundancy and with redundancy. . . .	135
5-18	Residue plots when using 2 bit decision comparators (1.58 bits/stage).	136
5-19	Residue Plots when using 3 bit decision comparators (2.0 bits/stage)	137
5-20	Residue Plots for gain $G = 4$ and number of bit decision comparators $n = 9$ . This yields a 3.3 bit/stage pipeline ADC with gain reduction.	139
5-21	Complete ZCBC fully differential pipeline stage. . . . .	140
5-22	First stage of ZCBC fully differential pipeline ADC. . . . .	141
5-23	Switch matrix implementation of input sampling circuit. . . . .	142
5-24	First stage sub-ADC implementation utilizing bottom plate sampling.	144
5-25	Resister string to generate nine sub-ADC reference voltages. . . . .	144
5-26	Sub-ADC implementation for all stages except the first. . . . .	145
5-27	Possible BDC implementations compared for offset, noise, and speed. BDC B is used in this design. . . . .	146
5-28	BDC comparison simulation results. . . . .	147
5-29	Bias current versus $\alpha$ . . . . .	155
5-30	Pre-amplifier Simulation Results . . . . .	160
5-31	Single Ramp Timing . . . . .	161
5-32	Dual Ramp Timing . . . . .	162
5-33	Die photo of fully differential ZCBC ADC in 90nm CMOS. . . . .	164
5-34	Measured Linearity . . . . .	165

5-35 Measured Frequency Response . . . . .	165
5-36 SNDR versus input amplitude . . . . .	166
5-37 Measured 1st stage programmable ZCD offset range. See Figure 5-8 for definition of $off_a$ and $off_b$ nets. . . . .	166
5-38 ADC noise sensitivity comparisons to I/O voltage and drive strength.	167
5-39 ADC noise sensitivity to I/O voltage for original single-ended ZCBC design described in Chapter 3. . . . .	167
5-40 Measured performance using BDC C and BDC B (see Figure 5-27). .	169
6-1 ZCBC implementation shown in the transfer phase utilizing propor- tional feedback control to the current source. . . . .	173
6-2 Virtual ground node dynamics for various ZCBC ramping schemes. .	173
6-3 Power Efficiency Comparison of Single-Ended Design . . . . .	177
6-4 Power Efficiency Comparison of Fully-Differential Design . . . . .	178



# List of Tables

2.1	Simulation mismatch parameters. . . . .	58
2.2	Simulation Results. . . . .	59
3.1	Summary of key DZCD quantities. . . . .	83
3.2	ADC Performance Summary . . . . .	87
5.1	ADC Performance Summary . . . . .	170



# Chapter 1

## Introduction

Cost reductions and performance improvements from transistor scaling continue to advance in the semiconductor industry at a rapid pace. Both digital and analog circuit design have benefited from the speed, power, cost, and area improvements associated with technology scaling. The advent of the nano-scale era, however, has brought with it the emergence of a many new issues for analog circuit design. Device leakage, mismatch, and modeling complexity are increasing while intrinsic device gain and voltage supplies are decreasing [2]. While historically the optimality of a technology node has served to improve both analog and digital circuits, the nano-scale era is beginning to see the divergence of an optimal technology node able to serve the needs of both digital and analog applications simultaneously [39].

For example, consider the trend analysis for published pipelined ADCs shown in Figure 1-1. In these plots, the blue dots represent the performance of individual ADCs extracted from publications\*. The red line is a plot of the median of the data for each technology node, and the black line is a plot of the trend line obtained from a linear regression of the data. Shown are three plots of sampling frequency, power consumption, and effective number of bits (ENOB) versus technology node. This data shows that ADCs are increasing in speed by a factor of 1.3x per process node and decreasing in power consumption by a factor of 1.5x per process node. Both of these are desirable trends and align with the trends of technology scaling in general.

---

\*ADC performance data provided by Brian Ginsburg of MIT.

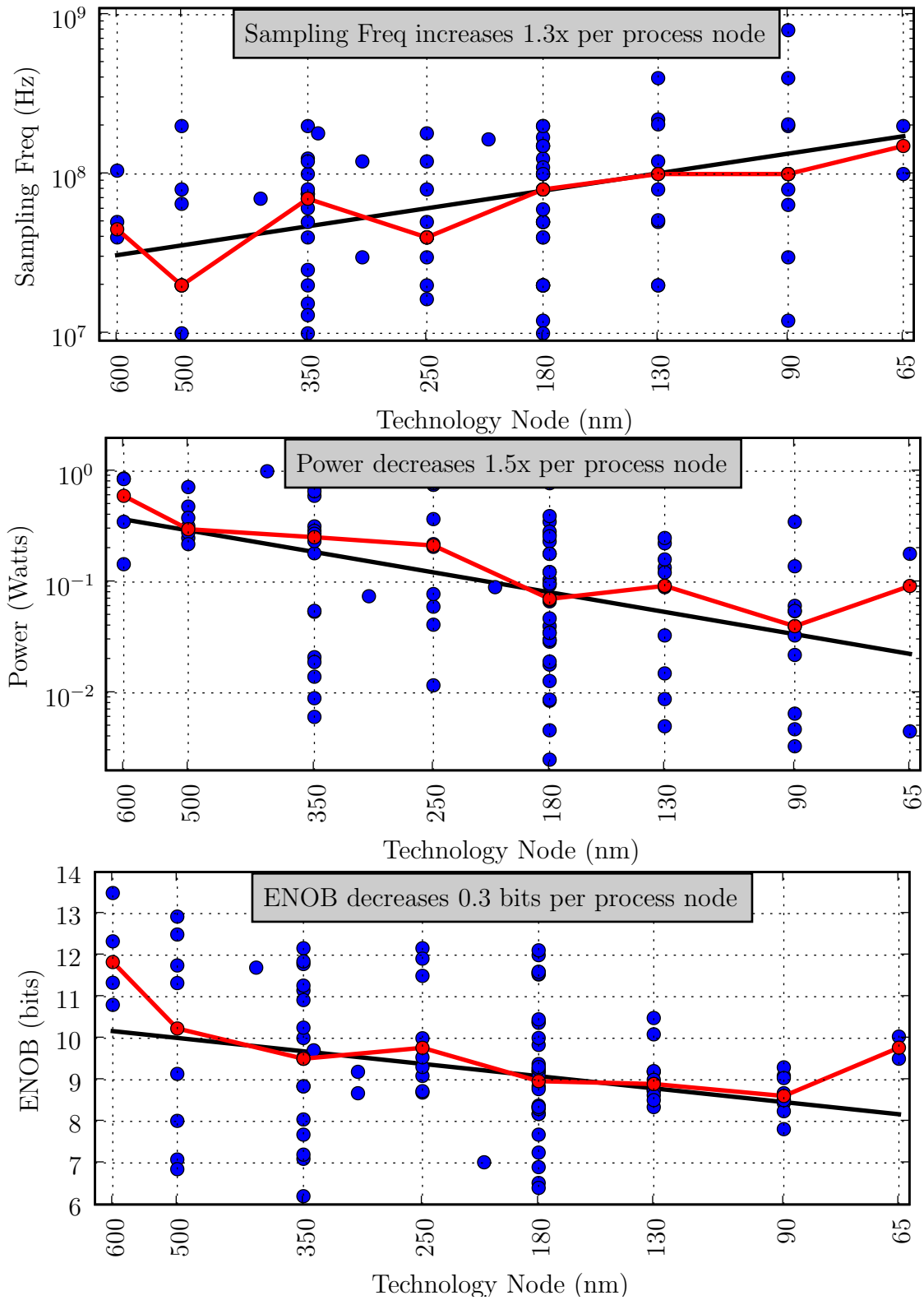


Figure 1-1: Trend analysis for published pipelined ADCs.

The disturbing trend, however, is that ADC resolution has been decreasing by 0.3 bits per process node. Furthermore, observe that below 130nm, no pipelined ADC with an effective resolution higher than 10 bits has been published.

This trend highlights one of the major issues analog designers are facing as technology scaling continues. Decreasing device gain and voltage supplies are increasing the difficulty of realizing high-gain amplifiers. In the case of switched capacitor circuit design, this translates into difficulty realizing a precision charge transfer via a high-gain, high-speed operational amplifier (opamp) in feedback. The methods of designing an opamp to maintain the necessary gain and bandwidth as device gain decreases are cascading and/or cascoding gain stages. Cascading gain stages introduces complexity and issues of stability versus bandwidth/power consumption [17]. Cascoding, on the other hand, exacerbates the issues of voltage supply scaling as it reduces available signal swing.

It has been speculated that because of these issues it will be both economically and technically impossible to implement high resolution circuits such as data converters in low-voltage, deeply scaled technologies and that the optimality of “System on Chip” (SoC) integration may be ending in favor of “System in Package” (SiP) solutions, where functionality from different die are assembled in a single package [39]. The issues associated with taking signals “off-chip,” however, severely limit this approach, especially at higher speeds and resolutions.

Another product of technology scaling has been the gradual transition of analog circuit implements to digital implementations. Digital implementations typically provide increases in flexibility, robustness, testability, scalability, and automated design capabilities. Because technology scaling is geared heavily toward optimizing digital circuit metrics, moving a digital design into a new process node will likely result in a lower power, faster, smaller, cheaper and all around better implementation with much less design effort than a corresponding analog circuit.

Since there is and always will be a need to interface digital circuits to the analog world, however, one area of analog circuit design that continues to thrive is that of mixed-signal data converters such as analog-to-digital converters (ADCs) and digital-

to-analog converters (DACs). ADCs, however, are rather power inefficient [30]. Analog circuit processing prior to the ADC is still common in many applications as a means of realizing more power efficient systems, and the widening gap between analog and digital circuit performance is not helping the cause of removing these blocks.

Therefore, the focus of this thesis is that of circuit techniques and architectures that not only deal with but also take advantage of technology scaling trends. Because pipelined ADCs perform well at high speeds and high resolutions, they are a popular architecture for a broad class of applications. For this reason, the principles and techniques developed in this thesis are specifically applied to pipelined ADCs. However, many of them can be applied on a broader level to other ADC architectures, switched capacitor circuits, and analog circuits in general. For example, the zero-crossing based circuits described in Chapters 3 and 5 can be applied to switched capacitor filters, DACs,  $\Delta$ - $\Sigma$  modulators and ADCs.

The innovations of this research can be broadly categorized into two different approaches. One is providing digital algorithms that can leverage scaling trends to ease the requirements of critical analog circuits. The other is developing new architectures of analog circuits that align better with the trends of scaling.

In Chapter 2, a digital estimation technique called Decision Boundary Gap Estimation (DBGE) is introduced as a method of digital calibration to static nonlinearities in pipelined ADCs. Calibration of such static non-linearities has been a very popular research topic and the ideas and methods demonstrated in [29, 44] have formed the basis for many techniques such as open-loop amplification [35], incomplete settling [26], and low-gain closed-loop amplification [22]. These all have the goal of reducing the requirements of the critical analog components by providing digital calibration circuits. DBGE is a very simple background calibration with many compelling advantages over other traditional approaches.

In Chapter 3, a new switched capacitor circuit architecture called Zero-Crossing Based Circuits (ZCBC) is introduced as a generalization of Comparator-Based Switched-Capacitor (CBSC) circuits [18]. This architecture replaces the function of an opamp with the combination of a zero-crossing detector and a current source to realize the

same functionality without an amplifier in feedback. With opamps completely eliminated from the design, there is no high-gain, high-speed feedback loop to stabilize. This not only reduces complexity but also eliminates the associated stability versus bandwidth/power trade off. Furthermore, such circuits are more power efficient [30,41] and provide design constraints that align better with scaling trends. The result is an 8 bit, 200MS/s pipelined ADC implemented in TSMC's 180nm CMOS technology node.

One of the major limitations of this initial ZCBC design was that it lacked a power efficient offset compensation required to make it production worthy. This spurred the developed of a digital offset estimation technique called Chopper Offset Estimation (COE) that is presented in Chapter 4. COE is based on Chopper Stabilization but significantly relaxes the filtering requirements and provides a method to null the offset in the analog domain to recover signal range lost due to the offset. Furthermore, it is compatible with a much broader class of circuits than traditional auto-zeroing techniques. Once again, because COE is based on digital estimation techniques, it aligns well with scaling trends and does not require significant power consumption as other auto-zeroing methods do.

One of the other major limitations of the initial ZCBC design was that it did not achieve its designed resolution due to suspected noise coupling paths from digital output drivers. In Chapter 5 zero-crossing based circuits are revisited with a second design whose goal was to demonstrate COE offset compensation and develop ZCBC circuits with improved noise rejection and a significantly higher resolution than the initial design. This second design a realized a fully differential 12 bit, 50MS/s pipelined ADC with COE offset compensation in IBM's 90nm CMOS technology node.

## 1.1 Pipelined ADC Overview

Because an understanding of the pipelined ADC and its critical design issues are a necessary foundation to this thesis, the remainder of this chapter provides a back-

ground of pipelined ADCs.

A pipelined ADC consists of lower resolution stages, as shown in Figure 1-2, concatenated together to form the desired resolution.  $N_j$  is the resolution of the sub-ADC in stage  $j$ . The input voltage is quantized to  $N_j$  bits to produce the bit decisions  $D_j$ . These bit decisions are then converted back into an analog voltage and subtracted from the input voltage to produce the quantization error. The quantization error is gained by  $2^{N_j}$  to produce the residue or output voltage  $v_O$ . Residue amplification basically takes the quantization error and maps it back to the full scale range of the next stage.

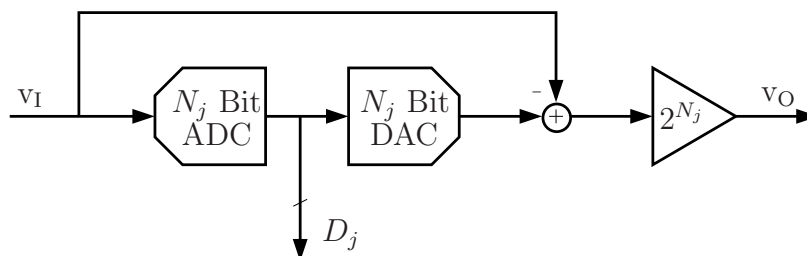


Figure 1-2: Block diagram of an  $N_j$  bit/stage pipeline stage.

To reconstruct the digital output code, one must digitally gain the bit decision  $D_j$  by  $2^{N_j}$  and add it to the bit decisions  $D_{j+1}$  of the next stage. That result must then be gained by  $2^{N_{j+1}}$  and added to the bit decisions  $D_{j+2}$  of the next stage. This continues until all  $B$  stages have been recombined. This can be expressed mathematically as:

$$\begin{aligned}
 x &= (((D_1 2^{N_1}) + D_2) 2^{N_2} + D_3) 2^{N_3} + \dots \\
 &= \sum_{i=1}^B D_i 2^{N_1 + N_2 + \dots + N_i}
 \end{aligned} \tag{1.1}$$

Typically each stage is implemented such that the residue gain  $2^{N_j}$  is a power of two so that during reconstruction, multiplying by  $2^{N_j}$  can be done with a simple bit shift. This reconstruction rule holds even when redundancy or over-range protection (see Sections 1.4 and in 5.4) is used.

When  $N_j = 1$  for all stages, then the pipelined ADC is called a 1.0 bit/stage ADC.



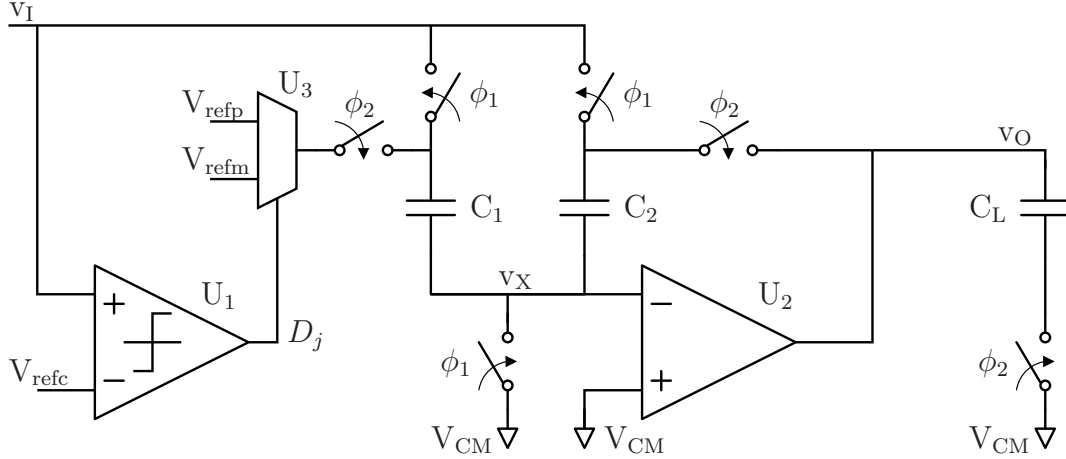


Figure 1-3: Typical circuit implementation of 1 bit/stage pipeline stage. Single-ended version shown for simplicity.

A typical opamp-based implementation of a 1.0 bit/stage pipeline stage is shown in Figure 1-3. The bit decision comparator (BDC)  $U_1$  makes up the sub-ADC and outputs the bit decision  $D_j$ . The two non-overlapping clocks  $\phi_1$  and  $\phi_2$  configure the circuit for two different functions. When  $\phi_1$  is high, the stage is configured in the *sampling* phase. During the sampling phase, the input voltage  $v_I$  is sampled with respect to the common mode voltage  $V_{CM}$  onto the sampling capacitors  $C_1$  and  $C_2$ . When  $\phi_2$  is high, the stage is configured in the *transfer* phase. The virtual ground node  $v_X$  becomes high impedance, and the output voltage  $v_O$  can be expressed as:

$$v_O = \frac{C_1 + C_2}{C_2}(v_I + v_X) - d \frac{C_1}{C_2} V_{ref}, \quad (1.2)$$

where  $V_{ref} = V_{refp} - V_{refm}$ ,  $d = 1$  when the comparator output  $D_j$  is high, and  $d = -1$  when  $D_j$  is low. Without a loss of generality, this result assumed  $V_{CM} = 0$  to simplify the math.

The analog multiplexer  $U_3$  implements the DAC and subtraction functionality to generate the quantization error, and the opamp  $U_2$  is used to force the virtual ground condition

$$v_X = V_{CM} \quad (1.3)$$

without removing or adding charge from it. When  $C_1 = C_2$  and when the virtual

ground condition is realized precisely, then the voltage  $v_O$  realized on the load capacitor (which will be the sampling capacitors of the next stage) can be expressed as

$$v_O = 2v_I - dV_{\text{ref}}. \quad (1.4)$$

This result is the ideal transfer function for a 1.0 bit/stage pipeline stage and is plotted in Figure 1-4. Also plotted in Figure 1-4 is the complete ADC transfer when digital output code of many such ideal pipeline stages are concatenated and reconstructed according to Equation 1.1.

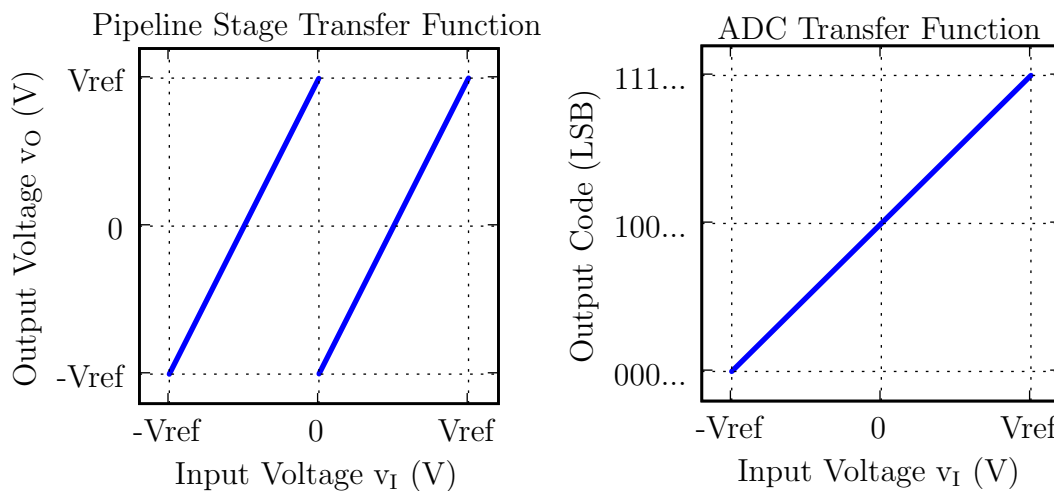


Figure 1-4: Ideal stage voltage transfer function (left) and ADC transfer function (right).

## 1.2 Comparator Based Switched Capacitor Circuits

An alternative to the opamp-based implementation is an architecture called Comparator Based Switched Capacitor (CBSC) circuits introduced in [18, 41]. This architecture replaces the opamp with a continuous time comparator and a current source as shown in Figure 1-5.

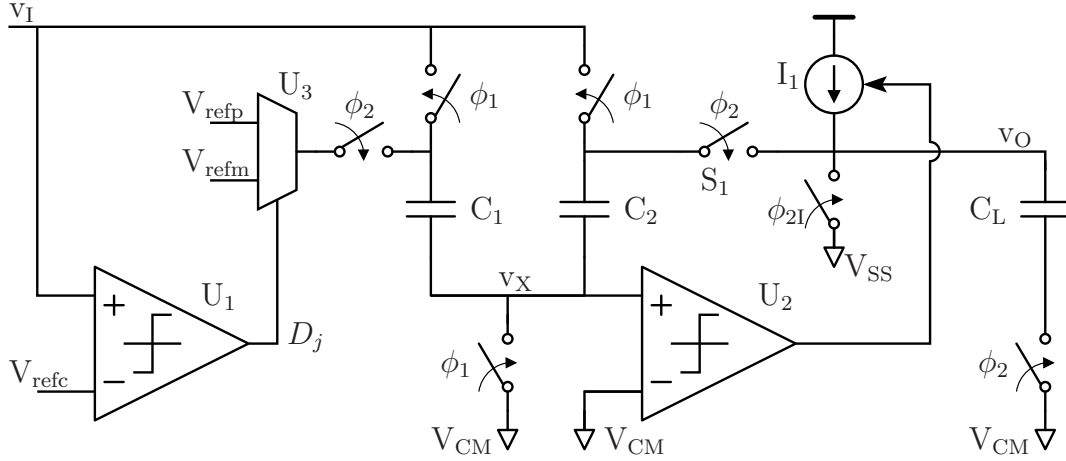


Figure 1-5: Implementation of a 1 bit/stage CBSC pipeline stage.

## 1.3 Pipelined ADC Error Models

There are many different circuit effects that can create static non-linearities in pipelined ADCs [4, 7, 29, 34]. Following is a discussion of the dominant sources.

### 1.3.1 Finite Opamp Gain

When an opamp-based architecture is used to realize the charge transfer in a pipelined ADC, there will be an error in the virtual ground condition due to the finite DC gain  $A$  of the opamp. This error can be expressed as

$$v_X = -\frac{v_O}{A}.$$

Substituting this into Equation 1.2 and solving for  $v_O$  when  $C_1=C_2$  yields

$$v_O = \frac{2v_I - dV_{\text{ref}}}{1 + \frac{2}{A}}. \quad (1.5)$$

This can be rewritten as

$$v_O = \frac{2v_I - dV_{\text{ref}}}{1 + \epsilon_{\text{op}}}, \quad (1.6)$$

where  $\epsilon_{\text{op}} = \frac{2}{A}$ . Finite opamp gain causes a gain reduction in pipeline stage transfer function as shown in the plot of Figure 1-6. The solid line represents the transfer

function with finite opamp gain and the dashed line is the ideal transfer function where the gain is exactly 2. In the second plot of Figure 1-6, the ADC transfer

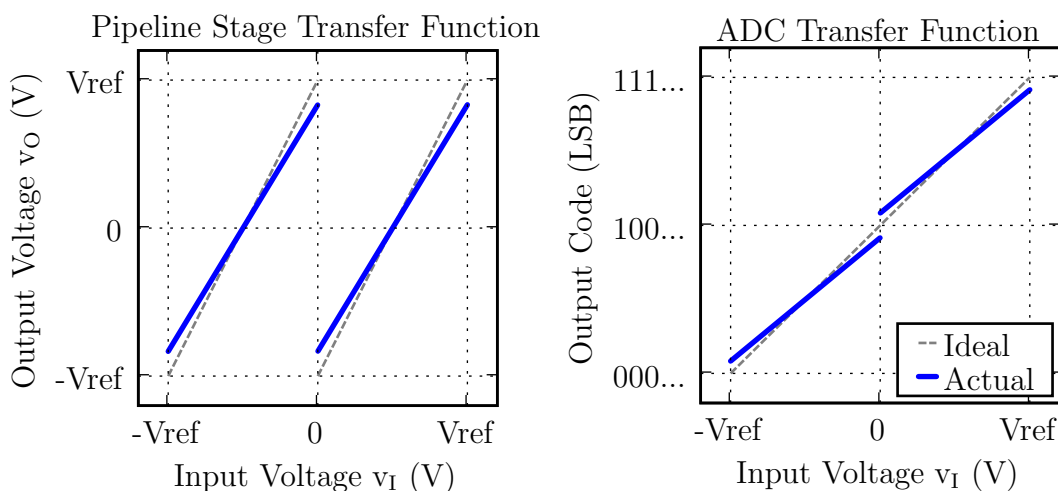


Figure 1-6: Single stage and ADC transfer function from finite op-amp gain or finite current source output impedance.

function is shown for the case of finite opamp gain only effecting the first stage. The result is a static non-linearity in the form of a missing code gap at the bit decision boundary. The amount of gain reduction and thus the size of the missing code gap is a function of the DC gain  $A$  of the opamp, and thus one must design the opamp with sufficient gain to meet the desired resolution.

### 1.3.2 Finite Current Source Output Impedance

When CBSC circuits are used to realize the charge transfer then the finite output impedance of the current source and the finite delay of the comparator will produce an effect that is very similar to finite gain in an opamp based circuit.

The output voltage ramp rate can be expressed as

$$\frac{dv_O}{dt} = \frac{I(v_O)}{C_T}, \quad (1.7)$$

where  $C_T$  is the total load capacitance of the current source ( $C_T = C_L + (C_2 \parallel C_1)$ ) and  $I(v_O)$  is the current provided by  $I_1$  when the output voltage is  $v_O$ . Suppose that the comparator has a finite delay  $t_d$ , then the voltage overshoot due to the finite

switching time of the comparator can be approximated as

$$\begin{aligned} v_{os} &= t_d \frac{dv_O}{dt} \\ &= t_d \frac{I(v_O)}{C_T}. \end{aligned} \quad (1.8)$$

If the output current source is modeled to first order as having an effective Early voltage of  $V_A$ , then the output current can be expressed as

$$I(v_O) = I_0 \left( 1 - \frac{v_O}{V_A} \right). \quad (1.9)$$

Substituting this result into Equation 1.8 gives

$$v_{os} = \frac{t_d I_0}{C_T} \left( 1 - \frac{v_O}{V_A} \right) \quad (1.10)$$

The baseline overshoot is the first term in this result and is  $\frac{t_d I_0}{C_T}$ . Since this baseline overshoot generates a constant offset that is not output voltage dependent, it can either be nulled using offset compensation techniques (see Chapter 4) or simply tolerated as it does not produce non-linearities at the output. The residual overshoot, however, is the second term in this result and is  $\frac{t_d I_0 v_O}{V_A C_T}$ . This is output voltage dependent and thus cannot be nulled by offset compensation and will produce a non-linearity at the output. Subtracting this term from the ideal voltage transfer function of Equation 1.4 and solving for  $v_O$  gives the residual error as:

$$v_O = \frac{2v_I - dV_{\text{ref}}}{1 + \frac{t_d I_0}{V_A C_T}} \quad (1.11)$$

By defining

$$\epsilon_{\text{zcd}} = \frac{t_d I_0}{V_A C_T} \quad (1.12)$$

then Equation 1.11 can be expressed as

$$v_O = \frac{2v_I - dV_{\text{ref}}}{1 + \epsilon_{\text{zcd}}} \quad (1.13)$$

Comparing this result with Equation 1.6 shows that the finite output impedance of the current source in a CBSC implementation produces a similar effect to that of finite opamp gain in an opamp-based circuit. The plots of Figure 1-6 also show how the finite output impedance in a CBSC implementation effects the residue amplification.

From Equation 1.12 a designer can see the parameters at his disposal to minimize this error. In an application where the overall speed of the ADC is specified, the baseline ramp rate of the output voltage, which is  $\frac{I_0}{C_T}$ , is fixed. This leaves the designer free to maximize the Early Voltage  $V_A$  of the current source and/or to minimize the comparator delay  $t_d$  in order to minimize the error  $\epsilon_{zcd}$ .

Equation 1.12 reveals that the overall speed of the ADC also effects the error caused by finite output impedance.  $I_0$  is the baseline current and needs changed proportionally with any change to the ADC speed.  $t_d$  is the delay of the comparator and may also be sensitive to the ramp rate, depending on the specific comparator architecture. For the case of the zero-crossing detector described in Chapter 3, the delay is inversely proportional to the cube root of the square of the ramp rate (see Equation 3.7). The net effect is that the error  $\epsilon_{zcd}$  will change by the cube root of the ramp rate. Thus, as one increases the speed of the ADC the overall linearity will get worse by a cube root factor. Compared to a zero-crossing detector used in the design described in Chapter 5, when the time-constant  $\tau$  of the system is fast enough compared to the sampling rate, the delay of the zero-crossing detector is fixed at  $t_d \approx \tau$ . In that case, the delay is independent of the ramp rate, so the linearity is inversely proportional ramp rate.

For both opamp based systems with finite gain and CBSC systems with finite output impedance the output voltage error percentage is  $\epsilon_{op}$  and  $\epsilon_{zcd}$  respectively. So for a 10 bit, 1.0 bit/stage pipelined ADC, the input referred error percentage ( $\epsilon/2$ ) would need to be less than 0.05% to yield an error less than 1/2 an LSB. For the opamp case, the gain would have to be  $A > 2000$ . For the CBSC case,  $\epsilon_{zcd} < 1000$ , so if the overshoot ( $t_d \frac{I_0}{C_T}$ ) is 100mV, then the Early voltage  $V_A$  must be greater than 100V.

### 1.3.3 Capacitor Mismatch

Capacitor mismatch results when the capacitor ratios deviate from their desired value due to variation inherent in manufacturing. Capacitor mismatch can cause both die-to-die variation from random etching variation and systematic variation from mask and structural density variation.

In the example shown in Figure 1-3, capacitor mismatch occurs when  $C_1$  and  $C_2$  are not equal. By defining the amount of capacitor mismatch as

$$\epsilon = \frac{C_1}{C_2} - 1,$$

the stage voltage transfer function becomes

$$v_O = (2 + \epsilon)v_I - (1 + \epsilon)dV_{\text{ref}}.$$

If  $\epsilon$  is negative, then a code gap results at the decision boundary of the digital output as depicted in Figure 1-7. If  $\epsilon$  is positive, then a duplicate or wide code region results in the digital output transfer function as shown in the example of Figure 1-8.

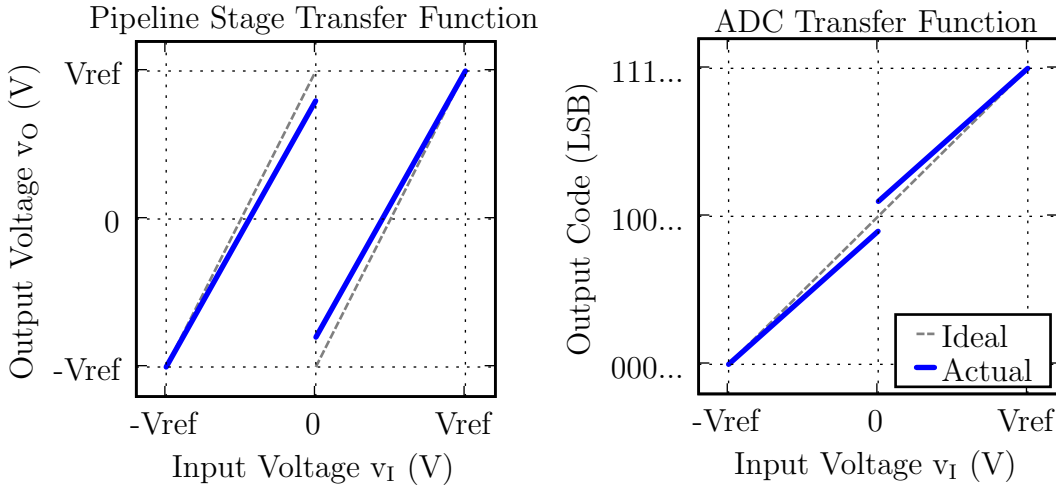


Figure 1-7: Single stage and ADC transfer function from capacitor mismatch when  $\epsilon < 0$ .

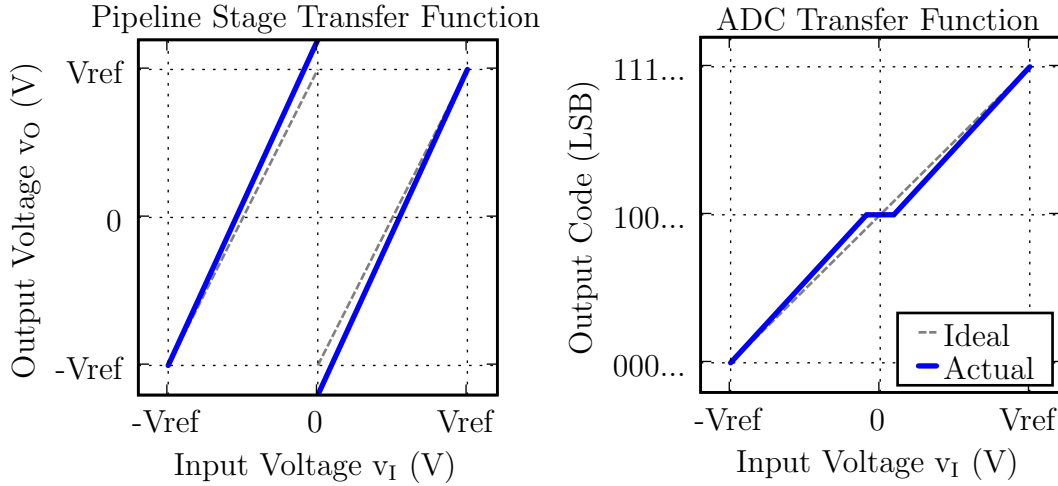


Figure 1-8: Single stage and ADC transfer function from capacitor mismatch when  $\epsilon > 0$ .

### 1.3.4 Charge Injection and Stage Offset

To the extent that any offset produced by charge injection or offset in the opamp (in opamp-based architectures) or comparator (in comparator-based architectures) is not signal dependent, any offset  $v_{os}$  in the residue amplification can be expressed as

$$v_O = 2v_I - dV_{ref} + v_{os}. \quad (1.14)$$

This result is plotted in Figure 1-9 for the case when  $v_{os}$  is positive, and the case when  $v_{os}$  is negative is plotted in Figure 1-10. Just like the case when the capacitor mismatch is positive, charge injection or stage offset causes a wide code at the decision boundary. The reason for this is that the output voltage goes out of range near the decision boundary.

### 1.3.5 Bit Decision Comparator Offset

When the bit decision comparator has positive offset, it produces the result plotted in Figure 1-11 and negative offset produces the results shown in Figure 1-12. Because the stage output voltage goes out of range, the ADC transfer function has a wide code and missing codes at the bit decision boundary.



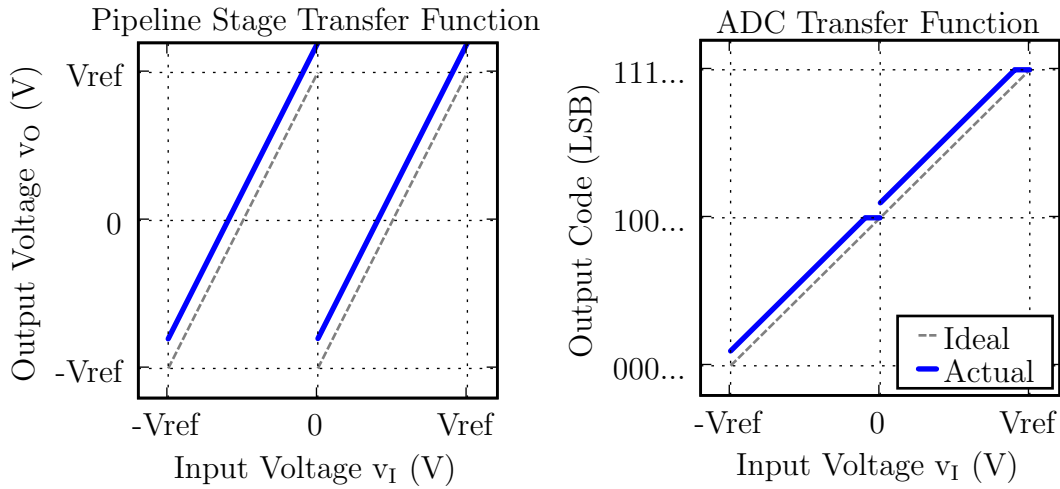


Figure 1-9: Single stage and ADC transfer function from positive charge injection or stage transfer offset.

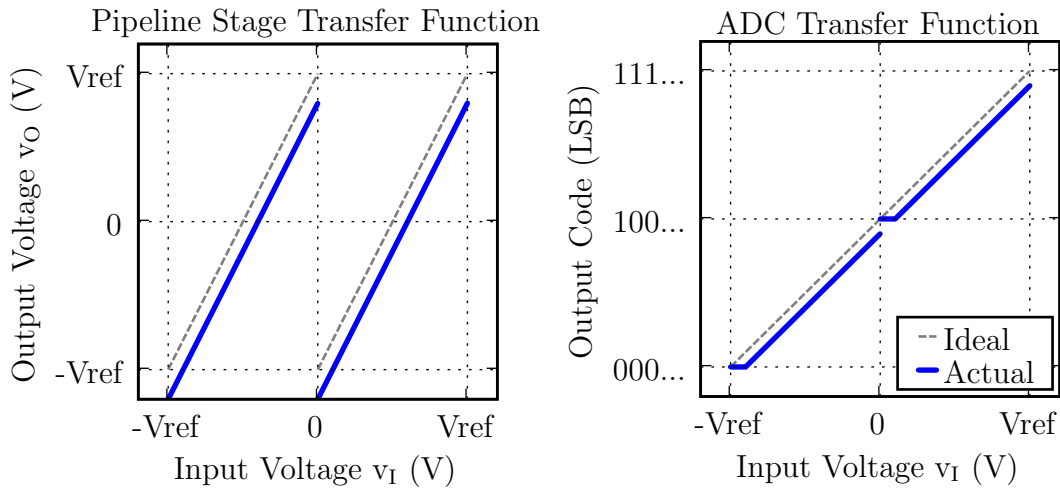


Figure 1-10: Single stage and ADC transfer function from negative charge injection or stage transfer offset.

### 1.3.6 Errors from Multiple Stages

The preceding examples showed the ADC transfer function when only the first stage had the static non-linearity and assumed the remaining stages were ideal. The effect of each additional stage, however, will also manifest itself as shown in the ADC transfer function example of Figure 1-13 where the first two stages are given the same low finite opamp gain. The missing code gap from the first stage is the largest and in the middle. The missing code gap from the second stage further divides each segment and produces a gap half the size of that from the first stage. The missing code gap

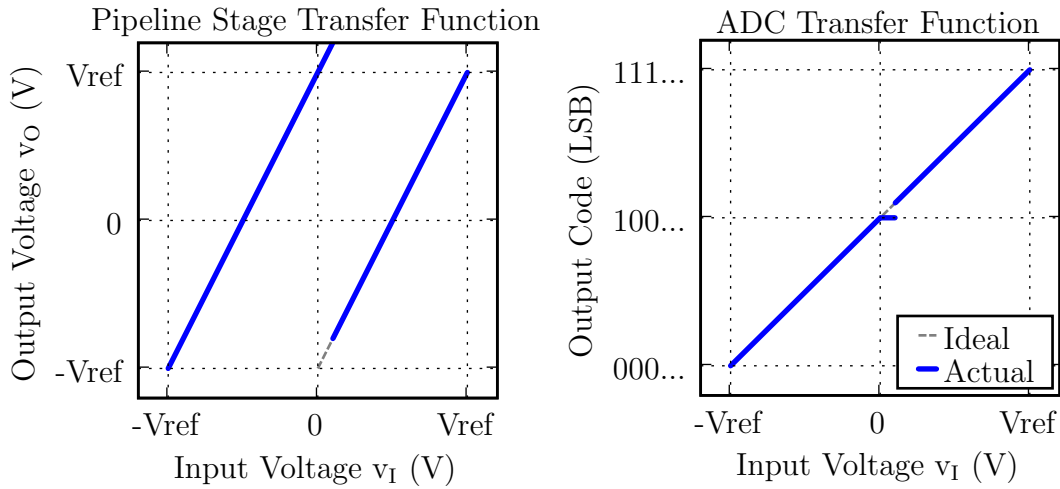


Figure 1-11: Single stage and ADC transfer function from a positive bit decision comparator offset.

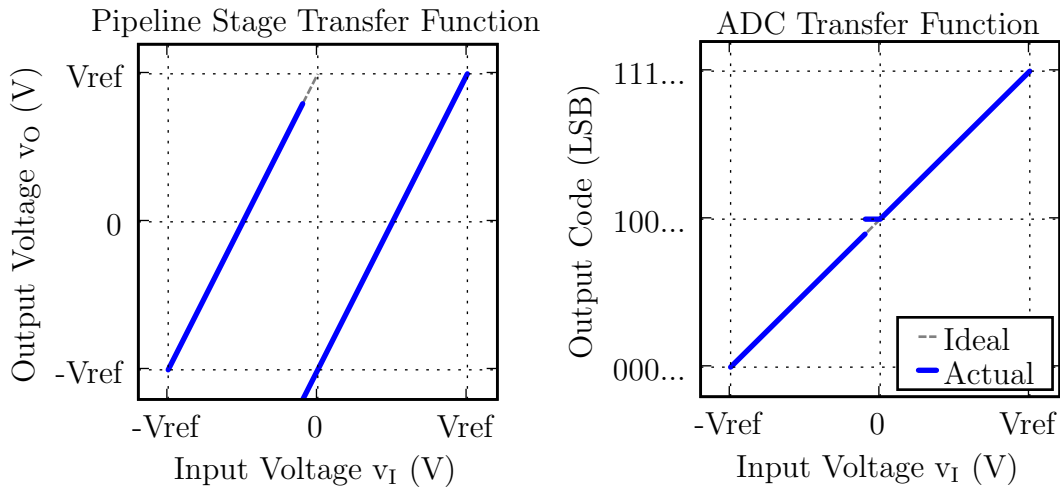


Figure 1-12: Single stage and ADC transfer function from a negative bit decision comparator offset.

from each additional stage will continue to be half that of the previous stage and further subdivide each segment.

## 1.4 Redundancy

When the output voltage goes out of range as in the examples in Figures 1-8 through 1-12, it produces a wide or duplicate code region. One significant development discussed in [31] is a method of over-range protection or redundancy to prevent wide codes.

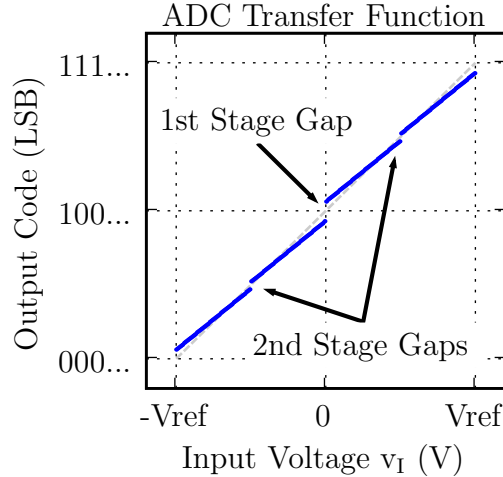


Figure 1-13: ADC transfer function when first 2 stages have finite opamp gain.

Figure 1-14 shows the block diagram of a pipeline stage with over-range protection. An  $M_j$  bit ADC and DAC, where  $M_j > N_j$ , are used instead of an  $N_j$  bit ADC and DAC.

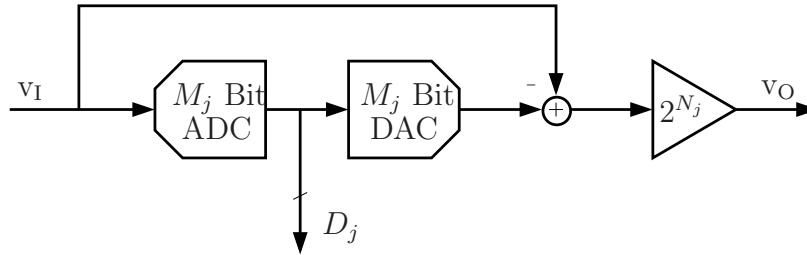


Figure 1-14: Block diagram of an  $M_j$  bit/stage pipeline stage. Over-range protection is offered when  $M_j > N_j$ .

In the simplest case,  $M_j = 1.5$  and  $N_j = 1$  for all stages. This is known as 1.5 bit/stage pipelined ADC. The bit decisions  $D_j$  for each stage are reconstructed in the same manner as before according to Equation 1.1, and in the ideal case, it produces a pipeline stage transfer function as shown in Figure 1-15.

In Figure 1-16 one can see that the over-range protection removes the wide code region in the middle of the ADC transfer function that plagues a 1.0 bit/stage with offset. The ADC transfer function does still have the input-referred offset, but this is not typically an issue for many ADC applications as the non-linearity of the wide-code region as been removed. Figure 1-17 shows that over-range protection also removes the

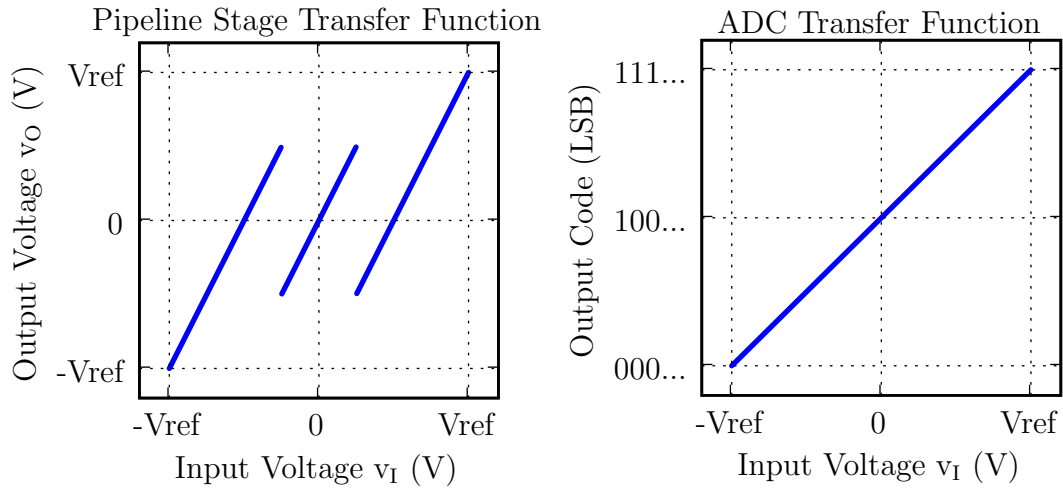


Figure 1-15: Ideal stage voltage transfer function (left) for a 1.5 bit/stage pipeline stage and resulting ADC transfer function (right).

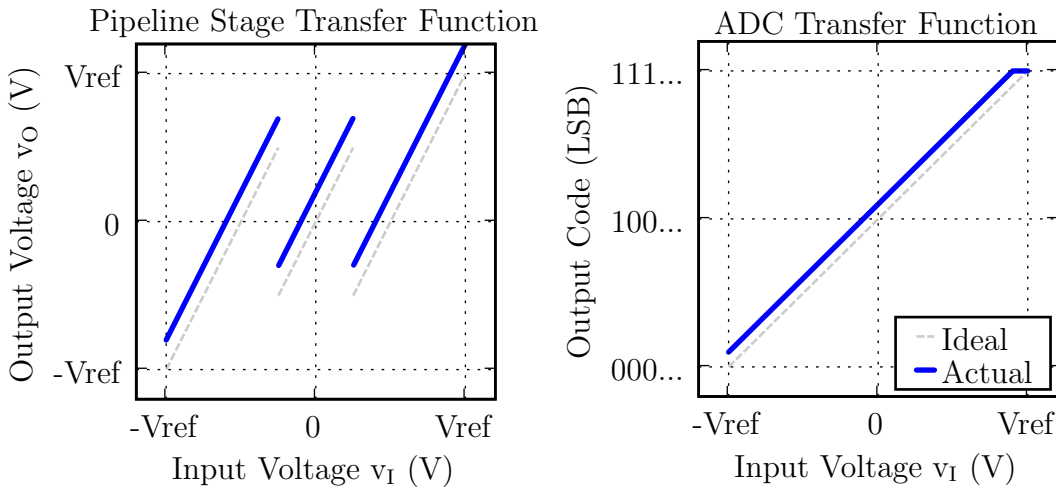


Figure 1-16: Single stage and ADC transfer function from positive charge injection or stage transfer offset.

wide code region completely with no remaining artifacts when bit decision comparator offset is present.

In Figure 1-18 one can see that over-range protection transforms wide codes into duplicate code regions. This introduces the possibility for the ADC transfer function to be non-monotonic. This may seem problematic for some applications, however, it enables digital calibration schemes that would otherwise not work if the non-linearity were a wide code. For example, Decision Boundary Gap Estimation is a digital calibration technique introduced in Chapter 2 that works by estimating the size of

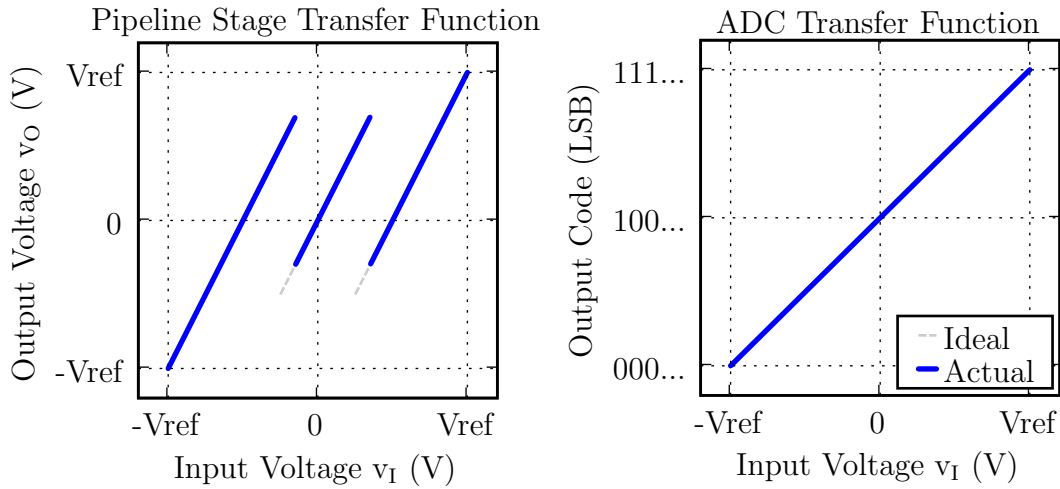


Figure 1-17: Single stage and ADC transfer function from a positive bit decision comparator offset.

the gaps that result at the decision boundaries and removing them by subtracting the gap away from the digital output codes. It cannot correct for wide codes because the information is lost, however, it can correct for duplicate or overlapping code regions.

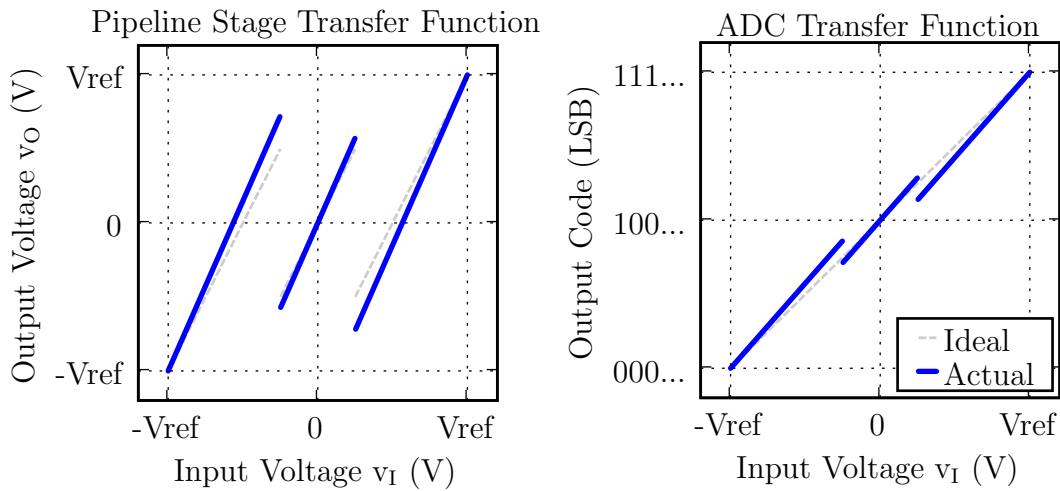


Figure 1-18: Single stage and ADC transfer function from capacitor mismatch when  $\epsilon > 0$ .



## Chapter 2

# Decision Boundary Gap Estimation

Since pipelined ADCs perform well at high speeds and moderate to high resolutions, they are a popular design choice and have been widely researched. In the absence of trimming or calibration, pipelined ADCs typically suffer from the static non-linearities described in Section 1.3 that limit the resolution to 8 to 10 bits [7, 29, 34].

These non-linearities have spurred many circuit and calibration techniques for realizing higher resolutions. Analog circuit techniques such as those in [33, 45] use analog components in the signal path to generate higher linearity at the expense of conversion speed. Digital calibration techniques, which realize the benefits of device scaling, have also been developed and can be categorized into foreground and background techniques.

Foreground calibration measures non-linearities during a calibration phase which usually occurs during startup. The method demonstrated in [29] measures the non-linearities by driving the bit decision boundary conditions during calibration to measure the non-linearities. Many other test-based or statistical-based methods have been developed that measure the non-linearities using code density or histogram measurements. For example, in [42], the reference voltages of the last pipeline stage are laser trimmed to produce ideal code densities. Likewise, in [9, 10, 19, 28], digital correction is performed based on foreground code density measurements of the non-linearities. Since these techniques use foreground calibration, they require interrupting normal ADC operation for calibration. To minimize the interruptions, the calibration phase

can be limited to manufacturing or ADC startup, but then calibration drift can result.

One class of background calibration measures circuit errors with calibration signals during hidden calibration time slots . A “skip-and-fill” approach is used in [45] where the input samples are interpolated during the hidden calibration phase. A queue-based approach is used in [5]. The drawback of these approaches is that they require redundant channels/stages that consume additional power and/or their overall accuracy is a function of the coverage of the calibration signal, which cannot follow the same path as the signal exactly.

Another popular background calibration approach, called Gain Error Correction (GEC) [22, 32, 35, 43, 44], additively injects an uncorrelated analog calibration signal into the ADC during normal operation. The known calibration signal is then subtracted from the ADC output and the calibration parameters are adjusted to null the correlation of the calibration signal to the corrected ADC output. Since the signal path must be able to accommodate the superposition of the input and the calibration signal, these techniques either reduce the available signal range or over-range protection of the ADC. Furthermore, its accuracy is tied to accuracy of the injected analog calibration signal.

Indirect methods of background calibration overcome the calibration signal coverage and accuracy issues by estimating the errors from the input signal itself without the use of calibration signals. In [7, 46] the dominant non-linearities of pipelined ADCs are modeled and corrected using adaptive equalization techniques prevalent in digital communications. It requires an additional “slow-but-accurate” ADC for reference to estimate and correct the errors. In [15] they note that when an input signal has a low-pass input histogram, the non-linearities of the ADC will generate high-pass components in the output histogram. Thus they collect an output histogram, low-pass filter it, and generate a correction map from the raw histogram space into the smoothed histogram space. In [14] they also use code densities or histograms with a second ADC to generate a correction map. These techniques are to varying degrees either algorithmically or hardware intensive.

Indirect calibration requires making assumptions about the input signal and pos-



sibly the errors themselves. For example, [15] assumes the input signal distribution is low-pass. The technique presented here is called Decision Boundary Gap Estimation (DBGE) for indirect digital background calibration. DBGE removes the dominant non-linearities of pipelined ADCs that appear as code gaps at decision boundaries. DBGE, therefore, models these gaps and relies on the input signal to exercise the codes in the neighborhood of these gaps to estimate and remove them. Much like the test-based or statistical-based methods, this technique estimates the non-linearities using code-density measurements. The estimation techniques, however, only require code-densities measurements in the regions surrounding the bit decisions of each stage and have been developed to run continuously in the background using the input signal itself as the stimulus rather than calibration signals.

The calibration procedure of DBGE can be broken into two steps. The first is an *estimation* phase where the digital output of the ADC is used to estimate the size of the missing code gaps for each stage. The second step is a *correction* phase where the gaps are digitally removed from the raw samples. The correction technique is described first in Section 2.1 under the assumption that accurate gap estimates have been measured. Then in Section 2.2 the gap estimation techniques of DBGE are described. Finally, in Section 2.3 simulation results are provided to show the effective performance of DBGE.

## 2.1 Gap Correction

The resolution of a pipelined ADC is set by the number of bits per stage and the number of stages. Suppose that an ADC with  $B$  stages is limited in resolution such that the first  $k$  stages need calibrated due to any number of the circuit issues described in Section 1.3. This means that the last  $B - k$  stages produce a linear output that does not contain any missing code gaps.

Calibration starts with stage  $k$ . The block diagram of Fig. 2-1 shows the calibration procedure. When stage  $k$  produces a bit decision output  $D_k$ , it is combined with the reconstructed output (see Equation 1.1) of the later stages to produce the

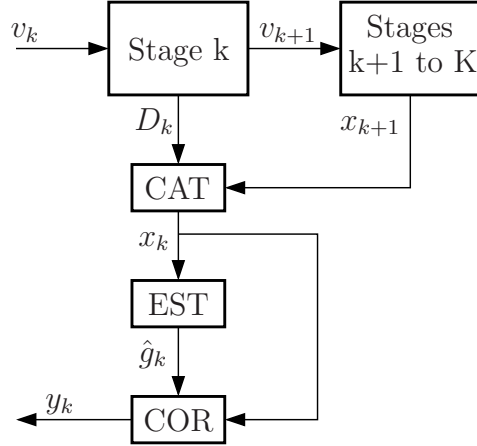


Figure 2-1: Block diagram of correction scheme for a single stage.

raw sample  $x_k$ .  $x_k$  is passed to the estimator to produce an estimate of the gap size. Assuming the estimator produces a good estimate  $\hat{g}_k$  of the gap size, then the non-linearity is removed from  $x_k$  by subtracting  $\hat{g}_k$  from all samples above the gap. Expressed mathematically, the linearized or corrected sample  $y_k$  is

$$y_k = \begin{cases} x_k, & \text{when } D_k = 0 \\ x_k - \hat{g}_k, & \text{when } D_k = 1 \end{cases} \quad (2.1)$$

An example of a raw and corrected ADC transfer function is plotted in Fig. 2-2. The dashed line represents the raw data and contains a missing code gap at bit decision boundary of the first stage. The solid line shows the corrected response. Observe that the gap or non-linearity has been removed but that the transfer function does not completely match the ideal response. In fact the resulting response has a residual offset and gain error. This residual offset and gain error is not an issue for many ADC applications as they do not cause any non-linear effects. However, for some applications, such as time-interleaved ADCs, an unknown offset and gain is not tolerable and will need further correcting with other techniques such as those presented in [12].

After correction, sample  $y_k$  is free of the non-linearity that was limiting the overall resolution, and the preceding stage  $k - 1$  can then be corrected in the same manner as stage  $k$  by using the corrected sample  $y_k$ . This will produce the corrected sample

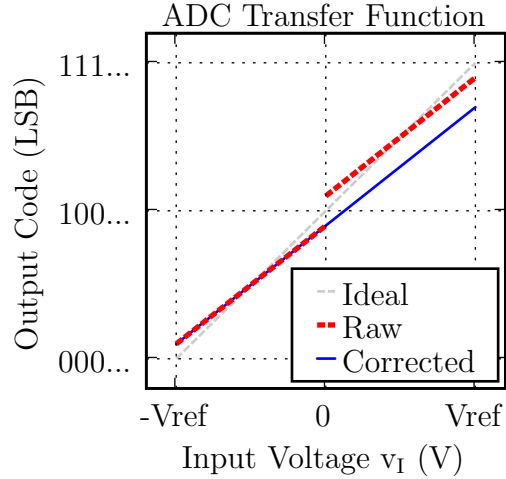


Figure 2-2: Transfer function of raw and corrected samples.

$y_{k-1}$  which can then be used by stage  $k - 2$ . A block diagram depicting this scheme of successive stage calibration is shown in Fig. 2-3.

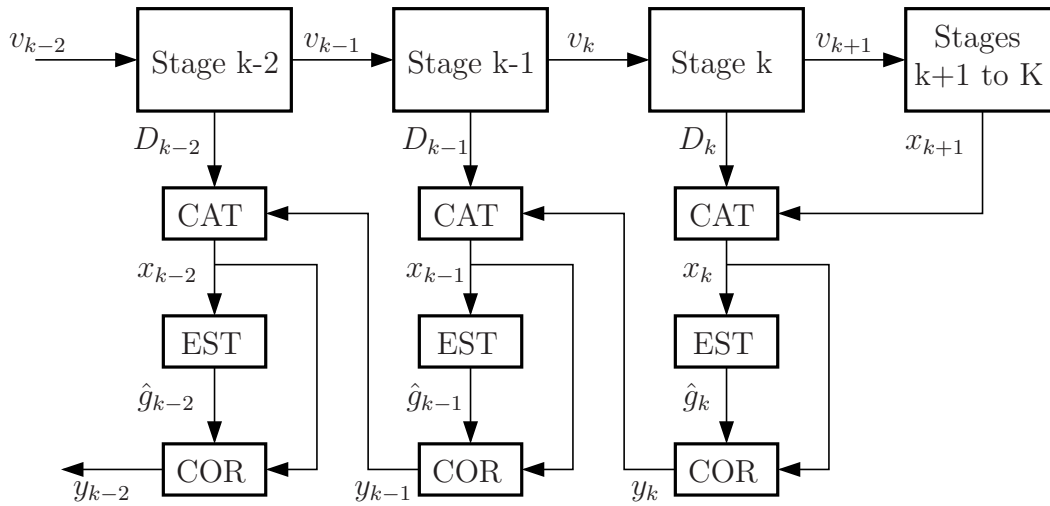


Figure 2-3: Block diagram of concatenated stages utilizing DBGE.

One can use this correction scheme for as many stages as necessary. If bit decision gaps were the only non-linearity in the ADC implementation, then this procedure could be used to achieve any arbitrary resolution. In practice, however, eventually other sources of non-linearity, such as signal dependent charge-injection, non-linear sampling capacitors, or non-constant opamp gain, will at some point become dominant and limit the linearity of the ADC.

This correction scheme has been demonstrated previously in [29]. There a sub-radix-2 pipelined ADC was designed and the gap was measured directly during a foreground calibration phase by driving the decision boundary voltage into each stage. This technique works well as demonstrated by the 15 bit ADC. The drawback is that foreground calibration requires taking the ADC out of service for calibration. Thus it suffers from calibration drift and/or service interruptions.

DBGE uses this same correction scheme with the slight extension that if redundancy is used then the stage radix does not need reduced. Redundancy prevents the signal from going out of range and thus allows the code gap  $g_k$  to be negative. Without redundancy the digital code gap gets clamped to be positive.

## 2.2 Gap Estimation

DBGE differs from the work presented in [29] in the gap estimation method. DBGE is an indirect background calibration technique and relies on the statistics of the input signal to estimate the code gap of each stage. The static non-linearities described in Section 1.3 cause the code gaps and can be modelled by the signal flow graph of Figure 2-4. Here the analog input voltage  $v_k$  into stage  $k$  is corrupted with an unknown, nonrandom parameter  $e_1$  or  $e_0$  when the MSB decision  $D_k$  is 1 or 0 respectively. The resulting analog voltage is then quantized by the remaining stages of the ADC, and the output  $x_k$  is the raw output sample and the observation variable. This model initially neglects the effect of circuit noise which will be considered later.

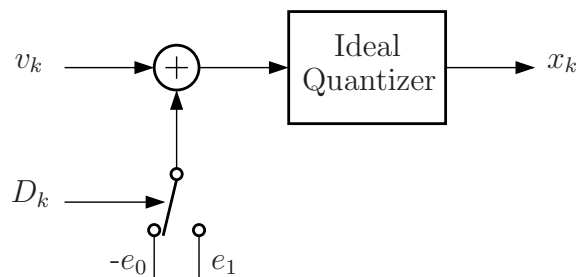


Figure 2-4: Signal flow graph modelling the code gap of stage  $k$  of a 1 bit/stage pipelined ADC.

Figure 2-5 shows an example of a histogram collected when the first stage has code gaps of  $e_0 = 4$  and  $e_1 = 5$  and when the input voltage  $v_k$  is uniformly distributed in a region near the bit decision boundary. Observe that no codes appear in the histogram within the region of the code gap.

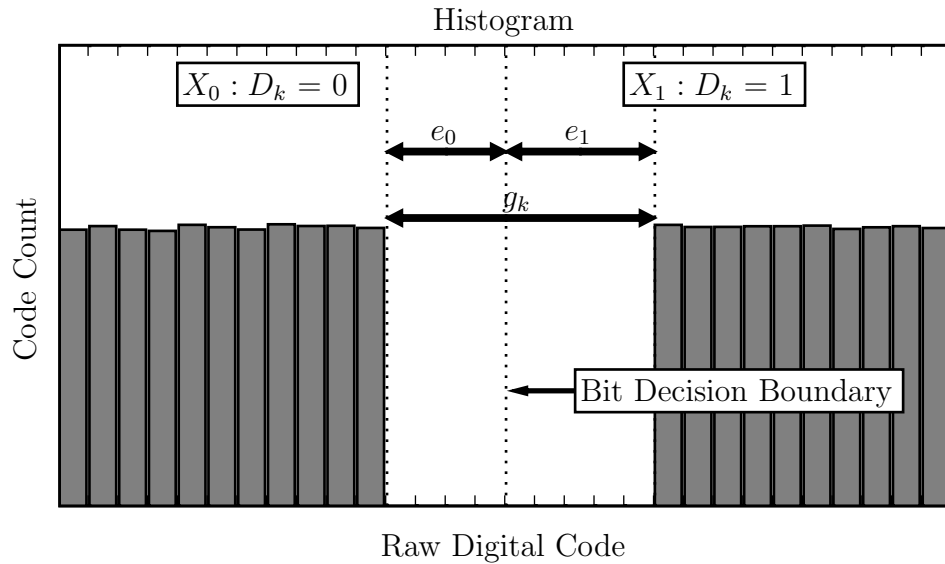


Figure 2-5: Histogram of an example data set (in the absence of noise) corrupted by unknown offsets.

The goal of DBGE is to estimate the gap size  $g_k$ , where  $g_k = e_1 + e_0$ . Although the example of Figure 2-5 uses parameters  $e_1$  and  $e_0$  that are integers, in reality they are not likely integers. Since DBGE corrects the digital output and not the source of the non-linearity, there is little advantage to estimating or correcting the gap size to a finer precision than an integer. Initially consider the case when the error parameters are integers is considered and more realistic parameters are considered in the simulation results presented in Section 2.3.

Following are several different gap estimation techniques of varying performance, hardware complexity, and robustness to circuit noise. For simplicity they are all described for the case of a 1 bit/stage ADC where each stage has a single code gap. These techniques, however, are general to higher resolution stages where each additional bit decision comparator produces an additional gap. For example, since a 1.5 bit/stage ADC contains 2 bit decision comparators, there will be two bit decision

boundaries and thus two independent code gaps that need estimated and corrected separately.

### 2.2.1 Max-Min Gap Estimator

The Max-Min gap estimator utilizes a very simple algorithm for estimating the code gap. Receive a block of  $N$  samples. Split it into two sets  $X_1$  and  $X_0$  where  $X_1$  is the set of all samples with an MSB  $D_k = 1$  and  $X_0$  is the set of all samples with  $D_k = 0$ . Estimate the gap  $\hat{g}_{mm}$  as

$$\begin{aligned}\tilde{e}_1 &= \min\{X_1\} \\ \tilde{e}_0 &= \max\{X_0\} \\ \hat{g}_{mm} &= \tilde{e}_1 + \tilde{e}_0.\end{aligned}\tag{2.2}$$

In other words, the Max-Min estimator watches the data stream to find the maximum sample received below the decision boundary and minimum sample received above the decision boundary and subtracts the two to form the estimate  $\hat{g}_{mm}$ . Once corrected, the effect on the histogram will be to shift the bins on the right side of the code gap to the left to close the gap and remove the non-linearity. Depending on the probability distribution of input voltage  $v_k$ , this estimate has varying degrees of performance. Whenever the probability distribution of  $v_k$  peaks or shares a peak at the decision boundary (which is midscale for a 1 bit/stage ADC), then this estimate is a Maximum-Likelihood (ML) estimate. Qualitatively, the more likely the input signal is to exercise the codes at the decision boundary, the better this estimation performs and vice versa. This is a desirable trend given that the impact of the non-linearity is a function of the code density of the input near the non-linearity. Furthermore, if the input signal has finite probability to be within one LSB of the decision boundary, then it can be shown that as the number of samples approaches infinity, the bias of this estimate approaches 0. How quickly it converges depends on the probability density in the region of the decision boundary.

The Max-Min estimator has a very efficient implementation in either hardware or

software. A hardware implementation requires 2 registers for storing the minimum  $\tilde{e}_1$  and maximum  $\tilde{e}_0$  estimates and comparison logic to determine when to update these registers. Estimation proceeds as each sample is received. First the bit decision  $D_k$  is checked. If it is 1, then the sample is compared against the minimum register and the minimum is updated if necessary. If  $D_k$  is 0, then the maximum register is compared and updated if necessary. To track changes in the gap that result from environmental changes, the minimum and maximum registers can be reset at a rate that matches the desired adaptation rate.

The Max-Min gap estimate provided in Eq. 2.2 suffers from a problem when one includes the effects of additive circuit noise in the analog processing path. Fig. 2-6 shows the addition of circuit noise to the signal flow graph as a random sample  $w_k$ . It has the effect of smearing the sharp edges of histogram at the code gap of the raw output samples. This can be seen in the example of Fig. 2-7 where Gaussian circuit noise with a standard deviation of  $\sigma_w = 1.0$  LSBs is added to the signal.

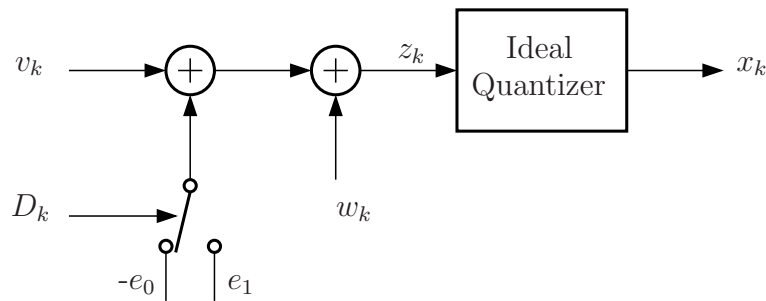


Figure 2-6: Signal flow graph of error model including circuit noise  $w_k$ .

With the additive noise smearing the sharp edges of the histogram, the Max-Min estimator will under compensate for the actual gap because the noise smears samples into the missing code region. The example histogram of Figure 2-7 shows how samples at the edge of the histogram have spilled into the missing code region and that the minimum and maximum samples according to Equation 2.2 no longer yield the correct estimate. Therefore, one must ensure that the circuit noise is lower than the quantization noise to minimize the bias that results on the gap estimate when using the Max-Min estimator. In ADCs where circuit noise is not lower than

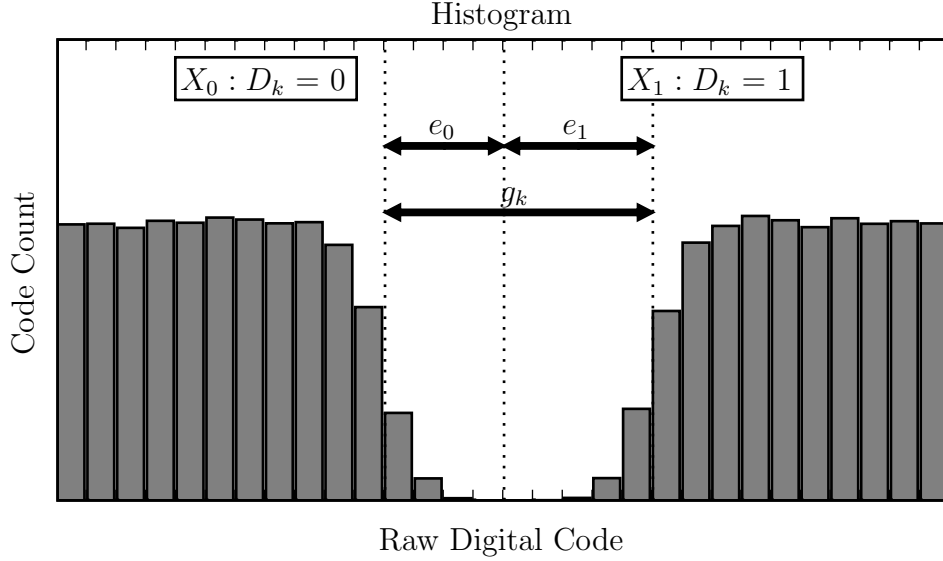


Figure 2-7: Histogram of an example data set corrupted by a code gap and additive circuit noise.

quantization noise, the Max-Min estimator does not likely perform adequately.

### 2.2.2 Bin-Reshaping Gap Estimator

An additional compensation calculation can be employed to improve the performance of the Max-Min estimator. This technique is called the Bin-Reshaping gap estimator. Consider the case when there is no circuit noise and  $e_0 = 3.5$ . A sample histogram of such a case is shown in Figure 2-8 for the case of a uniformly distributed input in the region of the bit decision boundary. The error parameter  $e_0$  causes the input to only span half of the right-most bin of set  $X_0$ , which will only fill half as much as its neighbor. The ratio of these two bins tells the fractional part of the error parameters  $e_0$ .

The basic concept behind Bin-Reshaping is to first quantize the input data to yield a coarse histogram where quantization noise is larger than the circuit noise. Although this meets the noise requirement of the Max-Min gap estimator, the Max-Min gap estimate will be of lower resolution and thus of limited effectiveness. However, one can extract the fractional part of this lower resolution estimate by taking the ratio of adjacent bins and interpolate back to the original resolution.



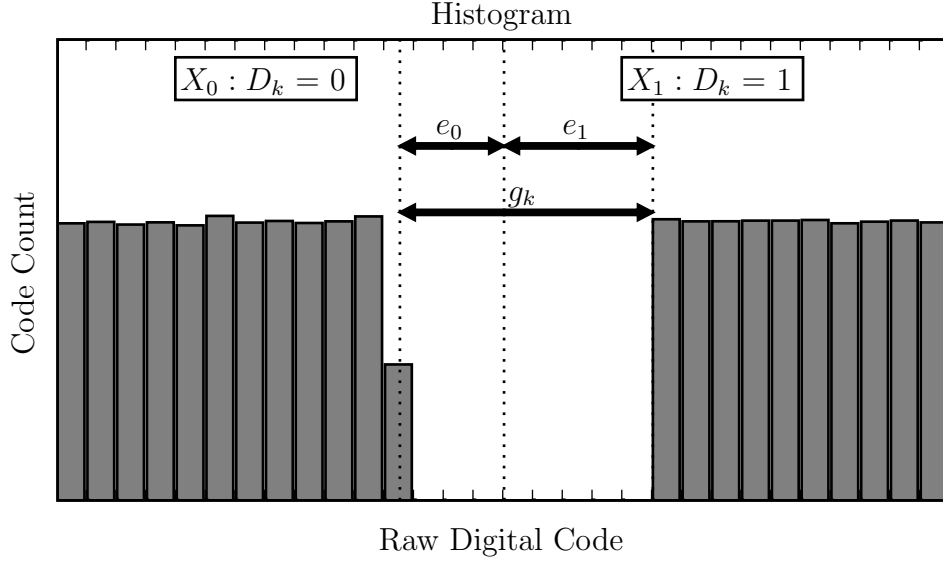


Figure 2-8: Example of a histogram resulting from a uniformly distributed input when gap size is not an integer.

Geometrically this technique reshapes the inner most histogram bins as shown in the example in 2-9. Here the high-resolution histogram of Figure 2-7 is quantized by merging adjacent bins. This can be done by simply dropping the noisy bits prior to binning or by summing  $s$  adjacent bins of the high resolution histogram to produce a lower resolution histogram. Expressed mathematically, this is

$$h_s[m] = \sum_{i=m}^{m+s-1} h[i],$$

where  $h_s[m]$  and  $h[i]$  are the bin counts of the lower and higher resolution histogram respectively. The bins labelled  $A_0$ ,  $A_1$ ,  $B_0$ , and  $B_1$  in Figure 2-9 make up the low resolution histogram.

The second step is to interpolate the value of the error parameters  $e_0$  and  $e_1$  across the two edge bins. Consider the case of estimating  $e_1$ . The bins labels  $A_1$  and  $B_1$  make up the two edge bins. Bin  $A'_1$  is created from bin  $A_1$  by reshaping it to the same height as  $B_1$  while preserving the area. The width of  $A'_1$  is taken as the effective minimum sample and thus the edge of the missing code gap. A similar procedure on bins  $A_0$  and  $B_0$  and can be used to find the effective maximum sample and thus the

other edge of the missing code gap. The Bin-Reshaping gap estimate  $\hat{g}_{br}$  is expressed mathematically as

$$\begin{aligned}\hat{e}_1 &= \tilde{e}_1 + s \left( 1 - \frac{h_s[\tilde{e}_1]}{h_s[\tilde{e}_1 + s]} \right) \\ \hat{e}_0 &= \tilde{e}_0 + s \left( 1 - \frac{h_{-s}[\tilde{e}_0]}{h_{-s}[\tilde{e}_0 - s]} \right) \\ \hat{g}_{br} &= \hat{e}_1 + \hat{e}_0,\end{aligned}\tag{2.3}$$

where  $\tilde{e}_1$  and  $\tilde{e}_0$  are the Max-Min estimates from the same data set.

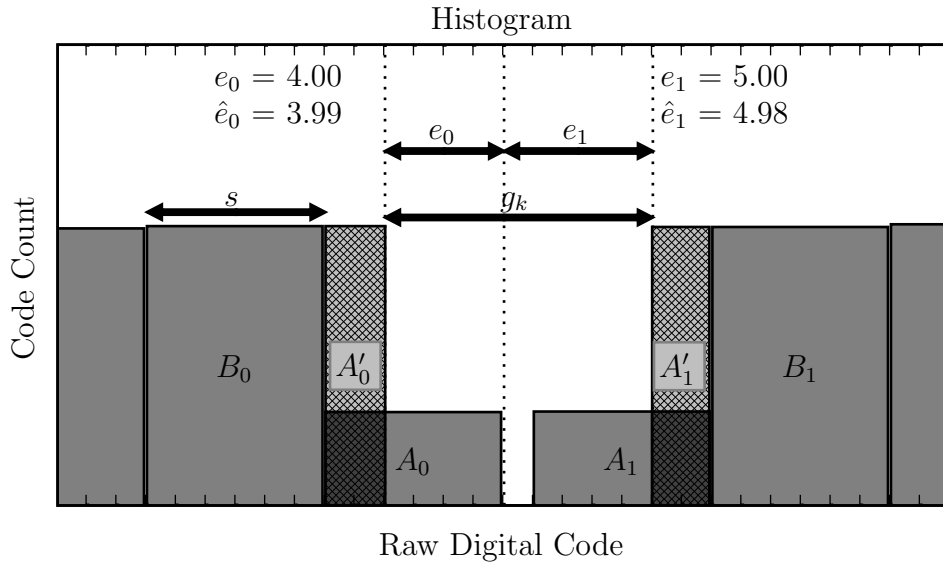


Figure 2-9: Histogram showing geometric interpretation of the Bin-Reshaping estimation method.

If  $s$ , the number of histogram bins to merge, is not picked large enough to adequately cover the spread in the histogram caused by the circuit noise, then the estimate will continue to under compensate. Thus  $s$  should be selected large enough to span the circuit noise to within good engineering tolerances (e.g.  $s > 3\sigma_w$ ). However, since the Bin-Reshaping gap estimator makes the approximation that the input is uniformly distributed over a width of  $2s$  codes,  $s$  should be chosen as small as possible. In practice  $s$  should be selected after characterizing the amount of circuit noise. In the example of Fig. 2-9, an extremely conservative choice of  $s = 6\sigma_w$  is used.

The Bin-Reshaping gap estimator makes the approximation that the input voltage

is uniformly distributed across the two inner-most bins on each side of the code gap region. This approximation is reasonable for many applications, especially high resolution ADCs, and is similar in nature to the approximation used when modelling quantization noise as uniformly distributed.

The Bin-Reshaping gap estimator is still very computationally friendly. Each estimate  $\hat{e}_0$  and  $\hat{e}_1$  requires an additional two registers for accumulating two lower resolution histogram bins. A division of these two registers must be performed, but since the estimate will be running at a very slow rate compared to that of the ADC, it can be implemented serially using shifts and subtractions for minimal gate count.

### 2.2.3 Cost-Minimizing Estimator

The traditional manner in which ADC linearity is characterized using code density measurements [13, 25] provides the inspiration for another more flexible gap estimator. Code density methods calculate the differential non-linearity (DNL) and integral non-linearity (INL) of the ADC by comparing the histogram or code density of the measured response to the theoretical response. When the ADC is stimulated with a uniformly distributed input, then a perfectly linear ADC will produce a histogram with uniform bin counts or code densities. Any non-linearities in the ADC will produce nonuniform bin counts as seen in the example histograms of Figure 2-7. From the bin counts, the DNL is derived from the ratio of adjacent bins and the INL is the cumulative sum of the DNL.

The Cost-Minimizing gap estimator takes an iterative approach to estimating an optimal code gap based on a predetermined cost function run on the histogram response of the ADC in the window of the bit decision boundary. The algorithm is as follows:

1. Receive a block of data from ADC.
2. Divide data into two sets.  $X_0$  is the set where  $D_k = 0$  and  $X_1$  is the set where  $D_k = 1$ .
3. Calculate the histogram of each set.

4. Select an initial gap estimate.
5. Shift the  $X_1$  histogram to the left by the gap estimate amount and add it to the  $X_0$  histogram. This combined histogram is equivalent to the histogram that would result if one corrected the samples with the selected gap estimate.
6. Evaluate the cost function on the combined histogram.
7. Increment the gap estimate and return to step 5. After sweeping the gap estimate over the desired range, select the gap estimate  $\hat{g}_{cm}$  that minimizes the cost function and stop.

The plots of Fig. 2-10 show the histogram manipulations of this procedure for 3 different gap estimates. This example corresponds to the original data set displayed previously in Fig. 2-7 where circuit noise was introduced into the simulation. The actual gap used in this example is 9 LSBs. In the first plot, a gap estimate of  $\hat{g}_{cm} = 8$  LSBs is selected. The histogram of the  $X_0$  is shown as the line marked with circles. The histogram from set  $X_1$  is shown as the line marked with triangles. This histogram get shifted to the left by 8 LSBs and added to the  $X_0$  histogram to produce the grey shaded histogram. For this example, the cost function is selected as the root mean square (RMS) of the DNL over an  $8\sigma$  circuit noise window where the two sets overlap at the bit decision boundary. The samples used in the DNL calculation of this example are marked with squares. Observe the dip in the histogram for this gap estimate. In the next plot, the gap estimate is updated to  $\hat{g}_{cm} = 9$  LSBs. The resulting histogram is flat, which is indicative of a histogram from a linear ADC. In the last plot, the gap estimate is updated to  $\hat{g}_{cm} = 10$  LSBs. Observe the mound that results in the histogram. Qualitatively these plots show that a gap estimate of  $\hat{g}_{cm} = 9$  LSBs produces the most linear ADC. The RMS DNL is a quantitative metric for determining this. In Figure 2-11 the RMS DNL is plotted for this example as a function of the gap estimate. As expected, it is minimized at  $\hat{g}_{cm} = 9$  LSBs, which corresponds to actual gap error used in the simulation. Thus, for this example, the gap estimate of  $\hat{g}_{cm} = 9$  would be selected as it minimizes the cost function.

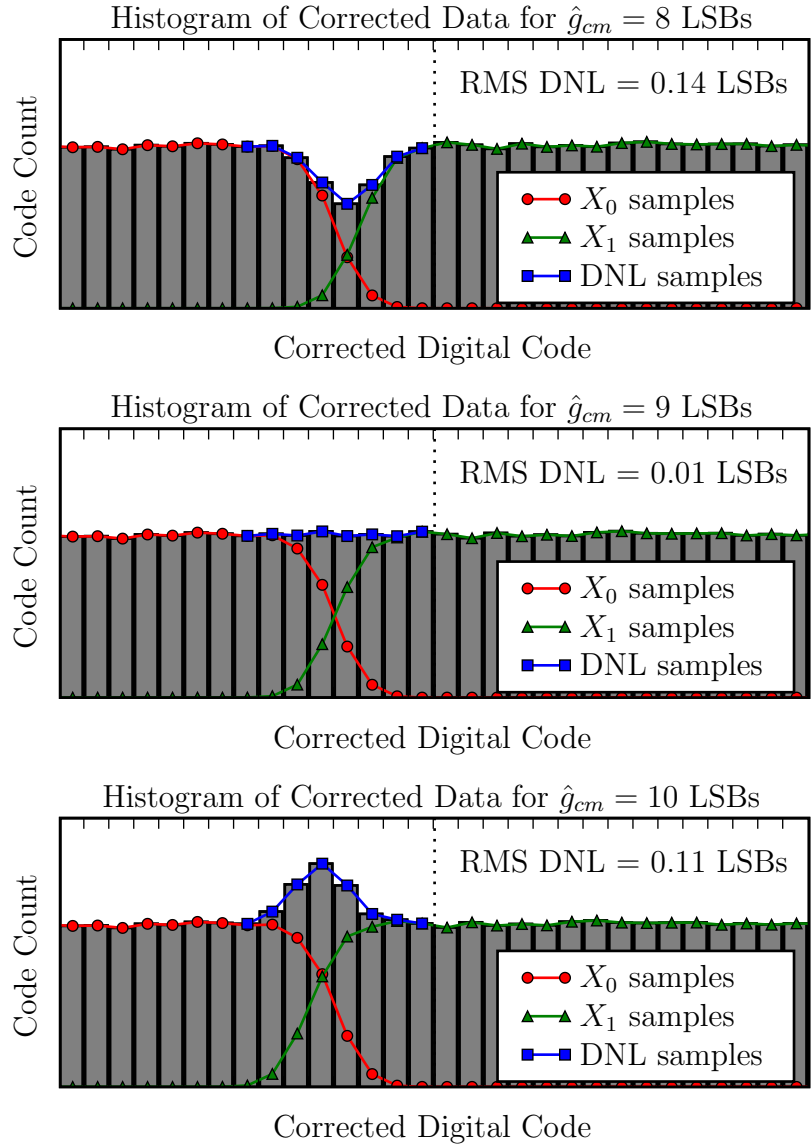


Figure 2-10: Histograms under various  $\hat{g}$  estimates. Actual  $g = 9$ .

The size of the window over which the RMS DNL should be calculated is governed by similar constraints to that of the Bin-Reshaping estimator. It should be wide enough to span the spread in the histogram caused by the circuit noise but it should be as narrow as possible to ensure that the input is approximated as well as possible by a uniform distribution. For the example shown in Figures 2-10 and 2-11 a spread of 8 bins is used, which is 8 standard deviations of the circuit noise. This example, therefore, assumes the input can be approximated as uniformly distributed over 8

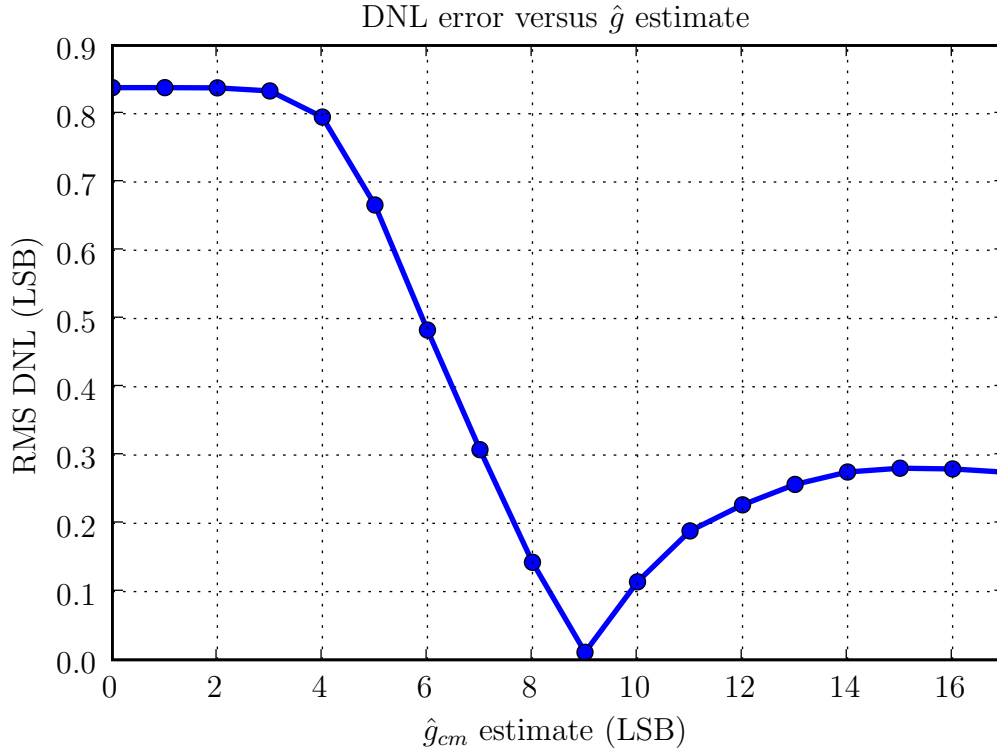


Figure 2-11: DNL vs  $\hat{g}$ .

LSBs.

Even if the input is not well approximated as uniform over the spread of the circuit noise, however, the Cost-Minimizing estimator offers the flexibility of selecting a cost function that is more appropriate for the given input signal. For example, another technique is to run a linear regression of the combined histogram over the desired window and select the gap estimate that produces the lowest RMS error or has the highest coefficient of determination  $R^2$ . This first order regression would then allow for inputs with distributions of constant gradients over the spread of the circuit noise. Another variation of this idea that is less complex would be a cost function that calculates the RMS value of the difference between adjacent bins.

The trade-off for the increased flexibility of the Cost-Minimizing estimator is an increase in complexity and hardware. It requires an increased register count to store histogram bins and also additional logic to perform the iterative search for the gap estimate that minimizes the selected cost function. Despite this, however, this estima-

tor is still relatively simple and would not require a large digital footprint compared to the overall size of the ADC.

#### 2.2.4 Estimator Discussion

Because DBGE is an indirect background calibration technique, it does not require service interruptions or suffer from calibration drift as foreground technique do. However, since it is dependant on the statistics of the input signal, it may not be appropriate for applications with input statistics that do not exercise codes in the vicinity of the decision boundaries of the ADC. Such applications, however, can use a combination of foreground and background techniques where at startup the initial gap estimates are measured during a direct foreground calibration phase using a technique like that described in [29]. Then after initialization, DBGE can then be used in the background to track parameter changes to eliminate calibration drift and avoid service interruptions or redundant hardware.

The previous discussions focused primarily on a single stage of a 1 bit/stage ADC. When going to higher resolution stages, unless the code gaps are systematic, each bit decision comparator of the sub-ADC will require independent hardware to estimate each code gap. Furthermore, each stage will require independent gap estimation. For example, suppose the first 4 stages of a 1.5 bit/stage ADC require calibration. Then 8 code gap estimates will be required for the 2 bit decision comparators in each of the 4 stages. Since the estimator updates at slower rate than the sampling frequency of the ADC, it is possible to share hardware between the various stages and perform updates in a serial fashion rather than running parallel estimates.

It is also possible to run this algorithm on a processor in a block based fashion. In this approach a block of raw data is collected. Then the processor sweeps through the data producing a gap estimate for each stage and correcting each stage in succession.

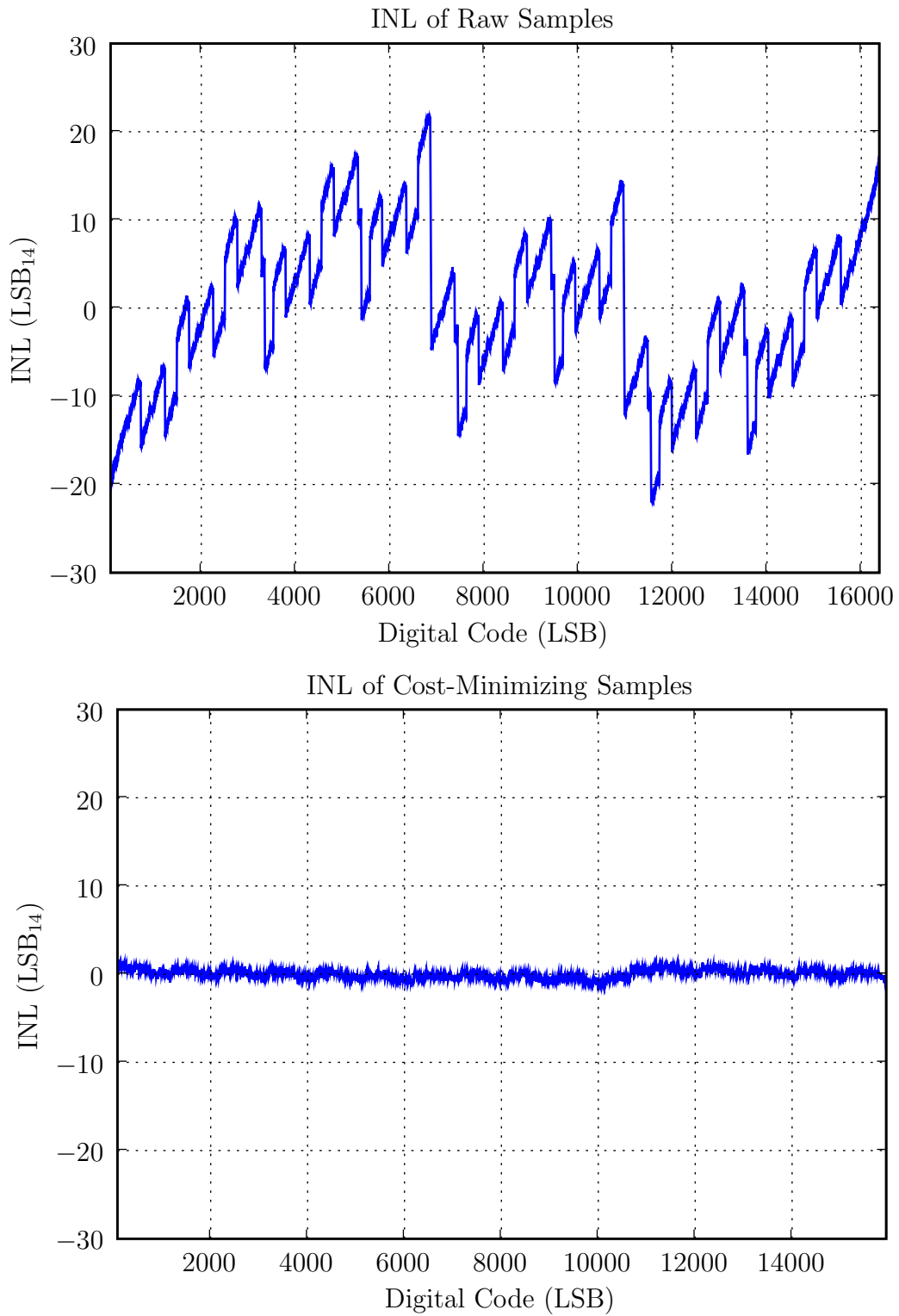


Figure 2-12: Raw and calibrated INL of 13 stage 1.5 bit/stage ADC with mismatch parameters specified in Table 2.1.



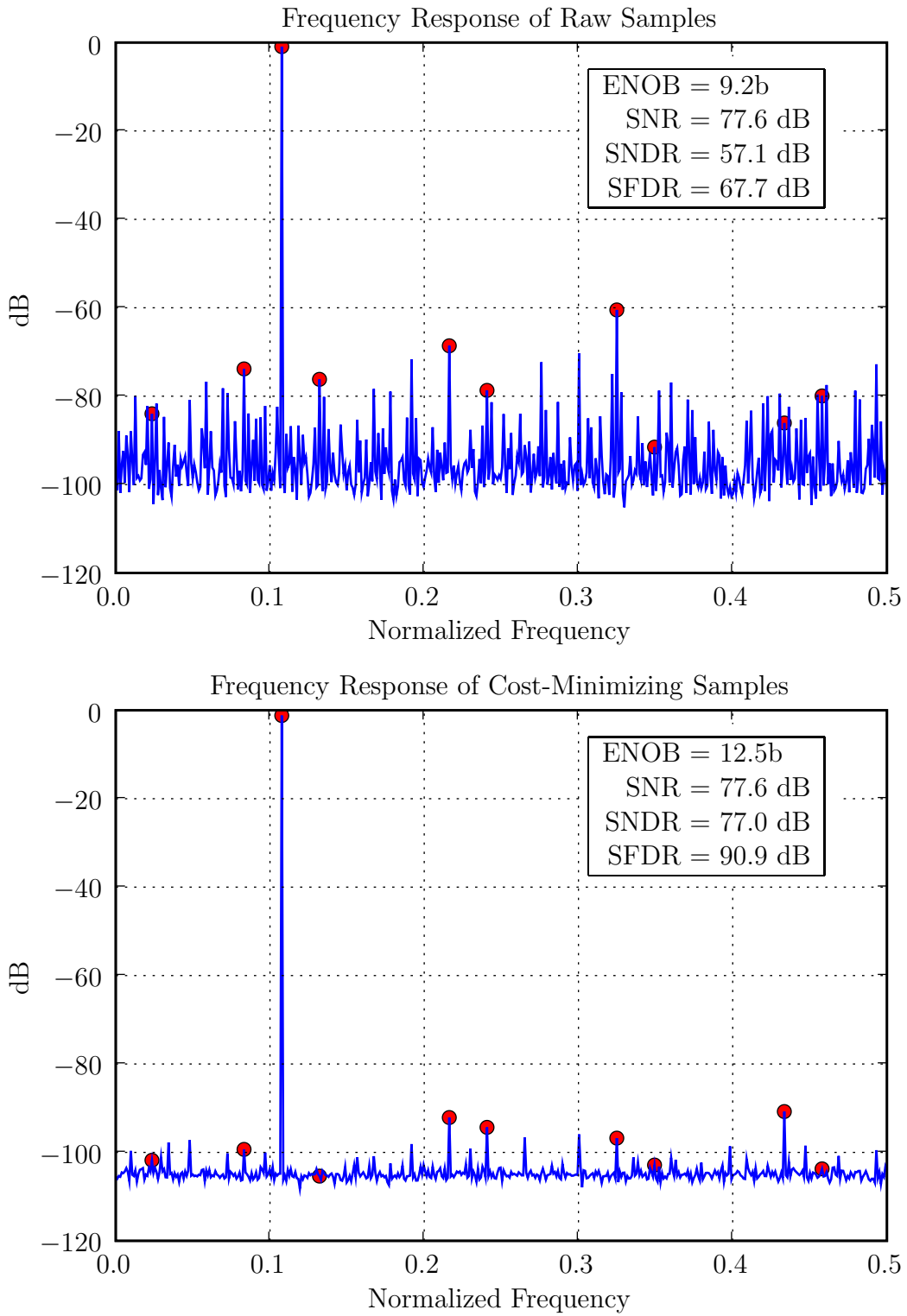


Figure 2-13: Raw and Calibrated DFT response of 13 stage 1.5 bit/stage ADC with mismatch parameters specified in Table 2.1.

Table 2.1: Simulation mismatch parameters.

Stage	Capacitor Mismatch	Opamp Gain	Comparator Offset	Voltage Offset
13	0.19%	542	0.24%	-0.41%
12	0.04%	606	-0.06%	-0.30%
11	-0.01%	597	4.72%	0.16%
10	-0.15%	454	-2.07%	0.39%
09	0.07%	421	2.71%	-0.15%
08	-0.18%	762	0.26%	-0.43%
07	0.21%	460	2.69%	-0.48%
06	-0.09%	651	-0.99%	-0.04%
05	0.51%	243	3.91%	-0.43%
04	-0.54%	299	-2.16%	-0.26%
03	0.55%	998	-1.47%	0.47%
02	-0.05%	705	3.07%	0.40%
01	-0.12%	535	4.19%	0.35%

## 2.3 Simulation Results

DBGE has been simulated under many different conditions. Shown here are the results of a 13 stage 1.5 bit/stage pipelined ADC simulated with the mismatch parameters specified in Table 2.1. Circuit noise was included in each stage to limit the effective resolution to 12.5 bits. The INL and DFT plots of uncalibrated ADC are shown in Figures 2-12 and 2-13. These show that the static non-linearities due to the mismatch parameters of Table 2.1 lower the effective resolution to 9.2 bits.

DBGE was performed on the first 6 stages. 200,000 samples from a zero mean Gaussian input were sent into the ADC. The results of the Cost-Minimizing estimator are shown the INL and DFT responses in Figures 2-12 and 2-13. The effective resolution has been raised to 12.5 bits. This means the resolution is limited by the additive circuit noise and is no longer limited by static non-linearities. The spurious free dynamic range (SFDR) goes from 67.7 dB to 91.0 dB after calibration, and the INL

Table 2.2: Simulation Results.

	ENOB (bits)	SNDR (db)	SFDR (db)	INL (LSB <sub>14</sub> )
Raw	9.2	57.1	67.7	$\pm 23$
Max-Min Estimator	11.8	72.7	85.6	$\pm 4$
Bin-Reshaping Est.	12.6	77.5	91.1	$\pm 1.5$
Cost-Minimizing Est.	12.5	77.0	91.0	$\pm 1.5$

goes from  $\pm 23$  LSB<sub>14</sub> to  $\pm 1.5$  LSB<sub>14</sub>. The ENOB (Effective Number of Bits), SNDR (Signal to Noise and Distortion Ratio), SFDR, and INL were calculated according to the procedures in [25]. Table 2.2 summarizes the results for both the raw and corrected ADC samples and shows the performance of the various estimators to this setup. Observe that the Max-Min estimator does not perform as well as the others, and this is due to the additive circuit noise introducing a bias. The Bin-Reshaping and Cost-Minimizing estimators, however, perform similarly.

Similar results are obtained with a wide range of inputs including sine wave, ramp, and uniformly random. The performance and speed of convergence of DBGE are input signal dependent. For a given estimation performance, the speed of convergence will scale with the probability of the input in the vicinity of a particular code gap. This means that decision boundaries corresponding to inputs with a low probability will take longer to collect enough samples to converge than those with a higher probability.

An input with zero probability at a particular code boundary is problematic if it has finite probability on both sides of the boundary. In this case, the input has a missing code gap, and DBGE will close the gap as it is unable to discern whether gaps come from the input signal or from the ADC. Clearly, applications with such inputs characteristics are not good candidates for DBGE. There is no problem if the input has zero probability at a particular decision boundary and has finite probability on only one side of the boundary. This corresponds to the case that a particular input does not fill the full input range of the ADC. Any decision boundaries outside of the range of the input signal will have wrong estimates, but since the input does not exercise those codes, their wrong estimates do not matter.

## 2.4 Conclusion

The motivation for DBGE came from the observation that the non-linearities that dominate CMOS switch-capacitor circuit design cause code gaps at each bit decision boundary of the sub-ADC. This technique, however, is general to a broader class of both implementations and architectures. It applies to any situation where the amplified error or residue from each stage causes a decision boundary gap.

An appropriate follow-up question to the work presented herein is what estimator and cost function achieves optimal performance. The answer to this question and others such as convergence time is beyond the scope of this thesis. One reason is that this requires specifying the statistics of the input signal and an additional cost function over which to define optimality. Instead, this work presents a general framework for performing indirect background calibration of the common static non-linearities in pipelined ADCs. The estimator and cost function should be selected and analyzed based on the specific application and the statistics of the input signal and remains as an open research question.

In its general form, Decision Boundary Gap Estimation is an adaptive, digital, indirect method of background calibration. The advantages of DBGE are numerous. There is no need for additional analog hardware, such as a redundant channels/stages or a reference converter to calibrate against. The calibration is highly accurate because the transition points are directly aligned. Furthermore, its simplicity makes it amenable to VLSI and/or processor based implementations. Thus, DBGE is a calibration approach that can be implemented to improve existing ADC designs or to shape new designs by relaxing analog circuit requirements for high gain opamps, matched capacitors, and low offset comparators. Reducing these design constraints allows the designer to reduce power and/or increase conversion speed, and perhaps most importantly, it can be an enabling factor for ADC design in deep sub-micron technologies.

# Chapter 3

## Zero-Crossing Based Circuits

While DBGE can ease the analog design requirements for an high-gain opamp, it has limitations such that it can only correct for non-linearities at the bit decision boundaries of each stage. Therefore, it can not correct for errors such as signal dependant gain variation in the opamp. Furthermore, it requires the input signal exercise codes in the vicinity of the non-linearities. This chapter changes the focus away from calibration to study an alternative circuit architecture that can be applied more generally to solve the design issues associated with device scaling.

### 3.1 Background

#### 3.1.1 Opamp-Based Switch Capacitor Circuits

The typical implementation of a opamp-based pipeline ADC stage was shown in Figure 1-3. When  $\phi_1$  is high, the circuit is configured in the *sampling* phase and the input voltage  $v_I$  is sampled with respect to  $V_{CM}$  onto capacitors  $C_1$  and  $C_2$ . When  $\phi_1$  falls and  $\phi_2$  rises, the circuit is configured in the *transfer* phase. The role of the opamp is to *force* the virtual ground condition by driving the output voltage  $v_O$  until the  $v_X$  node equals  $V_{CM}$ . The accuracy of the transfer phase is determined by how well the virtual ground condition is realized. If the error in the virtual condition is not signal dependent, then an offset results that can be nulled with any number of auto-zeroing

techniques [16]. When the error is signal dependent, gain errors and/or non-linearities will result. In the case of an opamp-based implementation, finite open-loop opamp gain and insufficient settling are two effects which cause such signal dependent errors in the virtual ground condition.

In the case of finite opamp gain, the accuracy of the virtual ground condition is inversely proportional to the open-loop gain of the opamp (see Equation 1.5). The gain, therefore, must be large enough to ensure the signal dependent error in the virtual ground condition is small enough for the specific application.

In the case of insufficient settling, the feedback loop from the output of the opamp, through  $C_2$ , and back to the input of the opamp must be given ample time to settle to avoid a signal dependent error in the virtual ground condition. The typical exponential settling of  $v_O$  and  $v_X$  in an opamp-based implementation is shown in the transient response of Figure 3-1a.

These issues create the stability versus bandwidth/power trade off for the opamp-based system because of the fundamental constraints associated with increasing gain and bandwidth simultaneously. Furthermore, the bandwidth requirements significantly decrease the power efficiency of an opamp-based system as the noise bandwidth of signal path is determined by the bandwidth of the feedback loop, which can be several times larger than the signal bandwidth to ensure sufficient settling time [18, 26].

### 3.1.2 Comparator-Based Switched Capacitor Circuits

Comparator-Based Switched Capacitor (CBSC) circuits as shown in simplified schematic of Figure 1-5 do not suffer from the above issues. Observe that the opamp is replaced with a comparator and current source. As with the opamp-based implementation, when  $\phi_1$  is high during the sampling phase, the input voltage  $v_I$  is sampled onto  $C_1$  and  $C_2$ . When  $\phi_2$  goes high to enter the transfer phase, a short pulse  $\phi_{2I}$  is used to initialize the charge transfer by closing switch  $S_2$  to pre-charge the output voltage  $v_O$  to ground. Following this pulse,  $S_2$  opens and the current source  $I_1$  charges the capacitors to generate a constant voltage ramp on the output voltage  $v_O$ . This causes

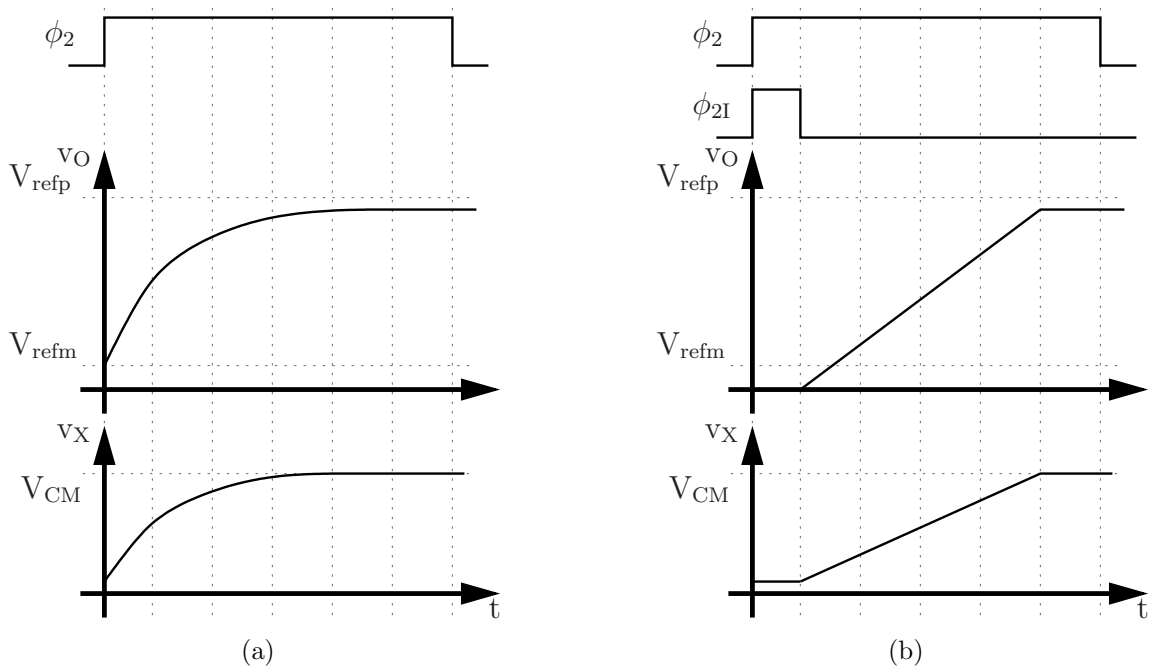


Figure 3-1: Sample transient response of (a) an opamp-based and (b) a CBSC switched capacitor gain stage.

the virtual ground voltage  $v_X$  to ramp with it via the capacitor divider consisting of  $C_1$  and  $C_2$ . As the voltage ramp proceeds, the comparator will *detect* when the virtual ground condition has been reached and then turn off the current source to realize the same charge transfer as the opamp-based implementation. The resulting transient response for voltages  $v_O$  and  $v_X$  is shown in Figure 3-1b.

It is important to realize that the shape of the transient response does not matter for switched-capacitor circuits. The critical time in the transfer phase is when the sampling switch opens to sample the output voltage  $v_O$  onto the load capacitor  $C_L$ . In fact, depending on the implementation of the opamp, two different opamp-based systems may have dramatically different transient responses depending on effects such as slewing and ringing. It is the accuracy of the virtual ground condition when the sampling switch opens that matters. Thus, whereas an opamp-based implementation forces the virtual ground condition, the CBSC implementation sweeps the output voltage and searches for the virtual ground condition. Both, however, realize the same charge transfer despite their dramatically different transient responses.

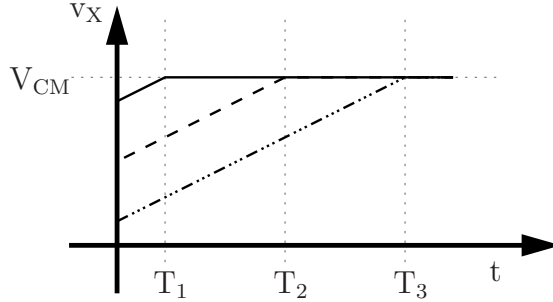


Figure 3-2: Sample input waveforms into a CBSC comparator.

## 3.2 Zero-Crossing Based Circuits

Just as the opamp in an opamp-based design, the comparator in a CBSC design contributes most significantly to the speed, power efficiency, and Figure of Merit (FOM) of the overall circuit. Generally, a comparator must resolve the difference between two arbitrary voltage waveforms. The input into the comparator of a CBSC circuit, however, is not arbitrary. As shown in the sample waveforms of Figure 3-2, the input into the comparator of a CBSC circuit is a constant slope voltage ramp, so the comparator actually performs a uni-directional zero-crossing detection. Therefore, a general purpose comparator is not strictly necessary. This work generalizes CBSC circuits into zero-crossing based circuits (ZCBC) by replacing the general purpose comparator with a zero-crossing detector. As discussed in Section 3.5, this generalization allows for implementations of zero-crossing detectors that are more power efficient than general purpose comparators.

Figure 3-3 shows a simplified implementation of the zero-crossing based circuit that is used in this work. The general purpose comparator of the CBSC implementation has been replaced with dynamic zero-crossing detector (DZCD) that consists of devices  $M_1$  and  $M_2$ . The circuit functions similarly to the CBSC circuit shown in Figure 1-5. During the sampling phase when  $\phi_1$  is high the input voltage is sampled onto  $C_1$  and  $C_2$ . Then, as shown in the timing diagram of Figure 3-4,  $\phi_2$  and  $\phi_{2I}$  go high to start the transfer phase.  $\phi_{2I}$  turns on  $M_4$  to pre-charge the output voltage  $v_O$  to ground. This pushes the virtual ground node voltage  $v_X$  down to turn off  $M_1$ . Simultaneously,  $\bar{\phi}_{2I}$  turns on  $M_2$  to pre-charge the voltage  $v_P$  high and turn on the



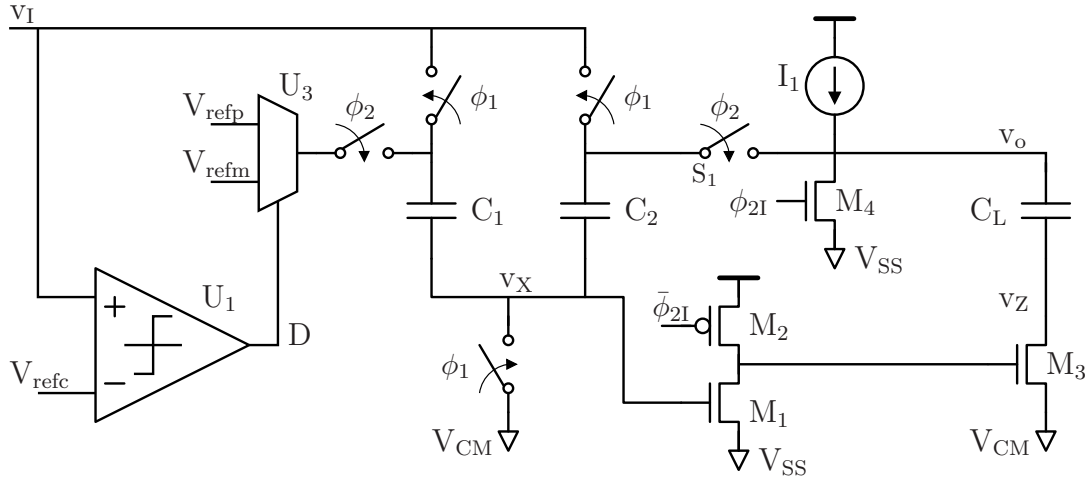


Figure 3-3: Zero-crossing based switched capacitor pipelined ADC stage.

sampling switch  $M_3$ . This initializes the load capacitor  $C_L$  below the full scale output range.

When  $\phi_{2I}$  drops, node  $v_P$  is left floating high to keep the sampling switch on, and the output voltage  $v_O$  begins to ramp from the current source pulling it up.  $v_X$  ramps with it according the capacitor divider established by  $C_1$  and  $C_2$ . As  $v_X$  ramps up it will at some point give  $M_1$  sufficient gate drive to start pulling down the floating  $v_P$  node. When  $v_P$  is pulled down sufficiently to turn off the sampling switch  $M_3$ , the voltage on the load capacitor  $C_L$  is sampled and the charge transfer is complete. Opening  $M_3$  to define the sampling instant minimizes signal dependent charge injection by performing bottom plate sampling [21].

The dynamic zero-crossing detector consisting of  $M_1$  and  $M_2$  is not suitable as a general purpose comparator. It can not detect differences in two arbitrary voltages. It is, however, suitable as a zero-crossing detector in this architecture because the constant slope voltage ramp created by the current source  $I_1$  ensures that  $M_1$  switches consistently at the same voltage. The switching threshold of  $M_1$  is temperature, process, and ramp-rate dependent, but since the switching threshold is not signal dependent, it creates a constant offset that can be nulled with any number of traditional auto-zeroing circuits [16]. This initial implementation did not employ an auto-zeroing technique but rather globally adjusted the  $V_{CM}$  voltage externally

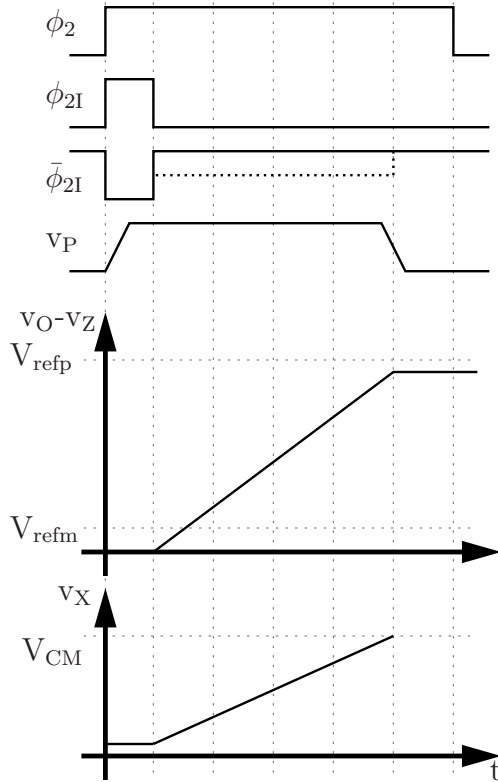


Figure 3-4: Sample transient response of a ZCBC switched capacitor gain stage.

to null the cumulative offset of the complete ADC. It must be noted, however, that power efficient auto-zeroing techniques need to be developed for this architecture to take the full advantage of the power efficiency of the DZCD.

One significant limitation to this DZCD is that it is inherently single-ended and does not have a natural differential extension. Thus, depending on the amount of power supply and substrate noise present in a particular system, this architecture may be not be suitable for high resolution applications.

Despite these limitations, this DZCD has several compelling advantages. It is fast, simple, and amenable to scaling. It produces a rail-to-rail digital logic level in a single stage while drawing no static current. It consumes only the power necessary to switch the capacitance on node  $v_P$ , which will be shown in Section 3.5 to offer an improvement in power efficiency.

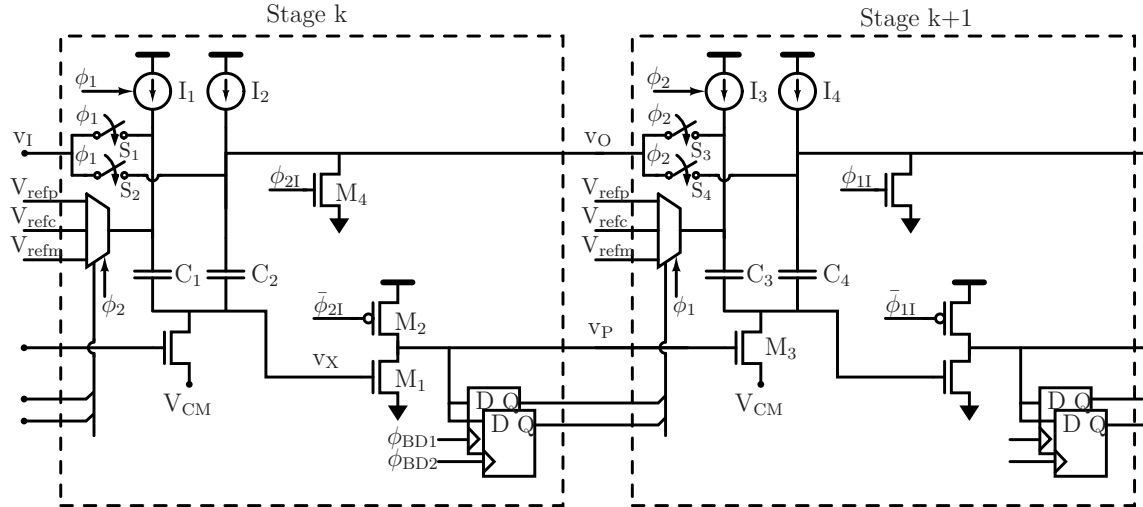


Figure 3-5: Two stages of the 1.5 bit/stage zero-crossing based pipelined ADC.

### 3.3 ZCBC Pipelined ADC Implementation

A 1.5 bit/stage pipelined ADC was implemented to demonstrate this ZCBC architecture. The schematic of two adjacent stages (stages  $k$  and  $k+1$ ) is shown in Figure 3-5. The implementation details that follow apply to the general case when stage  $k$  is not the first stage. The subtle differences imposed on the first stage are discussed in Section 3.3.7.

#### 3.3.1 DZCD Design

One significant issue that arises when  $v_P$  is left to float while the  $v_X$  voltage ramps is that feed-through from the  $C_{gd}$  of  $M_1$  pushes a signal dependent amount of charge onto the  $v_P$  node. This charge has to be removed by  $M_1$  when it switches and creates a signal dependent delay. Such a signal dependent delay produces a gain error similar to capacitor mismatch at the output. To eliminate this issue, rather than turning  $M_2$  off completely while the voltage ramps, the gate of  $M_2$  is biased so that  $M_2$  can sink the feed-through current and prevent  $v_P$  from accumulating a signal dependent amount of charge. The dashed line for  $\bar{\phi}_{2I}$  in the timing diagram of Figure 3-4 shows this scenario. After  $v_P$  switches, however,  $M_2$  is shut off to ensure no static current is drawn.

### 3.3.2 Current Source Splitting

The single current source  $I_1$  shown in Figures 1-5 and 3-3 has been divided in this implementation into  $I_2$ ,  $I_3$ , and  $I_4$  to charge each capacitor separately. This removes the series switch  $S_1$  in Figures 1-5 and 3-3 and improves the linearity and output swing. When implemented as a single current source, the charging current must pass through the series switch, which creates a voltage drop due to the finite on-resistance of the switch. This voltage drop reduces the output swing. More importantly, however, since the on-resistance of a typical CMOS switch is not constant, the voltage drop is also not constant and creates a signal dependent non-linearity at the output. Since the ramp rate must be increased as the speed of the ADC increases, this problem gets worse as the ADC runs faster. Rather than sizing the switches up to reduce the on-resistance to acceptable levels, one can divide the current sources up as shown in Figure 3-5 and remove the series switches to eliminate this issue. Since all other switches are connected to DC voltages, they do not produce signal dependent voltage drops and do not contribute non-linearities to the output.

### 3.3.3 Shorting Switches

When dividing the current source, current mismatch and capacitive load differences will create different voltage ramp rates on each capacitor. Shorting switches  $S_1$ ,  $S_2$ ,  $S_3$  and  $S_4$  of Figure 3-5 have been added to carry any mismatch current to ensure that each capacitor charges at the same rate. When  $\phi_1$  is high, stage k is in the sampling phase and  $I_2$  charges  $C_2$  directly. When  $\phi_2$  is high, stage k is in the transfer phase and  $I_2$  charges half the capacitive load because  $C_1$  and  $C_2$  are configured in series\*. To maintain the same voltage ramp rate, the charging current provided by  $I_2$  should be reduced by two during the transfer phase. For this implementation the charging current of  $I_2$  was not changed between the sampling and transfer phases for simplicity. This means that the  $\frac{1}{5}$  the current supplied by  $I_2$  during the transfer phase actually

---

\*This discussion applies to the case of a uniformly scaled 1.5 bit/stage ADC where the sampling capacitors are equal and  $C_1 = C_2 = C_3 = C_4$ . The exact numbers change depending on stage scaling or resolution when the sampling capacitors are not equal, but the technique still applies.

goes through each of the shorting switches  $S_3$  and  $S_4$  to keep the voltage ramp rate constant. Thus, in this implementation, the sizing requirement of the switches was reduced by a factor of 5 over using a single current source and a single series switch.

To further reduce the sizing of the shorting switches, these switches were implemented as nMOS only switches with a gate boosting circuit shown in Figure 3-6. The corresponding timing diagram is shown in Figure 3-7. In the schematic,  $M_1$  is the actual shorting switch, and the remaining circuitry is the driver. Thus, during the pre-charge phase when  $\phi_{2I}$  is high, the source and drain of  $M_1$  is reset to ground. Simultaneously the gate is charged to  $V_{DD}$  via  $M_2$ . Since  $M_2$  is an nMOS, its gate voltage must be boosted to give it sufficient gate drive to switch it to  $V_{DD}$ . This boosted gate drive is generated via the global switch driver circuit also shown in Figure 3-6. This circuit is based on the circuits found in [1, 11], and it ensures no device is stressed above the supply voltage. So during the pre-charge phase,  $C_1$  is charged to  $V_{DD}$ . When  $\phi_{2I}$  drops to end the pre-charge phase, the gate of  $M_1$  is left floating. Since the source and drain of  $M_1$  are connected to the output voltage of the ZCBC stage, they will then begin to ramp due to the current sources charging the sampling capacitors. The feed-through from  $C_1$  will pull the floating gate with them as they ramp and provide a constant  $V_{GS}$  of  $V_{DD}$  on  $M_1$ . A constant  $V_{GS}$  provides a much more constant resistance than a complementary switch and thus further reduces the sizing requirements of the shorting switch. At the end of the transfer phase when  $\phi_1$  rises,  $M_4$  discharges the floating gate and turns off the shorting switch.  $M_3$  ensures that the source-drain voltage of  $M_4$  never exceeds  $V_{DD}$  and no devices are stressed above their voltage rating.

Two global switch drivers as shown in Figure 3-6 are implemented on chip and shared between all the shorting switches of all the stages of the same phase. Current source splitting and switch gate boosting allow for minimum sized nMOS shorting switches.

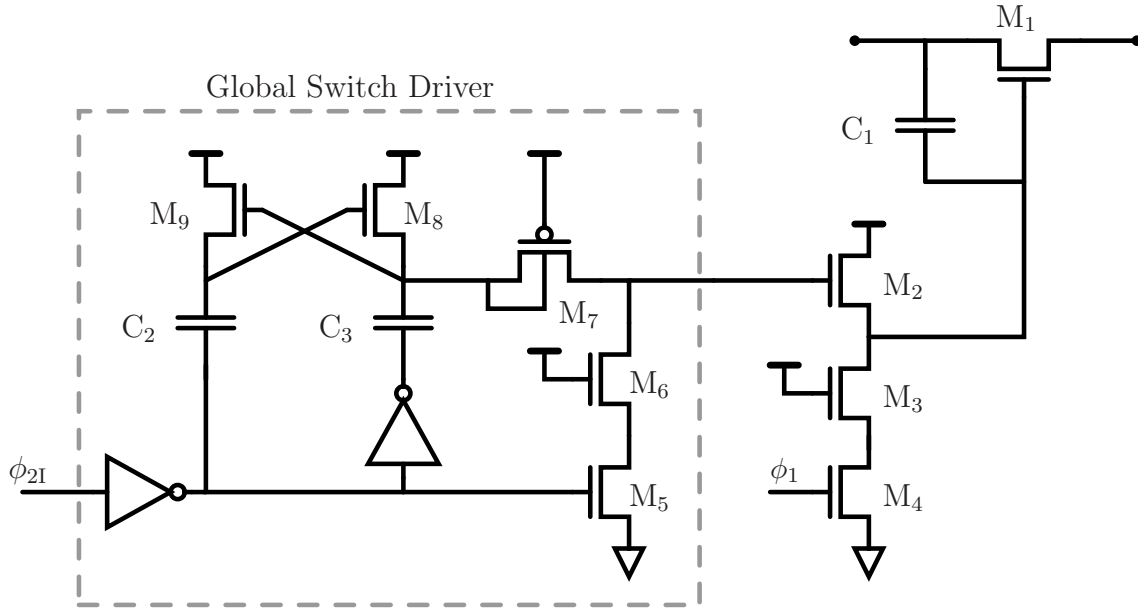


Figure 3-6: Shoring switch implementation.

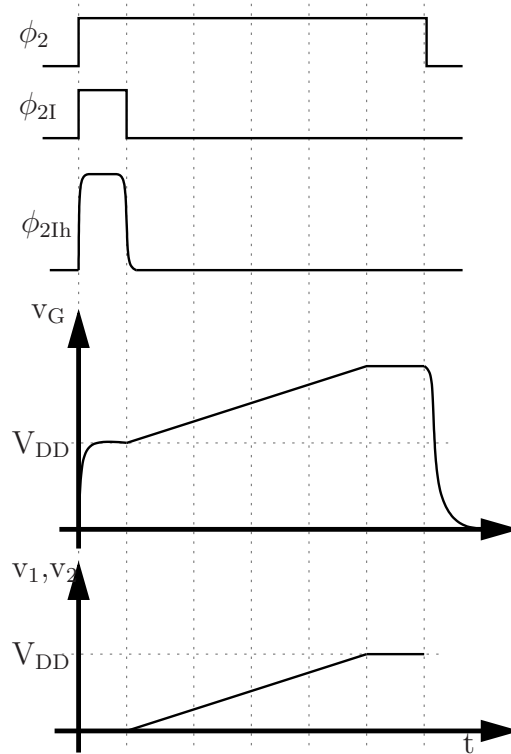


Figure 3-7: Shoring switch timing diagram.

### 3.3.4 Reference Voltage Switches

The reference voltage multiplex switches ( $V_{\text{refx}}$  switches in Figure 3-5) subtract the quantized voltage from the input to produce the residue. In the case of a 1.0 bit/stage

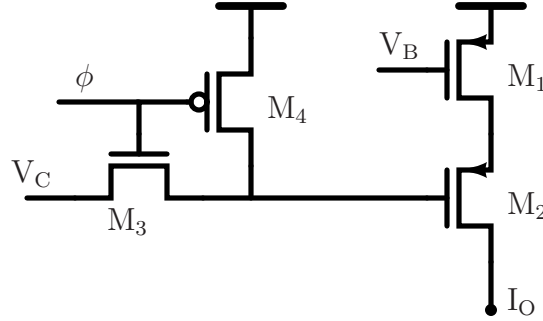


Figure 3-8: Current source implementation.

implementation, they only switch between two voltage levels, and they are inherently linear. In the case of a 1.5 bit/stage implementation, however, they must switch between three different reference voltages, and a non-linearity can result if the reference voltages themselves are non-linear. In the case of an opamp based implementation, the feedback loop must settle and thus the voltage drop across the reference switches is not a significant issue. In this ZCBC implementation, however, there is a constant current through the  $V_{\text{refx}}$  switches that produces a voltage drop due to its finite resistance. If each switch has a different resistance, then each will have a different voltage drop and create a non-linearity at the output. To ensure sufficiently matching switch resistance, the gate boosting circuit described in [11] is used to implement the  $V_{\text{refx}}$  switches. This circuit does not reduce reliability as it ensures that no device is stressed above the power supply and it boosts the gate to ensure each switch has the same  $V_{\text{GS}}$ . This same circuit is also used for the input sampling switch.

### 3.3.5 Current Source Implementation

The current sources ( $I_1$ ,  $I_2$ ,  $I_3$ , and  $I_4$  of Figure 3-5) were implemented as pMOS cascoded current sources as shown in Figure 3-8. The cascode device also doubles as the enable switch. Sufficient settling of the cascode voltage on the gate of  $M_2$  is not difficult to achieve when the enable is overlapped with the pre-charge phase. Not only does this give it extra time to settle but the pre-charging of  $v_O$  pulls the drain down and the feed-through from the  $C_{\text{gd}}$  of  $M_2$  helps its gate reach the cascode bias level faster.

In Section 1.3.2 the effect on the residue amplification due to the finite output impedance of the current source was calculated.

### 3.3.6 Bit Decision Flip-Flops

The bit decision comparators of the sub-ADC of a pipelined ADC provide a coarse quantization of the output voltage  $v_O$  and are traditionally implemented as clocked comparators. When the bit decision comparators are implemented in this manner in ZCBC architectures, they lie in the critical path because they must make their decision after one stage completes its transfer and before the next stage can begin. Thus they can limit the overall speed of the ADC and create meta-stability issues when they are not given ample time to make their decision. To remove the bit decision logic from the critical path, this design does not use traditional bit decision comparators but rather uses bit decision flip-flops as shown in Figure 3-5.

Since the output voltage  $v_O$  ramps up linearly until the DZCD switches, the time at which the DZCD switches is proportional to the output voltage. Therefore, in a manner analogous to a single slope ADC, sampling the DZCD output with flip-flops whose clock is phase-aligned with the decision threshold yields an equivalent coarse quantization of the output voltage.

To generate the clock phase that corresponds to the desired  $\pm V_{\text{ref}}/4$  bit decision levels necessary for a 1.5 bit/stage ADC, the feedback circuit of Figure 3-9 is used. The clock  $\phi$  goes through a voltage-controlled delay line (VCDL) to produce the reference clock phase  $\phi_{\text{BD}}$ .  $\phi_{\text{BD}}$  along with the bit decision voltage  $V_{\text{ref}}/4$  goes into a replica pipeline stage, and the output bit of this replica stage is then fed back to the VCDL to adjust the phase of  $\phi_{\text{BD}}$  for the next sample.

The actual circuit implementation of the VCDL is also shown in Figure 3-9. The voltage  $v_G$  controls the delay of the current-starved inverter consisting of  $M_1$ ,  $M_2$ , and  $M_3$ . Suppose  $v_G$  starts at  $V_{\text{DD}}$  such that  $C_1$  is fully charged. This gives the VCDL minimal delay and causes the bit decision flip-flop in the replica stage to sample the DZCD output immediately to yield a high decision output  $D$ . This will cause the VCDL to discharge  $C_2$  to ground. When  $\phi$  falls,  $C_1$  and  $C_2$  get shorted together to



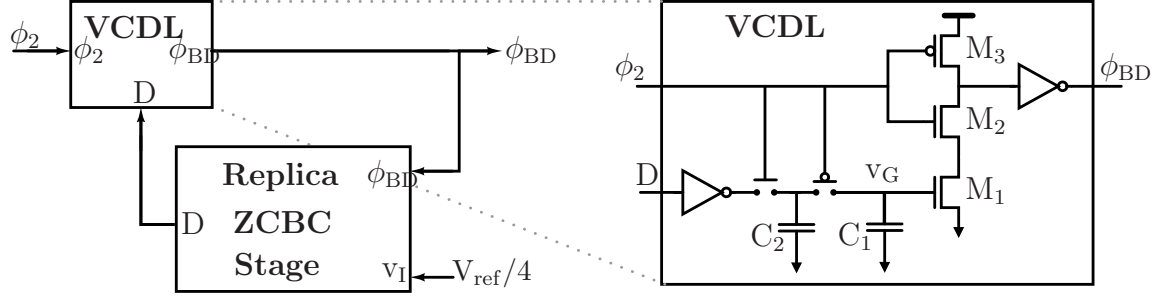


Figure 3-9: The bit decision flip-flop phase generation circuit, including the voltage-control delay line implementation.

decrement the voltage  $v_G$  on  $C_1$  and increase the delay. On each clock cycle the delay will continue to increase until the phase of  $\phi_{BD}$  passes the  $V_{ref}/4$  threshold and causes the bit decision flip-flop in the replica stage to sample the low DZCD output. At that point  $C_2$  will be charged to  $V_{DD}$  and when  $\phi$  falls and  $C_2$  and  $C_1$  are shorted,  $v_G$  will increment to decrease the delay. In steady state the bit decision flip-flop of the replica stage will toggle high and low to keep  $\phi_{BD}$  aligned to the falling edge of the DZCD in the replica stage. The small amount of jitter from such toggling is not an issue due to the over-range protection offered by a 1.5 bit/stage ADC. The over-range protection also eliminates any offset differences between the flip-flops of the replica stage and the actual pipeline stages from being problematic.

Using bit decision flip-flops removes the bit decision logic from the critical path because the bit decisions are made in parallel with the voltage ramp and are ready by the time the voltage ramp ends. This removes the meta-stability issues that can arise from using traditional clock comparators. Furthermore, the bit decision flip-flops do not have any unusual requirements and can be taken from a digital standard cell library.

### 3.3.7 First Stage Considerations

Since the first pipeline stage is not driven by a ZCBC stage, it requires several slight modifications to the schematic shown in Figure 3-5. The input voltage  $v_I$  of the first stage is not a voltage ramp but the actual low-impedance ADC input. This means

that current sources  $I_1$  and  $I_2$ , which generate the voltage ramp during the sampling phase, are not needed.  $I_1$  can be removed completely.  $I_2$  is still needed during the transfer phase when  $\phi_2$  goes high, so it is implemented as an enabled current source for the first stage. Furthermore, the first stage does not have a previous stage to control the sampling switch ( $M_5$  of Figure 3-5) and the  $V_{\text{refx}}$  switches. Since the sampling capacitors are driven with a low-impedance source, the gate of the sampling switch of the first stage is tied to  $\phi_1$  to give maximum settling time and to perform bottom-plate sampling. Lastly, without a voltage ramp input and a zero-crossing detector, bit decision flip-flops cannot be used to drive the analog multiplexer of the first stage. Therefore, traditional clocked comparators are used for the first stage and the input sampling period of the gate-boostered nMOS sampling switch is reduced to give them ample time to make their decision. Since the input switch does passive sampling, this reduction in time is not an issue.

### 3.4 Experimental Results

This design was implemented as ten equally sized pipeline stages in a  $0.18\mu\text{m}$  CMOS technology in an active die area of  $0.05\text{mm}^2$ . The die photo is shown in Figure 3-10. Figure 3-11 shows the DNL and INL is  $\pm 0.5\text{LSB}$  and  $\pm 0.75\text{LSB}$  at  $100\text{MS/s}$  and  $\pm 0.75\text{LSB}$  and  $\pm 1.0\text{LSB}$  at  $200\text{MS/s}$ . Figure 3-12 shows the frequency response to a near Nyquist rate input tone for  $100\text{MS/s}$  and  $200\text{MS/s}$ . From the frequency response the ENOB is measured at 6.9b and 6.4b for  $100\text{MS/s}$  and  $200\text{MS/s}$  respectively. The spectral response carries many aliased harmonics due to static non-linearities that cause distortion, but these harmonics carry very little power. The SNDR is dominated by temporal circuit noise as is further discussed in Section 3.5.3.

The power consumption is plotted as a function of sampling frequency in Figure 3-13. At  $200\text{MS/s}$  the ADC consumes  $8.5\text{mW}$  ( $2.9/5.6\text{mW}$  analog/digital) from a  $1.8\text{V}$  power supply. Figure 3-13 shows that the complete ADC draws only dynamic power. The current sources do not draw static power because they provide only the charge necessary to realize the charge transfer and then turn off.

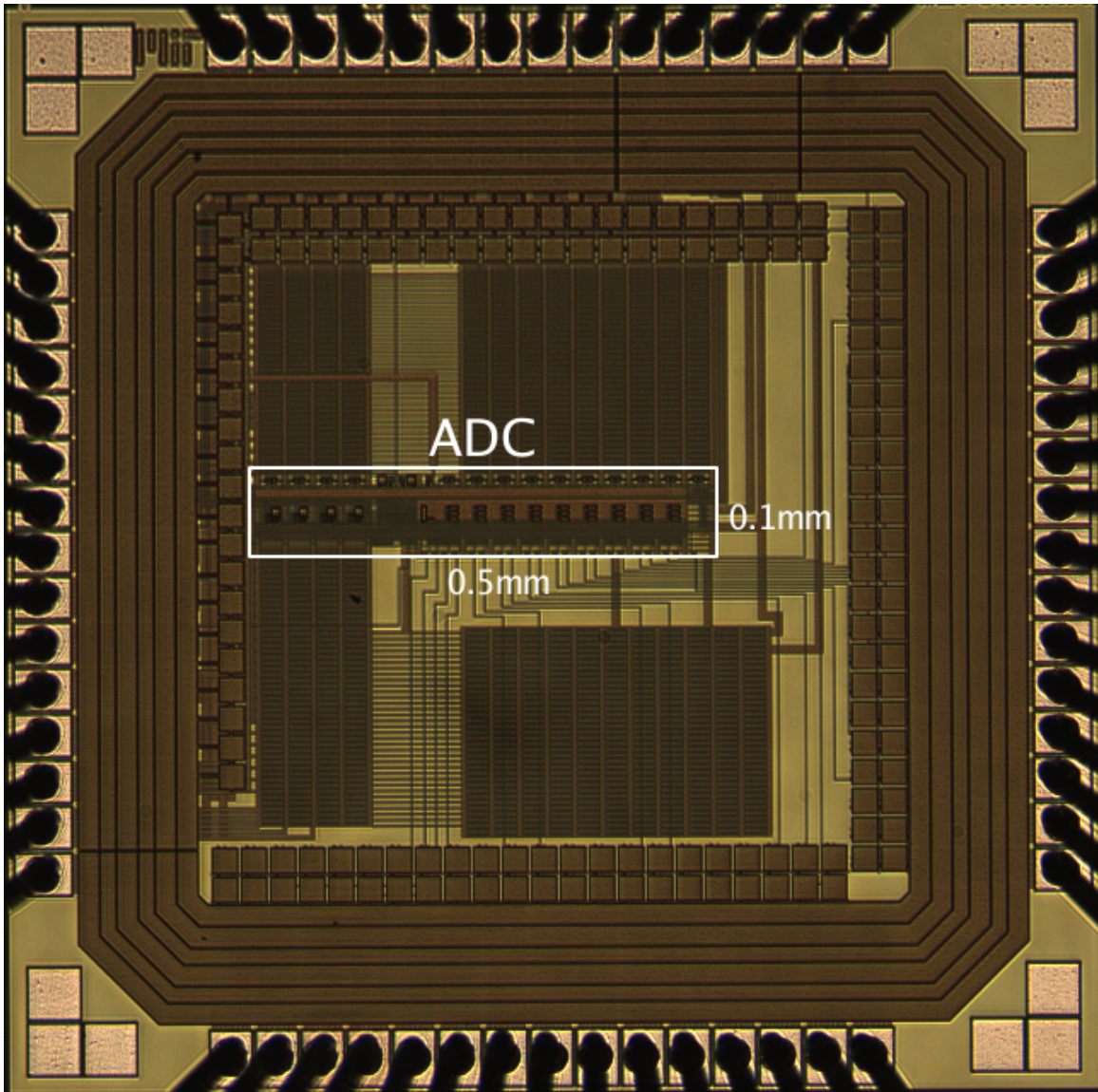


Figure 3-10: Die photo of 0.05mm<sup>2</sup> ADC in 0.18µm CMOS.

The corresponding Figure of Merit ( $FOM = \frac{P}{2f_{in}2^{ENOB}}$ ) is 380 fJ/step at 100MS/s and 510 fJ/step at 200MS/s. These results are summarized in Table 3.2.

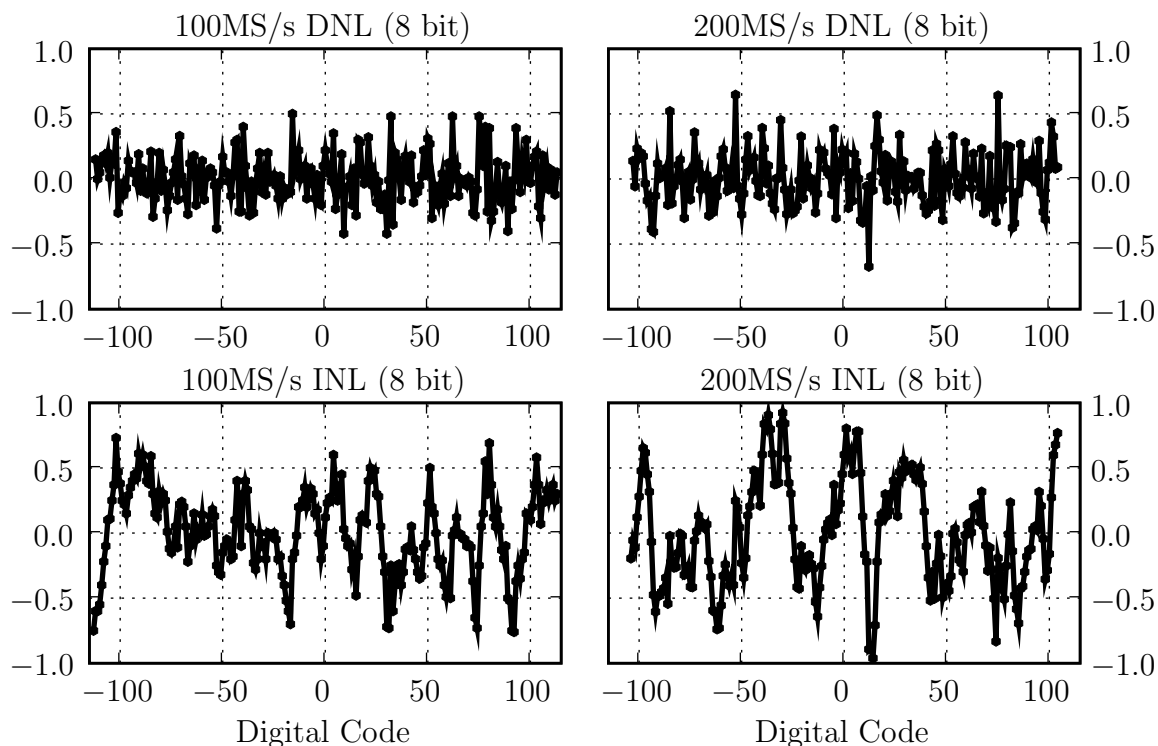


Figure 3-11: DNL and INL plots for 100MS/s and 200MS/s operation.

## 3.5 Power Efficiency Analysis

### 3.5.1 DZCD Noise Analysis

A thorough analysis of the noise performance of CBSC circuits, including the contribution of the threshold detecting comparator, current sources, and sampling switches, has been presented in [18,41]. Like CBSC circuits, the most significant source of noise for this circuit is the DZCD. Noise from the DZCD causes timing jitter on the falling edge of  $v_P$ , which creates uncertainty in when the sampling switch opens. Because the sampling switch opens at an uncertain time, an uncertain voltage, or noise, will be sampled as the output voltage ramps. Device  $M_1$  of the DZCD in Figure 3-5 contributes most significantly to this source of noise.

Figure 3-14 shows the waveforms obtained from a transient simulation of a single pipeline stage. The waveform names correspond to voltages shown in the schematic of Figure 3-5. The first waveform shows the transient response of  $\phi_2$  and  $\phi_{2I}$ . The

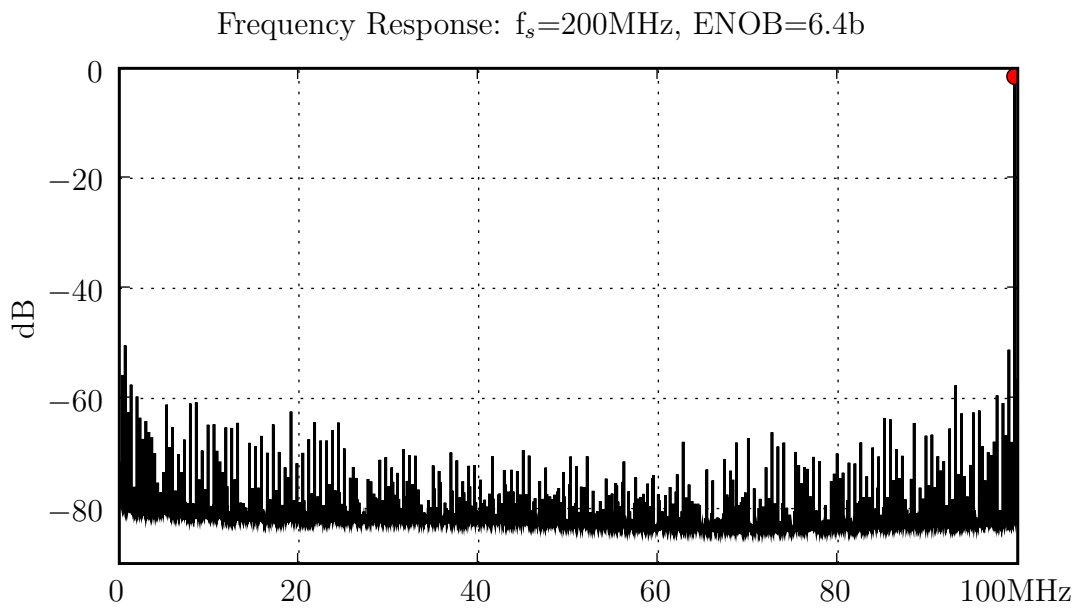
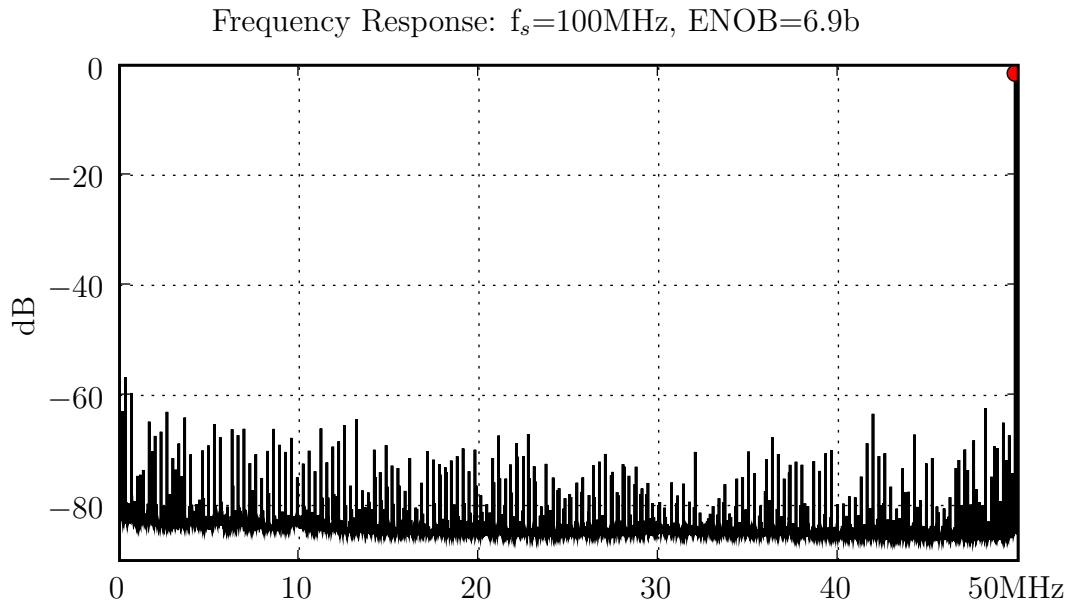


Figure 3-12: Measure frequency response to near Nyquist rate input tone.

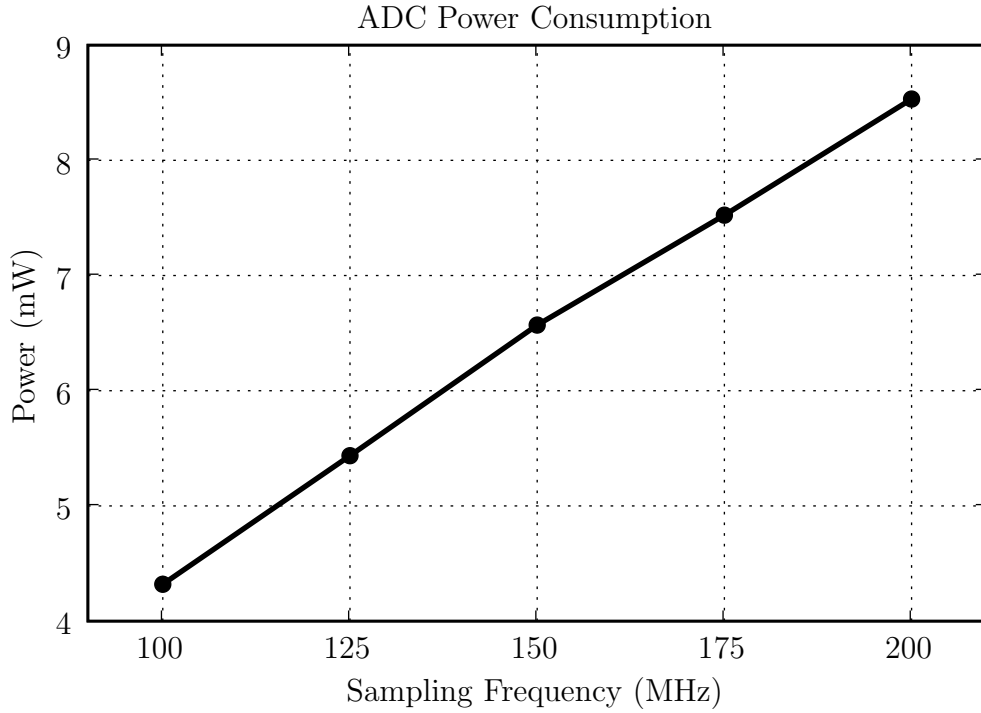


Figure 3-13: Measured power consumption versus sampling frequency.

second plot shows the transient response of  $v_P$ ,  $v_X$ , and  $v_O$ . The third plot shows  $I_D$ , the transient current drawn by  $M_1$ . This current draw is insignificant while the voltage ramp proceeds until  $v_X$  gets high enough to start turning on  $M_1$ . At that point the current level rises rapidly until  $v_P$  is completely discharged at which point the current draw returns to zero. The shaded area under the  $I_D$  waveform represents the total charge consumed while the sampling switch is closed (i.e.  $v_P$  is high enough to provide  $M_3$  sufficient gate drive to be on). It is during this period that the noise generated by  $M_1$  integrates onto the capacitance on node  $v_P$  and causes timing jitter on the falling edge of  $v_P$ .

Approximating the shaded area of the current spike as a box of equal area simplifies the noise calculation. If the height of this box is  $\bar{I}_D$  and the width is  $t_d$ , then the effective noise bandwidth is  $\frac{1}{2t_d}$  and the input referred noise spectral density is  $\frac{8kT}{3\bar{g}_m}$ , where  $\bar{g}_m$  is the transconductance resulting from a bias current of  $\bar{I}_D$  in device  $M_1$ . The total input referred noise is the product of the bandwidth and the spectral density

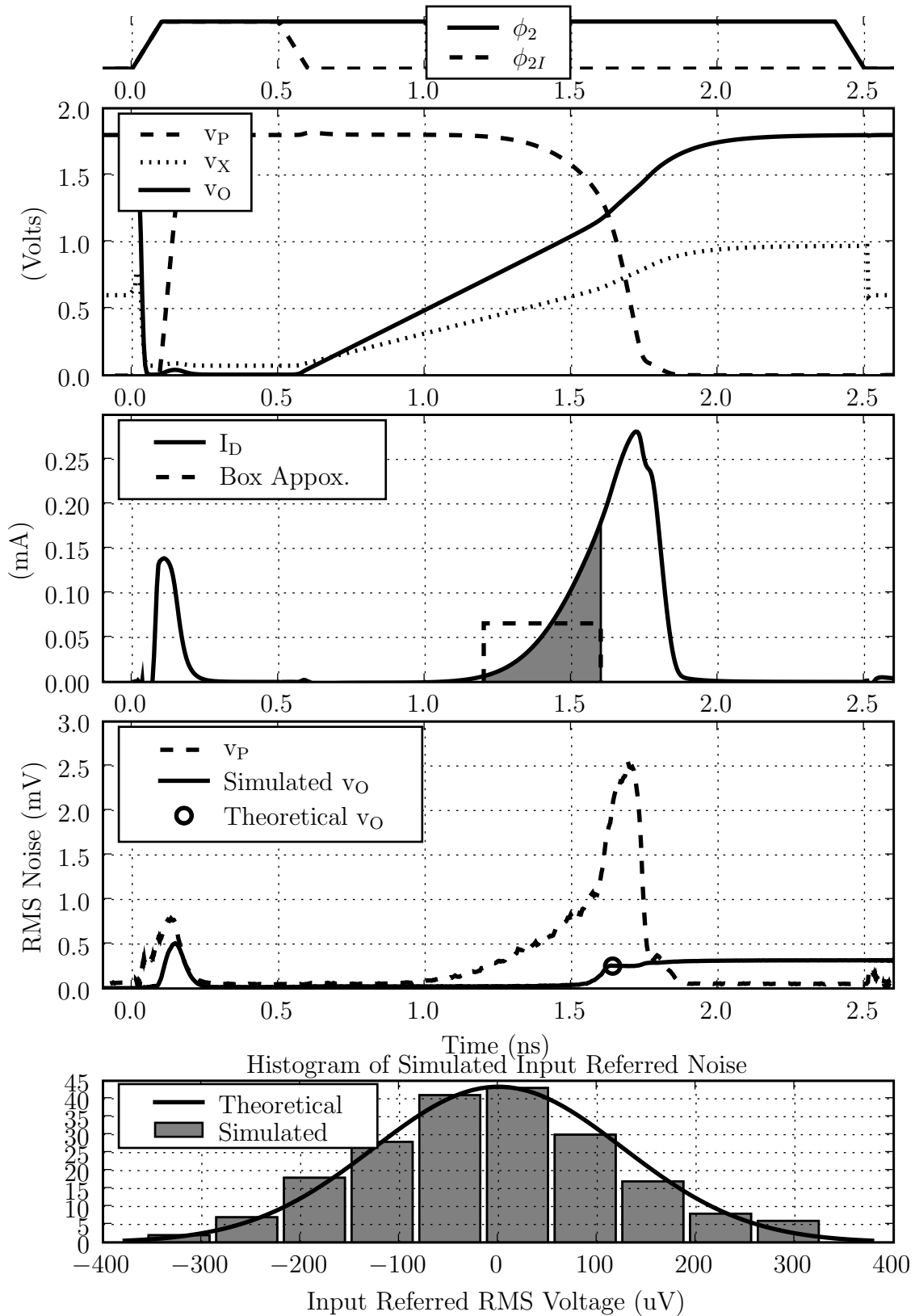


Figure 3-14: Simulated transient response used for noise analysis verification.

and equals

$$\bar{v}_{O,ZCBC}^2 = \frac{4kT}{3\bar{g}_m t_d}. \quad (3.1)$$

For this design with a ramp rate for 200MS/s operation, simulation shows  $t_d = 400\text{ps}$  and  $\bar{g}_m = 870\mu\text{S}$ . This gives  $250\mu\text{V}$  of RMS noise on the output<sup>†</sup>. To verify this result, a transient noise simulation was run with 200 parallel transient responses to yield the fourth plot of Figure 3-14. The dashed line shows the RMS noise on  $v_P$  and the solid line shows the RMS noise on  $v_O$  as a function of time. The noise on  $v_O$  is insignificant until  $v_P$  switches to open the sampling switch. After the switch opens the output referred noise rises to  $250\mu\text{V}$ , which matches the theoretical calculation. In this simulation noise generation is enabled in all devices including the current sources and switches, and this verifies that the DZCD noise is the dominant source of noise.

The final plot of Figure 3-14 shows the histogram of the input referred output voltage for the 200 parallel noise simulations. The theoretical Gaussian distribution is overlaid to show that the response is indeed approximated well by a Gaussian distribution.

One additional source of noise that is investigated in [8] is the positive feedback loop that exists during the transient response from  $M_1$  through  $M_3$  and back through capacitors  $C_{3,4}$  and  $C_2$ . The transient noise simulation for this implementation did not show this feedback loop contributed any significant increase in noise.

### More Rigorous DZCD Noise Analysis

The box approximation of the noise in the DZCD calculated in Equation 3.1 turns out to be equivalent to the result of a more rigorous derivation using both square-law and velocity saturated device characteristics. Such a derivation requires a transient noise analysis of device  $M_1$  of Figure 3-3.

Suppose the input voltage  $v_X$  into the DZCD is a ramp with slope  $a$ . If  $V_T$  is the threshold voltage of  $M_1$ , then the effective gate drive of  $M_1$  can be expressed as  $v_e = v_X - V_T$ . Assuming square-law device physics, the drain current of  $M_1$  can then

---

<sup>†</sup>The RMS voltage is obtained by taking the square root of Equation 3.1. To refer it to output requires multiplying the RMS noise by 2, which is the gain of the pipeline stage.



be expressed as

$$I_D = \kappa v_e^2 \quad (3.2)$$

where  $\kappa = \mu C_{ox} \frac{W}{2L}$ .

By defining the time when  $v_e = 0$  as  $t = 0$ ,  $v_e$  can further be expressed as

$$v_e = at, \quad (3.3)$$

and the transconductance of  $M_1$  can be calculated from Equations 3.2 and 3.3 as

$$g_m = 2\kappa at. \quad (3.4)$$

Since the output voltage  $v_P$  is reset to  $V_{DD}$  during the initialization phase,  $v_P$  will be at  $V_{DD}$  at time  $t = 0$ . The drain current  $I_D$  will begin to discharge  $v_P$  at  $t = 0$  according to the equation  $v_P = V_{DD} - \frac{1}{C_p} \int_0^t I_D dt$ , where  $C_p$  is the parasitic capacitance on the  $v_P$  node. Defining  $v_y = V_{DD} - v_P$  yields the transfer function from the drain current  $I_D$  to the effective DZCD output voltage  $v_y$  as

$$v_y = \frac{1}{C_p} \int_0^t I_D dt. \quad (3.5)$$

Combining this result with Equations 3.3 and 3.2 yields

$$v_y = \frac{\kappa a^2 t^3}{3C} \quad (3.6)$$

This shows that the linear input voltage ramp creates a squared current response and thus a cubic voltage response on the output.

Suppose the sampling switch  $M_3$  of Figure 3-3 has a switching threshold of  $V_{DD} - V_{tp}$ . Then the time  $t_d$  at which the DZCD detector switches is the time when  $v_y = V_{tp}$  and can be found by evaluating Equation 3.6 at  $t = t_d$  when  $v_y = V_{tp}$  and solving for  $t_d$ . This gives

$$t_d = \sqrt[3]{\frac{3C_p V_{tp}}{\kappa a^2}}. \quad (3.7)$$

$t_d$  is the time it takes  $M_1$  to switch  $v_P$  from  $V_{DD}$  and turn off the sampling switch and is thus the delay of the DZCD.

It is the noise on the output voltage  $v_y$  at time  $t_d$  that matters because it defines the sampling instance. This noise, however, is not stationary because the circuit is not in steady state. Since the channel current noise generated by  $M_1$  is integrated to produce the output voltage, the noise will grow as a function of time as a random walk. Specifically, suppose the that noise spectral density of the channel current  $I_D$  is  $N = \frac{8}{3}kTg_m$ , then using the current to voltage transfer function of Equation 3.5, the output noise at time  $t_d$  will be

$$\begin{aligned}\bar{v}_y^2 &= \frac{1}{C_p^2} \int_0^{t_d} \frac{N}{2} dt \\ &= \frac{8}{3}kT \frac{\kappa a t_d^2}{2C_p^2}\end{aligned}\quad (3.8)$$

From  $v_y$  the input referred noise of the output voltage  $v_O$  can be calculated as

$$\bar{v}_{O,ZCBC}^2 = \frac{\bar{v}_y^2}{A^2}\quad (3.9)$$

where  $A$  is the dynamic gain of the DZCD at time  $t_d$  and is the ratio of the DZCD output voltage slope to the input voltage slope  $a$  evaluated at the switching time  $t_d$ .

$A$  can be expressed as

$$\begin{aligned}A &= \left. \frac{\partial v_y / \partial t}{\partial v_e / \partial t} \right|_{t=t_d} \\ &= \frac{\kappa a t_d^2}{C_p}.\end{aligned}\quad (3.10)$$

Furthermore, the mean transconductance from time 0 to  $t_d$  can be calculated from Equation 3.4 as

$$\bar{g}_m = \kappa a t_d.\quad (3.11)$$

Combining Equations 3.8 through 3.11 gives

$$\bar{v}_{O,ZCBC}^2 = \frac{4kT}{3\bar{g}_m t_d}, \quad (3.12)$$

which is the same result calculated using the approximations to yield Equation 3.1. Following this same procedure under a velocity saturated region where  $I_D \propto v_e$  also yields the same result. The key quantities for weak inversion, square-law strong inversion, and linear strong inversion are summarized in Table 3.1.

Table 3.1: Summary of key DZCD quantities.

Definitions:  $\phi_t = \frac{kT}{q}$ ,  $n$  is weak inversion ideality factor,  $\kappa = \frac{W}{2L}\mu C_{ox}$  for square-law strong inversion,  $\kappa = v_{sat}WC_{ox}$  for velocity saturated strong inversion,  $a$  is input voltage slope,  $t$  is time,  $C_p$  is capacitance on output of DZCD,  $\bar{g}_m = \frac{1}{t_d} \int_0^{t_d} g_m(t) dt$  is the average transconductance from time 0 to  $t_d$ .

		Weak Inversion	Square Law	Velocity Sat.
Input Voltage	$v_e$	$at$	$at$	$at$
Output Current	$I_D$	$I_0 e^{\frac{v_e}{n\phi_t}}$	$\kappa v_e^2$	$\kappa v_e$
Output Voltage	$v_y = \frac{1}{C_p} \int_0^t I_D dt$	$\frac{n\phi_t}{aC_p} I_D$	$\frac{\kappa a^2 t^3}{3C_p}$	$\frac{\kappa a t^2}{2C_p}$
Time ( $v_y = V_{tp}$ )	$t_d$	$\frac{n\phi_t}{a} \ln \frac{aC_p V_{tp}}{I_0 n\phi_t}$	$\sqrt[3]{\frac{3C_p V_{tp}}{\kappa a^2}}$	$\sqrt{\frac{2C_p V_{tp}}{\kappa a}}$
Effective Gain	$A = \frac{\partial v_y / \partial t}{\partial v_e / \partial t}$	$\frac{I_D}{aC_p}$	$\frac{\kappa a t_d^2}{C_p}$	$\frac{\kappa t_d}{C_p}$
Transconductance	$g_m$	$\frac{I_D}{n\phi_t}$	$2\kappa a t$	$\kappa$
Output-referred NSD	$\frac{N_o}{2}$	$4kT \frac{n}{2} g_m$	$\frac{8}{3} kT g_m$	$\frac{8}{3} kT g_m$
Output-referred Noise	$\bar{v}_y^2 = \frac{1}{C_p^2} \int_0^{t_d} \frac{N_o}{2} dt$	$\approx \frac{N_o(t_d) n\phi_t}{C_p^2 a}$	$\frac{8kT\kappa a t_d^2}{3C_p^2}$	$\frac{8kT\kappa t_d}{3C_p^2}$
Input-referred Noise	$\bar{v}_i^2 = \frac{\bar{v}_y^2}{A^2}$	$\frac{N_o(t_d) a}{g_m^2(t_d) n\phi_t}$	$\frac{8kT}{3\kappa a t_d^2}$	$\frac{8kT}{3\kappa t_d}$
Input-referred NSD	$\frac{N_i}{2}$	$4kT \frac{n}{2} \frac{1}{g_m(t_d)}$	$\frac{8}{3} kT \frac{1}{g_m}$	$\frac{8}{3} kT \frac{1}{g_m}$
Input-referred Noise	$\bar{v}_i^2$	$\frac{N_i a}{2n\phi_t}$	$\frac{N_i}{2t_d}$	$\frac{N_i}{2t_d}$

### 3.5.2 Comparison to Original CBSC Implementation

In the original CBSC implementation described in [18] a general purpose comparator was used for the zero-crossing detection. The first stage of this comparator was a differential pair with a constant bias current. It was shown in [18, 41] that for this setup the noise bandwidth is  $\frac{1}{2t_i}$  where  $t_i$  is the delay of the the first stage of the comparator and can be expressed as  $t_i = \alpha \frac{T_{CLK}}{2}$ . Both devices of the input pair contribute noise and thus the input referred noise spectral density is  $\frac{16kT}{3g_m}$ , where  $g_m$  is the transconductance of the input devices biased at  $I_D$ . Thus the total noise for the original CBSC implementation is

$$\bar{v}_{O,CBSC}^2 = \frac{8kT}{3g_m t_i} \quad (3.13)$$

Since the static bias current drawn by the differential pair is  $2I_D$  for the entire half clock period  $T_{CLK}/2$ , the energy consumed by the input pair is

$$E_{CBSC} = V_{DD} I_D T_{CLK}. \quad (3.14)$$

The energy consumed for this ZCBC implementation is

$$E_{ZCBC} = V_{DD} \bar{I}_D t_d. \quad (3.15)$$

Multiplying the input referred noise together with the energy consumption gives a Noise-Energy product that tells how energy efficient each architecture is for a given noise. Assuming square-law device characteristics where  $g_m = \frac{2I_D}{V_{GS} - V_T}$ , the Noise-Energy product, NEP, of the CBSC implementation can be calculated by multiplying Equations 3.13 and 3.14 to give

$$NEP_{CBSC} = \frac{8kT}{3\alpha} V_{DD} (V_{GS} - V_T) \quad (3.16)$$

The NEP for this ZCBC implementation comes from multiplying Equations 3.1 and 3.15 to give

$$\text{NEP}_{\text{ZCBC}} = \frac{2kT}{3} V_{\text{DD}} (V_{\text{GS}} - V_{\text{T}}) \quad (3.17)$$

When  $\alpha = \frac{1}{2}$ , this ZCBC implementation operates 8x more power efficiently than the original CBSC implementation for the same noise level. The original CBSC implementation, however, does have the capabilities to be made fully differential, which would improve a Noise-Energy product normalized to the signal energy by a factor of 4. However, this derivation does not include the power that the additional gain stages in the original CBSC implementation would consume.

The original CBSC used a two phase ramping scheme where first a fast ramp provided a coarse charge transfer and then a slower ramp followed to provide a fine adjustment. The two phase approach improved the power efficiency of the differential pair input stage. The DZCD used in this implementation, however, does not consume static power, thus the dual ramp scheme does not offer the same benefit. Furthermore, a single ramp scheme simplifies the design and enables higher speeds. The trade off for using a single ramp scheme is that the current levels are higher at the sampling instant. Higher currents can reduce linearity and output swing. Since neither linearity nor output swing were limiting issues in this implementation due to the circuit techniques described in Section 3.3, a single ramp scheme was used to take advantage of the complexity reduction and speed improvements.

### 3.5.3 FOM Discussion

Input referring the  $250\mu\text{V}$  of DZCD noise calculated in Section 3.5.1 yields  $125\mu\text{V}$ , which for a 1V full scale input corresponds to 69dB of SNR (11 bit). The total input stage sampling capacitance is 50fF, which corresponds to  $287\mu\text{V}$  of  $\frac{kT}{C}$  noise or 62dB of SNR (10 bit). The total input referred noise from both of these contributions would be  $313\mu\text{V}$  or 61dB of SNR (9.8 bits). The measured SNR, on the other hand, is 40dB (6.4 bit), which is more than a factor of 10 lower than the theoretical and simulated SNR, and this extra noise raises the FOM by the same factor. This extra noise is

not likely fundamental but appears to be coming from power supply or substrate noise. As stated in Section 3.2, the DZCD is inherently single-ended, giving it limited rejection from these sources. A strong correlation is found between the I/O output driver voltage level and the noise floor. This indicates that noise induced from the output drivers is at least one source of this extra noise. Improved I/O driver design, less inductive packaging, and deep NWEELL implants for better substrate isolation are options that could reduce the impact of this noise and yield a higher SNR and improved FOM.

Given the correlation between the I/O voltage level and the noise floor, one other potential noise source would be code dependent noise on the power supply, ground, substrate, reference voltages, or/and bias voltages due to the asynchronous switching of each ZCBC stage. For example, if the DZCD of one stage switches just before another, ground bounce from switching one stage may corrupt the other.

The power consumption for the reference and bias voltages of this implementation is negligible because they are by-passed externally with large capacitors, and thus their power consumed was ignored in the previous discussions. In some applications, however, large external capacitors may not be practical and may require increased power consumption to generate the necessary reference and bias voltages.

The power consumption of the DZCD is simulated to be about 15% of the system power consumption. The digital power makes up approximately 66% of the total power consumption in this design. The Figure of Merit, therefore, for this implementation will improve in further scaled technologies as digital parasitic switching power consumption reduces. The rest of the power is consumed to switch various capacitors in the circuit including the sampling capacitors  $C_1$ ,  $C_2$ .

## 3.6 Conclusion

Zero-crossing based circuits were introduced as a generalization of comparator-based switched-capacitor circuits. Zero-crossing based circuits offer advantages over traditional opamp-based designs both from a theoretical power efficiency and from an

Table 3.2: ADC Performance Summary

Technology	0.18 $\mu$ m CMOS	
Area	0.05 mm <sup>2</sup>	
Input Voltage Range	1V (single ended)	
Power Supply: V <sub>DD</sub>	1.8V	
Sampling Frequency	100MS/s	200MS/s
DNL	$\pm 0.50$ LSB <sub>8</sub>	$\pm 0.75$ LSB <sub>8</sub>
INL	$\pm 0.75$ LSB <sub>8</sub>	$\pm 1.00$ LSB <sub>8</sub>
ENOB	6.9b	6.4b
Power Consumption	4.5mW	8.5mW
Figure of Merit: $\frac{P}{2f_{in}2^{ENOB}}$	0.38 pJ/step	0.51 pJ/step

amenability to scaling perspective. The implementation of an 8b, 200MS/s pipelined ADC was presented that demonstrates this generalization. It includes a dynamic zero-crossing detector that is fast, simple, and power efficient. Furthermore, current source splitting was introduced as means of removing series switches to improve linearity and output swing. Bit decision flip-flops were also used in place of traditional clocked comparators to improve speed and eliminate meta-stability issues.

There are two major outstanding issues with this implementation. One is that to make it production worthy, it requires a power efficient method of offset compensation. The other is that it has poor noise rejection performance. The remainder of this thesis describes techniques to deal with these issues.





# Chapter 4

## Chopper Offset Estimation

It is clear that the dynamic zero-crossing detector used to implement the ZCBC introduced in Chapter 3 requires offset compensation to make it production worthy. Therefore, this chapter presents a theoretical study of a generic circuit offset compensation technique called Chopper Offset Estimation (COE). Offset compensation is studied in this work not only because it is essential to making ZCBC circuits more practical but also because it offers additional advantages of improving signal range and reducing low frequency corruption such as flicker noise. As flicker noise increases [50] and signal range decreases with technology scaling, offset compensation is becoming increasingly important for realizing high resolution circuits such as ADCs in scaled technologies.

A thorough analysis of the traditional offset cancellation techniques of auto-zeroing (AZ), correlated double sampling (CDS), and chopper stabilization (CHS) is presented in [16]. The basic idea of auto-zeroing is to sample the offset during a calibration phase and subtract it from signal during the processing phase. Open-loop techniques of output offset storage (OOS) store the static offset at the output of the open-loop circuit. This technique can be impractical, however, when using high-gain amplifiers as the limited output range of an open-loop amplifier can be insufficient to capture the offset. Furthermore, OOS typically requires doubling the power consumption as the amplifiers must remain active during both the calibration and processing phases.

A traditional closed-loop technique for input offset storage (IOS) also exists that

eliminates the limited amplifier range issue. This technique, however, also doubles the power consumption as the opamp must remain active for a calibration phase. One further issue with this technique is that it also doubles the wide-band circuit noise power as the amplifier noise gets sampled twice—once to sample the offset and twice to sample the signal. Both the increased power consumption and noise adversely effect the power efficiency of the overall circuit.

Another auto-zeroing technique is multistage offset storage where several single-stage amplifiers are cascaded and cancelled in succession to reduce wide-band noise injection and eliminate residual charge injection errors [37]. An additional technique for applications utilizing transconductance amplifiers is to use an auxiliary transconductance amplifier to inject a closed-loop sampled offset current at the output of the amplifier to achieve offset compensation [3]. While these approaches have the advantage of not significantly increasing the wide-band circuit noise, they do require increased power consumption and are only applicable to very specific circuit implementations.

The development of zero-crossing based circuits (ZCBC) [6, 18] as an opamp-free method of switch capacitor circuit design has raised further compatibility issues with traditional offset compensation techniques. For example, the traditional method of closed-loop IOS is not compatible with ZCBC circuits because a ZCBC circuit cannot simultaneously drive both sides of the sampling capacitor. Furthermore, the power efficiency of ZCBC requires giving special consideration to any potentially power inefficient closed loop amplification stages to measure the offset. Lastly, the offset of concern in the zero-crossing detector in a ZCBC circuit is its dynamic offset, which is not necessarily equal to the static offset that can be measured using traditional auto-zeroing techniques.

One promising traditional technique for offset cancellation that is compatible with both traditional opamp-based and ZCBC circuits is Chopper Stabilization [23, 47]. This approach uses modulation, or chopping, prior to analog processing to frequency translate the input signal out of band with the offset and low frequency noise that gets introduced during analog processing. After processing, the signal is demodulated back

down and the low frequency noise and offset are filtered away. One advantage of this technique is that the amplifiers can be disabled or shared during the sampling phase to realize power savings over AZ techniques. Furthermore, unlike AZ implementations, it is indiscriminate to the sources of offset and is not susceptible to second order circuit issues such as charge injection or finite open-loop gain. CHS has been used in ADC applications [20, 24, 27, 48] which have the further advantage that the filtering can be performed digitally.

There are, however, several disadvantages to CHS. Since the amplifiers are not offset nulled, the reduced signal range due to the offset cannot be recovered. Moreover, the traditional CHS topology requires high performance filtering to remove the offset, and such filtering can be area and power intensive. In addition, extra sampling bandwidth is required to remove flicker noise.

To overcome these issues, a derivative technique called Chopper Offset Estimation (COE) is presented that retains the advantages of CHS but reduces the filtering requirements and can recover the complete signal range. Section 4.1 reviews traditional Chopper Stabilization in more detail and introduces the COE architecture. Section 4.2 studies the effects of using random chopping vectors in the COE architecture. Section 4.3 then introduces additional derivative COE architectures including input-referred COE that recovers the complete signal range and also several architectures specific to pipelined ADCs. Section 4.4 then concludes and summarizes the results of this work.

The work present here focuses on offset compensation for ADCs where the processing can be done digitally. Just as with CHS, however, much of this work is applicable to a purely analog circuits where analog processing can be used to remove the offset. In that scenario, however, it is critical that the signal be gained sufficiently prior to such analog processing to ensure that the input-referred offset and noise added by the additional analog processing is negligible.

## 4.1 Chopper Offset Estimation

### 4.1.1 Traditional Chopper Stabilization

A block diagram of traditional CHS for offset compensation for an ADC application is shown in Fig. 4-1. The input voltage  $v$  is modulated by a chopping vector  $p$  prior to quantization by the ADC. The vector  $p$  is a tone at  $f_s/2$ , where  $f_s$  is the sampling frequency. It takes the form

$$p = [+1, -1, +1, -1, \dots]$$

whose  $n^{\text{th}}$  element can be expressed as  $p_n = e^{-j\pi n}$ . After modulation the signal is quantized by the ADC. The ADC adds an unknown offset  $z$  while generating the digital output  $q$ . After quantization,  $q$  is digitally demodulated by the same chopping vector  $p$  and then low-pass filtered (LPF).

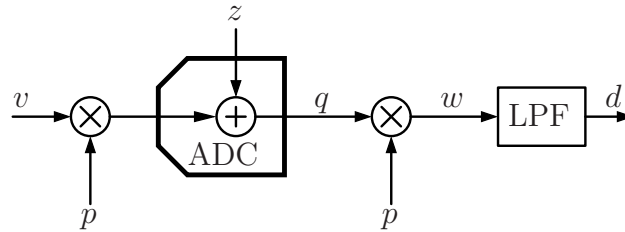


Figure 4-1: Traditional Chopper Stabilization for offset compensation

The frequency domain plots of Fig. 4-2 provide insight into the CHS method. The Discrete Time Fourier Transform (DTFT)  $V(f)$  of an example input voltage vector  $v$  is shown first. Likewise the DTFT  $P(f)$  of the modulation vector  $p$  is shown second. The third plot shows the DTFT  $Q(f)$  of the ADC output where the input signal has been modulated up to  $f_s/2$  and an impulse of area  $z$  has been introduced by the ADC. Observe that modulation shifted the signal out of band with respect to the offset. The final plot shows the DTFT  $W(f)$  after demodulation with vector  $p$  where the signal gets restored back to DC and the ADC offset gets modulated up to  $f_s/2$ . The final step in chopping is to low pass filter the demodulated output to remove the offset that has been shifted up to  $f_s/2$ .

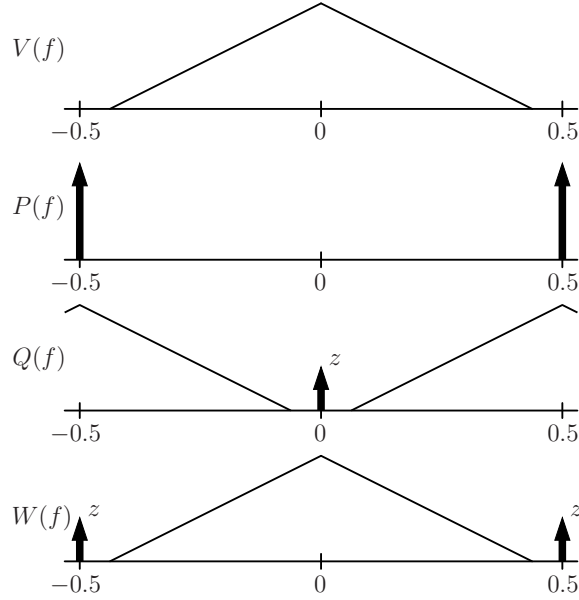


Figure 4-2: Frequency domain view of Chopper Stabilization

For a fully differential design, modulation by the vector  $p$  can be done with negligible hardware and distortion as it can be implemented with two extra switches that invert the input as appropriate [23]. The demodulation of the ADC output is a simple matter of digitally inverting the appropriate samples. The low pass filter, on the other hand, introduces a significant trade-off between hardware complexity and adequate frequency response. The LPF must be able to meet the frequency response requirements of the application in terms of transition band steepness, pass-band ripple, phase response, and latency while also trying to limit the extra sampling bandwidth requirement.

#### 4.1.2 Chopper Offset Estimation (COE)

Chopper Offset Estimation (COE) can be used to reduce the filtering requirements of traditional CHS. The block diagram manipulations in Fig. 4-3 introduce the initial transformations required to convert from CHS to COE. The first block diagram in Fig. 4-3 shows the traditional case where demodulation is followed by an LPF with frequency response  $H(f)$ . Shown to the right of this block diagram is an example ideal frequency response for  $H(f)$ . In the second block diagram, the filter and de-

modulation blocks have been swapped, and in order to maintain the same overall frequency response, the LPF must be frequency shifted to become a high-pass filter with response  $H(f + \frac{1}{2})$ . In the third and final block diagram, the high-pass filter is magnitude inverted to become an LPF of the form  $1 - H(f + \frac{1}{2})$  whose output is subtracted from the signal to preserve the same overall frequency response.

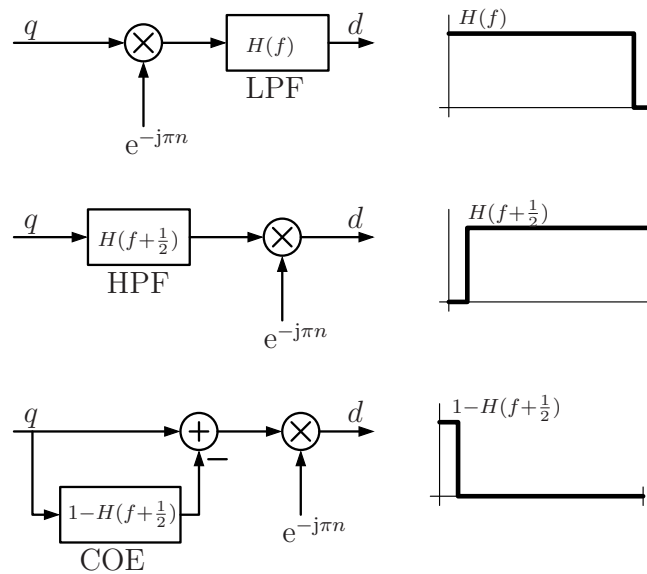


Figure 4-3: Block diagram manipulations with corresponding filter responses that all yield the same overall response.

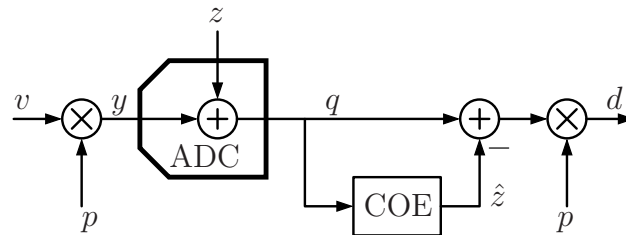


Figure 4-4: Alternate chopping technique utilizing a Chopper Offset Estimation (COE) block.

Applying these block diagram manipulations to traditional CHS yields the COE architecture shown in Fig. 4-4. The block labeled COE is a low-pass filter that produces an estimate  $\hat{z}$  of the ADC offset that is then subtracted from the signal prior to demodulation. In this form, one can see that if the offset estimate is perfect, i.e.  $\hat{z} = z$ , then the offset is removed immediately after it is added.

### 4.1.3 COE Decimation

As the filter responses of Fig. 4-3 show, traditional CHS must implement a wide-band low-pass filter whereas the COE must implement a narrow-band low-pass filter. This was realized in the ADC implementation in [27] where tunable single pole infinite impulse response (IIR) filter was used to implement the COE filter. While this approach is simple, it causes the frequency response of the ADC to have non-linear phase and ripple in the pass band. An alternative approach that can realize similar hardware savings with linear phase and more controlled pass band ripple is to employ polyphase decimation finite impulse response (FIR) filters [40] that sub-sample the offset estimate.

Consider an example where a  $N$  tap moving-average or box-car low-pass filter is designed to implement the COE. In this case, the COE would produce an offset estimate  $\hat{z}$  at the same rate as the ADC. Since the offset is of low bandwidth, however, a polyphase decimation filter that decimates by a factor  $N$  would reduce the tap requirement from  $N$  to 1. Such a polyphase decomposition requires a single register that accumulates the mean of the signal over a block length of  $N$ . If  $N$  is a power of 2, then the need for a multiplier is also eliminated as the required multiplication can be implemented via a bit shift.

With the selection of a proper interpolation filter, the impulse response of the system can be preserved exactly when the offset estimate  $\hat{z}$  is interpolated back up to the rate of the ADC. For the case of the box-car filter discussed previously, a zero-order hold register is such a filter that preserves the impulse response of the system exactly. Even though the impulse response of the system can be preserved, aliasing of the signal into the decimation band of the offset estimate will likely occur in all applications regardless of the decimation rate and filter selection. Since the magnitude of the aliasing, however, can be made arbitrarily low with appropriate decimation rate and filter design, a polyphase decimation implementation of the COE block will likely realize significant hardware savings regardless of the application.

## 4.2 Random Chopping Vector

The COE architecture of the Fig. 4-4 provides insight into an alternative modulation method. Instead of a deterministic tone for modulation, a random vector that is uncorrelated with the input can be used. As will be discussed later, this allows for full bandwidth input signals and enables further COE architectures.

Consider the case when the chopping vector  $p$  is selected as a random Bernoulli vector whose elements are independent and randomly selected from the set  $\{1, -1\}$  such that  $p$  and  $v$  remain uncorrelated. Further suppose that the elements of  $v$  are also identically distributed with a probability density function  $f_v(x)$ . When  $v$  is independent of  $p$ , the resulting distribution of  $y$  will be

$$f_y(x) = \frac{1}{2}(f_v(x) + f_v(-x)). \quad (4.1)$$

Sending this through an ADC with offset  $z$  will then shift the distribution by  $z$  to yield the distribution of  $q$  as

$$f_q(x) = \frac{1}{2}(f_v(x - z) + f_v(z - x)). \quad (4.2)$$

The plots of Fig. 4-5 provide an example of each of these probability distributions. In the first plot a sample distribution  $f_v(x)$  is provided for an input voltage  $x$ . The second shows the distribution  $f_p(x)$  for the elements of  $p$ . The third is the probability distribution  $f_y(x)$  after modulation, and the fourth shows how the ADC offset shifts that result by  $z$  to produce the distribution  $f_q(x)$  of the ADC output.

Eq. 4.1 reveals the important property that modulating with a random Bernoulli vector produces a distribution that is even and thus zero-mean, regardless of the shape of the input voltage distribution  $f_v(x)$ . Furthermore, since  $p$  is a white vector, chopping will whiten the input to produce a white random vector  $y$ . Thus, chopping with a random vector  $p$  generates a white, zero-mean process whose variance equals the expected square value of the input. The ADC then adds an unknown offset  $z$  to this random vector  $y$  to shift the mean.



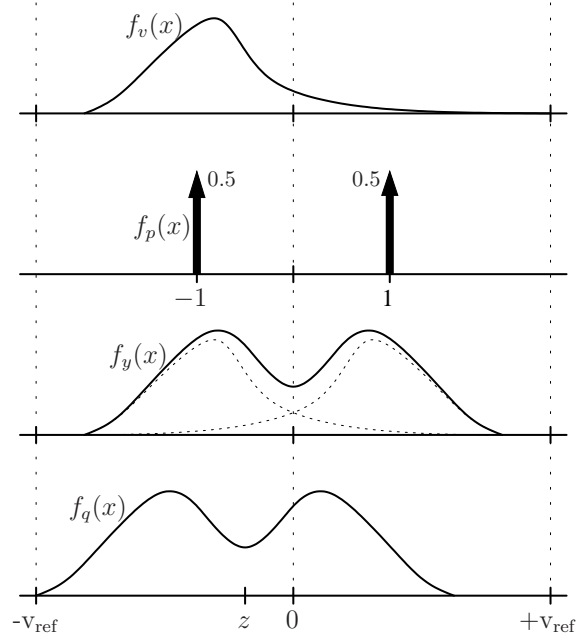


Figure 4-5: Sample probability density functions of signals when chopping vector  $p$  is a random Bernoulli vector.

Since the ADC output is

$$q = y + z, \quad (4.3)$$

where  $y$  is a zero-mean, white random vector, then estimating  $z$  based on observations of  $q$  is a classic estimation problem of an unknown parameter corrupted with additive noise.

### 4.2.1 Minimum Variance Linear Unbiased Estimator

Consider the minimum variance linear unbiased (MVLU) estimator. Given a block of  $N$  samples, a linear estimator will take the form

$$\hat{z} = \alpha \cdot q, \quad (4.4)$$

where  $\alpha$  is a linear weighting vector of length  $N$ . Further requiring the estimator to be unbiased\* yields a constraint on the elements of  $\alpha$  of

$$\sum_{n=1}^N \alpha_n = 1. \quad (4.5)$$

Using this constraint and the fact that the samples of  $q$  are uncorrelated, the variance of the estimation error  $e_z = z - \hat{z}$  can be found to be

$$\text{E}[e_z^2] = N\sigma_v^2(\alpha \cdot \alpha). \quad (4.6)$$

Lagrange multipliers can be used to find the  $\alpha$  vector that minimizes Eq. 4.6 subject to the constraint of Eq. 4.5. The result is the MVLU estimator

$$\hat{z}_{\text{MVLU}} = \frac{1}{N} \sum_{n=1}^N q_n = \bar{q}, \quad (4.7)$$

which means all elements of  $\alpha$  are equal to  $\frac{1}{N}$ . The MVLU estimator simply calculates the mean of the ADC output vector. When implemented in real time in a streaming fashion, this is equivalent to a moving-average or box-car filter. Thus, as discussed previously, polyphase decomposition can be applied to the MVLU estimator to realize the same significant hardware savings.

The variance of the MVLU estimator can be found by substituting  $\alpha_n = \frac{1}{N}$  into Eq. 4.6 to give

$$\text{E}[e_z^2] = \frac{\sigma_v^2}{N}. \quad (4.8)$$

Observe that the MVLU estimator as defined in Eq. 4.7 is independent of the probability distribution of the input signal  $f_v(x)$  while the performance as defined in Eq. 4.8 is a function of the variance provided by  $f_v(x)$ .

When one considers the ADC input vector  $y$  as a wide-band noise source that

---

\*An unbiased estimate  $\hat{z}$  has an expected estimation error of zero

$$\text{E}[z - \hat{z}] = 0.$$

corrupts the observations of the ADC offset  $z$ , the MVLU estimator filters the ADC output with a box-car filter of length  $N$  to reduce the bandwidth and thus the noise. Other filters can be used besides the box-car if another frequency response is desired, but the box-car filter produces the minimum variance estimator as it reduces the bandwidth most significantly for a given number of taps.

### 4.2.2 MVLU Performance

Since error in the offset estimate has unity gain to the output, the noise  $\sigma_d^2$  that results on the output vector  $d$  due to the offset estimation error is the same as the offset estimation variance as derived in Eq. 4.8 and is

$$\sigma_d^2 = \frac{\sigma_v^2}{N}. \quad (4.9)$$

Achieving an equivalent  $B$  bit noise level for a given input can be obtained by substituting the traditional quantization noise expression  $\sigma_d^2 = \frac{V_{\text{FS}}^2}{2^{2B}12}$  into Eq. 4.9 and solving for  $N$  to give the block length constraint

$$N = 2^B \sqrt{12} \frac{\sigma_v}{V_{\text{FS}}}, \quad (4.10)$$

where  $V_{\text{FS}}$  is the full-scale input voltage range of the ADC. Since the bandwidth of the offset estimate is inversely proportional to the block length, this result reveals the fundamental constraint between the resolution and bandwidth of the offset estimate when using a random chopping vector. Depending on the power spectral density of the input, it is possible to use a shaped random chopping vector [49] to realize better performance.

### 4.2.3 MVLU Example

As an example, consider the case when the input  $v$  is a random vector whose elements are independent and uniformly distributed over the entire range of the ADC. The

mean of such an input is 0 and the variance  $\sigma_v^2$  is

$$\sigma_v^2 = \frac{V_{\text{FS}}^2}{2^{2B}12}. \quad (4.11)$$

Substituting this result into Eq. 4.10 gives the block length requirement

$$N = 2^{2B}. \quad (4.12)$$

If 10 bits of accuracy is required for the offset estimate, the block length should be  $N = 2^{20} \approx 1$  mega-sample (MS). If the sampling rate of the ADC is 100MS/s, then the offset update rate is 100 Hz. Thus, any noise generated within the ADC with a bandwidth less than 100 Hz will be removed by this chopping technique. Since a full scale uniformly distributed input is rather extreme, this example provides a conservative view of the capabilities of COE bandwidth requirements.

The results derived for this example in Eq. 4.12 have been verified in simulation with an ideal 10 bit ADC (i.e.  $z = 0$ ). The input vector  $v$  is set as a full-scale uniformly distributed random vector. With a block length  $N = 2^{20}$ , 100 blocks of data are sent through the ADC. The offset estimate of each block is shown in Figure 4-6. As expected, this choice of input and block length yields a 10 bit accurate estimate of the offset.

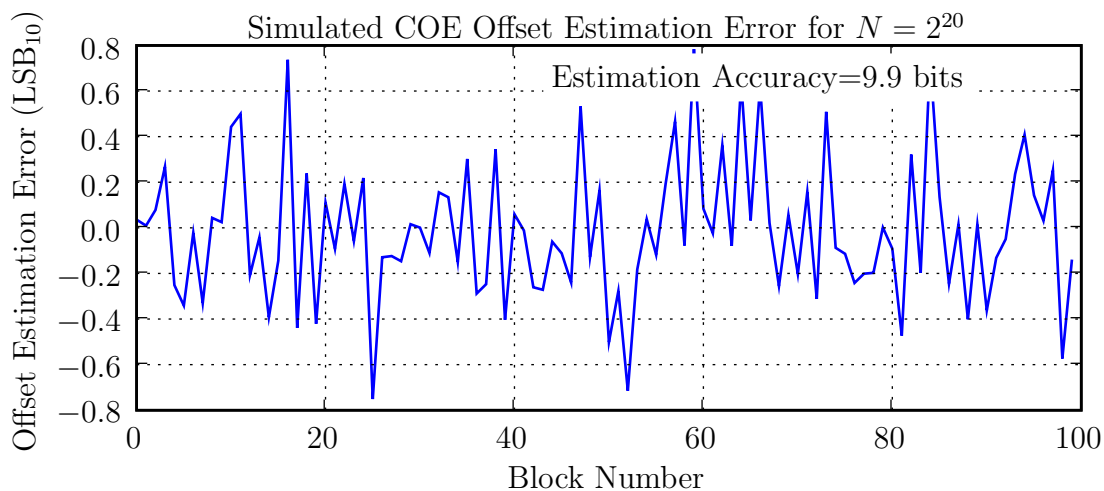


Figure 4-6: Simulated offset estimate.

#### 4.2.4 Distortion Performance

Suppose that in addition to an offset that the ADC has static non-linearities defined by a distortion function  $g(x)$  that includes both the offset and quantization introduced by the ADC. The  $n^{\text{th}}$  ADC output sample is then

$$q_n = p_n v_n + g(p_n v_n). \quad (4.13)$$

If the distortion is much smaller than the signal, then its effect on the offset estimate will be negligible. Chopping in the presence of distortion, however, affects the output. The distortion function  $g(x)$  can be decomposed into its even and odd components,  $g_e(x)$  and  $g_o(x)$  respectively, to give  $g(x) = g_e(x) + g_o(x)$ . The output after demodulation can then be found to be

$$d_n = v_n + g_o(v_n) + p_n g_e(v_n).$$

This shows that odd order distortion gets passed unaltered through the system but that even order distortion gets modulated by the chopping vector. The sample frequency responses of a simulated ADC in Figures 4-7 and 4-8 show this effect. The ADC in this simulation is given second and third order distortion. In the first plot chopping is disabled, and a second and third harmonic result from a pure tone input. In the second plot, random chopping is enabled, and the second harmonic, which is generated by the second order even distortion, gets modulated by the chopping vector and spread out as temporal noise. As these plots show, this modulation of the even order distortion does not change the effective number of bits (ENOB) or signal to noise and distortion ratio (SNDR) but does change the signal to noise ratio (SNR) and spurious free dynamic range (SFDR) as it converts even-order distortion into temporal noise. Observe also that chopping does not increase the quantization noise power as quantization is an odd order distortion.

When the chopping vector is not random but a deterministic tone at  $f_s/2$ , the even order harmonics get modulated by the tone to get frequency inverted. Thus any

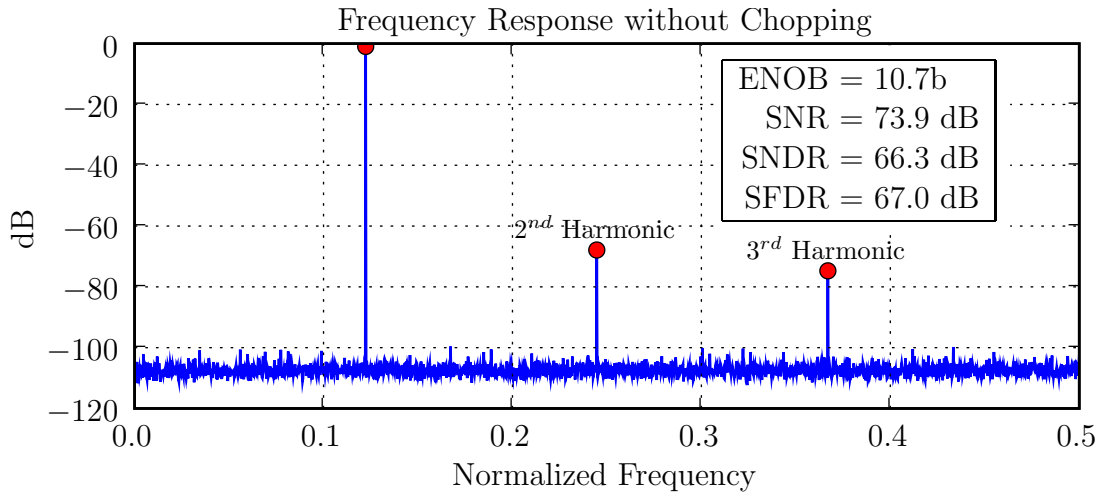


Figure 4-7: Frequency response of an ADC with second and third order distortion. Chopping disabled.

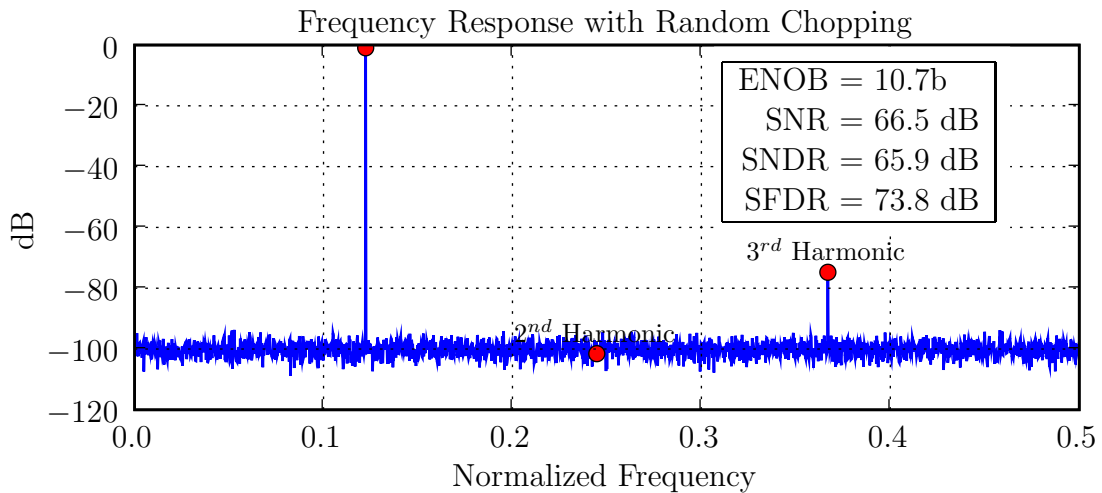


Figure 4-8: Frequency response of an ADC with second and third order distortion. Random chopping enabled.

tone produced from even order distortion has the same magnitude but gets moved to a different frequency.

#### 4.2.5 Random vs. Deterministic Chopping

From an implementation perspective, generating a deterministic chopping tone at  $f_s/2$  requires a single register with an inverter in feedback. Generating a random vector can also be done efficiently with a short series of registers and XOR gates [38]. Thus neither approach carries significant complexity, hardware, or power consumption

overhead in generating the chopping vector.

Bandwidth, however, is an area with a clear distinction in performance when comparing deterministic versus random chopping. Deterministic chopping requires a band-limited input signal so that ADC offset and low frequency noise can be injected out of band with the signal. It can remove offset noise of arbitrary bandwidth as long as the bandwidth of the input signal is narrow enough to ensure that the signal and offset do not overlap in frequency content. Random chopping, on the other hand, does not require limiting the bandwidth of the input signal but does limit the bandwidth of the offset estimate as governed by Eq. 4.10 to achieve the necessary accuracy.

Consider the previous example of offset compensating a 10 bit, 100MS/s ADC. Suppose that flicker noise needs nulled up to a corner frequency of 1MHz. It was found previously that for a full scale uniformly distributed input that random chopping required limiting the offset estimate to a 100 Hz bandwidth, thus random chopping is not a good candidate for this application. Instead, deterministic chopping requires limiting the bandwidth of the input to at least 49MHz to give 1MHz of bandwidth to the flicker noise. On the other hand, if the ADC is implemented with low flicker noise devices, such as bipolar transistors, then random chopping may be more appropriate as it does not require limiting the bandwidth of the input signal and enables additional architectures such as multistage chopping as will be discussed later. Thus the trade offs of each approach must be evaluated under the bandwidth requirements of the application to determine which is most appropriate.

### **4.3 Additional COE Architectures**

In addition to offering significant hardware savings, the technique of using the COE to estimate the ADC offset offers several derivative architectures that provide further advantages.

### 4.3.1 Input Referred Offset Compensation with COE

One disadvantage of traditional CHS is that the offset is not removed in the analog domain, so the offset reduces the available signal range. Since the COE block, however, produces a digital offset estimate  $\hat{z}$ , an alternative offset correction scheme is to pass the offset estimate to an Offset Controller (OC) to null the offset in an input-referred fashion as shown in Fig. 4-9. The OC must generate an analog estimate  $\hat{z}_a$  of the input referred offset based on the digital measurement of the offset at the output.

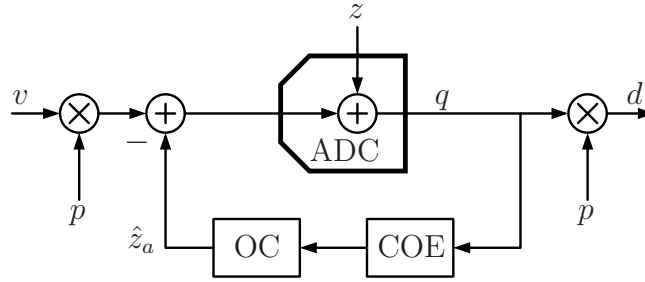


Figure 4-9: Block diagram using Chopper Offset Estimation (COE) and Offset Controller (OC) blocks to null the ADC offset in the analog domain.

The offset controller can be implemented in any number of ways. A simple charge-pump based method is shown in the schematic of Fig. 4-10. The series capacitor  $C_1$  stores the offset. After receiving a block of  $N$  samples, the COE makes an offset estimate. Capacitor  $C_2$  is then be charged to either positive or negative full scale depending on the polarity<sup>†</sup> of the offset estimate.  $C_2$  is then shorted to  $C_1$ , which causes an increase on the voltage across  $C_1$  when the measured offset is positive and decrease when the measured offset is negative. Since  $C_1$  is in the inverted channel of the ADC, the offset on  $C_1$  is subtracted from the input signal as desired. The ratio of  $C_1$  to  $C_2$  should be large enough that the offset voltage across  $C_1$  changes less than a single LSB in magnitude when in steady state. This ratio provides additional filtering of the offset estimate such that the block length  $N$  can be reduced by the same ratio to achieve the desired noise performance.

Observe that one potential issue with this approach is that the single series ca-

<sup>†</sup>The Signum() function calculates the sign of its input to determine if the input is positive or negative.



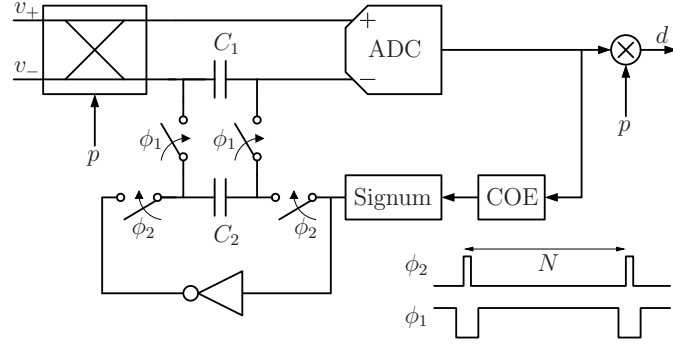


Figure 4-10: Example charge-pump based input referred COE offset compensation implementation.

capacitor on the inverted input causes a shift to the common mode of the input signal. A more balanced approach with a series capacitor on each path that subtracts equal and opposite amounts from each path could be used. Also observe that second order offset effects such as charge injection from the switches are nulled with this approach because the offset estimate is based on the digital output which includes all offset sources.

### 4.3.2 COE for Pipelined ADCs

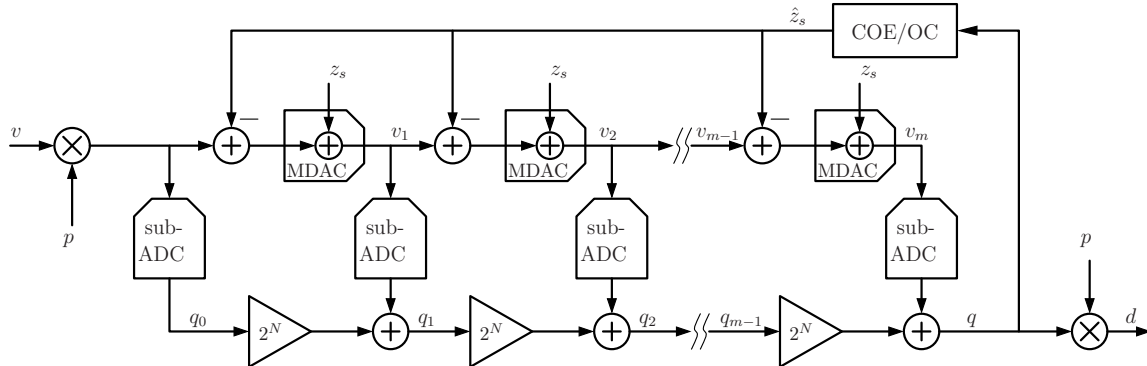


Figure 4-11: Block diagram of an  $m$  stage pipelined ADC with identical COE offset correction distributed to each pipeline stage.

An input referred COE approach was applied to a two-step ADC architecture in [48]. Consider the following methods of applying it to pipelined ADCs. It is critical with input referred offset compensation that the offset correction be injected at the appropriate spot in the analog processing chain to recover the signal range that

is lost due to the offset. For example, consider a 1.0 bit/stage pipelined ADC where a systematic offset  $z_s$  affects each stage. In this case, the total input referred offset  $z$  will be the weighted sum of the offsets of each stage according to

$$z = z_s + \frac{1}{2}z_s + \frac{1}{4}z_s + \frac{1}{8}z_s + \dots,$$

where each additional stage contributes half as much as the previous.

For a pipelined ADC that is dominated by systematic offset, an appropriate offset cancellation approach is to distribute the same offset correction factor to the input of each stage as shown in the block diagram of Fig. 4-11. A ZCBC implementation is an example of a pipelined ADC that is likely to be dominated by systematic offset. The reason is that each stage will suffer from a voltage ramp overshoot due the finite delay of the zero-crossing detector. Even if there is random offset variation for each stage, this approach will null the complete ADC offset.

### 4.3.3 Per-Stage COE for Pipelined ADCs

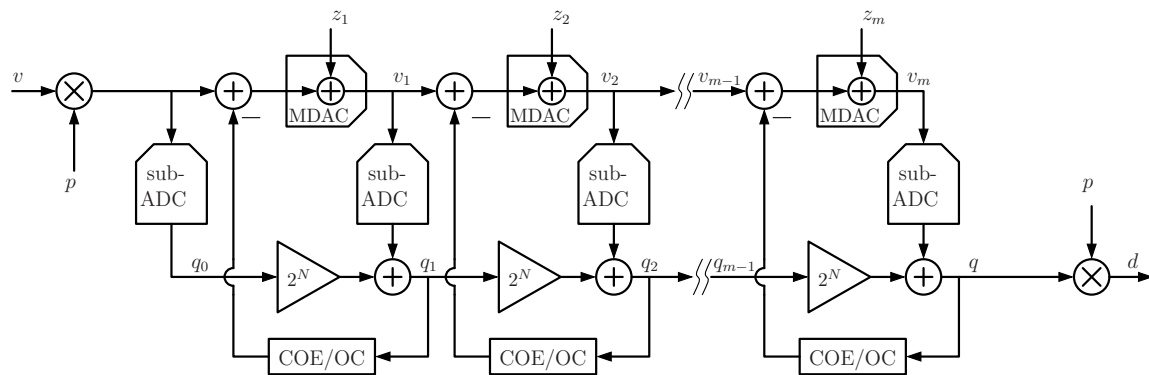


Figure 4-12: Block diagram of an  $m$  stage pipelined ADC utilizing individual Chopper Offset Estimate (COE) and Offset Control (OC) blocks for each stage for per-stage offset compensation.

If random offset variation in each pipeline stage is appreciable, in order to realize signal range recovery for each stage, it is necessary to offset compensate each stage individually. One such technique for providing per-stage offset compensation is shown in Figure 4-12. Here the Chopper Offset Estimator (COE) and the Offset Controller

(OC) have been replicated for each stage. The COE uses the output of each sub-ADC as the measurement vector of the offset for the preceding stage.

While the block diagram of Figure 4-12 offset compensates every stage, this may not be necessary and/or practical as the offset contribution of the last stages may be smaller than the resolution of the ADC and not worth compensating. In practice one may only provide individual compensation to the first few stages and then either cumulatively offset compensate or skip offset compensating the final stages.

One issue with this technique of per stage offset compensation comes from quantization noise. The signal into the COE of the first stage is a very low resolution signal. For example, in the case of a 1.0 bit/stage ADC, the signal  $q_1$  used for estimation is only 2 bits wide. This means that for the estimate to be valid, the input must provide plenty of dither across quantization boundaries. When that is the case, then the quantization noise becomes an additional noise source whose variance must be considered when deciding the block length.

Offset in the bit decision comparators (BDCs) that implement the sub-ADC of each stage poses another issue for this technique. When redundancy is used in a pipelined ADC, the offset in the BDCs gets corrected by the later stages; however, in the case of per-stage COE, the bits from the later stages are not used in the estimation. Although auto-zeroing the BDCs using traditional approaches can be done more power efficiently than auto-zeroing an opamp or zero-crossing detector, there are several other options to consider to diminish the effect of their offset.

Since chopping is performed by switching the fully differential inputs, the BDC offsets will actually produce equal and opposite errors in the COE accumulation register. Thus, if the probability density of the input of the ADC is symmetrically distributed, the BDC offsets will cancel out and not produce a residual bias on the ADC offset estimate. Furthermore, the BDCs offsets cause problems only to the degree of asymmetry in the ADC input source.

In situations where the input cannot be guaranteed to be symmetric and where BDC offsets will be problematic, a hybrid foreground and background calibration approach can be used on each stage. During startup, a symmetric calibration signal can

be sent into the ADC so that per-stage offsets can be measured and removed. Then after startup, digital calibration of the output according to Fig. 4-4 can continue to null flicker noise and track any parameter drift. With coarse offset cancellation performed in the analog domain in the foreground and fine precision cancellation performed in the digital domain in the background, as long as parameter drift is not excessive, this approach achieves near maximal signal range recovery without suffering from calibration drift. If offset estimation bandwidth is not a critical design constraint, the COE and OC logic can be multiplexed between stages to realize additional hardware savings.

### 4.3.4 Multistage Chopping

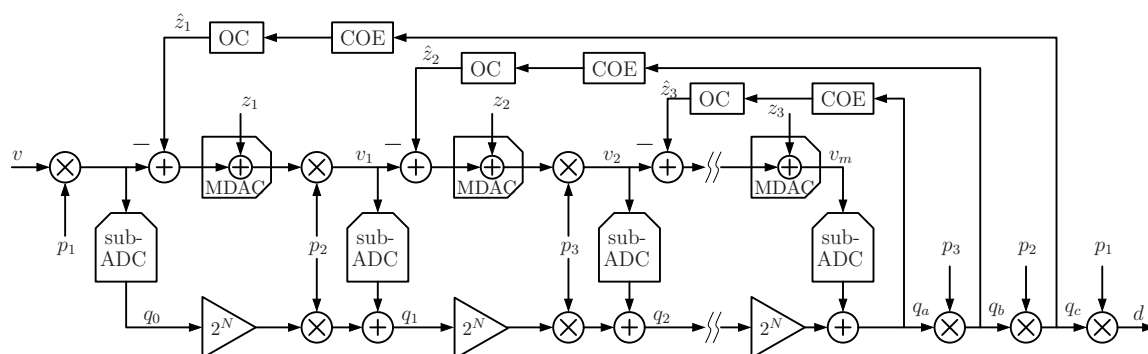


Figure 4-13: Block diagram of an  $m$  stage pipelined ADC utilizing multistage chopping vectors to estimate and null the offset of each stage individually.

An alternative method of offset compensating each stage individually that avoids the BDC offset issues is a multistage chopping technique as shown in Fig. 4-13. Here uncorrelated chopping vectors  $p_1$ ,  $p_2$ , and  $p_3$  are used to modulate the input of each stage. The output is demodulated in reverse order with individual COE and OC blocks supplied to each stage. Suppose the MDAC of each stage adds an offset  $z_1$ ,  $z_2$ ,  $z_3$ ,  $\dots$ . The outputs  $q_a$ ,  $q_b$ , and  $q_c$  will equal

$$q_a = vp_1p_2p_3 + z_1p_2p_3 + z_2p_3 + z_3$$

$$q_b = vp_1p_2 + z_1p_2 + z_3p_3 + z_2$$

$$q_c = vp_1 + z_2p_2 + z_3p_3p_2 + z_1$$

Since the offset of each stage is isolated by each chopping vector, each COE will produce a unique offset estimate that corresponds to the offset injected by each stage. Thus the random offset of each stage can be independently nulled to maximize the available signal range. Since this technique requires each chopping vector to be uncorrelated or orthogonal with the others, this technique is not compatible with deterministic chopping and requires random chopping vectors.

## 4.4 Conclusion

In summary, COE provides a method of offset compensation with several key advantages. When compared with traditional CHS, COE can use polyphase filtering techniques to significantly reduced hardware requirements and can use feedback to null the offset at its source to maximize the signal range. When compared to other offset compensation techniques, COE has clear advantages in that it estimates the actual and complete offset out the output and includes all sources of offset, whether it is from charge injection or device mismatch or whether it is dynamic or static offset. This is what makes it compatible with a very broad class of circuit architectures including both opamp-based and zero-crossing based circuits. Thus, COE is a very general and power efficient offset compensation technique well suited to deal with the traditional and newer circuit architectures and also the challenges of device and voltage scaling.



# Chapter 5

## ZCBC Revisited

The major limitations of the initial ZCBC design presented in Chapter 3 included the lack of a power efficient offset compensation technique and the poor noise performance. To implement the Chopper Offset Estimation ideas presented in Chapter 4 and to develop ZCBC circuits with improved noise rejection, a second ZCBC design was implemented in IBM's 90nm CMOS process. The design goal was a 50 MS/s, 12 bit pipelined ADC.

This chapter is organized as follows: System level changes are discussed in Section 5.1. The circuit changes including a fully differential implementation, voltage reference switch improvements, increased redundancy for increased signal range, and switched capacitor sampling techniques for low offset bit decision comparators are discussed in Sections 5.2, 5.3, 5.4, and 5.6 respectively. The complete circuit including these changes is then introduced in Section 5.5. A noise analysis is provided in Section 5.7, and finally the chip results are presented in Section 5.8.1.

### 5.1 System Level Improvements

Because the exact noise source(s) that gave the initial ZCBC design such poor noise performance were not isolated, many precautions were taken with this second design to try and eliminate them. While the later sections in this chapter discuss the circuit techniques developed to improve the robustness of the ZCBC circuits to system

conditions, for this design the following system level changes were implemented to both study the sensitivities of the circuits to the system as well as to provide the best possible system conditions to accomplish the design goal.

### **5.1.1 Embedded SRAM and Programmable Output Drivers**

To be able to study how much noise from the chip output drivers of the digital output codes couples back into the ADC, an embedded SRAM was implemented on this chip. The SRAM is capable of storing a 16K block of continuous samples from the ADC while the output drivers are disabled. While the SRAM will consume power that can cause power supply and/or substrate noise for the ADC, the power consumption of the SRAM will be significantly less than the output drivers. In addition, the output drivers have been implemented with eight programmable drive strengths to further study the effect of output driver coupling to the ADC.

### **5.1.2 Triple Well for Improved Substrate Isolation**

The trend toward SoC design has brought with it a trend toward manufacturers offering a triple well option to provide better substrate isolation between different circuit blocks. By providing a deep NWELL implant, PWELLS can be put inside an NWELL to create NMOS devices with an isolated bulk. This option was used on this chip to put the I/Os and ancillary digital logic like the embedded SRAM in isolated PWELLS. Furthermore, each zero-crossing detector from each stage was also put in an independent well to minimize any interstage substrate coupling due to the asynchronous nature of ZCBC switching.

### **5.1.3 On-chip Bias and Voltage Generation**

On the first chip, all voltage references and current biases were generated on the PCB for maximal configurability. This includes the ability to use large bypass capacitors to minimize noise and power consumption. The problem with this approach is that bond wire inductance can cause ringing and allow bounce on the on-chip signals. To



eliminate this possibility on this second chip, all bias, voltage, and current generation were done on chip with DACs. A 64 byte register file was implemented on-chip to program the state of these DACs as well as the rest of the internal configuration options. This register file uses a serial interface requiring two input pins to configure.

Only the ADC reference voltages  $V_{\text{refp}}$  and  $V_{\text{refm}}$  are implemented off-chip on this design. This was done for simplicity as on-chip reference generation requires further research to develop power efficient methods that can meet the stringent design requirements of the reference voltages. The following steps were used help to minimize the effects of bond-wire inductance causing ringing and bounce on the on-chip references.

**Fully Differential:** Because the design is fully differential, the ringing and bounce on the reference voltage is largely symmetric and thus common mode.

**Large On-chip Bypass Capacitance:** A large metal finger bypass capacitor that uses all available 8 metal layers that is approximately 1 nF and consumes approximately 0.5mm x 1.5mm of area was put between  $V_{\text{refp}}$  and  $V_{\text{refm}}$ . The design is pad limited and this capacitor consumed all available spare area.

**Adjacent Pads:** Since the current draw on the reference voltages flows in through  $V_{\text{refp}}$  and out through  $V_{\text{refm}}$ , these pads were located adjacent to each other on the chip to minimize the area of the current loop. Furthermore, power and ground pads were also placed adjacent to  $V_{\text{refp}}$  and  $V_{\text{refm}}$  pads to minimize the loop for any current that flows from these signals into the power supply.

**Off-center Packaging:** As shown in the bonding diagram of Figure 5-1, the die has been placed off center in the package to minimize the length of the  $V_{\text{refp}}$  and  $V_{\text{refm}}$  bond wires. The length of the bond wires for these pins is less than 1mm. The digital data bus pins were made longer, but with the embedded SRAM and programmable output driver strength, the risk to these signals is minimal.

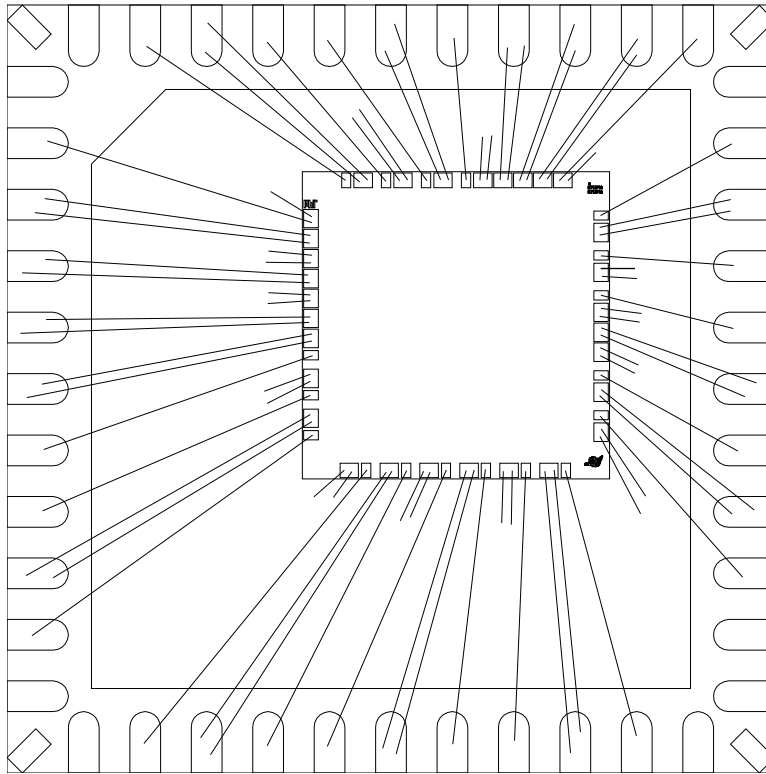


Figure 5-1: Bonding Diagram of Second ZCBC Chip.

#### 5.1.4 Single Ground

The initial ZCBC design featured a non-EPI substrate with three isolated grounds for analog, digital, and I/O circuits. This approach makes sense when the ground reference is off-chip as it isolates the current for independent circuits to independent paths. This means that ground bounce on one circuit cannot effect the others.

When putting all the bias and voltage generation on-chip, however, chip ground becomes the reference, and in order to create as low as impedance ground on chip as possible, all grounds are connected together on-chip on this design. As will be discussed next, ground is also down-bonded on this chip to an exposed paddle which also significantly reduces impedance on the ground network to significantly reduce ground bounce.

### 5.1.5 Packaging Considerations

The smallest possible package was desired for this design to keep the bond wire length as short as possible. A small 48 pin QFN package that is 5mm x 5mm on a side was selected. This is a leadless package with 0.4mm pitch pins. To keep the number of die pins below 48, the digital output bus was made DDR (double data rate) to reduce the databus count to 10 pins. Another advantage of this QFN package is that it has an exposed paddle to which ground is down-bonded directly as shown in the bonding diagram of Figure 5-1.

These packaging changes represent significant changes over the initial ZCBC design which was packaged in a 10mm x 10mm package. Since that die was centered in that package, all signals traveled through bond wires that were at least 4.5mm. Furthermore, it did not have an exposed paddle, so ground also travelled through these long bond wires as well. When compared to the packaging of this second chip, the critical analog nets such as  $V_{\text{refp}}$  and  $V_{\text{refm}}$  are bonded with wires less than 1mm and ground is down bonded directly to the exposed paddle.

## 5.2 Fully Differential ZCBC

When compared to its single-ended counterpart, a fully differential circuit implementation typically doubles the signal amplitude without effecting the noise level. A 2x amplitude gain results in a 4x signal power gain, so the SNR of a fully differential circuit will be 4x that of its single-ended counterpart. Furthermore, if the common-mode feedback circuit of a fully differential implementation does not consume significant power, a fully differential implementation will consume the same power as that of its single-ended counterpart. In total, a 4x SNR increase without an increase in the power consumption yields a 2x improvement in the Figure of Merit (FOM). Coupled with the opportunities for better power supply and substrate noise rejection, a fully differential approach was taken on this second ZCBC design.

A simplified schematic of two of the fully differential ZCBC pipeline stages designed for this chip is shown in Figure 5-2. The corresponding timing diagram is

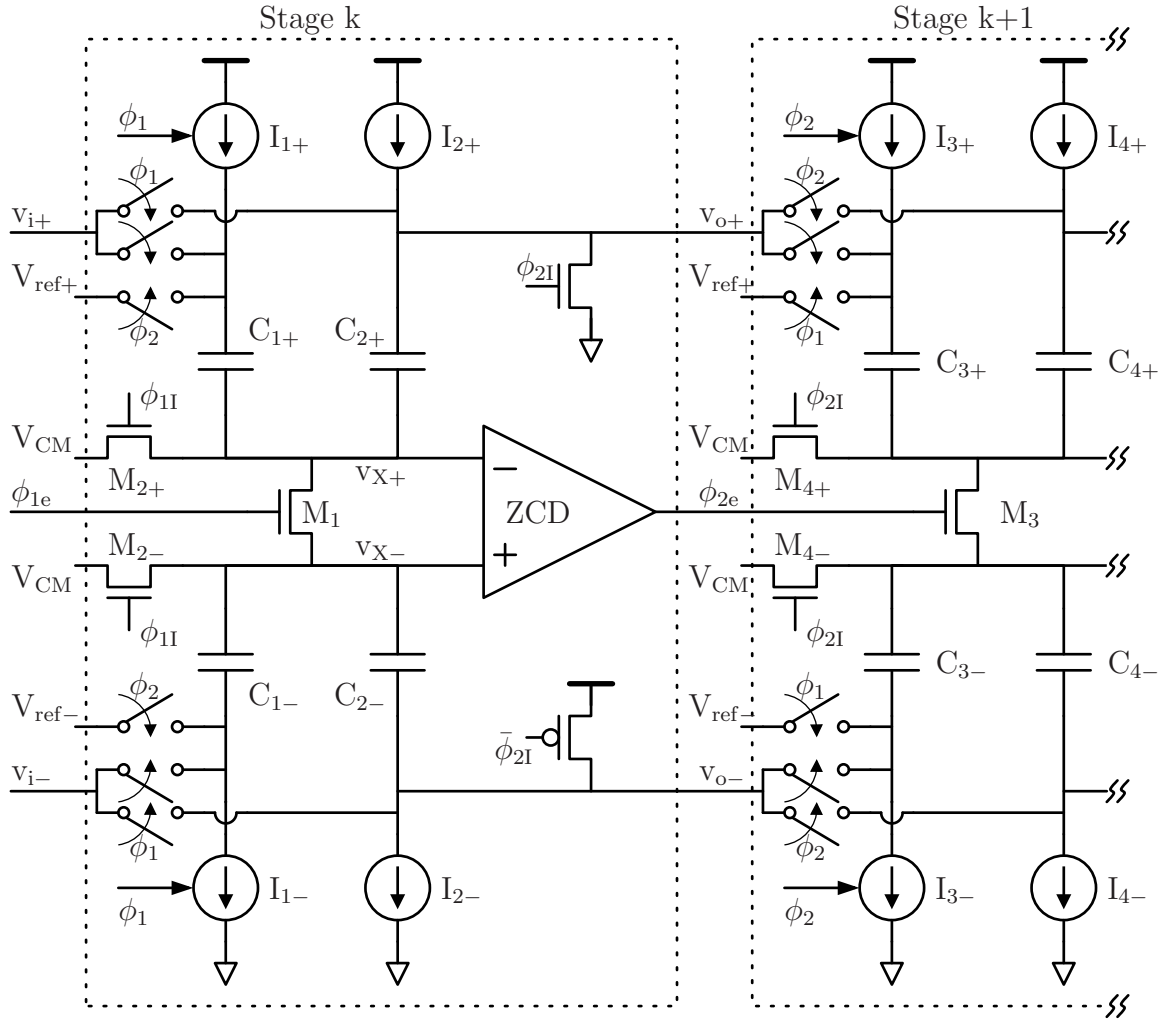


Figure 5-2: Fully differential implementation

shown in Figure 5-3. When compared to a fully differential implementation of a traditional opamp-based circuit, one significant change besides the replacement of the opamp with a zero-crossing detector is in this implementation of the sampling circuit. Specifically, a common-mode error reset mechanism has been introduced with the addition of switch  $M_3$  and a slight modification to the traditional timing of switches  $M_{4+}$  and  $M_{4-}$ .

When  $\phi_1$  is high and stage  $k$  is in the sampling phase, the input into stage  $k$  is sampled on capacitors  $C_{1\pm}$  and  $C_{2\pm}$ . When  $\phi_2$  goes high and stage  $k$  then enters the transfer phase, stage  $k+1$  enters the sampling phase and capacitors  $C_{3\pm}$  and  $C_{4\pm}$  become the load of stage  $k$ . In an opamp-based implementation, the inside plate

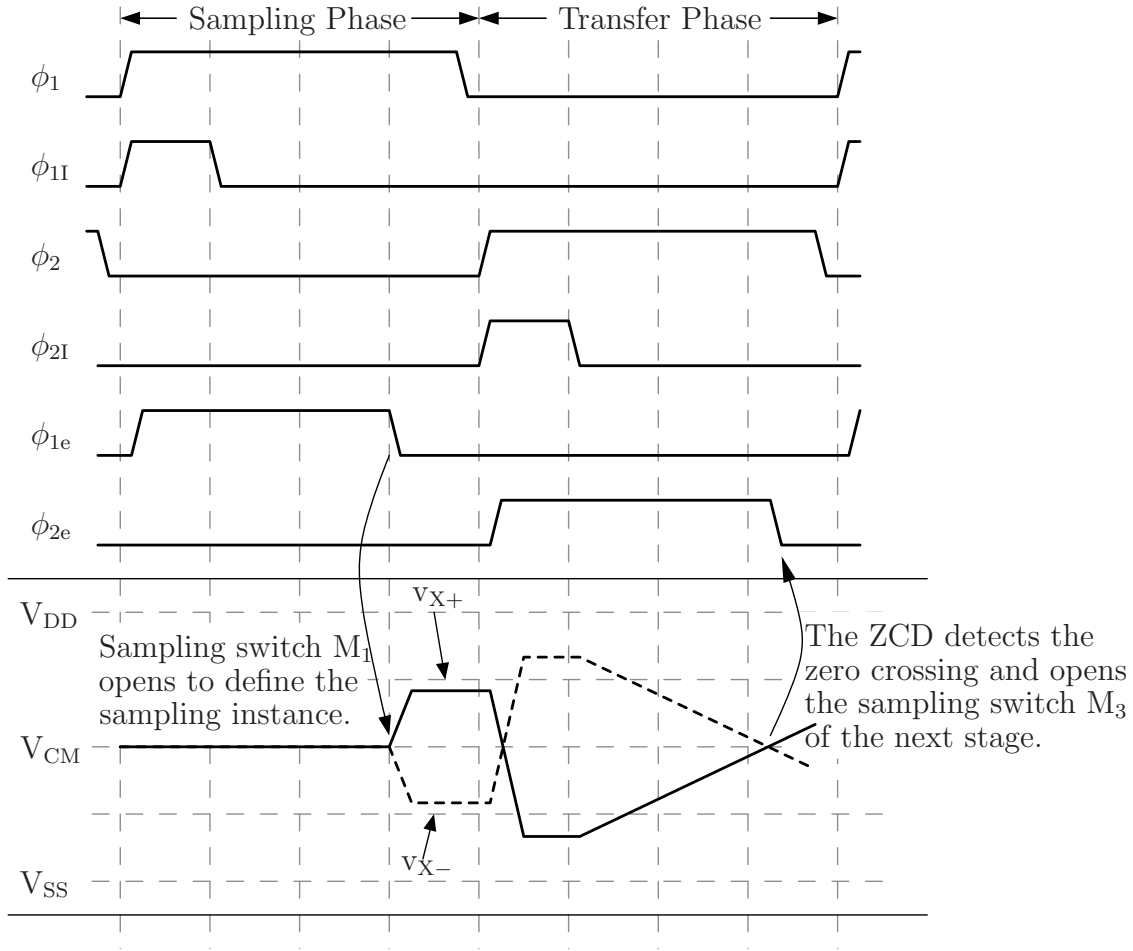


Figure 5-3: Fully differential timing diagram

sampling switches  $M_{4+}$  and  $M_{4-}$  are closed for the duration of the charge transfer to provide a low impedance connection to the common mode voltage  $V_{CM}$ . In this implementation, however, these switches are only closed during the pre-charge phase ( $\phi_{2I}$ ). After pre-charge, switch  $M_3$  is left connecting the inside plates of the sampling capacitors. When the zero-crossing detector switches, it opens switch  $M_3$  to lock the charge on  $C_{3\pm}$  and  $C_{4\pm}$  to realize the desired charge transfer. Thus  $M_3$  becomes the sampling switch that ties the inside plates together but allows the voltage on that node to float.

In the traditional opamp-based implementation when the inside plates of the sampling capacitors are held at  $V_{CM}$  for the entire transfer phase, the common mode voltage error of each stage accumulates down the pipeline onto the output capacitors. In

this implementation, however, if we ignore the effects caused by parasitic capacitance, the inside plates of the sampling capacitors are not held at  $V_{CM}$  during the charge transfer and the common-mode error will not accumulate in the sampling capacitors. This is because  $C_{3+}$  and  $C_{4+}$  have to charge at the same rate as  $C_{3-}$  and  $C_{4-}$  regardless of any current source mismatch in the current sources. A common-mode error can occur on the output *voltage*, but the floating inside plates ensures that no common mode error occurs on the *charge* sampled onto each capacitor. Therefore, when stage  $k+1$  advances into the transfer phase, the output voltage is reset during the pre-charge phase and the common-mode error is reset with it. Parasitic capacitance on the bottom plate will allow common mode charge error to accumulate on the sampling capacitors, but the error will be attenuated by the capacitor ratio of the parasitic capacitance to the sampling capacitors.

### 5.2.1 Common Mode Control

A continuous time common mode feedback circuit is essential in a traditional fully differential opamp-based implementations. The reason is that the common mode of the output voltage of a fully differential opamp is a function of both the differential and common mode of the input signal. In the case of a ZCBC implementation, however, the output voltage common mode is set by the relative strengths of positive and negative current sources and does not depend on the common mode performance of the zero-crossing detector. Any mismatch in the relative strength of the positive and negative current sources will produce an output voltage common mode error that grows with time, but it does not require a continuous time common mode feedback circuit as an opamp-based implementation does. Coupled with the fact that the common mode error gets reset after each stage, the constraints for a ZCBC common mode feedback circuit are significantly reduced when compared to an opamp-based implementation.

For this ZCBC implementation, no continuous time common mode feedback circuit is implemented on the signal path. Instead, the strength of each current source can be digitally programmed by extra fingers on each current source. For simplicity

on this chip, the relative strengths of each current source is adjusted to give optimal performance during a startup calibration procedure. Procedures for automatic background calibration of current source mismatch do still need developed. Something as simple as using an auto-zeroed comparator on each stage to measure the common mode error and to control a charge pump that adjusts the relative strengths of each current source would likely be sufficient. This chip demonstrates that continuous time common mode feedback on the signal path is not necessary for fully differential ZCBC implementations and leaves the development of automatic background calibration for future research.

The previous discussion regarding common mode feedback applies to controlling the common mode of the signal path. Even though the common mode performance of the zero-crossing detector does not effect the common mode performance of the signal path, the zero-crossing detector does have a differential input and may need internal common-mode feedback control. For this implementation, the zero-crossing detector performs a differential to single-ended conversion with a pre-amplifier and thus avoids the need for common mode feedback control. See Section 5.2.3 for further details on the zero-crossing detector implementation.

## 5.2.2 Symmetry for Improved Power Supply Noise Rejection

The power supply noise that corrupts the signal through the current sources feeds in through a few different mechanisms as shown in Figure 5-4. Any noise modulating the  $V_{GS}$  of  $M_2$  can be mitigated with sufficient by-pass capacitance  $C_B$  and/or using a reference current mirror  $M_1$ . The more problematic issue from power supply noise is the path through the finite output impedance and the drain-to-bulk parasitic junction capacitance of  $M_2$ .

The small signal model including these effects is shown in Figure 5-5 where the power supply voltage noise source is labelled  $v_{PS}$  and the output impedance of the current source is labelled  $r_o$ . In addition, the load capacitance  $C_L$  and the resistance of the sampling switch  $r_{ds}$  have been added as well. Without changing the behavior of the circuit for this analysis, the sampling switch resistance  $r_{ds}$  is put on top of the

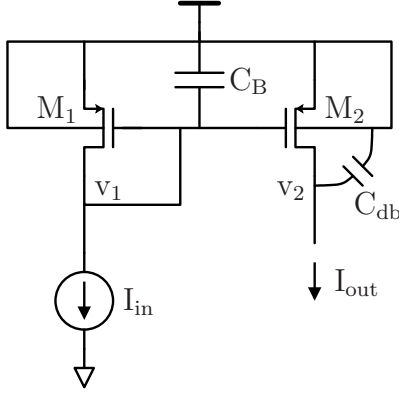


Figure 5-4: Large Signal Current Source

load capacitance so that the output voltage  $v_O$  is referenced to ground. In reality the sampling switch is on the bottom and performs bottom plate sampling, but putting it on top simplifies the math. The voltage transfer function from  $v_{PS}$  to  $v_O$  under the

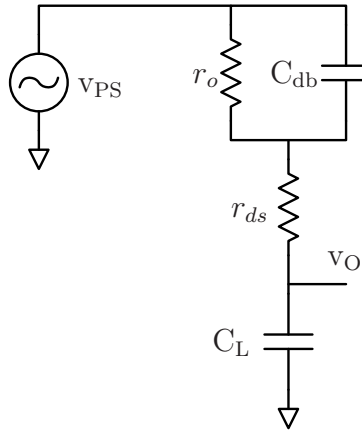


Figure 5-5: Small Signal Current Source

assumptions that  $C_L \gg C_{db}$  and  $r_o \gg r_{ds}$  is

$$\frac{v_O(s)}{v_{PS}(s)} \approx \frac{sr_o C_{db} + 1}{(sr_o C_L + 1)(sr_{ds} C_{db} + 1)}. \quad (5.1)$$

This is plotted in Figure 5-6 for simulated parameters extracted from this design. With 2 poles and 1 zero there are 4 different regions to the frequency response. At DC the capacitors are open and the power supply noise feeds directly to the output with unity gain through  $r_o$ . The first pole occurs at  $1/r_o C_L$  when the impedance of



$C_L$  becomes active and the power supply noise rolls off with first order slope through the low pass filter of  $r_o$  and  $C_L$ . The third region occurs due to the zero at  $1/r_o C_{db}$  when the frequency is high enough to activate the parasitic junction capacitance  $C_{db}$ . At this point the impedance of  $r_o$  becomes negligible, the frequency response flattens, and capacitor divider ratio from  $C_{db}$  to  $C_L$  sets the gain. Finally the second pole at  $1/r_{ds} C_{db}$  due to the resistance of the sampling switch activates at the highest frequencies and provides further first order attenuation.

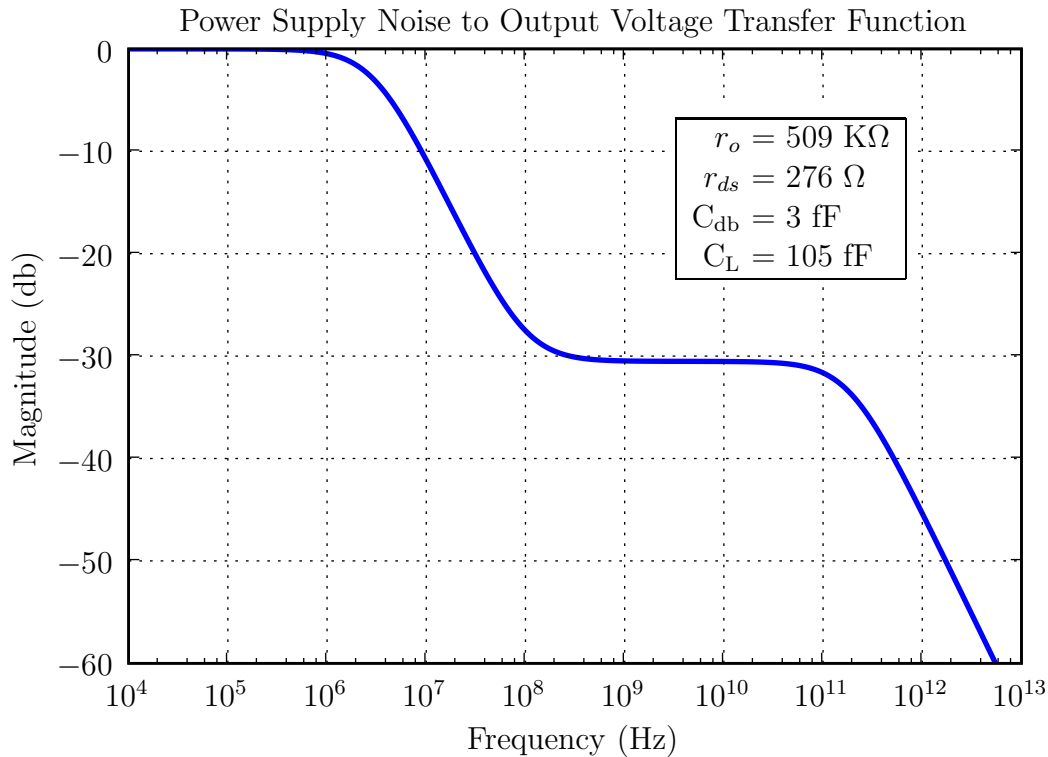


Figure 5-6: Power supply to output voltage transfer function from parameters extracted via simulation.

How does noise on the output voltage  $v_o$  effect the dynamics of the ZCBC circuits and the final sampled voltage? In simple terms, the zero-crossing detector can track and null noise on the output slower than its open-loop bandwidth, however, it is unresponsive to any frequency content higher. Since the zero crossing detector is unresponsive to high frequency noise on the output, this noise will get sampled during the sampling instance. As revealed in the transfer function of Equation 5.1, the high frequency noise comes in through the parasitic junction capacitance  $C_{db}$ . Minimizing

the capacitance ratio of  $C_{db}$  to  $C_L$  to maximize the attenuation of the high frequency power supply noise to the output does improve power supply noise rejection, but this comes at the expense of signal range as reducing the width of the current source device raises its saturation voltage.

One way to effectively eliminate the high frequency power supply noise from corrupting the differential signal is to put the same parasitic junction capacitance on both channels of the fully differential signal path. Then to first order, the power supply noise feeds equivalently into both channels and appears as a common mode voltage error. For a fully differential ZCBC implementation, however, the parasitic junction capacitance is not inherently symmetric because the positive channel uses a PMOS-based pull-up implementation and the negative channel uses an NMOS-based pull-down implementation. The PMOS device introduces a reversed bias NWELL/p+ junction between  $V_{DD}$  and the output of the positive channel, and the NMOS device introduces a PWELL/n+ reverse biased junction between  $V_{SS}$  and the output of the negative channel. To make the parasitic junction capacitance equivalent on both channels, however, dummy current sources that are permanently disabled have been added to each channel as shown in the partial circuit diagram of Figure 5-7. This figure extracts the current sources  $I_{1\pm}$  and the sampling capacitors  $C_{1\pm}$  from the complete circuit of Figure 5-2 and added dummy current sources  $I_{dum\pm}$  to create complete parasitic symmetry for both the positive and negative channel.

### 5.2.3 Differential Zero-Crossing Detector

A fully differential ZCBC requires a differential zero crossing detector. The dynamic zero-crossing detector (DZCD) used in the initial single-ended design and described in Section 3.2 seems inherently single-ended and does not have a natural extension to a differential implementation. Thus, the differential zero-crossing detector shown in Figure 5-8 was used in this fully differential implementation. The first stage is a differential to single-ended pre-amplifier followed by a dynamic threshold detecting latch (DTDL).

The pre-amplifier is implemented with an NMOS differential pair ( $M_1$  and  $M_2$ )

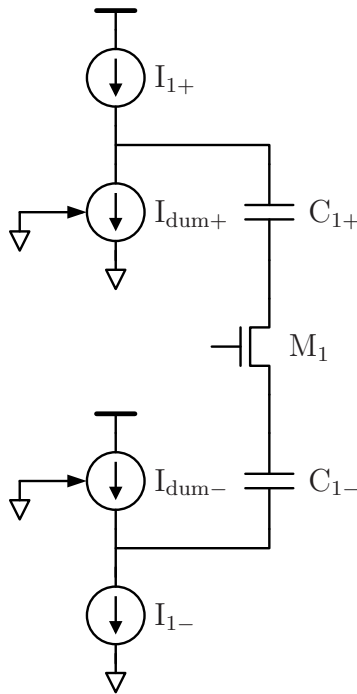


Figure 5-7: Permanently disabled dummy current sources ( $I_{dum\pm}$ ) are added to provide symmetric parasitic capacitance for improved power supply noise rejection.

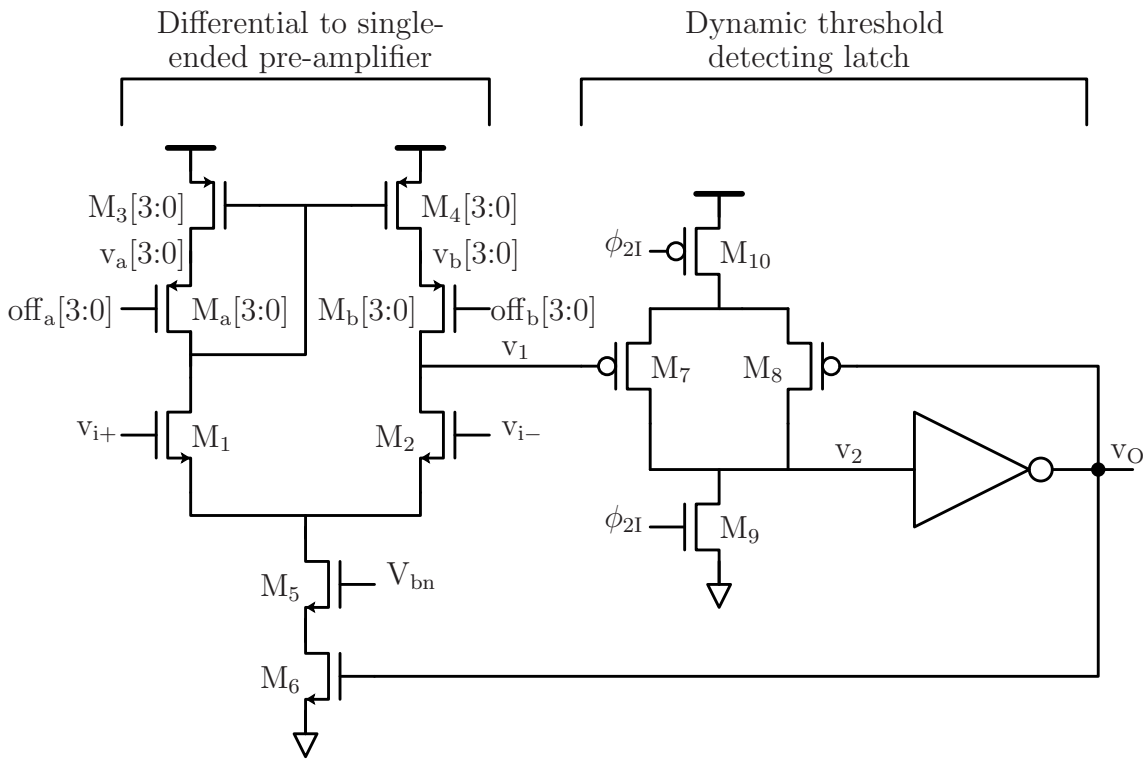


Figure 5-8: Differential zero crossing detector

input. A current mirror ( $M_3$  and  $M_4$ ) is used to convert from a differential input to a single-ended output. Devices  $M_3$ ,  $M_4$ ,  $M_a$  and  $M_b$  utilize iterated instance notation to show that there are actually 4 devices draw in parallel. Nets  $v_a[3:0]$  and  $v_b[3:0]$  use bus notation to show that these are actually 4 different nets hooked up to the individual iterative device instances. This notation helps with schematic readability. The binary weighted widths of devices  $M_3$ ,  $M_4$ ,  $M_a$ , and  $M_b$  creates a programmable current gain by enabling or disabling devices  $M_a$  and  $M_b$  independently. This programmable current gain creates an offset programmable pre-amplifier that is used for offset compensation.

The DTDL is composed of devices  $M_7$ - $M_{10}$  and is like the DZCD used previously in that it is a dynamic logic circuit that draws no static current. During the pre-charge phase when  $\phi_{2I}$  is high, the latch is reset due to  $M_{10}$  turning off and  $M_9$  turning on. In this state, the current to the pre-amplifier is turned on via switch  $M_6$ . When  $\phi_{2I}$  drops to enter the ramping phase, voltage  $v_1$  will start ramping with an amplified slope. When  $v_1$  ramps sufficiently and the virtual ground condition has been realized, it will have turned on  $M_7$  sufficiently to flip the state of the latch to signal the zero-crossing has been detected. This will also turn off the current source to the pre-amplifier by disabling device  $M_6$ . Thus, while the pre-amplifier draws static current prior to the threshold detection, the current is turned off after the detection to save power.

#### 5.2.4 Chopper Offset Estimation

Chapter 4 introduced Chopper Offset Estimation (COE) as an general offset compensation technique that is compatible with ZCBC. Because the zero-crossing detector in a ZCBC implementation will have a finite delay, the voltage ramp of each pipeline stage will overshoot to give an output referred offset equal to  $v_{os} = at_d$  where  $t_d$  is the finite delay and  $a$  is the nominal ramp rate. If each stage is scaled appropriately, then in addition to random offset from device mismatch, each stage will have the same systematic offset due to this overshoot.

Because the supply voltage is only 1.2V in this 90nm technology node, signal range is limited and the input-referred COE offset estimation technique as shown

in Figure 4-11 was selected as the offset compensation method for this design. By cancelling the offset of each stage at its source, this approach recovers the lost signal range of each stage and improves the output signal range to allow for cascaded current sources.

To actually adjust the offset of the zero-crossing detector, the current mirror made of the devices  $M_3$  and  $M_4$  in Figure 5-8 had been implemented with programmable gain. The offset controller simply adjusts the gain of the current mirror digitally to null the dynamic offset of the complete zero-crossing detector to 0.

### 5.3 Voltage References

During the transfer phase an analog multiplexer must switch between the reference voltages. The ZCBC architecture imposes two distinct challenges both for the reference voltage source and the analog multiplexer when compared to the opamp-based architecture. When implemented in an opamp-based architecture, the reference voltage has similar settling requirements to that of the opamp in the signal path. They both must settle sufficiently by the end of the clock period. Insufficient settling from either source will cause signal dependent errors in the virtual ground condition and non-linearities at the output. Although the settling characteristics of both the opamp and the reference voltage source must be sufficient, the reference voltage source has the advantage that it can be loaded with large by-pass capacitance and has a fixed output range. These both generally ease the design requirements and increase the power efficiency of generating the voltage references.

These advantages also hold true for the case of the ZCBC architecture, however, the ZCBC architecture requires that the reference voltage settle within the pre-charge clock phase, which will typically be considerably less time than the entire clock phase. This can be seen in the timing diagram of Figure 5-3 where  $\phi_{21}$  is the pre-charge signal and  $\phi_2$  is the clock signal of for the entire transfer phase. Any settling of the reference voltage after the start of the voltage ramp will appear directly on the voltage ramp and will cause signal dependant errors on the virtual ground condition as discussed

in Section 1.3.2 regarding finite output impedance in the current sources. Thus the settling time requirements for the reference voltage sources are shorter for a ZCBC implementation over an opamp-based implementation.

The second issue that the ZCBC architecture introduces is that the reference voltage sources must source and/or sink the voltage ramp current. In an opamp-based system the current drops as the dynamics settle. In a ZCBC implementation, however, the current is constant for the entire ramping period. Furthermore, while the current load is constant for a given code, each code produces a different current requirement as the bit decisions of each stage determine whether current is sourced or sinked by the reference voltage source. Because each stage switches asynchronously to the others, the reference voltage source must be able to hold the reference voltage to within an LSB of precision when one stage switches and its current load turns off.

### 5.3.1 Off-chip Reference Voltage Issues

When the reference is off chip, the parasitic inductance of the bond wire can be extremely problematic for both of these issues. For this design, the 1 nF on-chip bypass capacitance between  $V_{\text{refp}}$  and  $V_{\text{refm}}$  provided the solution. Figure 5-9 shows transient simulation results when the bond wire impedance was modelled with 1  $\Omega$  of series resistance and 1 nH of series inductance on both  $V_{\text{refp}}$  and  $V_{\text{refm}}$ . The clock in this simulation is running with an 8 ns period, so every 4 ns a new load get switched onto the reference voltages. The pre-charge period lasts 1 ns.

The error on  $V_{\text{refp}}$  and  $V_{\text{refm}}$  is plotted in the first graph. The ringing peaks at approximately 40mV on each signal. The differential error is plotted in the second graph. The differential error has no ringing on it, although it does have settling and corruption due to ZCD switching. With a 2 V differential input range, a 12 bit LSB will be about  $500\mu\text{V}$ , so the grid lines on the differential error plot correspond to the size of a single 12 bit LSB. With approximately 1nF of bypass from  $V_{\text{refp}}$  to  $V_{\text{refm}}$ , the disturbances on the differential signal due to ZCD switching are smaller than an LSB. The bigger issue is the settling time after the clock switches. Even after 1 ns pre-charge completes there is still some settling that is occurring that is on the

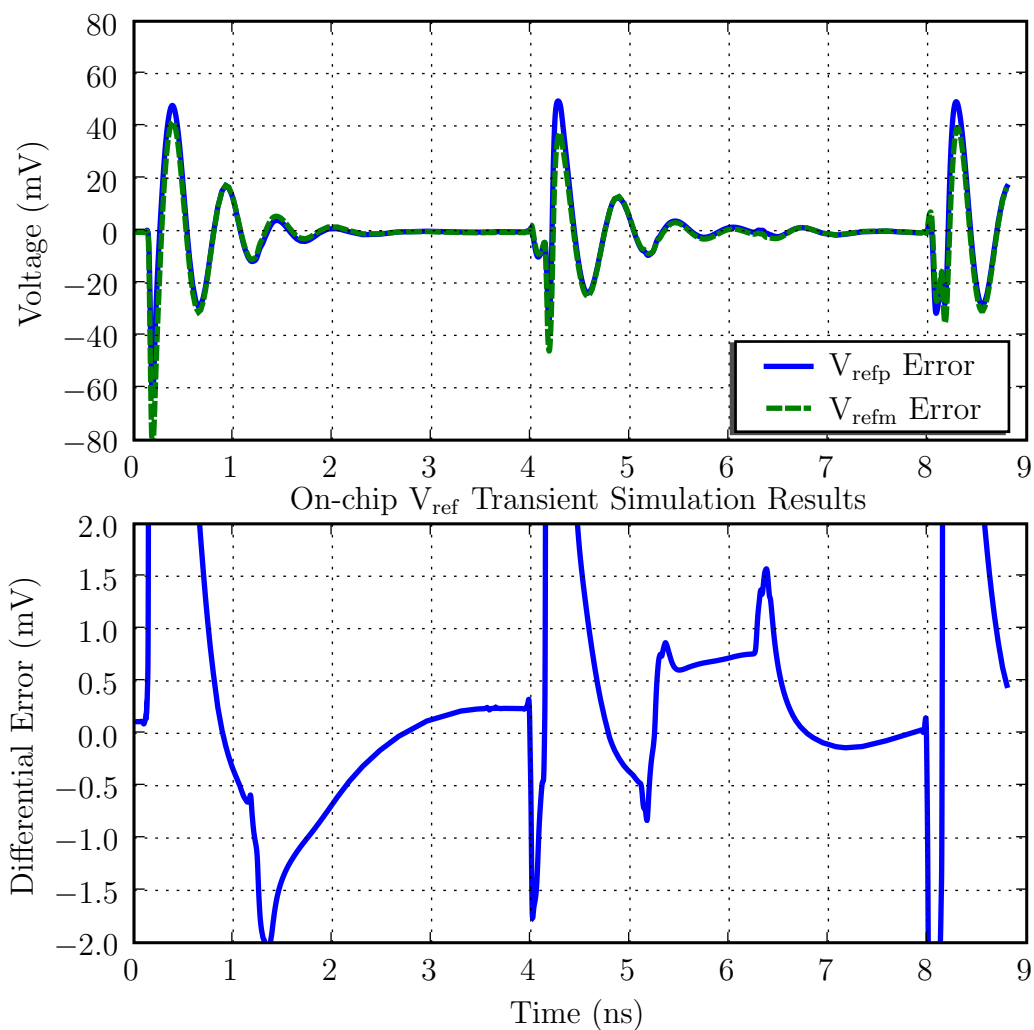


Figure 5-9: On-Chip Transient Reference Voltage Simulation Results

order of several LSBs. There is, however, about 1 ns of “runway” prior to the voltage ramp reaching the minimum of the output voltage range, and by 2 ns into the phase the settling is on the order of an LSB. To verify this further, a 16 sample transient simulation with a small signal sine wave input that straddled an MSB transition point measured an SNDR performance of 11.5 bit.

The issues with generating the reference voltages stand as open areas of research. Section 6.1.1 discusses these trade offs and provides additional ideas that may prove useful for future research in this area.

### 5.3.2 Voltage Reference Switching via Capacitor Splitting

One additional issue to consider with regards to the reference voltages is the series on-resistance introduced by the switches that implement the analog multiplexer that selects the appropriate reference to apply during the transfer phase. Recall that the voltage drop in the reference switches for the single-ended case was not an issue for the 1.0 bit/stage case because two reference points are inherently linear. Only when additional reference points are introduced can a non-linearity result. This posed the problem that for the initial design because the 1.5 bit/stage implementation required a third reference voltage.

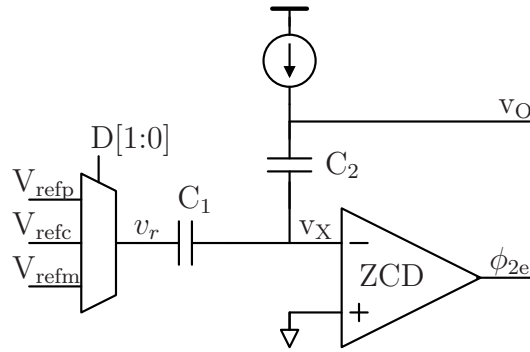


Figure 5-10: Traditional implementation of voltage references for a 1.5 bit/stage pipeline stage.

The schematic of Figure 5-10 shows a traditional 1.5 bit/stage implementation in the transfer phase. The analog multiplexer selects between three voltage references. At the end of an ideal voltage transfer, the output voltage will have the form

$$v_o = 2v_i - v_r, \quad (5.2)$$

where  $v_i$  is the voltage that was sampled on both capacitors during the sampling phase and  $v_r$  is the output of the reference voltage multiplexer. There are three different possible values for  $v_r$  that correspond to the three possible bit decision states, and each results in a corresponding different reference voltage selection. Suppose each switch to produces a voltage drop of  $\Delta_p$ ,  $\Delta_c$ , and  $\Delta_m$  corresponding to the switch associated with  $V_{refp}$ ,  $V_{refm}$ , and  $V_{refc}$  respectively. Solving for  $v_r$  under these three



conditions yields

$$\begin{aligned}
 v_r &= V_{\text{refp}} + \Delta_p && \text{when } D[0] = 1 \text{ and } D[1] = 1 \\
 v_r &= V_{\text{refc}} + \Delta_c && \text{when } D[0] = 1 \text{ and } D[1] = 0 \\
 v_r &= V_{\text{refm}} + \Delta_m && \text{when } D[0] = 0 \text{ and } D[1] = 0.
 \end{aligned}$$

Substituting these into Equation 5.2 gives the three possible output voltage states as

$$\begin{aligned}
 v_o &= 2v_i - (V_{\text{refp}} + \Delta_p) && \text{when } D[0] = 1 \text{ and } D[1] = 1 \\
 v_o &= 2v_i - (V_{\text{refc}} + \Delta_c) && \text{when } D[0] = 1 \text{ and } D[1] = 0 \\
 v_o &= 2v_i - (V_{\text{refm}} + \Delta_m) && \text{when } D[0] = 0 \text{ and } D[1] = 0
 \end{aligned}$$

For these to produce a linear response, the center equation must subtract a quantity that is exactly the average of the outer two:

$$V_{\text{refc}} + \Delta_c = \frac{1}{2}(V_{\text{refp}} + \Delta_p + V_{\text{refm}} + \Delta_m). \quad (5.3)$$

When this constraint is satisfied, we get the ideal residue plot and complete ADC transfer function as shown in Figure 5-11. When this constraint is not satisfied, we get a response like that of Figure 5-12 where  $\Delta_p$ ,  $\Delta_c$ , and  $\Delta_m$  were given values of 2%, 10%, and 4% respectively. One can see in the complete ADC transfer function that the center segment is misaligned due to the voltage drop mismatch.

The approach of the initial ZCBC design was to use gate-boostered switches for the reference voltage to generate switches with matched on-resistance to satisfy the constraint of Equation 5.3. For this second design, however, an alternative method was developed to use switch capacitor techniques to generate the middle voltage reference and eliminate the series on-resistance issue completely. The schematic for this approach is shown in Figure 5-13. Here capacitor  $C_1$  has been split in half and driven with two reference voltage multiplexers that can interpolate the middle voltage as necessary. Thus, when the bit decisions  $D[1:0]$  require driving either  $V_{\text{refp}}$  or  $V_{\text{refm}}$ , both analog multiplexers drive the corresponding voltages. When the third

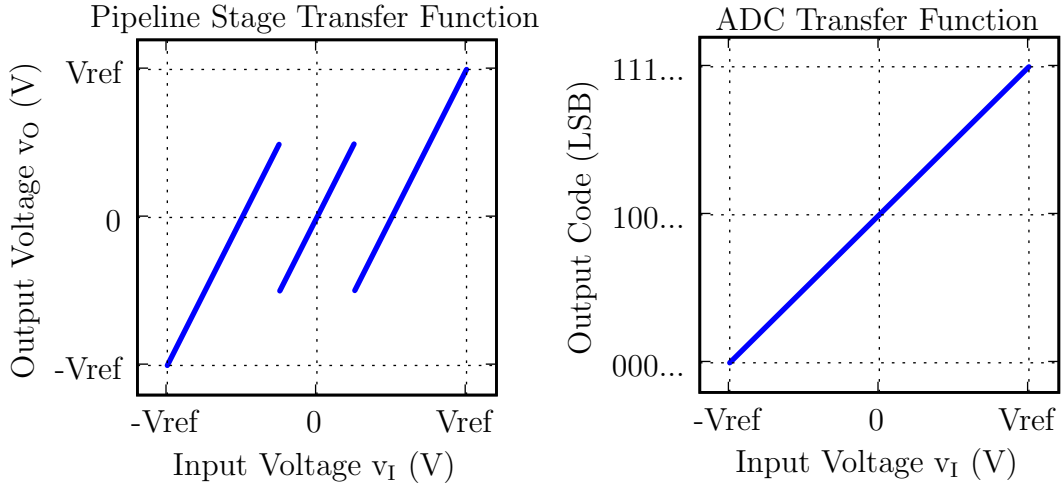


Figure 5-11: Ideal stage voltage transfer function (left) and ADC transfer function (right) for a 1.5 bit/stage ADC.

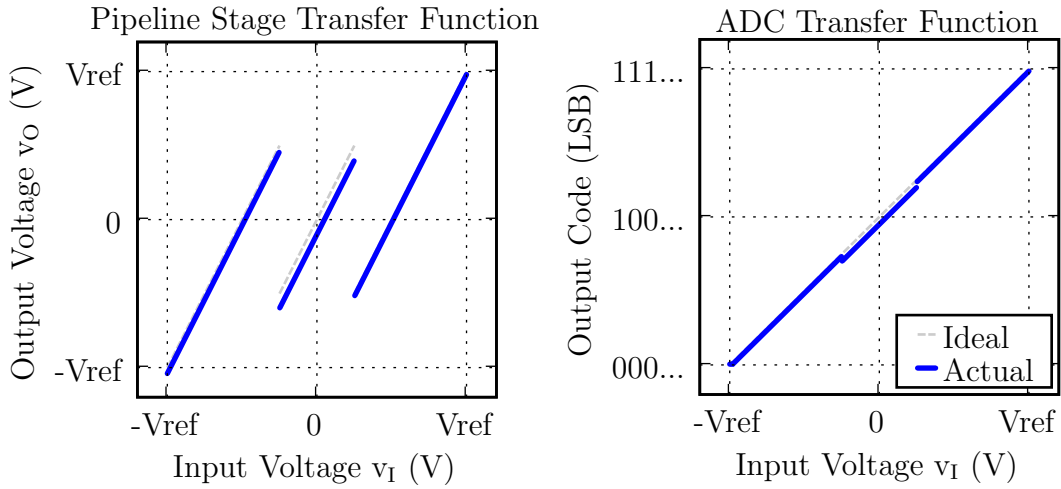


Figure 5-12: Voltage transfer function (left) and ADC transfer function (right) for a 1.5 bit/stage ADC including series resistance mismatch for the voltage reference switches.

$V_{refc}$  voltage is required, one multiplexer drives to  $V_{refp}$  and the other to  $V_{refm}$  to interpolate the middle reference voltage.

Under this scheme, the ideal voltage transfer takes the form

$$v_o = 2v_i - \frac{1}{2}(v_{r1} + v_{r2}), \quad (5.4)$$

where  $v_{r1}$  and  $v_{r2}$  are the outputs of each multiplexer. Enumerating the possible

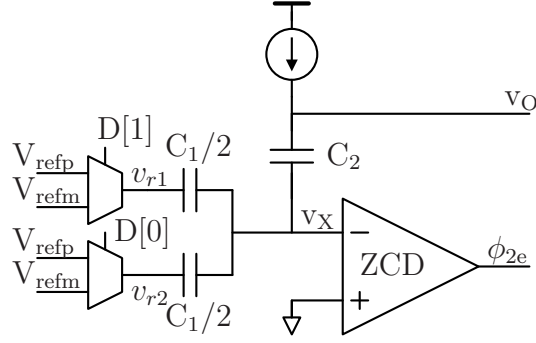


Figure 5-13: Alternative 1.5 bit/stage ZCBC implementation where  $C_1$  has been split to eliminate the  $V_{\text{refc}}$  voltage reference.

values for  $v_{r1}$  and  $v_{r2}$  under the three different bit decision states gives

$$\begin{aligned}
 v_{r1} &= V_{\text{refp}} + \Delta_p, & v_{r2} &= V_{\text{refp}} + \Delta_p & \text{when } D[0] = 1 \text{ and } D[1] = 1 \\
 v_{r1} &= V_{\text{refm}} + \Delta_m, & v_{r2} &= V_{\text{refp}} + \Delta_p & \text{when } D[0] = 1 \text{ and } D[1] = 0 \\
 v_{r1} &= V_{\text{refm}} + \Delta_m, & v_{r2} &= V_{\text{refm}} + \Delta_m & \text{when } D[0] = 0 \text{ and } D[1] = 0.
 \end{aligned}$$

Substituting this result into Equation 5.4 gives the output voltage under the three different states as

$$\begin{aligned}
 v_o &= 2v_i - (V_{\text{refp}} + \Delta_p) & \text{when } D[0] = 1 \text{ and } D[1] = 1 \\
 v_o &= 2v_i - \frac{1}{2}(V_{\text{refp}} + \Delta_p + V_{\text{refm}} + \Delta_m) & \text{when } D[0] = 1 \text{ and } D[1] = 0 \\
 v_o &= 2v_i - (V_{\text{refm}} + \Delta_m) & \text{when } D[0] = 0 \text{ and } D[1] = 0
 \end{aligned}$$

Now the voltage drop for the center equation is exactly the average of the other two, which makes it inherently linear. With this approach, the series resistance of the switches no longer needs to match for a linear response. The above results, however, do rely on being able to split capacitor  $C_1$  in half exactly, and so the linearity problem has been moved from one of matching switch resistors to that of matching capacitors. Implementing matched capacitors is much more reasonable than generating switches with a constant on-resistance at the three different reference voltages.

Given that this technique no longer requires switching to a third reference voltage, it simplifies both the design of the analog multiplexer and the selection logic. In the original implementation, the analog multiplexer required digital logic to turn on the

correct switch based on the value of  $D[1:0]$ . In the new approach, however, no logic is needed as each thermometer encoded bit controls each a multiplexer directly. Logic is required to only enable the switches during the transfer phase when  $\phi_2$  is high. The implementation is shown in Figure 5-14.

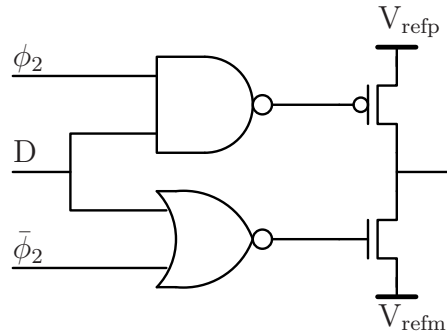


Figure 5-14: Analog multiplexer implementation

This technique has a natural extension to even higher resolutions. The rule is that capacitor  $C_1$  should be split up equally to match the exact number of bit decision comparators used in the design and each bit decision comparator controls the select logic for the multiplexer for each capacitor. The schematic of a ZCBC stage in the transfer phase in Figure 5-15 shows such an implementation for the case of when  $n$  bit decision comparators are used to create a  $\log_2(n + 1)$  bit/stage pipeline stage. This schematic uses iterative instance notation to denote multiple parallel instances and wide wires to denote buses of multiple nets. Notice that the thermometer encoded output  $D[n:1]$  of the bit decision comparators  $BDC[n:1]$  does not need converted into another format but can drive the select of the multiplexer  $U[n:1]$  directly.

### 5.3.3 Capacitor Splitting with Fully Differential Designs

Now consider the more general case of a fully differential implementation when  $n$  bit decision comparators are used in the implementation of a  $\log_2(n + 1)$  bit/stage ZCBC stage as shown in Figure 5-16. Here the analog multiplexer has been implemented as the parallel combination of an ideal switch and a series resistor where  $R_p$  is the resistance of switches connecting to  $V_{refp}$  and  $R_m$  to  $V_{refm}$ . The iterative instance notation denotes parallel instantiations of multiple instances and wide wires denote

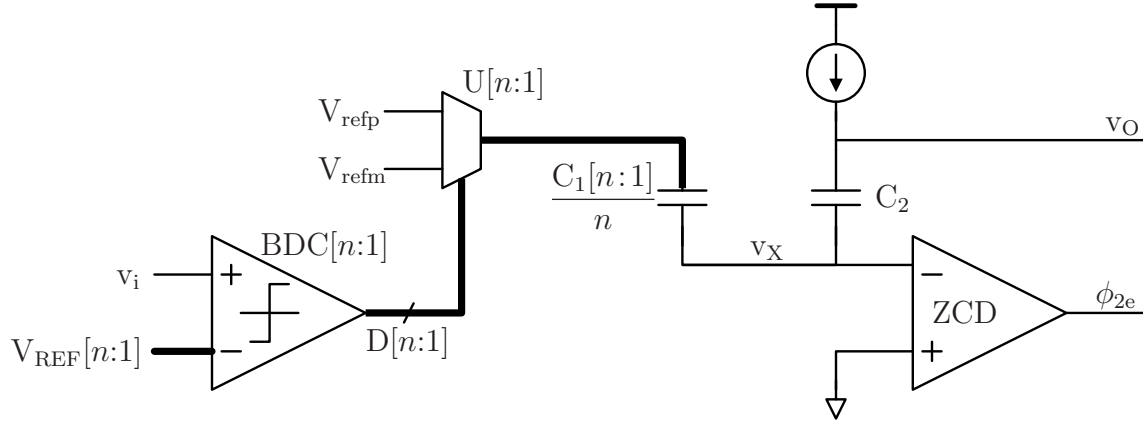


Figure 5-15: Schematic of a  $\log_2(n + 1)$  bit/stage ZCBC pipeline stage using capacitor splitting (only circuits active during the transfer phase have been included). Capacitor  $C_1$  is split into  $n$  equal parts.

buses of unique connections. As before, sampling capacitors  $C_{1\pm}$  are split into  $n$  equal parts. The voltage from nets  $v_{r+}[i]$  to  $v_r[i]$  is the reference voltage that matters to the

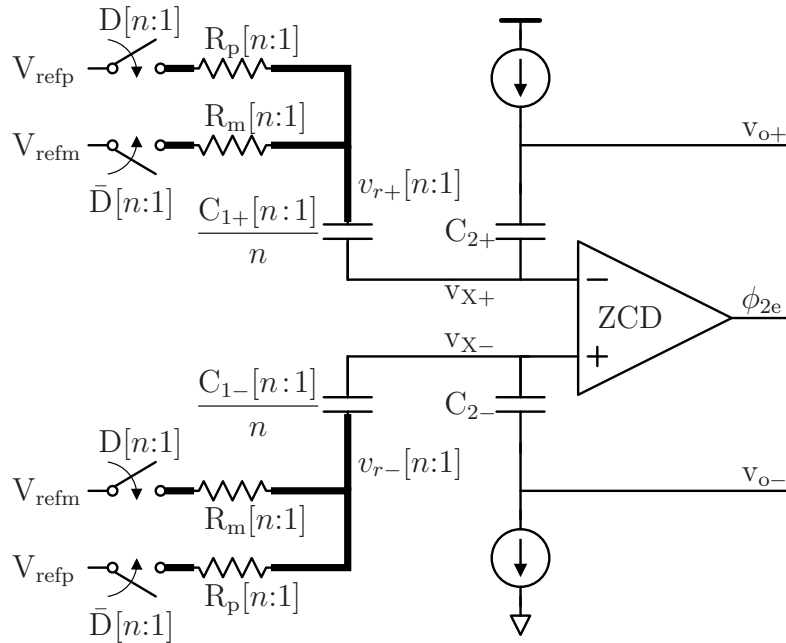


Figure 5-16: Differential ZCBC showing series on-resistance of reference switches

ZCBC circuit. The output or residue voltage when an ideal transfer phase is realized can be calculated as

$$v_o = 2v_i - \frac{1}{n} \sum_{i=1}^n v_r[i], \quad (5.5)$$

where the following differential voltage definitions have been used

$$\begin{aligned} v_o &= v_{o+} - v_{o-} \\ v_i &= v_{i+} - v_{i-} \\ v_r[i] &= v_{r+}[i] - v_{r-}[i]. \end{aligned}$$

To analyze the effect of the series resistance, initially assume the current sources provide equal and opposite amounts of current so that the voltage drop across  $R_p$  and  $R_m$  can be expressed as  $\Delta_p$  and  $\Delta_m$  respectively. Furthermore, using the definitions

$$\begin{aligned} V_{\text{ref}} &= V_{\text{refp}} - V_{\text{refm}} \\ \Delta &= \Delta_p + \Delta_m, \end{aligned}$$

the sum of each reference voltage  $v_r[i]$  when  $k$  bits of bit decision vector  $D[n:1]$  are high can be calculated as

$$\sum_{i=1}^n v_{ri} = (2k - n)V_{\text{ref}} + n\Delta \quad (5.6)$$

Substituting this result into Equation 5.5 gives the residue voltage as

$$v_o = 2v_i - \left( \frac{2k}{n} - 1 \right) V_{\text{ref}} - \Delta \quad (5.7)$$

This reveals several important aspects of the fully differential implementation of capacitor splitting. First is that the series voltage drop  $\Delta$  due of the on-resistance of the reference voltage switches adds an offset to the residue plot. This offset gets added to all the other sources of offset and is nulled in this design by COE offset compensation. Furthermore, since  $k$  is the number of bit decisions comparators tripped high, it is the decimal representation of thermometer encoded sub-ADC output, and Equation 5.7, which assumed perfect capacitor matching, shows that this approach produces an inherently linear response. Thus, even though the drop across the switches will reduce the available signal range, the fully differential implementation using capacitor

splitting also produces linear response.

## 5.4 Redundancy For Increased Signal Range

Redundancy or over-range protection is traditionally used to relax offset constraints in both the bit decision comparators and the residue amplification. A typical residue plot without redundancy and with redundancy is plotted in Figure 5-17. The gray area represents valid signal area, and adding extra bit decisions to create redundancy helps protect the signal from leaving the valid signal area in the presence of bit decision comparator offset or residue amplification offset.

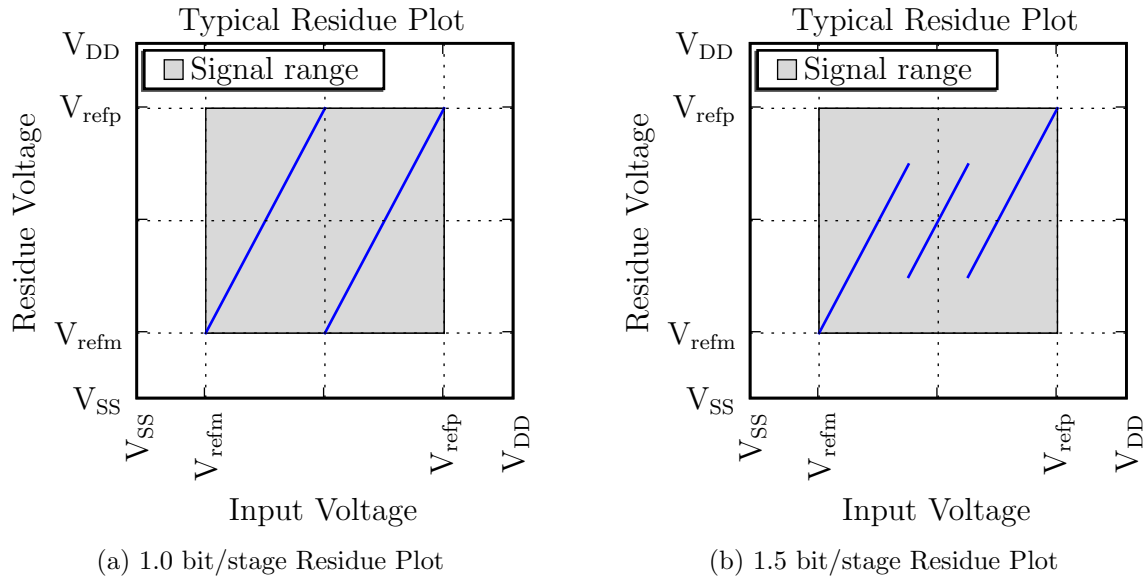


Figure 5-17: Typical residue plots without redundancy and with redundancy.

Traditional redundancy is used to keep the signal in range, but since the gray area is square, the output signal range must match the input signal range. Even though redundancy does reduce the output range at the bit decision boundaries, the extreme edges of the input near  $V_{refp}$  and  $V_{refm}$  still swing over the complete range, so the output linearity of the residue amplification stage must be designed to match the input range.

When designing in scaled technologies, however, the output range of the residue amplifier can be extremely limiting, especially if cascoded devices are used in the out-

put stage. The input range, on the other hand, is typically not so severely limited, especially if passive sampling is used. If the output range is severely limited, as shown in the example of the first plot of Figure 5-18, traditionally the input range is also reduced to match it. An alternative approach, however, is to grow the reference voltages until the output range of the interior step transition points reach the maximum output range. This, of course, grows the input range at the same rate and produce regions where the output can go out of range. Shrinking the input range ( $V_{irm}$  to  $V_{irp}$ ) can eliminate these invalid regions so that it only covers the valid output range. This is the technique used in the second plot of Figure 5-18. The grey box representing the valid signal range is no longer square but rectangular so that the input range is larger than the output range. Furthermore, comparing both plots of Figure 5-18 shows that the output range of both residue plots is identical, but the input range of the left-side plot is larger. In this example it has grown by 1.5x for the same output range. This change does not require changing anything in the circuit other than to increase the reference voltages  $V_{refm}$  and  $V_{refp}$  the appropriate amount. Thus, since the noise level and the power consumption stay the same while the signal range increases, the SNR and power efficiency improve. For the example of Figure 5-18, the reference voltages have been scaled by 2x to realize the 1.5x increase in input range for the same output range, which amounts to 2.25x increase in  $SNR^2$ .

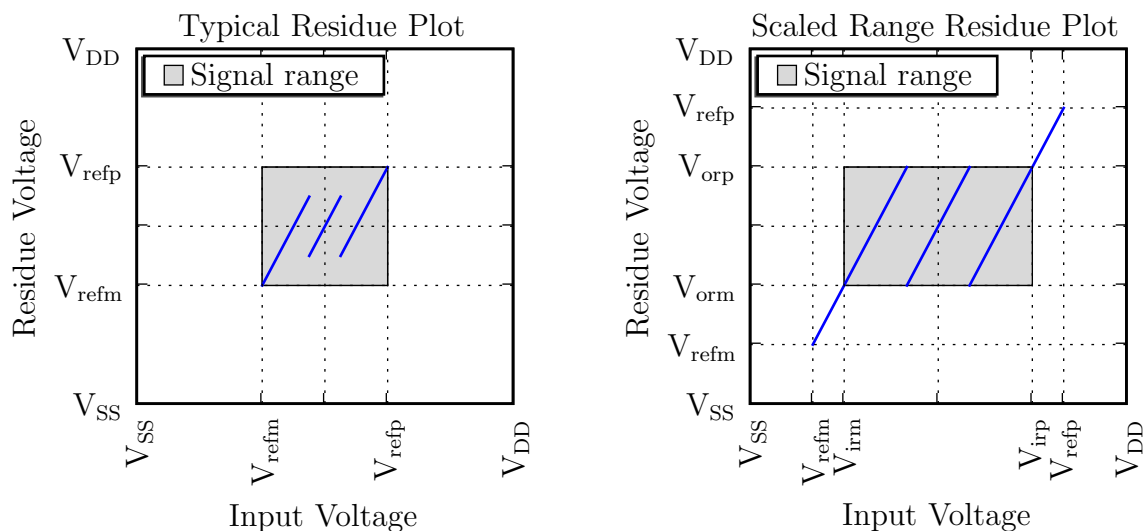


Figure 5-18: Residue plots when using 2 bit decision comparators (1.58 bits/stage).



Further redundancy can be employed to allow for further reference voltage scaling. This can be seen by comparing the residue plots of Figure 5-18 to Figure 5-19. Figure 5-18 corresponds to a stage with 2 bit decision comparators that implement a 1.5 bit/stage pipelined ADC. Figure 5-19, on the other hand, corresponds to a stage with 3 bit decision comparators, or 2 bits/stage. Both have the same output range, but the later has a larger input range. In this case the reference voltages were scaled by 3x to realize an 2x larger input range, which corresponds to a 4x improvement in SNR<sup>2</sup>. Adding additional redundancy for further improvements in this example would require scaling the reference voltages beyond the power supply range from  $V_{DD}$  to  $V_{SS}$ , which is probably impractical for most application, so this example has reached its limit in terms of scaling the reference for improved SNR. One side benefit that is also realized by using reference voltage scaling is that as the reference voltages push closer to the power supply it eases the switch sizing requirements that realize the analog multiplexer described in Section 5.3.

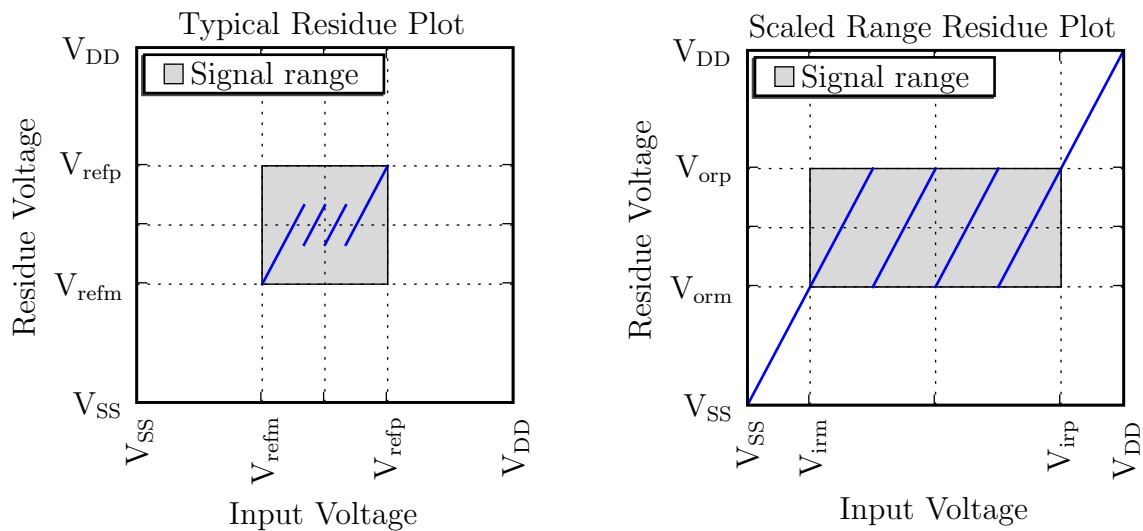


Figure 5-19: Residue Plots when using 3 bit decision comparators (2.0 bits/stage)

The problem with using reference voltage scaling as introduced to this point is that the over-range protection to BDC and ZCD offset has been reduced to nothing. Thus, when defining the available output range, one must include margin for all sources of offset that could affect the residue plot. For example, suppose a given

process has a 1.2V supply and that  $V_{\text{dsat}}$  is 175mV. If cascoded current sources are used, then  $2V_{\text{dsat}}$  must be removed from both sides of the power supply to reduce the available output range to 0.5V. Suppose further that the zero-crossing detector offset is nulled, and that the input referred BDC offset is  $\pm 25\text{mV}$  worst case. Then the output referred offset will be  $\pm 50\text{mV}$ . Taking 50mV away from both sides of the available output range reduces it to 0.4V. The typical implementation would then set  $V_{\text{refp}}=0.8\text{V}$  and  $V_{\text{refm}}=0.4\text{V}$  and limit the input range to 0.4V to match the available output range. Using reference voltage scaling, on the other hand, with a redundancy of 3 bit decision comparators allows for the reference voltages to scaled by 3x so that the  $V_{\text{refp}}=1.2\text{V}$  and  $V_{\text{refm}}=0\text{V}$ . The input range would then scale by 2x to 0.8V, and the  $\text{SNR}^2$  would increase by 4x. This is exactly the conditions that are plotted in the residue plots of Figure 5-19.

Voltage reference scaling can be generalized for the case when  $n$  bit decision comparators are used to realize a  $\log_2(n + 1)$  bits/stage pipeline stage and when the residue amplifier gain is  $G$ . The case of no redundancy is when  $\log_2(n + 1) = G$  and redundancy is introduced whenever  $\log_2(n + 1) > G$ . As compared to the case when no redundancy is used, using redundancy can increase the input range by a factor

$$x_{ir} = \frac{n + 1}{G}$$

when the reference voltages are scaled by a factor

$$x_{ref} = \frac{n}{G - 1}.$$

The  $\text{SNR}^2$  then scales as the square of the input range, so the

$$x_{\text{SNR}} = \left( \frac{n + 1}{G} \right)^2.$$

So in the example shown in Figure 5-18, which is a 1.5 bit/stage,  $G = 2$  and  $n = 2$ , so the input range scales by 1.5x, the reference voltages scale by 2x, and the  $\text{SNR}^2$  by 2.25x. In the example shown in Figure 5-19,  $G = 2$  and  $n = 3$ , which gives a input

range scaling of 2x, a reference voltage scaling of 3x, and an SNR scaling of 4x.

For this particular design, the power supply is  $V_{DD}=1.2V$ . With a  $V_{dsat}$  of 150mV and input referred BDC offset of 25mV, the available output range is 0.4V. By selecting  $G = 4$  and  $n = 9$ , the input voltage range can scale by a factor of 2.5x from 0.4V to 1.0V, the references can scale by 3x from 400mV to 1.2V, and the  $SNR^2$  scales by 6.25x. The residue plots both without and with reference voltage scaling for this particular case are shown in Figure 5-20.

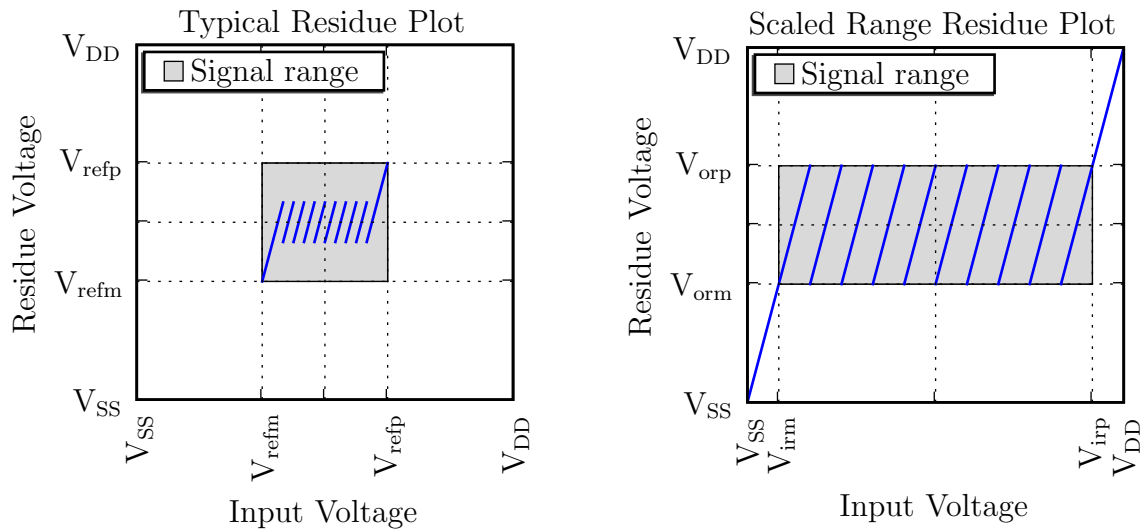


Figure 5-20: Residue Plots for gain  $G = 4$  and number of bit decision comparators  $n = 9$ . This yields a 3.3 bit/stage pipeline ADC with gain reduction.

## 5.5 Complete ZCBC Pipeline Stage

The schematic of a complete pipeline stage implemented for this fully differential design is shown in Figure 5-21. This schematic uses iterative instantiation notation, where, for example,  $U_{1+}[8:0]$  means there are nine unique instances of that symbol. Wide lines represent buses of with unique routes. As shown there are nine  $C_{1\pm}$  capacitors and three  $C_{2\pm}$  capacitors, which realizes a gain of 4. The twelve unit capacitors on each side driven with twelve current sources on each side. Not shown are the twelve dummy current sources for each side. The sub-ADC includes nine bit decision comparators that produces a ten level (3.3b) sub-ADC. The nine bits out

of the sub-ADC differentially drive nine analog reference voltage multiplexers of the type shown in Figure 5-14. The sampling switch  $M_1$  is connected to the zero-crossing detector output of the previous stage. The offset of the zero-crossing detector is digitally programmed into via the  $off[7:0]$  bus.

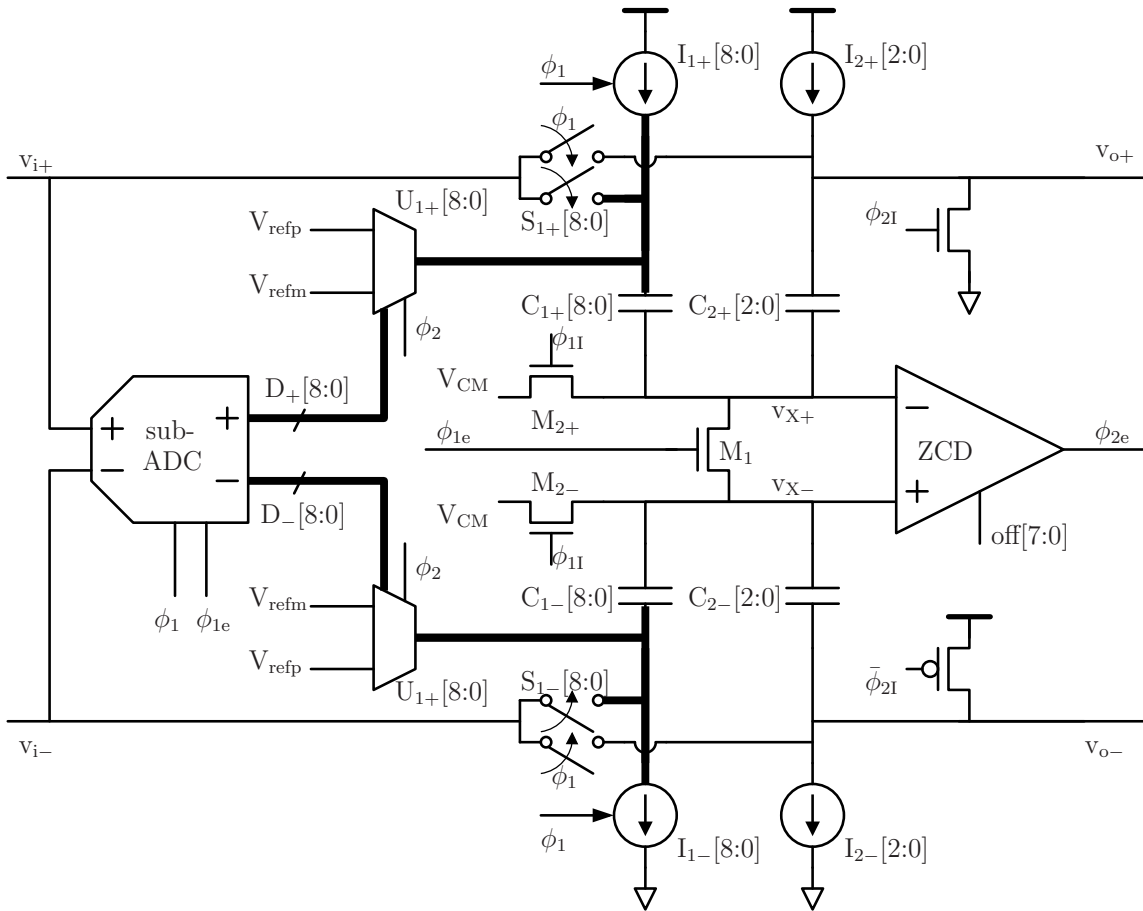


Figure 5-21: Complete ZCBC fully differential pipeline stage.

Current source splitting is used for each stage on this design as was introduced in Section 3.3.2, but the shorting switches are implemented as complimentary pass gate devices rather than gate-boosterd NMOS switches as in the initial ZCBC design. To reduce the current through these switches and to deal with the capacitive load changing between sampling and transfer phase (see Section 3.3.3), only two rather than all three of the current source labeled  $I_{2\pm}$  in Figure 5-21 are enabled during the transfer phase.

The first stage of the ZCBC pipelined ADC as shown in Figure 5-22 is a slightly

different than other stages. Sampling device  $M_1$  has been removed so that devices  $M_{2\pm}$  sample the input with respect to the common mode voltage for the entire duration of the sampling phase. Since  $\phi_1$  falls prior to  $\phi_{1d}$ , these switches open first to perform bottom-plate sampling. The switches  $S_{1\pm}$  use gate boosting to realize a constant  $V_{GS}$  NMOS switch. Because the sampling capacitors of the first stage sample the low impedance input voltage directly, these capacitors do not require current sources to generate voltage ramps during sampling. During the transfer phase, however, current source  $I_{2\pm}$  is required to generate the voltage ramp on the series connected sampling capacitors.

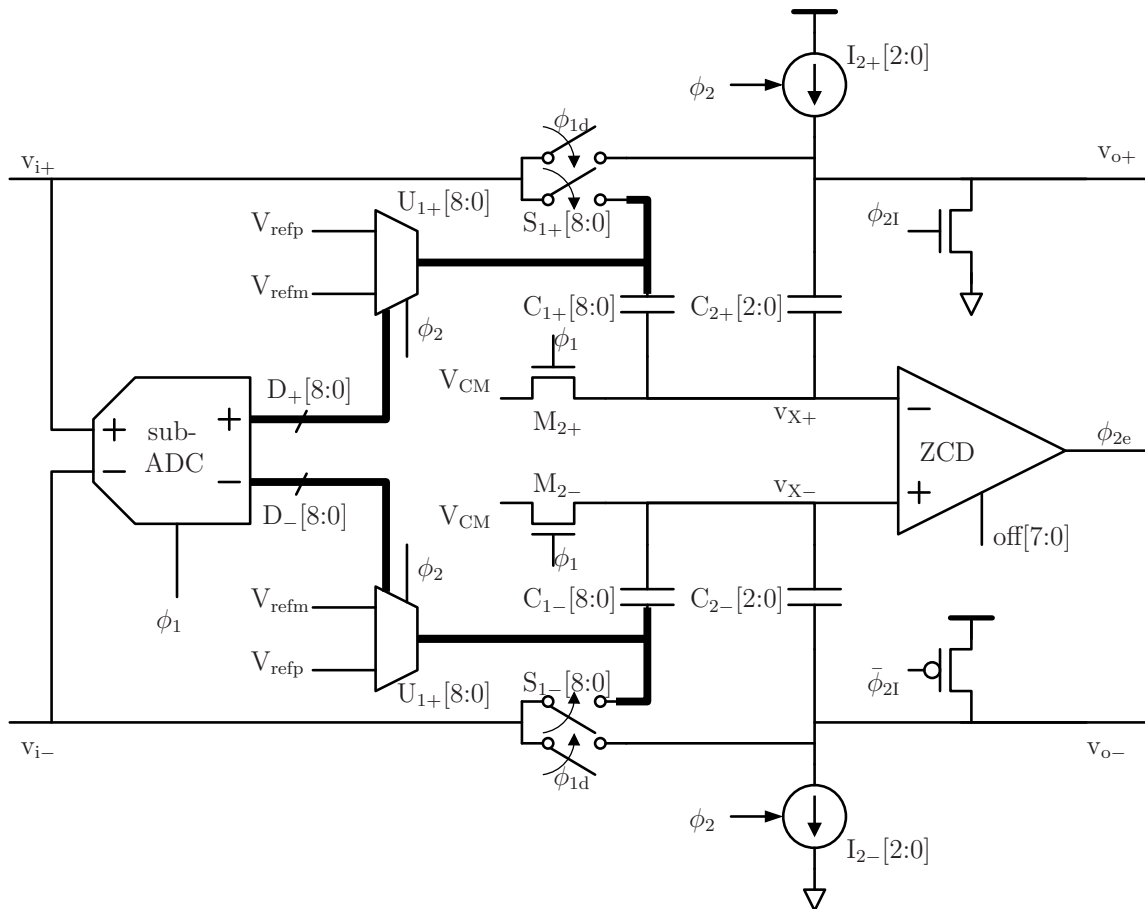


Figure 5-22: First stage of ZCBC fully differential pipeline ADC.

Although it is not shown in the first stage schematic of Figure 5-21, the input sampling switches  $S_{1\pm}$  are actually implemented with as a switching matrix as introduced in [23] to allow for input chopping or modulation as shown in Figure 5-23.

These switches are implemented in the same way as the initial ZCBC chip as described in Section 3.3.4. Except for these four input sampling switches, this design uses no additional gate-boostered switches.

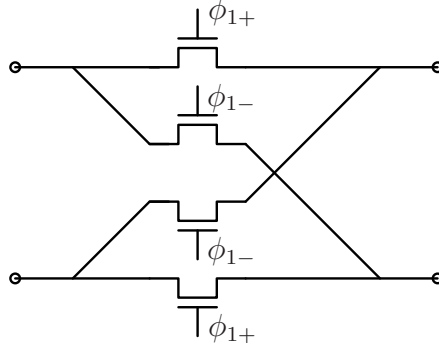


Figure 5-23: Switch matrix implementation of input sampling circuit.

To realize a 12 bit ADC, six pipeline stages are implemented. The first stage is scaled four times larger than the remaining stages which are identical. The unit capacitance of the first stage is 415 fF, meaning the total input capacitance is approximately 5 pF on each input terminal. The equivalent ENOB due to a dose of uncorrelated thermal noise sampled on each input terminal can be expressed as

$$\text{ENOB} = \frac{1}{2} \log_2 \left( \frac{V_{\text{FS}}^2}{12} \cdot \frac{C}{2kT} \right) \quad (5.8)$$

A full scale differential input range of  $V_{\text{FS}} = 2\text{V}$  yields an sampling ENOB of 13.63 bits, which leaves plenty of margin for the 12 bit design goal.

## 5.6 Sub-ADC Design

While over-range protection minimizes the impact of offsets in the sub-ADC of each stage, any offset will increase the required output range. Therefore, in this design, special care was given to ensure that no systematic offset was introduced into the sampling path of the sub-ADC of each stage.

Beginning with the sub-ADC of the first stage, special care must be given to the input sampling circuit to ensure that the voltage sampled by the sub-ADC matches

that of the first stage sampling capacitors. While over-range protection will be able to compensate for any voltage difference up to a certain point between the two paths, this design avoids the issue by using the same exact sampling circuitry and timing for both paths. After sampling, switched capacitor techniques are used to subtract the differential reference from the differential input so that it can be compared with a standard two terminal bit decision comparator.

Figure 5-24 shows the schematic of the sub-ADC used in the first stage. A comparison of the sampling capacitors in this sub-ADC with the sampling capacitors of the first stage (see Figure 5-22) shows that both utilize the same circuitry and timing for sampling where the switches and capacitor are scaled to have the same ratio. Bottom plate sampling [21] is used by turning off switches  $M_{2\pm}$  prior to switches  $S_{1\pm}$  to reduce signal dependant charge injection. After sampling completes when  $\phi_1$  falls,  $\phi_2$  rises to close switches  $S_{2\pm}$ . This subtracts and inverts the differential  $V_{REF}$  signal from the sampled input. The bit decision comparator then fires a short time later to produce the bit decisions  $D[8:0]$ . The schematic of Figure 5-24 uses iterative instantiation to show the parallel instantiation of the nine circuits that make up the sub-ADC. The nine different reference voltages  $V_{REF}[8:0]$  are generated using the resistor string shown in Figure 5-25.

The sub-ADC for the stages that follow the first must also sample using the same circuitry as the signal path to avoid systematic offset in the bit decision locations. Figure 5-26 shows the implementation used in this design. Comparing this to Figure 5-21 shows that the sampling circuitry between the two is identical. Furthermore, just as the sub-ADC for the first stage, the sub-ADC for the remaining stages uses switched capacitor techniques to subtract the reference voltage from the signal prior to comparison to generate the bit decisions  $D[8:0]$ .

Observe that the sub-ADC implementation for all the stages following the first does not implement the two outermost bits decisions and only utilizes seven parallel circuits to generate the seven inner bit decisions. A look at the scaled-range residue plot for a this design as shown in Figure 5-20 reveals that implementing the outermost bit decision comparators is unnecessary. This is because the output range is reduced

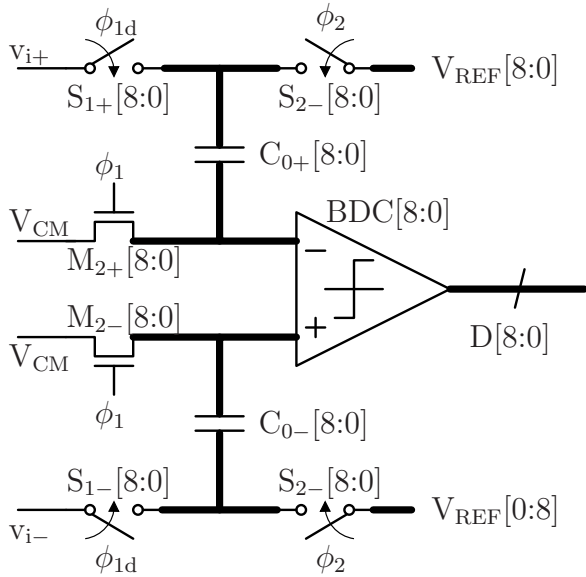


Figure 5-24: First stage sub-ADC implementation utilizing bottom plate sampling.

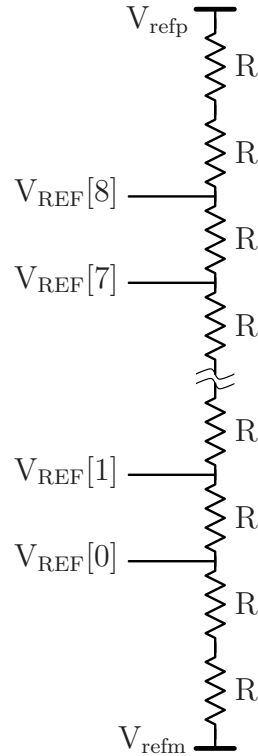


Figure 5-25: Resistor string to generate nine sub-ADC reference voltages.

for a factor of 2.5 over the input range, and the output range of the the first stage becomes the input range of the next stage. Thus, the input into the stages after the first cannot be in the outermost bit decision range unless there are severe over-range issues. It is true that more than just the two outermost bits can be dropped, but each bit dropped reduces the over-range protection by the size of the bit decision quantum. These bit decisions cannot just be dropped completely, however, because all nine bit decisions are required to drive the analog voltage reference selection multiplexer. Instead, the outermost bit decisions are simply hard-coded to eliminate the actual instantiation of a bit decision comparator to make a comparison. This saves a little power and area but most importantly for this design given that the raw bit decisions are not reconstructed on chip, it saves pins and SRAM size as the hard-coded bit decisions for each stage do not need to be sent off-chip. Dropping the number of bit decisions from nine to seven drops the number of bits after the thermometer to binary encoding from four to three.



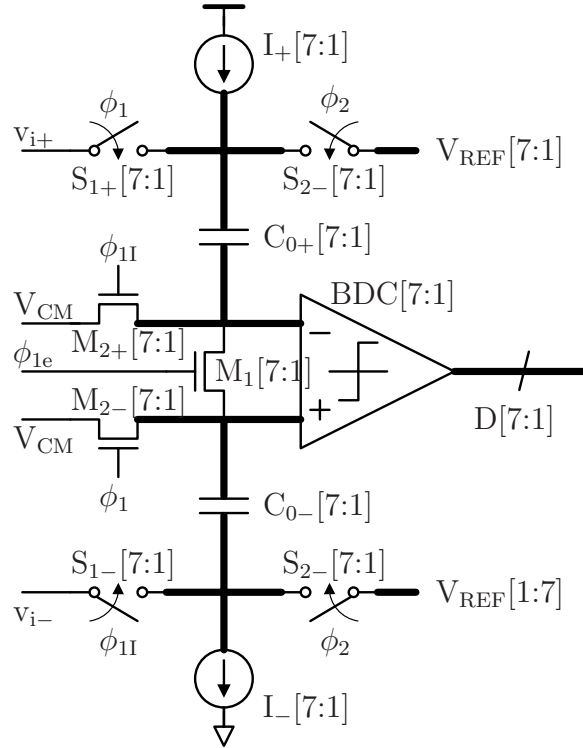


Figure 5-26: Sub-ADC implementation for all stages except the first.

### 5.6.1 Bit Decision Comparator Design

The four bit decision comparator architectures depicted in Figure 5-27 were considered for this design. Each is similar in nature in that devices  $M_3$ - $M_6$  make a cross coupled latch. The latch is reset when the clock  $\phi$  goes low to send both outputs  $v_{o-}$  and  $v_{o+}$  high. The reset is performed by the reset devices  $M_7$  and  $M_8$  turning on and the enable devices  $M_9$ - $M_{11}$  turning off. When the clock goes high, the comparator enters the evaluation phase and the input pair consisting of  $M_1$  and  $M_2$  differentially controls which way the unstable positive feedback of the latch tips to latch the decision state.

BDC A features a standard BDC with the enable device  $M_{11}$  at the bottom such that the input devices  $M_1$  and  $M_2$  immediately enter saturation when  $M_{11}$  turns on to enable the comparison. BDC B is a slight variant where the reset devices  $M_7$  and  $M_8$  have been split to provide explicit initialization to all internal nodes of the latch. As can be seen in the simulation results of Figure 5-28, this results in a slightly lower RMS offset as  $M_3$  and  $M_4$  start in triode and initially have a lower gain when

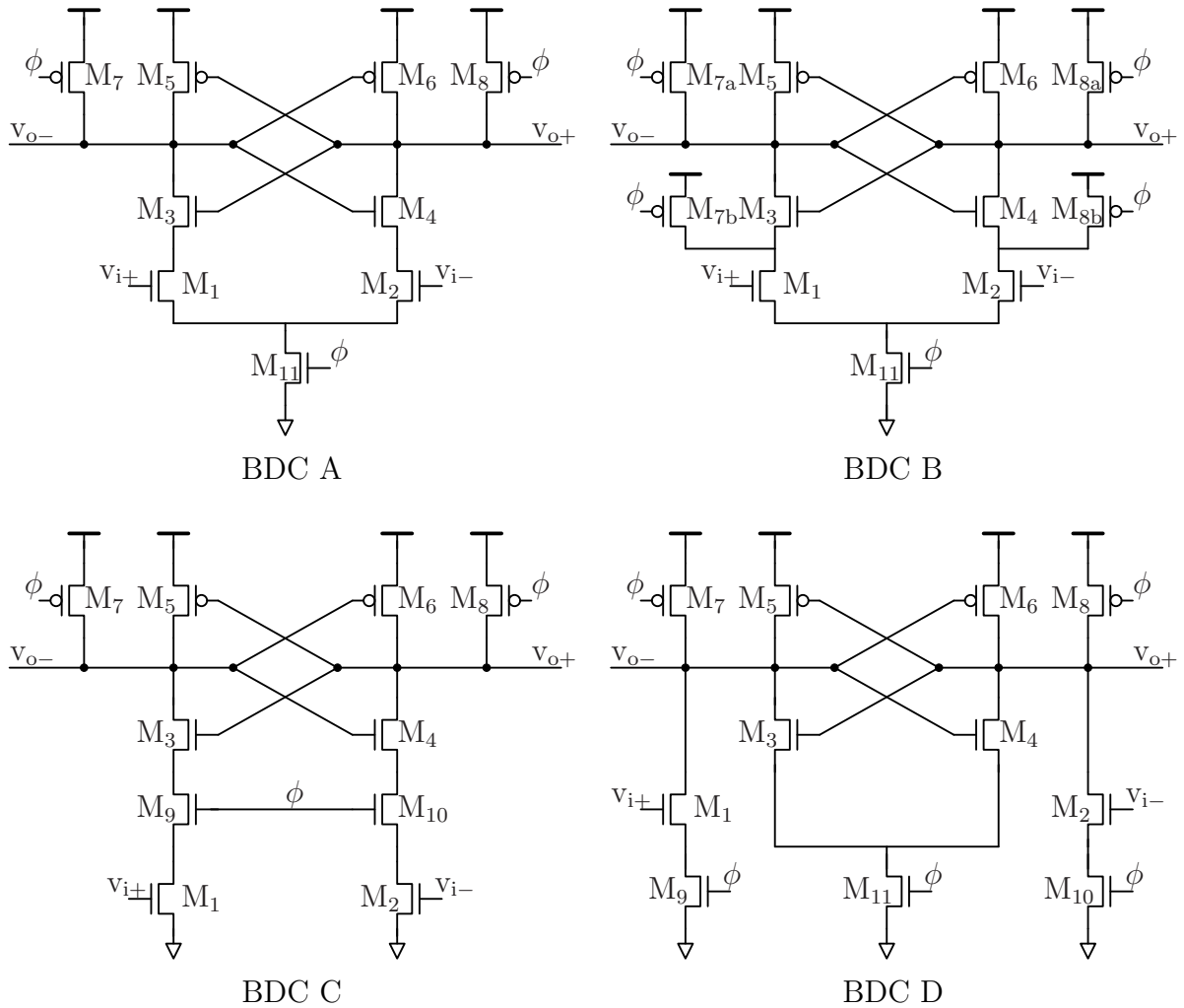


Figure 5-27: Possible BDC implementations compared for offset, noise, and speed. BDC B is used in this design.

compared to BDC A. Because they start with lower gain, their input referred offset is less, so the mismatch between these devices has less effect on the decision. The trade off for this lower gain is a slightly slower response.

BDC C is similar to BDC A with the exception that device  $M_{11}$  is split to form devices  $M_9$  and  $M_{10}$  and swapped in position with the input devices. The result is that during the reset phase, the input pair will enter triode because the enable devices  $M_9$  and  $M_{10}$  turn off. When the enable devices turn on, they start in saturation while the input pair is still in triode. Because of this, the enable devices start with the most dynamic gain into the latch, and the offset caused by the device mismatch between these parameters gets amplified by the gain ratio of these devices. As the

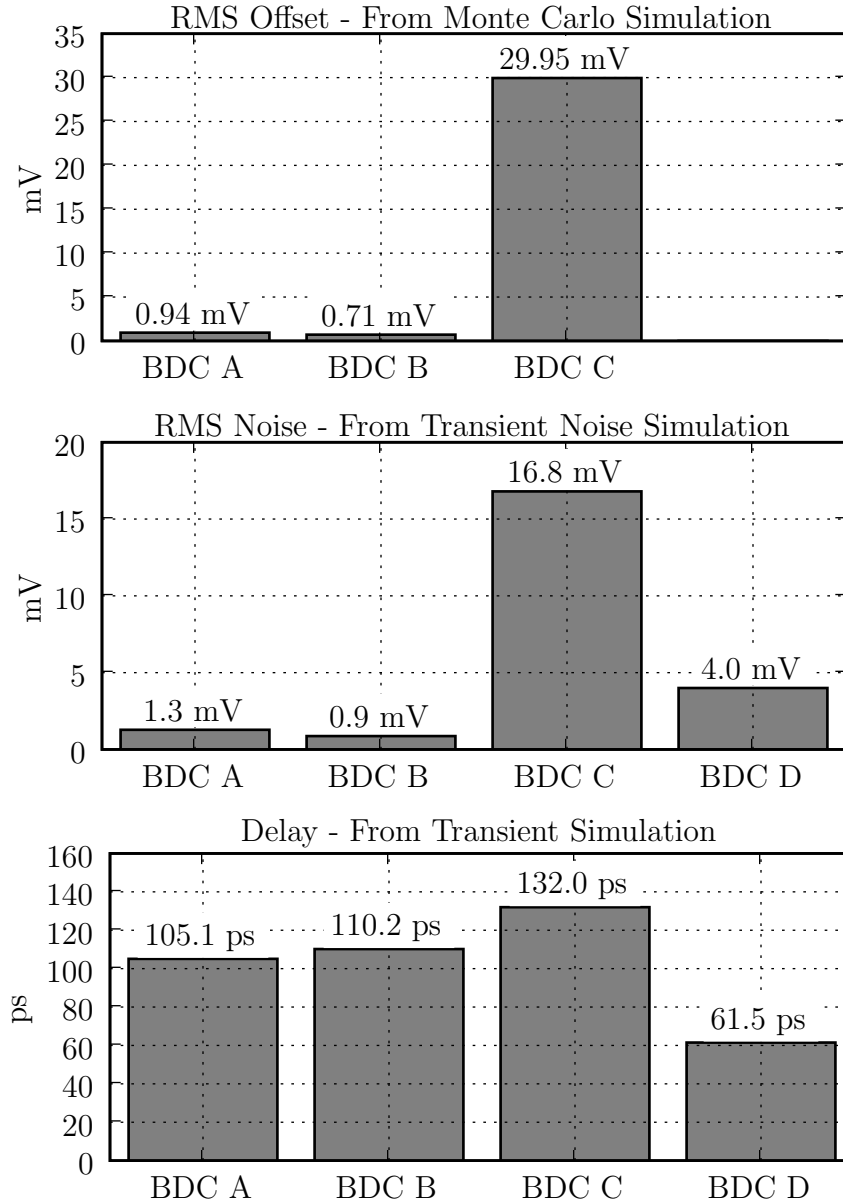


Figure 5-28: BDC comparison simulation results.

Monte Carlo offset simulations in Figure 5-28 show, BDC C has more than an order of magnitude more RMS offset than BDC A and B. It is also slower and noisier for the same reason that the input devices have much less gain into the latch at the start of the comparison. The big advantage of this topology is that it has much less kick back than the other topologies. This is because the drain of the input devices swings much less than the others.

BDC D is yet another variation of BDC A where the input devices are removed

from the series path of the latch and put in parallel instead. This increases their initial gain into the latch and thus increases the speed by a factor of almost 2. The problem is that this approach draws static current as one of the input devices will continue to draw current after the latch latches. An approach similar to this was used on the initial ZCBC design present in Chapter 3 where the enable devices  $M_9$  and  $M_{10}$  were further conditioned on the output so that as soon as the latch made a decision it turned off the input devices.

Each of these designs was simulated using the Eldo circuit simulator from Mentor Graphics to compare the performance of each. A Monte Carlo transient simulation was performed to obtain the results of the first plot. A transient noise simulation was performed in the second plot, and the delay for a zero voltage input was obtained using a transient simulation in the third plot. The trade-offs of each design can be seen from these results. An unfortunate turn of events led to an initial tapeout of this design that utilized BDC C. The reason is that a lack foresight to perform the above analysis and simulations won out over the time constraints of the tapeout deadline. BDC C is clearly the worst performer in all the regards shown in the simulation results. The resulting chip was unusable as the BDC offset and noise were so large that even despite the extreme over-range protection of this design, the signal would go out of range and cause serious distortion for almost all input codes (see Figure 5-40). As a result, a chip revision was performed after the above analysis, and BDC B was selected as it provided the best offset and noise performance while being sufficiently fast for this application.

## 5.7 Noise Analysis

### 5.7.1 Dynamics

If the dynamics of the zero-crossing detector are approximated well as a single pole response, then the response will take the form

$$H(s) = \frac{A}{s\tau + 1} \quad (5.9)$$

where  $A$  is the gain and  $\tau$  is the pole location.

The input ramp  $x(t)$  into the zero-crossing detector can be expressed as

$$x(t) = Ktu(t)$$

where  $K$  is the slope of the input ramp,  $t$  is time, and  $u(t)$  is a unit-step function. The Laplace transform of  $x(t)$  is  $X(s) = \frac{K}{s}$ , and thus the output will be  $Y_x(s) = X(s)H(s)$ . Using inverse Laplace transform properties, the time domain output signal  $y_x(t)$  can be found to be

$$y_x(t) = AK \left( t - \tau \left( 1 - e^{-\frac{t}{\tau}} \right) \right) u(t) \quad (5.10)$$

In reality the input ramp  $x(t)$  starts negative and ramps to zero. Calling this starting point  $v_s$  and including it in the  $x(t)$  and  $y_x(t)$  gives

$$x(t) = Ktu(t) - v_s \quad (5.11)$$

$$y_x(t) = AK \left( t - \tau \left( 1 - e^{-\frac{t}{\tau}} \right) \right) u(t) - Av_s. \quad (5.12)$$

The time  $T_1$  when the input crosses zero is

$$T_1 = \frac{v_s}{K}.$$

The time  $T_2$  when the output  $y_x(t)$  crosses zero is

$$T_2 = T_1 + \tau \left( 1 - e^{-\frac{T_2}{\tau}} \right).$$

Solving for  $T_2$  explicitly is difficult, however, by defining  $\alpha$  as the ratio of  $T_2$  to  $\tau$ ,

$$\alpha = \frac{T_2}{\tau},$$

$T_2$  can be expressed as

$$T_2 = T_1 + \tau(1 - e^{-\alpha}).$$

$\alpha$  represents the amount of settling that occurs on the output of the zero-crossing detector. The delay  $t_d$  of the zero-crossing detector is then  $t_d = T_1 - T_2$  and is

$$t_d = \tau(1 - e^{-\alpha}).$$

When  $\alpha$  is large and  $1 \gg e^{-\alpha}$ , the output has largely settled by the time zero-crossing detector switches and  $t_d \approx \tau$ . For a single ramp ZCBC architecture, the starting point  $v_s$  is unknown. In order to ensure a constant switching threshold, the dynamics of the zero-crossing detector must be either settled or consistently at the same point over the entire output signal range when the zero-crossing detector switches. In either case, since the zero-crossing detector cannot be infinitely fast, this requires starting the output signal below the valid signal range to give the dynamics zero-crossing detector an opportunity to play out even for the minimum output signal. This is precisely the reason why the pre-charge state resets the output voltage below the output range—it provides opportunity for the dynamics to have settled adequately for all possible output voltages.

### 5.7.2 Input Referred Noise Derivation

Just as the noise analysis of Section 3.5 for the dynamic zero-crossing detector, the fully-differential zero-crossing detector requires a non-stationary noise analysis to find an input-referred noise quantity for the noise generated internal to the zero-crossing detector. The dynamics of this zero-crossing detector are described by a single pole system.

The zero-crossing detector generates additive white Gaussian noise  $n(t)$  with a

spectral density of  $N/2$ . If the internal nodes of the comparator are initialized to a “noiseless” condition at the beginning of the ramping phase, then the additive noise can be modelled as  $v(t) = n(t)u(t)$ , where  $u(t)$  is the unit-step function. Given a single pole system is linear and time-invariant (LTI), the noise on the output of the zero-crossing detector will be independent of the input signal  $x(t)$  and can be included using the superposition principle. Call the noise at the output  $w(t)$ .  $w(t)$  is not wide-sense stationary (WSS), however, it is Gaussian. Thus, the auto-correlation  $R_{ww}(t_1, t_2)$  is sufficient to describe the statistics of  $w(t)$ . The time-domain impulse response of the zero-crossing detector is

$$h(t) = \frac{A}{\tau} e^{-\frac{t}{\tau}} u(t).$$

The auto-correlation function of the noise of the zero-crossing detector is

$$R_{xx}(t_1, t_2) = \frac{N}{2} \delta(t_1 - t_2).$$

The problem of finding the auto-correlation of the noise at the output for the white noise input being applied at time 0 has been solved in [36, Eq.9-96] and the result is that

$$R_{ww}(t_1, t_2) = \frac{NA^2}{4\tau} \left(1 - e^{-\frac{2t_1}{\tau}}\right) e^{-\frac{|t_2 - t_1|}{\tau}} \quad (5.13)$$

for  $0 < t_1 < t_2$ . The variance of  $w(t)$  at time  $t$  is  $R_{ww}(t, t)$ , which is

$$\sigma_w^2(t) = \frac{NA^2}{4\tau} \left(1 - e^{-\frac{2t}{\tau}}\right) u(t) \quad (5.14)$$

The zero-crossing detector will switch when its output reaches 0, which corresponds to the condition  $y_x(t) + w(t) = 0$ . If the deterministic time when the input crosses zero is  $T_c$ , then  $w(t)$  causes jitter on the time when the zero-crossing detector switches. Therefore, the actual crossing time can be defined as a random variable  $T_a = T_c + T_j$  where  $T_j$  is a random variable that captures the jitter. The probability

distribution of  $T_a$  can be found using cumulative distributions as follows:

$$\begin{aligned}\Pr(T_a < t) &= \Pr(y_x(t) < w(t)) \\ &= \int_{-\infty}^{y_x(t)} \mathcal{N}(0, \sigma_w(t)) dw\end{aligned}\quad (5.15)$$

where  $\mathcal{N}(m, \sigma)$  is the normal distribution with mean  $m$  and variance  $\sigma^2$ . The probability density  $f_T(t)$  of the random variable  $T_a$  is then the derivative of Equation 5.15 with respect to  $t$ ,

$$f_T(t) = \frac{\partial \Pr(T_a < t)}{\partial t}\quad (5.16)$$

Given that both  $\sigma_w(t)$  and  $y_x(t)$  are functions of  $t$ , solving for Equation 5.16 in closed form is difficult without making some simplifying approximations. A second order Taylor series expansion of  $y_x(t)$  about  $T_c$  gives

$$y_x(t) \approx AK \left( T - \tau \left( 1 - e^{-\frac{t}{\tau}} \right) \right) - v_s + AK \left( 1 - e^{-\frac{T}{\tau}} \right) (t - T)$$

Furthermore, approximating the noise power  $\sigma_w^2(t)$  of Equation 5.14 as a constant at time  $T_c$  as follows:

$$\sigma_w^2 \approx \frac{NA^2}{4\tau} \left( 1 - e^{-\frac{2T_c}{\tau}} \right)\quad (5.17)$$

Substituting these approximations into Equation 5.16 yields that the random variable  $T_a$  is a Gaussian with a mean  $T_c$  and variance

$$\sigma_T^2 \approx \frac{\sigma_w^2}{A^2 K^2 \left( 1 - e^{-\frac{T_c}{\tau}} \right)^2}\quad (5.18)$$

Given the input into the comparator is moving with a slope  $K$ , referring the jitter  $\sigma_T^2$  back to an input referred noise  $\sigma_v^2$  is

$$\sigma_v^2 = K^2 \sigma_T^2.\quad (5.19)$$



Substituting Eq. 5.18 into this result gives the input referred noise as

$$\sigma_v^2 = \frac{N}{4\tau} \left( \frac{1 + e^{-\frac{T_c}{\tau}}}{1 - e^{-\frac{T_c}{\tau}}} \right) = \frac{N}{4\tau} \coth \left( \frac{T_c}{2\tau} \right) \quad (5.20)$$

The approximation made in Equation 5.18 that the noise is constant at the time of the zero-crossing assumes that the system reduces the bandwidth of the noise sufficiently so that it looks constant in the region of the zero-crossing. This approximation means the noise is effectively filtered rather than a peak-detected when referred to the input. The same result of Equation 5.20 can be obtained in the more intuitive manner by calculating the input-referred noise from the output-referred noise using the dynamic gain of the zero-crossing detector. Expressed mathematically, this is

$$\sigma_v^2 = \left( \frac{\partial x / \partial t}{\partial y_x / \partial t} \right)^2 \sigma_w^2.$$

Further insight into the result of Equation 5.20 can be obtained by defining  $\alpha = \frac{T_c}{\tau}$ . Substituting this into Equation 5.20 gives the input referred noise as

$$\sigma_v^2 = \frac{N (1 + e^{-\alpha})}{4\tau (1 - e^{-\alpha})} \quad (5.21)$$

Looking at this in the two extremes when  $\alpha$  is much larger than 2 and when  $\alpha$  is much smaller than 2 gives the following approximations

$$\begin{aligned} \sigma_v^2 &\approx \frac{N}{4\tau} && \text{when } \alpha \gg 2 \\ \sigma_v^2 &\approx \frac{N}{2T_c} && \text{when } \alpha \ll 2 \end{aligned}$$

This means when  $\alpha$  is large, the dynamics of the zero-crossing detector have largely settled by the time the zero-crossing detector switches. The noise looks stationary under this condition, and this result is equivalent to that obtained from a stationary analysis of filtering white noise with a single pole filter with a time constant of  $\tau$ . On the other hand, when  $\alpha$  is small, the dynamics of the system are slow compared to

the time  $T_c$ . This means that  $T_c/2$  becomes the effective time constant of the system that sets the noise bandwidth.

When in the “slow” regime when  $\alpha < 2$ , the initial conditions of the zero-crossing detector become important to ensure the Zero Input Response (ZIR) of the system is sufficiently small when the zero-crossing detector switches to avoid introducing signal dependant errors or extra noise into the system. Because the ZIR settles exponentially with the time constant  $\tau$ , when operating in the “slow” regime, the initial conditions will still be in the process of settling out when the zero-crossing detector switches. Thus any signal history or noise on the initial conditions must be sufficiently small for a given performance constraint. Since the system is “slow,” however, additional circuitry is probably necessary that explicitly resets or clamps the output quickly during the pre-charge phase.

### 5.7.3 Substituting Real Circuit Parameters

Using the circuit architecture of the zero-crossing detector in Figure 5-8, the noise spectral density of the pre-amplifier can be expressed as

$$\frac{N}{2} = 4kT \frac{\gamma(2+b)}{g_m}, \quad (5.22)$$

where  $b$  is the effective number of devices in addition to the input pair that contribute noise,  $g_m$  is the transconductance of the input pair,  $\gamma$  is  $\frac{n}{2}$  for devices in weak inversion ( $n$  is the ideality factor for weak inversion), and  $\gamma$  is  $\frac{2}{3}$  for devices in strong inversion. Substituting the following definitions

$$\begin{aligned} V_p &= \frac{I_d}{g_m} \\ \beta &= 4kT\gamma(2+b)V_p \\ f(\alpha) &= \alpha \frac{1 + e^{-\alpha}}{1 - e^{-\alpha}}, \end{aligned}$$

into the input referred noise expression of Equation 5.21 gives

$$\sigma_v^2 = \frac{\beta f(\alpha)}{4I_d T_c}. \quad (5.23)$$

This gives the fundamental constraint on the relationship between SNR, power, and speed. Solving this for the drain current gives

$$I_d = \frac{\beta f(\alpha)}{4\sigma_v^2 T_c}. \quad (5.24)$$

Recall that  $T_c$  in the previous noise analysis was the time given to the zero-crossing detector to make its decision. Since this design uses a single ramp scheme, the minimum  $T_c$  is the time it takes for the output voltage to ramp from the pre-charged state ( $V_{SS}$ ) to the minimum possible output ( $V_{orm}$ ). This time is called the “runway” time as it corresponds to the time given to ensure that all possible output voltages see the same dynamics.  $T_c$  is constrained by the sampling rate and reference voltages of the ADC. If the sampling rate and resolution of the ADC are constrained, the only free parameter in Equation 5.24 is  $\alpha$ , which corresponds to how many time constants worth of settling will occur during the “runway” period.

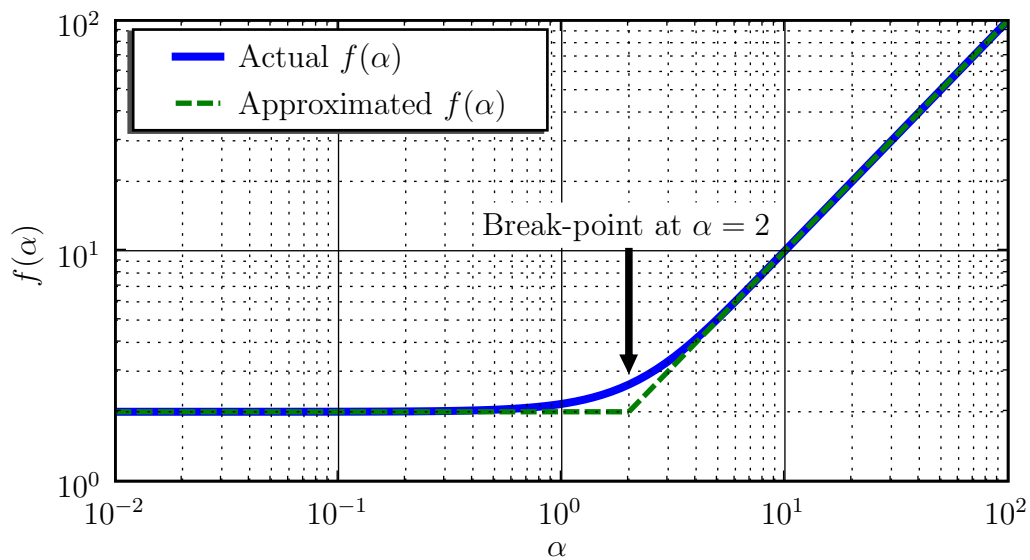


Figure 5-29: Bias current versus  $\alpha$

A plot of  $f(\alpha)$  versus  $\alpha$  is shown in Figure 5-29.  $f(\alpha)$  is a scale factor in Equation 5.24, and since  $I_d$  is proportional to the power consumption, this plot shows how the power consumption scales as a function of  $\alpha$  for a fixed sampling rate and resolution. As already discussed, there is a break-point in this plot at  $\alpha = 2$ . This point marks the transition from a “slow” zero-crossing detector to a “fast” one. This shows that power consumption scale factor levels off at 2 for speeds slower than  $\alpha = 2$ , and that power consumption scale factor increases linearly with  $\alpha$  for speeds faster than  $\alpha = 2$ . Since running the zero-crossing detector slow in the single ramp case can increase the linearity requirements of the current source as well as the difficulty in managing the dynamics of the zero-crossing detector itself, the optimal design will minimize  $\alpha$  while achieving sufficient linearity.

#### 5.7.4 Linearity from Finite Current Source Impedance

In Equation 1.12 the linearity relationship due to the finite current source impedance and delay of the zero-crossing detector was found to be

$$\epsilon_{zcd} = \frac{t_d I_0}{V_A C_T}.$$

If the linearity is desired to be constrained to an LSB, then

$$\epsilon_{zcd} = \frac{1}{2^B}.$$

Equating these two and solving for  $V_A$  gives

$$V_A = a t_d 2^B, \tag{5.25}$$

where  $a = \frac{I_0}{C_T}$  is the slope of the output voltage ramp. This gives a constraint on the output impedance required in the current sources when the ramp rate and zero-crossing delay are specified.

### 5.7.5 Differential ZCD Design Methodology

With a constraint on the input referred noise and the linearity, the zero-crossing detector and current sources can be optimally designed. The approach used in this design was to assume a given speed and resolution constraint and to design the lowest power consuming ADC possible to meet those constraints. The following design procedure, assuming the zero-crossing detector architecture shown in Figure 5-8, was used for this purpose:

1. **Select Sampling Capacitor Size.** Given the resolution constraint of a  $B$  bit ADC, the first step is to select the size of the sampling capacitors of the first stage to meet this constraint. For this design the total capacitance was selected at 5pF, which is equivalent to 13.8 bit, giving plenty of margin for a 12 bit ADC.
2. **Select ZCD Input Device Size.** The wider a transistor for a given drain current, the more power efficiently it will operate in terms of its  $\frac{I_d}{g_m}$  ratio. Therefore, the input devices of the zero-crossing detector should be as large as possible. These devices, however, set the parasitic capacitance looking into the zero-crossing detector. This parasitic capacitance amplifies the input referred noise calculated in Equation 5.23 by  $\left(1 + \frac{C_p}{C_T}\right)$ , where  $C_p$  is the parasitic capacitance and  $C_T$  is the total sampling capacitance (i.e. 5pF). To limit this noise amplification to about 5%, the input devices were limited to 250 fF. To give them improved output resistance and thus increased intrinsic gain, they were not made minimum length but rather 130nm. This limited their width to  $120\mu\text{m}$ .
3. **Select Other ZCD Device Sizes.** Minimizing the noise that devices  $M_3$  and  $M_4$  of the current mirror in the zero-crossing detector in Figure 5-8 contribute to the output requires minimizing their transconductance as much as possible with respect to the transconductance of the input pair. The amount of noise

they contribute (the parameter  $b$  in Equation 5.22) is

$$b = \frac{g_{m3}}{g_{m1}} + \frac{g_{m4}}{g_{m2}}.$$

On the other hand, these devices need adequate output impedance so as not to attenuate the gain of the input pair too severely. They also need a sufficiently low gate voltage to ensure sufficient signal range. All of these are competing constraints. For this design, low-Vt devices were used for devices M<sub>1</sub>-M<sub>4</sub> to maximize the signal range, and M<sub>3</sub> and M<sub>4</sub> were selected to have a total width of 50um and a length of 500nm. This made each contribute an additional 30% to the total mean square input-referred noise of the input pair, or  $b = 0.6$ .

The load of the pre-amplifier is a single device M<sub>7</sub>. This device is the input into the latch that must drive the sampling switches. Using a fan-out-of-4 rule for the digital logic where each logic stage is sized such that it drives a load 4x larger than itself, the size of M<sub>7</sub> is set to 20um. This makes the load of this device insignificant to the parasitic capacitance of the other devices.

4. **Select Bias Current.** With device sizes selected, the remaining free parameter is the bias current. A simulation of the pre-amplifier to perform a parametric sweep of the bias current  $I_B = 2I_d$  shows how the various circuit parameters. The first graph plots  $V_{GS} - V_T$  of the input pair and shows that they enter weak inversion at  $I_B = 300\mu\text{A}$ . The second graph plots the  $g_m$  of the input pair. Notice that the slope of  $g_m$  versus  $I_B$  does not show a strong bend at the transition from weak to strong inversion. The third graph plots  $V_p = \frac{I_d}{g_m}$  for the input pair. This shows that the devices at  $10\mu\text{A}$  are almost near the ideal of 25mV and that the efficiency degrades much quicker once the devices leave weak inversion. The fourth graphs plots the output impedance of the pre-amplifier. As expected it is inversely proportional to the bias current. The fifth graph plots  $\alpha$  for the cause when  $T_c = 1\text{ns}$ . The critical spot where  $\alpha = 2$  corresponds to to bias current of  $250\mu\text{A}$ . The sixth graph plots the pre-amplifier gain, which is  $A = g_m r_o$ . Observe that the gain of the preamplifier peaks at a

mere 18x. This is due to the low intrinsic gain of the devices. Also observe that the gain is not constant in weak inversion nor does it decrease as the square root of the bias current in strong inversion as first order device equations would predict. The seventh graph plots the effective number of bits (ENOB) from the input referred noise of the pre-amplifier based on Equation 5.23. This shows that this design achieves between 12.8 to 13.5 bits of SNR from thermal noise over the plotted bias current range. The final graph plots the corresponding Figure of Merit ( $FOM = \frac{I_B V_{DD}}{2^{ENOB} f_s}$ , where  $V_{DD} = 1.2$  V and  $f_s = 100$  MHz) of the pre-amplifier.

These plots show the trade offs that exist between power, SNR, and linearity.

5. **Scale Remaining Stages.** The remaining stages should be scaled in size and current consumption by the gain of the previous stage to minimize the power consumption and ensure that each stage contributes equal amounts of noise to total input-referred noise. This scaling relationship can be found using Lagrange Multipliers to minimize the total power consumption for fixed speed and resolution. For this design, it was only practical in terms of time and layout considerations to scale stages 2 through 6 to be 4 times smaller than the first stage. This means that the first and second stages contribute equal amounts to the input-referred noise, and the remaining stages contribute negligible noise while consuming only 4 times less power than the first stage.
6. **Calculate Current Source Output Impedance.** With  $\tau$  of the zero-crossing detector selected, the necessary output impedance of the current sources can be found using Equation 5.25 using the approximation that  $t_d \approx \tau$  as

$$V_A = a\tau 2^B,$$

where  $a$  is the output voltage ramp rate and  $B$  is the desired bit resolution.

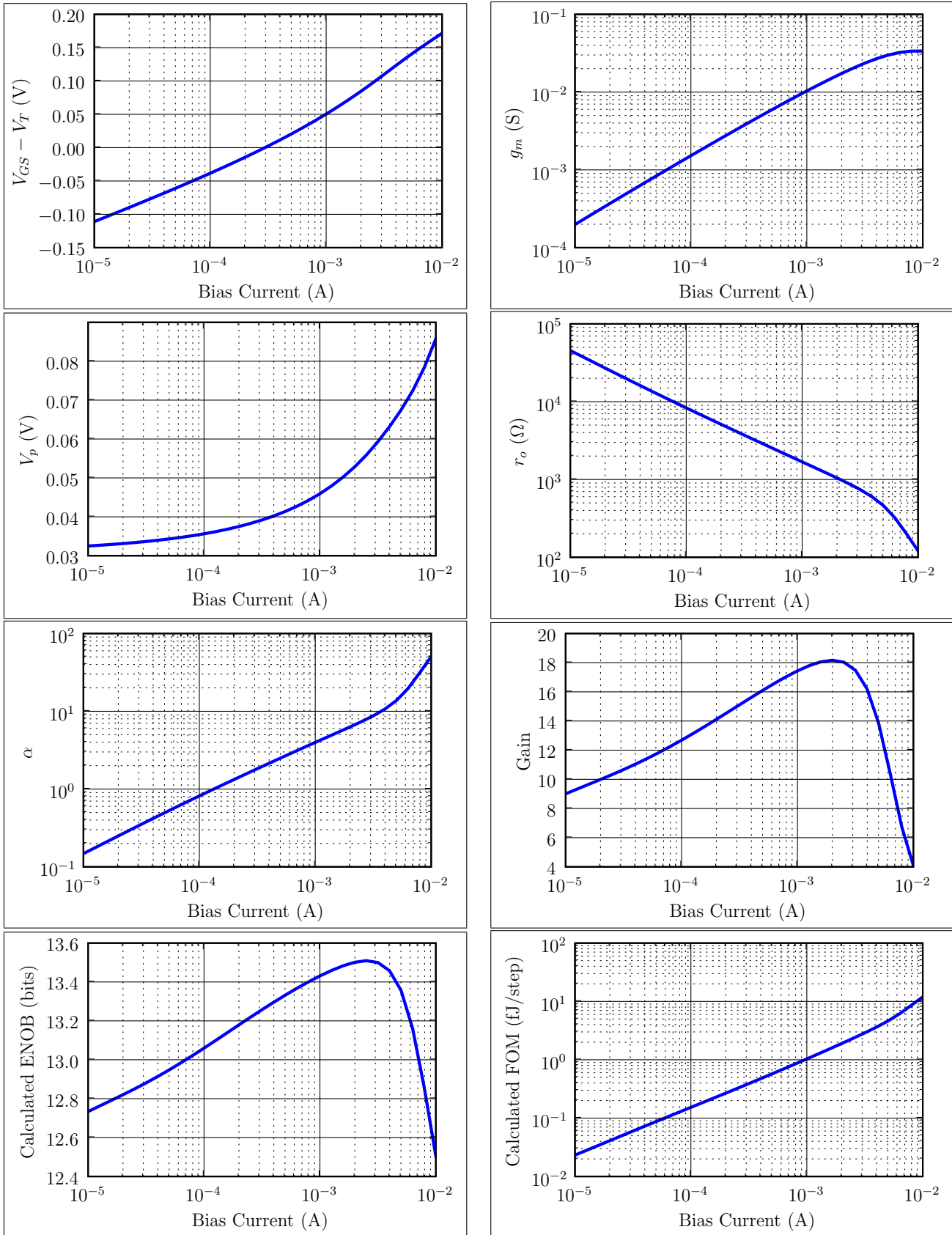


Figure 5-30: Pre-amplifier Simulation Results



### 5.7.6 Number of Ramps Analysis

The original CBSC design [18] utilized a dual ramping scheme. Equation 5.24 can be used to compare the energy consumption of a dual ramping and a single ramping scheme at the same noise and speed.

#### Signal Ramp

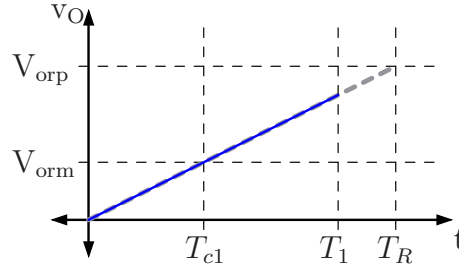


Figure 5-31: Single Ramp Timing

The timing diagram for the ramping output voltage  $v_O$  for the single ramp scheme is shown in Figure 5-31. The pre-charge phase resets the output voltage to start at ground. The output ramps until the zero-crossing detector switches at time  $T_1$ . The valid output voltage range is labeled  $V_{orm}$  and  $V_{orp}$ , so the earliest the zero-crossing detector can switch corresponds to the minimum output voltage  $V_{orm}$  and it is labeled time  $T_{c1}$ . The latest the zero-crossing detector can switch corresponds to the maximum output voltage  $V_{orp}$  and is labeled time  $T_R$ .

Because the bias current of the zero-crossing detector gets switched off after the zero-crossing detector switches, the expected amount of energy consumed can be expressed as

$$E_1 = I_d E[T_1] \quad (5.26)$$

where  $E[T_1]$  is the expected value of the time when the zero-crossing detector switches. If  $T_1$  is uniformly distributed, then the expected value of  $T_1$  is the midpoint between  $T_{c1}$  and  $T_R$ :

$$E[T_1] = \frac{T_R + T_{c1}}{2} = \frac{T_{c1}}{2} \left( \frac{V_{orp}}{V_{orm}} + 1 \right).$$

Substituting this and Equation 5.24 into Equation 5.26 gives the energy of single

ramp scheme as

$$E_1 = \frac{\beta f(\alpha)}{8\sigma_v^2} \left( \frac{V_{\text{orp}}}{V_{\text{orm}}} + 1 \right) \quad (5.27)$$

## Dual Ramps

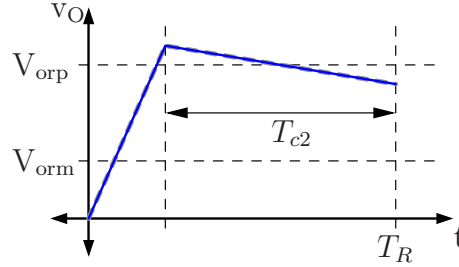


Figure 5-32: Dual Ramp Timing

The timing diagram for the dual ramp scheme is shown in Figure 5-32. Supposing the bias current is only enabled during the second phase for a time duration  $T_{c2}$ , then the energy consumed is

$$E_2 = I_d T_{c2}.$$

Substituting Equation 5.24 into this result gives

$$E_2 = \frac{\beta f(\alpha)}{4\sigma_v^2} \quad (5.28)$$

## Scheme Comparison

Comparing this result to Equation 5.27 shows that if all factors are equal, that the single ramp scheme will consume

$$\frac{E_1}{E_2} = \frac{1}{2} \left( \frac{V_{\text{orp}}}{V_{\text{orm}}} + 1 \right)$$

more energy than the dual ramp scheme. Notice, however, that as the output range is reduced, the efficiency of the single ramp scheme improves. In fact, if the output range is shrunk to nothing such that  $V_{\text{orp}} = V_{\text{orm}}$ , then both schemes consume the same amount of energy. Section 5.4 introduced the use of extra redundancy as a means of reducing the output range to give extra head room for cascoded current

sources, and this decreased output range further improves the power efficiency of the single ramp scheme. The intuition behind this result is that as the output range shrinks, the “runway” time increases, making it more power efficient.

The dual ramp scheme has an additional advantage that the dynamics of the zero-crossing detector during the second ramp phase can be much slower than those of the single ramp scheme, so  $\alpha$  can be made slower to yield additional energy improvements. Furthermore, running slower means that  $V_p$  reductions will also reduce  $\beta$  yielding further incremental improvements.

One aspect of the dual ramp scheme that this analysis neglected was the energy that would be consumed by the zero-crossing detector during the fast ramp phase. Even though the noise of the fast ramp detection can be large, this power is not negligible as it needs to be fast. Furthermore, the complexity increase and likely speed penalty of the dual ramping scheme will also add incremental energy to the system. Some of the biggest advantages of the dual ramp scheme comes when one considers other factors such as implementation of the reference voltage supplies and current sources. For this design, a the single ramp scheme was chosen for the reduced complexity and increased speed potential of the single ramp scheme.

## 5.8 Experimental Results

### 5.8.1 Overall Performance

The die photo for this design as implemented in a 90nm CMOS process is shown in Figure 5-33 in a active area of  $0.225\text{mm}^2$ . At  $50\text{MS/s}$ , the power consumption from a 1.2V supply is 4.5mW. The reference voltages are set to  $V_{DD}$  and ground to give an full scale input range of 2 volts. As shown in the linearity plots of Figure 5-34, the DNL and INL are  $\pm 0.5$  LSBs and  $\pm 3$  LSBs on a 12 bit scale. Furthermore, as shown in the frequency response plots of Figure 5-35, the SNR, SFDR, and SNDR were measured to be 72dB (11.7 bits), 68dB (11 bits), and 62dB (10 bits) respectively. Figure 5-36 plots the measured SNDR as a function of the input signal amplitude

showing that the circuit noise limit is effectively 11.7 bit and that distortion of a full scale input signal limits the resolution to 10 bits. This resulting figure of merit is 88 fJ/step.

### 5.8.2 ZCD Offset Performance

The offset range of the programmable offset ZCD as described in Section 5.2.3 is plotted in Figure 5-37. These are dependant on bias current, ramp rate, temperature,

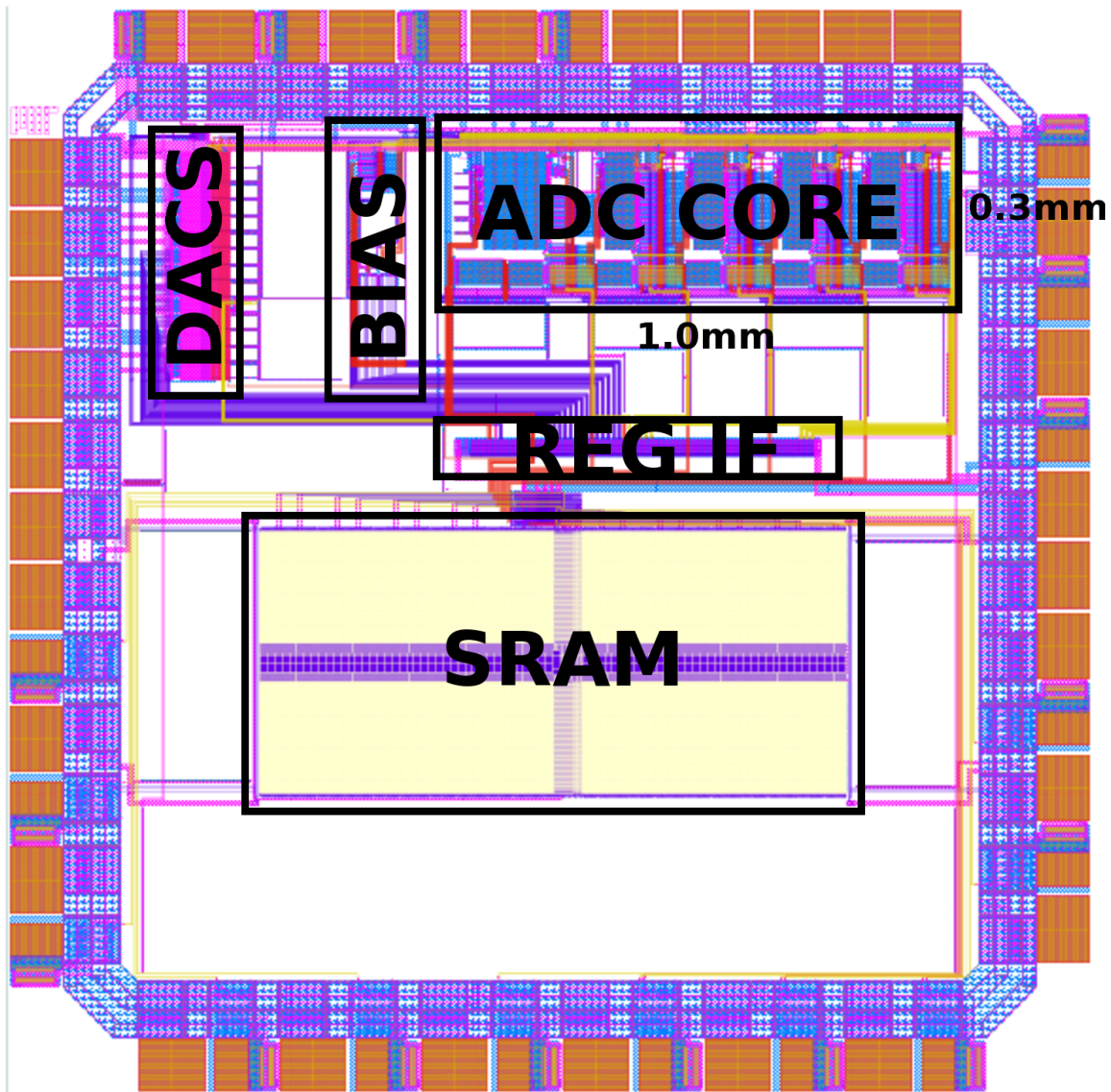


Figure 5-33: Die photo of fully differential ZCBC ADC in 90nm CMOS.

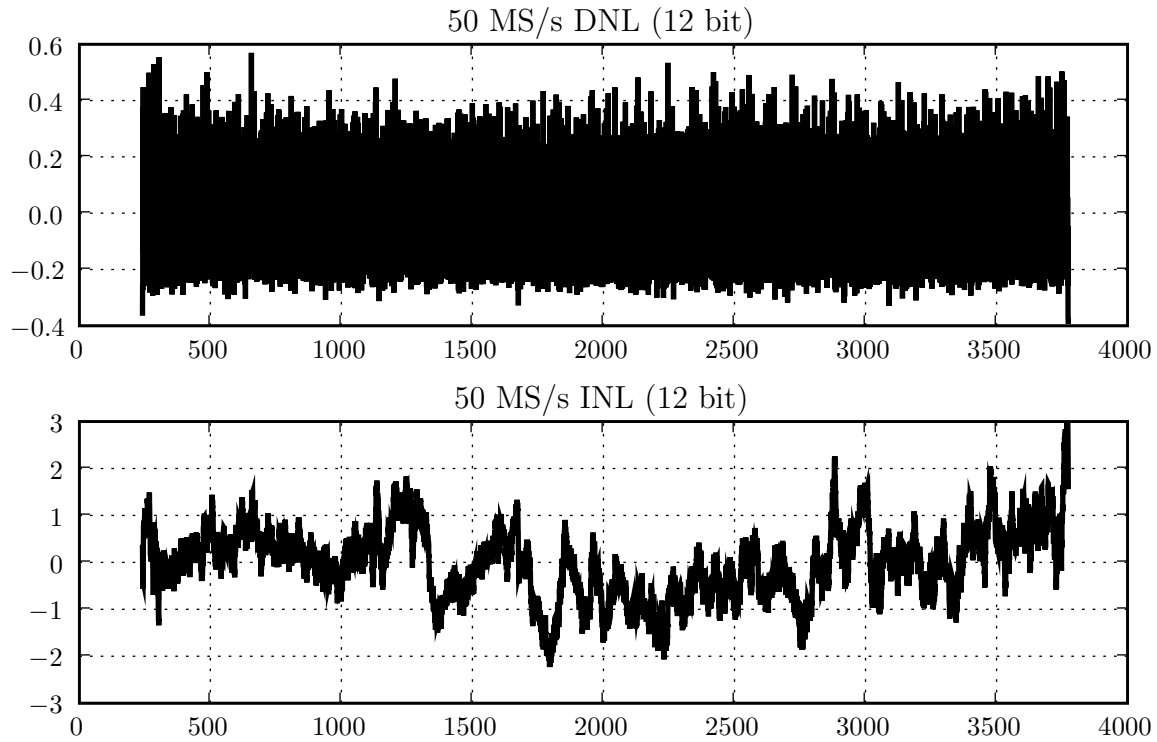


Figure 5-34: Measured Linearity

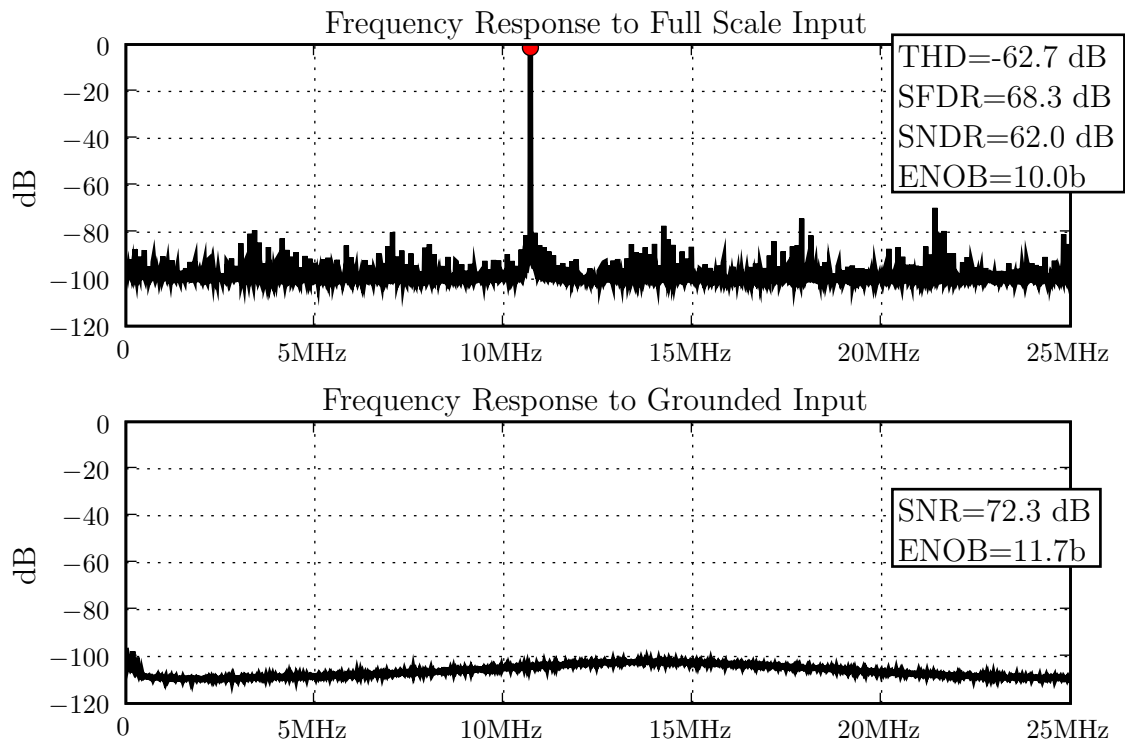


Figure 5-35: Measured Frequency Response

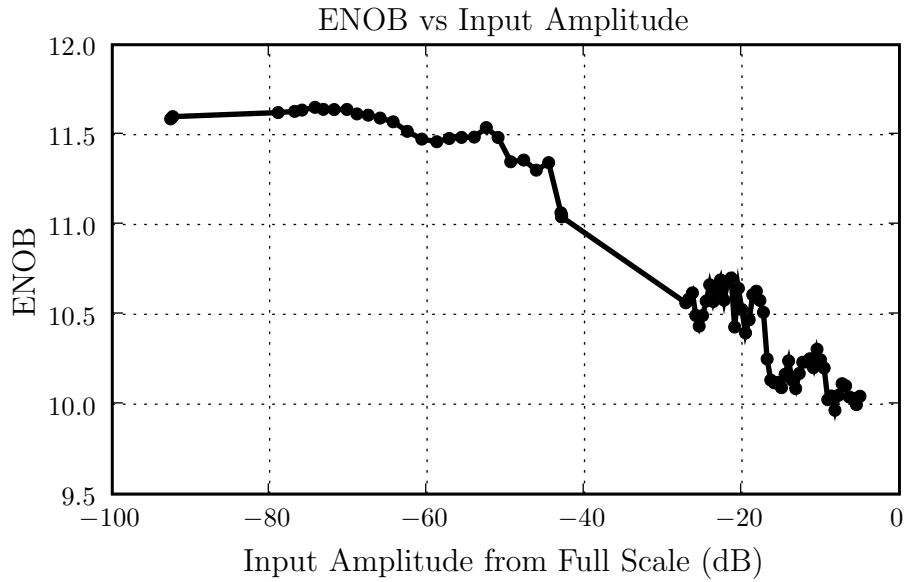


Figure 5-36: SNDR versus input amplitude

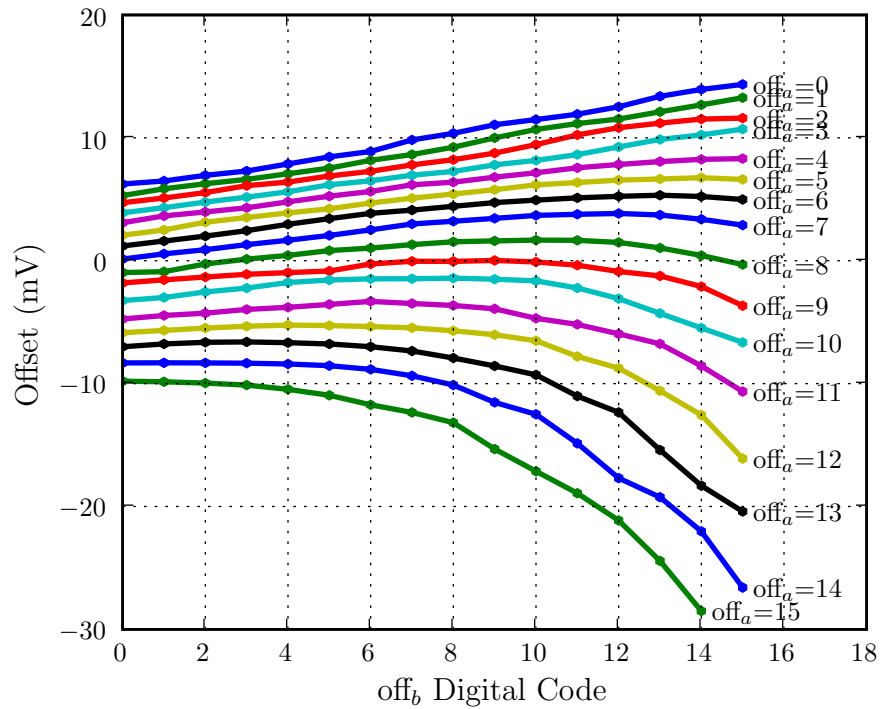


Figure 5-37: Measured 1st stage programmable ZCD offset range. See Figure 5-8 for definition of  $off_a$  and  $off_b$  nets.

process, and voltage, however, this does not matter to a COE feedback controller (see Chapter 4) as the feedback loop will adjust the ZCD offset until the overall ADC offset

is nulled.

### 5.8.3 I/O Noise Coupling

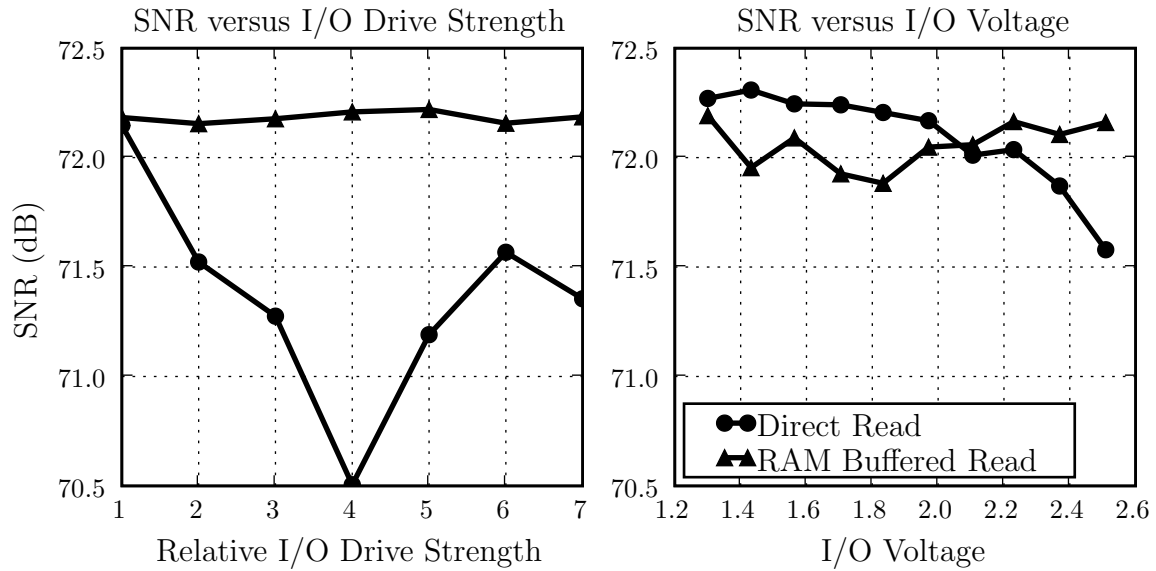


Figure 5-38: ADC noise sensitivity comparisons to I/O voltage and drive strength.

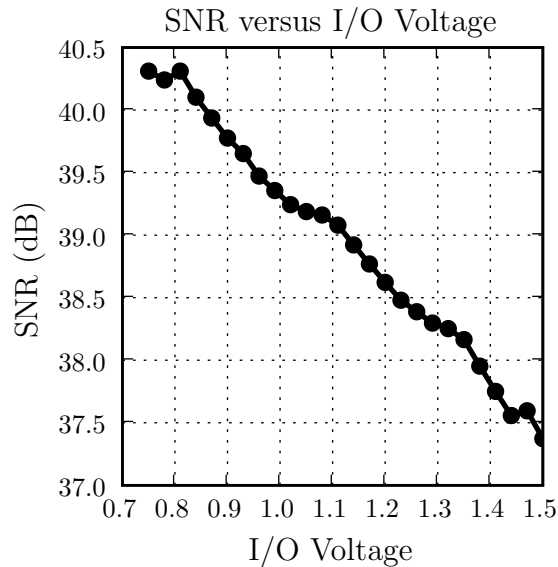


Figure 5-39: ADC noise sensitivity to I/O voltage for original single-ended ZCBC design described in Chapter 3.

With programmable I/O voltage and driver strength as well as the ability to turn off the I/Os completely and use the on-chip SRAM to buffer a block of data (see

Section 5.1), the sensitivity of the ADC noise to the I/O can be measured under all the various permutations. The results of these measurements are shown in Figure 5-38. Notice the SRAM buffered read is largely independent of the I/O voltage and drive strength. This is the expected behavior as the I/Os get disabled during the data block collection when using the on-chip SRAM as a data buffer. Regardless, there is on the order of 1dB of sensitivity on this chip to the I/O drive strength and less sensitivity to the I/O drive voltage. By comparison, the SNR sensitivity to the I/O voltage of the original single ended design described in Chapter 3 is shown in Figure 5-39. Observe that it has a much stronger correlation in that over the 500mV of I/O voltage change the SNR drops by 3dB where the fully differential design only moved 0.5dB over a 1.2V range.

#### 5.8.4 BDC Offset

Since the SNR of this design is approximately 12 bit accurate and the SNDR is 10 bit accurate, this design is clearly limited by distortion, and the dominant source of distortion is being caused by offset in the BDCs. This can be seen in Figure 5-40 where designs with two different BDC topologies as introduced in Figure 5-27 are compared. An initial version of this fully differential design was fabricated using BDC C and had such extreme BDC offsets that the design was completely unusable as shown in the various measured responses of the first column. The design was then changed to use BDC B and fabricated again. The bottom plots showing the first stage digital response show that BDC B has much less offset and noise. BDC B's offset, however, is not as low as the Eldo-based Monte Carlo simulations would predict (see Figure 5-28). The residue plot for BDC B in Figure 5-40 shows that the BDC offset is causing the residue output to go beyond the head room limits imposed by the cascode devices.



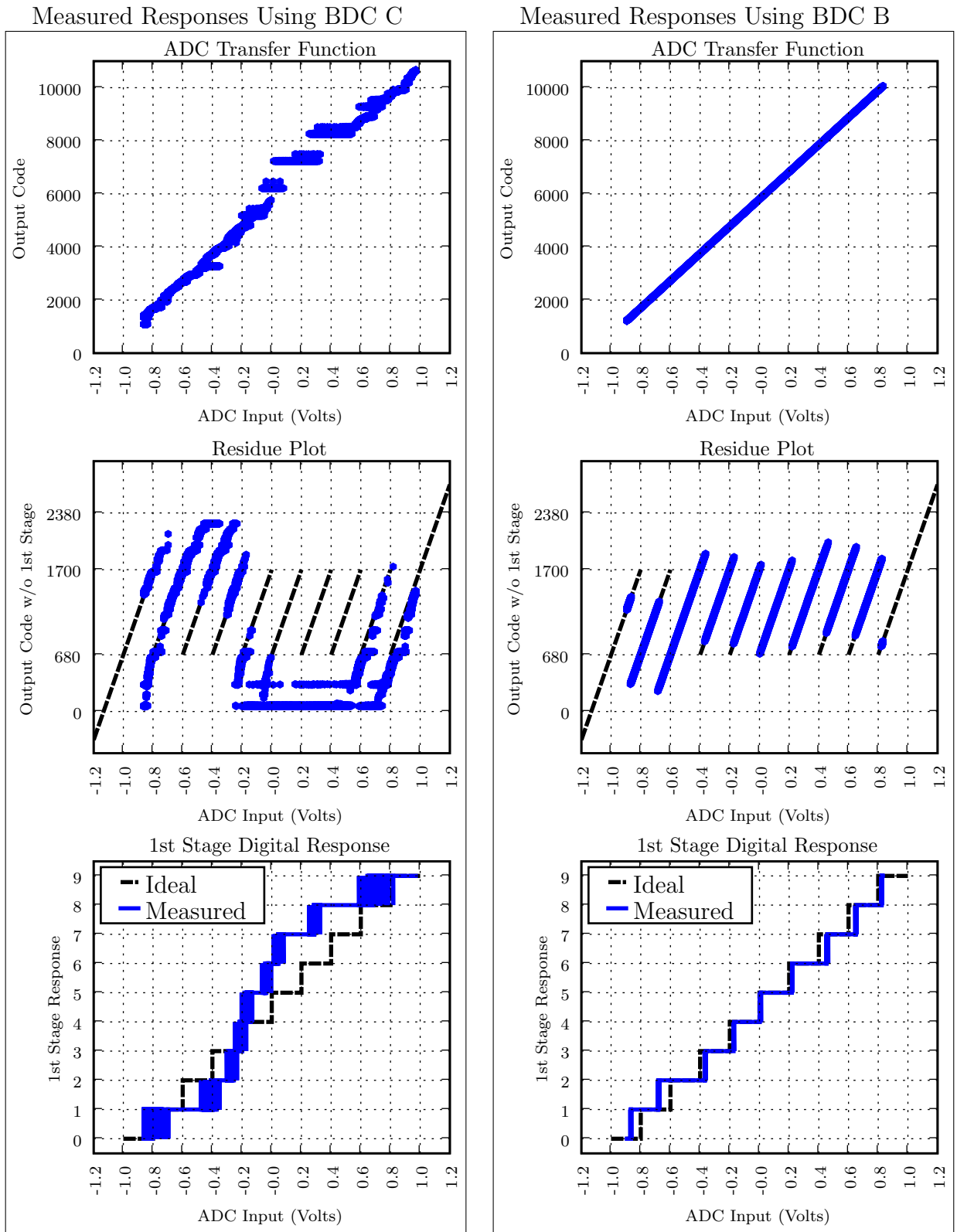


Figure 5-40: Measured performance using BDC C and BDC B (see Figure 5-27).

## 5.9 Conclusion

As shown in the performance summary of Table 5.1, this ADC represents a significant step forward in the performance of ZCBC pipelined ADCs. Furthermore, the fully differential implementation and offset compensation also represent a significant step forward in making ZCBC designs production worthy.

Table 5.1: ADC Performance Summary

Technology	90nm CMOS	
Area	0.225 mm <sup>2</sup>	
Input Voltage Range	2V (differential)	
Power Supply: V <sub>DD</sub>	1.2V	
Sampling Frequency	25 MS/s	50 MS/s
DNL	±0.5 LSB <sub>12</sub>	±0.5 LSB <sub>12</sub>
INL	±2.0 LSB <sub>12</sub>	±3.0 LSB <sub>12</sub>
Power Consumption	3.8 mW	4.5 mW
SNR	72 dB	72 dB
SFDR	73 dB	68 dB
SNDR	66 dB	62 dB
ENOB	10.6 bit	10 bit
Figure of Merit: $\frac{P}{2f_{in}2^{ENOB}}$	98 fJ/step	88 fJ/step

# Chapter 6

## Conclusion

While this thesis presents several algorithms and circuits for improving state-of-the-art performance of pipelined ADCs in scaled technology, there are still some outstanding issues that deserve further research.

### 6.1 ZCBC Future Work

The ZCBC architecture is still in its infancy and has several areas that require solutions to make a design production worthy. The following is a discussion of some of these areas along with some speculative ideas for potential solutions.

#### 6.1.1 Reference Voltages

As discussed in Section 5.3, integrating the reference voltages on-chip in a power efficient manor remains an open research topic. The constraints on the voltage references are that they must settle within the pre-charge phase and they must be able to hold a constant voltage to within an LSB of precision when any given stages switches off and the current load changes.

The dual ramping scheme used in the original CBSC [18] design can help with the changing current load problem because the current levels of the second ramp phase are lower than when a single ramping scheme is used. This is also the reason that the

dual ramping scheme can have better linearity performance due to the finite output impedance of the current source. The trade-off, however, for the dual ramping scheme is speed and complexity.

An alternative approach to using dual ramps is to use various current source linearization techniques [U.S. Patent 7253600]. One such method is to use a current source whose current level is proportional to the error in the virtual ground condition. A proof-of-concept schematic in Figure 6-1 shows such an implementation. The circuit is shown in the transfer phase where transistor  $M_2$  is biased to provide a small constant current for ramping and transistor  $M_1$  is biased to provide current that is proportional to the error in the virtual ground condition. The amplifier labeled  $U_1$  measures the error, amplifies it, and applies it to the gate of  $M_1$ . As the solid line of Figure 6-2 shows, this will initially cause exponential settling to occur on the virtual ground node while the current provided by  $M_1$  is dominant. When the error settles sufficiently that the current provided by  $M_2$  is dominant, then the dynamics will become a linear ramp. Figure 6-2 also shows the dynamics of the virtual node for both the single and dual ramping schemes. The slope of the single ramp scheme is largest of the three when at when the virtual ground condition is realized, meaning that it requires the highest amount of current. The slope of the dual ramp and proportional current scheme are both lower and thus offer improved linearity and eased requirements on the reference voltages.

There are several ideas that can perhaps improve the proof-of-concept schematic of Figure 6-1. One is to incorporate the amplifier  $U_1$  as a pre-amplifier within the zero-crossing detector ZCD. Another is to add a series capacitor on the  $v_C$  node to explicitly control the nominal drive strength of  $M_1$ .

Using a proportional current controller *may* allow of faster operation or simplified design over the dual ramping scheme while still allowing for reduced current levels when the virtual ground condition is realized. The proportional current scheme is somewhat like the combination of an opamp-base and a zero-crossing based system. An amplifier in feedback is used initially to make a quick adjustment of the virtual ground condition. Then a current source and a zero-crossing detector take over to

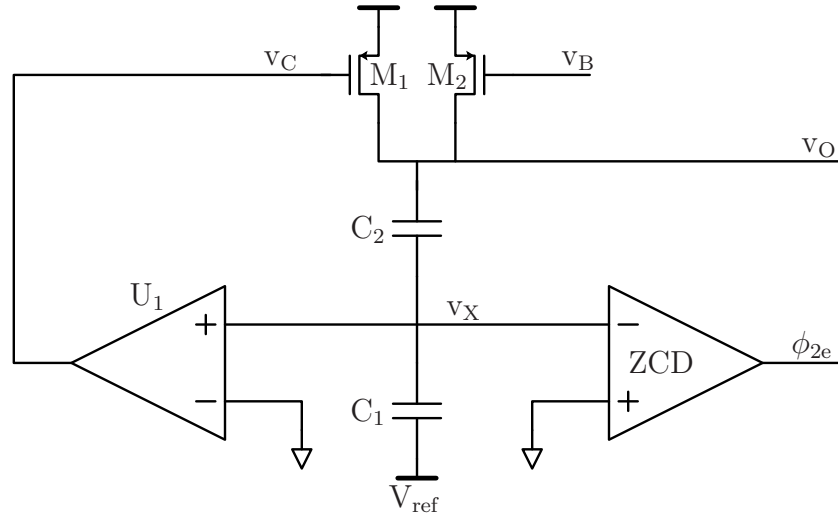


Figure 6-1: ZCBC implementation shown in the transfer phase utilizing proportional feedback control to the current source.

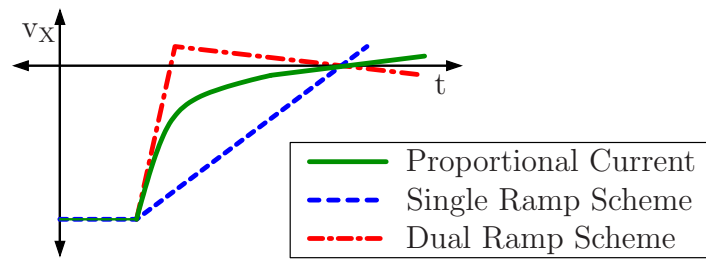


Figure 6-2: Virtual ground node dynamics for various ZCBC ramping schemes.

make the fine adjustment. Stability issues with the amplifier in feedback do need to be considered, but since the high-gain constraints on the amplifier in the proportional current scheme are greatly diminished over those of a traditional opamp-based system, stability should be much easier to obtain. Furthermore, because the amplifier is only used for coarse adjustment, it should also have much more relaxed noise and power constraints as well. Making this approach fully differential, however, does complicate the common-mode feedback implementation as both the positive and negative proportional currents need matched.

### 6.1.2 PVT Hardening

One reason that opamp-based design is so popular is that the sensitivity of the large open-loop gain of the opamp to process, voltage, and temperature (PVT) variation

can be transformed into a precision and largely PVT insensitive closed-loop gain.

ZCBC circuits have received little analysis in terms of their sensitivities to PVT variation. While there are some commonalities between opamp based and zero-crossing based circuits in terms of sensitivities to PVT variation, there are some clear differences that still need analyzed and confirmed in silicon.

In an opamp based circuit, the dynamics of the system must be given adequate time to settle over all PVT corners. Thus, the dynamics must be accounted for only in the worst case conditions. ZCBC circuits, however, are dynamic circuits by their very nature, and so the effects of PVT variation on the dynamics must be analyzed differently. For example, consider the case of the input referred offset of a ZCBC circuit. Since the delay of the zero-crossing detector is temperature sensitive, so too is the offset of the ZCBC. Thus offset compensation is critical to making ZCBC circuits robust to temperature variation.

No analysis is presented here to compare the sensitivities of ZCBC and opamp based systems to PVT variation. However, it is clear that ZCBC circuits do need more attention in this area. Areas that need analyzed regarding ZCBC circuits and PVT variation include the following:

**Ramp Rate Selection** : Setting the voltage ramp to use all of the available clock cycle will maximize performance. Generating a band-gap referenced current source is one way to ensure that the ramp rate stays constant over PVT variation, but in practice it may be desirable to have a feedback circuit pick the optimal ramp rate based on conditions even beyond PVT such as clock rate and reference range. Perhaps a replica stage that slaves the ramp rate to optimal setting is appropriate to automatically adjust the circuit performance based on the current conditions.

**Zero-Crossing Detector Bias Selection** : As previously discussed, for both the DZCD presented in Chapter 3 and the differential ZCD presented in Chapter 5, their dynamics are PVT dependant. Offset compensation can remove the offset sensitivity, but constant  $g_m$  biasing should be used as well if constant linearity

is desired. However, the trade offs between constant  $g_m$ , constant current, and constant overdrive voltage [51] biasing need analyzed and understood.

**Clock Phase Generation :** A pre-charge phase must be generated in ZCBC circuits prior to starting the voltage ramp. This should be made as short as possible to maximize the time for voltage ramping to maximize the linearity. The question remains on how to generate that clock phase. A method that ensures adequate time is given to the pre-charge clock phase and that is robust over PVT variation needs developed.

Generating this clock phase using a DLL is one way to make the duration of the pre-charge phase completely PVT insensitive. Under this approach the designer needs to dial in a duration that provides adequate pre-charge time under the worst case PVT corner. Picking this worst case condition poses a complicated and uncertain trade off between performance and margin. For example, since the linearity of the ADC is a function of the ramp rate, any extra time devoted to the pre-charge phase requires a proportional increase in the voltage ramp rate and thus a decrease in linearity.

Another approach is to use a replica circuit that tracks the PVT variation to ensure the pre-charge phase is always minimal for the given conditions. This would then give variable time to the voltage ramp. If the voltage ramp were also generated using a replica circuit that minimized the voltage ramp based on the time available, then this would always ensure maximal linearity performance over PVT variation. It does not guarantee what maximal performance is, but it removes the guess work in picking the desired operating point and trading performance for margin.

### 6.1.3 Common Mode Feedback

As discussed in Section 5.2.1 a common mode feedback (CMFB) circuit was not implemented in the fully differential ZCBC design. The common mode of that design was adjusted during a manual startup calibration routine. A production worthy design

may need to incorporate automatic CFMB. Given the circuit techniques developed in the fully differential design regarding common mode performance, however, it should be a straight forward to implement a power efficient common mode feedback controller.

One such approach would be to put an offset compensated clocked comparator off the virtual ground node of each stage. This comparator fires at the end of the sampling phase to measure whether the common mode is high or low. Because the virtual ground node floats during the sampling phase, it provides a measure of the signal common mode. This does introduce some timing complexities, so perhaps using additional capacitors rather than trying to reuse the sampling capacitors may be a better solution.

## 6.2 Conclusions

It has been speculated that a single technology node will not be able to optimally serve both digital and analog circuit design as we enter the nano-scale era [39]. As the trend data of Figure 1-1 shows, the issues of implementing high resolution circuits such as data converters in low-voltage, deeply scaled technologies in the traditional manner validate this speculation.

While the issues are severe, the work of thesis is to present new methods and architectures for switched capacitor circuit design that align with the strengths of technology scaling. On the digital front, Decision Boundary Gap Estimation and Chopper Offset Estimation were introduced as simple and purely digital methods of recovering linearity lost due to effects such as limited finite gain or output impedance and nulling offset and flicker noise.

On the architecture front, zero-crossing based circuits were introduced as a generalization to comparator-based switched capacitor circuits and two different designs were implemented that demonstrated state-of-the-art performance in their respective classes. A comparison of power efficiency of the initial single-ended design to other published ADCs in its class is shown in Figure 6-3. As stated, this design was still quite competitive despite its noise floor being 8 times higher than calculations and



simulations showed. A comparison for the power efficiency of the fully-differential design is shown in Figure 6-4. This design demonstrates state-of-the-art performance in its class of 12 bit converters.

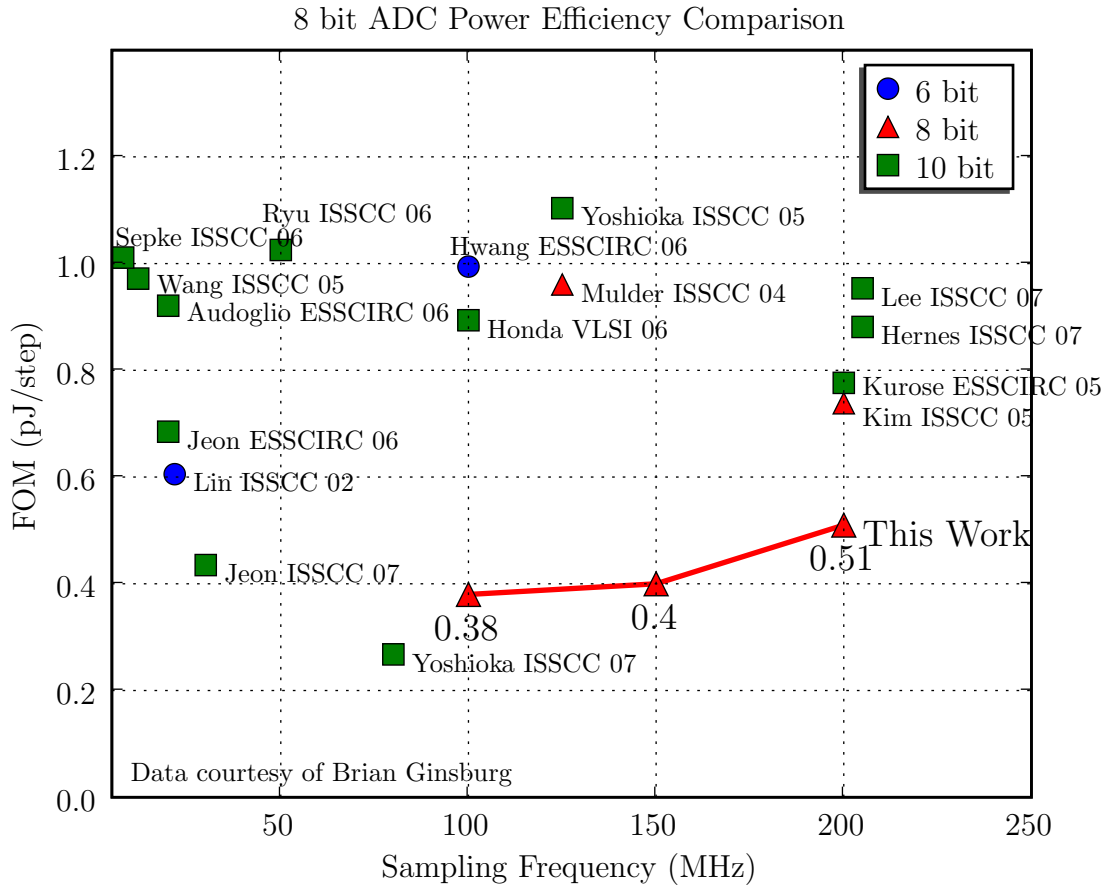


Figure 6-3: Power Efficiency Comparison of Single-Ended Design

All the techniques presented in this thesis can be used together or in isolation as a particular application demands. Furthermore, while they are applied specifically to pipelined ADCs, many have natural extensions to applications beyond pipelined ADCs. Thus, while the situation for analog circuit design in scaled technologies does look bleak, algorithms and architectural changes whose strengths align better with the scaling trends, such as those presented in this thesis, can be used to extend the optimality of a single process node being able to serve the needs both of digital and analog circuits simultaneously.

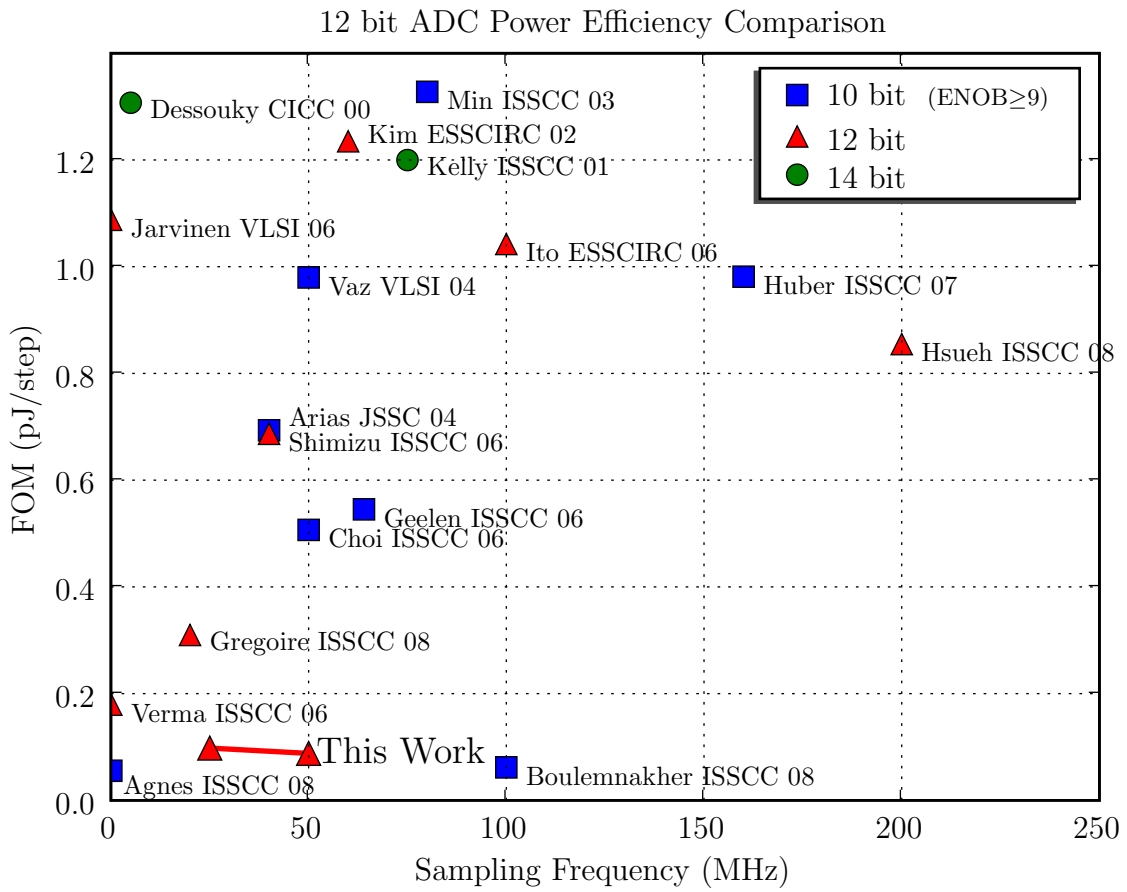


Figure 6-4: Power Efficiency Comparison of Fully-Differential Design

# Bibliography

- [1] Andrew M. Abo and Paul R. Gray. A 1.5v, 10-bit 14.3-ms/s CMOS pipeline analog-to-digital converter. *IEEE Journal of Solid-State Circuits*, 34(5):599–606, May 1999.
- [2] Anne-Johan Annema, Bram Nauta, Ronald van Langevelde, and Hans Tuinhout. Analog circuits in ultra-deep-submicron CMOS. *IEEE Journal of Solid-State Circuits*, 40(1):132–143, January 2005.
- [3] J.H. Atherton and H.T. Simmonds. An offset reduction technique for use with CMOS intergrated comparators and amplifiers. *IEEE Journal of Solid-State Circuits*, 27:1168–1175, August 1992.
- [4] Eulalia Balestrieri, Pasquale Daponte, and Sergio Rapuano. A state of the art on ADC error compensation methods. *IEEE Transactions on Instrumentation and Measurement*, pages 1388–1394, August 2005.
- [5] E.B. Blecker, T.M. McDonald, O.E. Erdogan, P.J. Hurst, and S.H. Lewis. Digital background calibration of an algorithmic analog-to-digital converter using a simplified queue. *IEEE Journal of Solid-State Circuits*, pages 1489–1497, June 2003.
- [6] Lane Brooks and Hae-Seung Lee. A zero-crossing based 8 bit, 200ms/s pipelined ADC. *IEEE Journal of Solid-State Circuits*, 43, December 2007.
- [7] Yun Chiu, Cheongyuen W. Tsang, Borivoje Nikolic, and Paul R. Gray. Least mean square adaptive digital background calibration of pipelined analog-to-

- digital converter. *IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications*, pages 38–46, January 2004.
- [8] Albert Chow and Hae-Seung Lee. Transient noise analysis for comparator-based switched-capacitor circuits. *Proc. ISCAS*, May 2007.
- [9] Xin Dai, Degang Chen, and Randall Geiger. A cost-effective histogram test-based algorithm for digital calibration of high-precision pipelined adcs. *IEEE ISCAS*, 5:4831–4834, May 2005.
- [10] A.C. Dent and C.F.N. Cowan. Linearization of analog-to-digital converters. *IEEE Transactions on Circuits and Systems*, 37:729–737, June 1990.
- [11] M. Dessouky and A. Kaiser. Input switch configuration suitable for rail-to-rail operation of switched opamp circuits. *Electronics Letters*, 35(1):8–9, January 1999.
- [12] Vijay Divi and Greg Wornell. Signal recovery in time-interleaved analog-to-digital converters. *IEEE ICASSP*, 2004.
- [13] Joey Doernberg, Hae-Seung Lee, and David Hodges. Full-speed testing of A/D converters. *IEEE Journal of Solid-State Circuits*, 19:820–827, December 1984.
- [14] Udaykiran Eduri and Franco Maloberti. Online calibration of a nyquist-rate analog-to-digital converter using output code-density histograms. *IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications*, pages 15–24, January 2004.
- [15] J. Elbornsson and J.-E. Eklund. Histogram based correction of matching errors in subranged ADC. *ESSCIRC 2001*, pages 555–558, September 2001.
- [16] Christian C. Enz and Gabor C. Temes. Circuit techniques for reducing the effects of op-amp imperfections: Autozeroing, correlated double sampling, and chopper stabilization. *Proceedings of the IEEE*, 84(11):1584–1613, November 1996.

- [17] Rudy G.H. Eschhausier and Johan H. Huijsing. *Frequency Compensation Techniques for Low-Power Operational Amplifiers*. Kluwar Academic Publishers, 1995.
- [18] John K. Fiorenza, Todd Sepke, Peter Holloway, Charles G. Sodini, and Hae-Seung Lee. Comparator-based switched-capacitor circuits for scaled CMOS technologies. *IEEE Journal of Solid-State Circuits*, 41(12):2658–2668, December 2006.
- [19] U. Gatti, G. Gazzoli, F. Maloberti, and S. Mazzoleni. A calibration technique for high-speed high-resolution a/d converters. *Advanced A/D and D/A Conversion Techniques and their Applications*, pages 168–171, July 1999.
- [20] Kush Gulati and Hae Seung Lee. A low-power reconfigurable analog-to-digital converter. *IEEE Journal of Solid-State Circuits*, 36(12):1900–1911, December 2001.
- [21] D.G. Haigh and B. Singh. A switching scheme for switched-capacitor filters, which reduces effect of parasitic capacitances associated with control terminals. *Proc. IEEE Int. Symp. on Circuits and Systems*, 2:586–589, June 1983.
- [22] Bjornar Hernes, Johnny Bjornsen, Terje N. Anderson, Anders Vinje, Havard Korsvoll, Frode Telsto, Atle Briskemyr, Christian Holdo, and Oystein Moldsvor. A 92.5mw 205 ms/s 10b pipeline IF ADC implemented in 1.2v/3.3v 0.13um CMOS. *ISSCC Digest of Tech. Papers*, pages 462–463, 2007.
- [23] K.-C. Hsieh, P. R. Gray, D. Senderowicz, and D. G. Messerschmitt. A low-noise chopper-stabilized differential switched-capacitor filtering technique. *IEEE Journal of Solid-State Circuits*, 16:708–715, December 1981.
- [24] Chun-Cheng Huang and Jieh-Tsorng Wu. A background comparator calibration technique for flash analog-to-digit l converters. *IEEE Transactions on Circuits and Systems—Part II: Analog and Digital Signal Processing*, 52(9):1732–1740, September 2005.

- [25] IEEE. IEEE standard for terminology and test methods for analog-to-digital converters. *IEEE STD 1241-2000*, December 2000.
- [26] Echere Iroaga and Boris Murmann. A 12-bit 75-ms/s pipelined ADC using incomplete settling. *IEEE Journal of Solid-State Circuits*, 42(4):748–756, April 2007.
- [27] Shafiq M. Jamal, Daihong Fu, Nick Chang, Paul Hurst, and Stephen H. Lewis. A 10-b 120-msamples/s time-interleaved analog-to-digital converter with digital background calibration. *IEEE Journal of Solid-State Circuits*, 37(12):1618–1627, December 2002.
- [28] Le Jin, Degang Chen, and Rnadal Geiger. A digital self-calibration algorithm for ADCs based on histogram test using low-linearity input signals. *IEEE International Symposium on Circuits and Systems*, pages 1378–1381, May 2005.
- [29] Andrew Karanicolas, Hae-Seung Lee, and Kantilal Bacrania. A 15-b 1-msample/s digitally self-calibrated pipeline ADC. *IEEE Journal of Solid-State Circuits*, pages 1207–1215, December 1993.
- [30] Hae-Seung Lee. Limits of power consumption in analog circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, June 2007.
- [31] Stephen Lewis, Scott Fetterman, George Gross, R. Ramachandran, and T.R. Viswanathan. A 10-b 20-msample/s analog-to-digital converter. *IEEE Journal of Solid-State Circuits*, pages 351–358, March 1992.
- [32] Jipeng Li and Un-Ku Moon. Background calibration techniques for multistage pipelined adcs with digital redundancy. *IEEE Transactions on Circuits and Systems—Part II: Analog and Digital Signal Processing*, 50:531–538, September 2003.
- [33] P.W. Li, M.J. Chin, P.R. Gray, and R. Castello. A ratio-independent algorithmic analog-to-digital conversion technique. *IEEE Journal of Solid-State Circuits*, pages 828–836, December 1984.

- [34] U.-K. Moon and B.-S. Song. Background digital calibration techniques for pipelined ADC's. *IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications*, pages 102–109, February 1997.
- [35] Boris Murmann and Bernhard E. Boser. A 12-bit 75-ms/s pipelined ADC using open-loop residue amplification. *IEEE Journal of Solid-State Circuits*, 38(12):2040–2050, December 2003.
- [36] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes*. Tata McGraw-Hill, 4 edition, 2002.
- [37] R. Poujois and J. Borel. A low drift fully integrated MOSFET operational amplifier. *IEEE Journal of Solid-State Circuits*, 13:499–503, August 1978.
- [38] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, 2007.
- [39] Jan M. Rabaey, Fernando De Bernardinis, Ali M. Niknejad, Borivoje Nikolic, and Albert Sangiovanni-Vincentelli. Embedding mixed-signal design in systems-on-chip. *Proceedings of the IEEE*, 94(6):1070–1088, June 2006.
- [40] Ronald Schafer and Alan Oppenheim. *Digital Signal Processing*. Prentice-Hall, Inc., 2 edition, 1999.
- [41] Todd Sepke. *Comparator Design and Analysis for Comparator-Based Switched-Capacitor Circuits*. PhD thesis, Massachusetts Institute of Technology, 2006.
- [42] Tzi-Hsiung Shu, Kantilal Bacrania, and Chong-In Chi. Statistical correction of a/d converter errors. *IEEE BCTM*, pages 189–191, September 1996.
- [43] Eric Siragusa and Ian Galton. Gain error correction technique for pipelined analog-to-digital converters. *Electron. Lett.*, 36:617–618, 2000.

- [44] Eric Siragusa and Ian Galton. A digitally enhanced 1.8-v 15-bit 40-msamples/s CMOS pipelined ADC. *IEEE Journal of Solid-State Circuits*, 39(12):2126–2138, December 2004.
- [45] B.-S. Song, M.F. Tompsett, and K.R. Lakshmi Kumar. A 12-bit 1-msample/s capacitor error-averaging pipelined A/D converter. *IEEE Journal of Solid-State Circuits*, pages 1316–1323, December 1988.
- [46] S. Sonkusale, J. van der Spiegel, and K. Nagaraj. True background calibration technique for pipelined adc. *Electron. Lett.*, 26:786–788, April 2000.
- [47] Laszlo Toth and Yannis P. Tsividis. Generalization of the principle of chopper stabilization. *IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications*, 50(8), August 2003.
- [48] Hendrick van der Ploeg, Gian Hoogzaad, Henk A.H. Termeer, Maarten Vertregt, and Raf L.J. Roovers. A 2.5-v 12-b 54-msample/s 0.25- $\mu\text{m}$  CMOS ADC in 1-mm<sup>2</sup> with mixed-signal chopping and calibration. *IEEE Journal of Solid-State Circuits*, 36:1859–1867, December 2001.
- [49] Stefano Vitali, Giampaolo Cimatti, and Riccardo Rovatti. Algorithm ADC offset compensation by non-white data chopping. *ISCAS*, pages 1425–1428, May 2007.
- [50] J.-W. Wu, C.-C. Cheng, K.-L. Chiu, J.-C. Guo, W.-Y. Lien, C.-S. Chang, G.-W. Huang, and T. Wang. Pocket implantation effect on drain current flicker noise in analog nMOSFET devices. *IEEE Transactions on Electron Devices*, 51(8):1262–1266, August 2004.
- [51] Masato Yoshioka, Masahiro Kudo, Toshihiko Mori, and Sanroku Tsukamoto. A 0.8v 10b 80ms/s 6.5mw pipelined adc with regulated overdriver voltage biasing. *ISSCC Dig. of Tech. Papers*, pages 452–453, February 2007.