A 12-b 50Msample/s Pipeline Analog to Digital Converter

by

Nathan Carter

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Electrical Engineering

April 24, 2000

Approved:

Prof. John McNeill
ECE Department
Thesis Advisor

Prof. Yusef Leblebici
ECE Department
Thesis Committee

Prof. Len Polizzotto
ECE Department
Thesis Committee

Prof. John Orr
ECE Department Head

Acknowledgments

I would like to thank the following people for making this thesis possible,

Prof. John McNeill	For funding, technical advice, professional advice, and giving me the freedom to make mistakes.
Prof. Yusuf Leblebici	For use of his lab, and his technical advice.
Prof. Len Polizzotto	For his work on my thesis committee.
1999/2000 Staff of the Analog Lab, VLSI Lab, and ECE Dept. -Frank Gurkaynak -Renato Baumgartner -Ilhan Hatirnaz -Wesley Blackstone -Bruce Lavoie	For many small things to numerous to say but in total require an acknowledgment in this thesis.
Bill and Sadie Goesch	For providing living arrangements.
David, Marcia, Chris Josh and Tricia	For flying from California all those times and showing your support.

Abstract

This thesis focuses on the performance of pipeline converters and their integration on mixed signal processes. With this in mind, a 12-b 50 MHz pipeline ADC has been realized in a 0.6-μm digital CMOS process. The architecture is based on a 1.5-b per stage structure utilizing digital correction for the first six stages. A differential switched capacitor circuit consisting of a cascode $g_m - C$ op-amp with 250MHz f_t is used for sampling and amplification in each stage. Comparators with an internal offset voltage are used to implement the decision levels required for the 1.5-b per stage structure. Correction of the pipeline is accomplished by measuring the offset and gain of each of the first six stages using subsequent stages. The measured values are used to calculate digital values that compensate for the inaccuracies of the analog pipeline. Corrected digital values for each stage are stored in the pipeline and used to create corrected output codes. Errors caused by measuring the first six stages using uncalibrated stages are minimized by using extra switching circuitry during calibration.

Contents

1 Introduction **1**

2 Architecture **3**

2.1 Flash 3

2.2 Folding 5

2.3 Pipeline and Multi Pass 7

2.4 Summary 9

3 Pipeline Architecture **10**

3.1 Bit Resolution Per Stage 10

3.1.1 Device Matching 11

3.1.2 Bandwidth 13

3.1.3 Latency 13

3.2 Pipeline Errors 15

3.2.1 Offset Errors 15

3.2.2 Gain Errors 16

3.2.3 Comparator Errors 18

3.3 1.5 bit Per Stage Architecture 19

3.3.1 Overlap and Error Correction 20

3.3.2 Overlap and Bit Resolution 22

3.4 Calibration 24

3.4.1 Mathematical Characterization 24

3.4.2 Offset 27

3.5 Behavioral Simulations 29

3.5.1 Linearity Characterization 29

3.5.2 Simulation Procedure 30

3.5.3 Simulation Coverage 33

3.5.4 Residue Plots 34

3.5.5 Simulation Results 35

3.6 Summary of Requirements 38

4 Analog Architectural Requirements **39**

4.1 Bandwidth 39

4.2 Stability 41

4.3 Open Loop Gain 45

4.4 Slew Rate 49

4.5 Noise 51

4.6 Summary of Requirements 52

5 Op-Amp Architectures 53

5.1 Introduction 53

5.1.1 Current Mirror Amplifier 54

5.1.2 Small Signal Operation 56

5.1.3 AC Response 58

5.1.4 Cascoded Current Mirror Amplifier 61

5.2 Folded Cascode 63

5.2.1 Small Signal Operation 63

5.2.2 AC Response 64

5.3 Cascode with Active Current Sources 66

5.4 Summary of Architectures 68

5.5 Common Mode Feedback 69

5.5.1 Continuous Time Common Mode Feedback 69

5.5.2 Switched Capacitor CMFB 71

5.6 Test Chip Amplifier 73

6 Comparator Architecture 76

6.1 Flashback 76

6.2 Reference Voltage 77

6.3 Test Chip Comparator 78

7 Digital Architectural Requirements 80

7.1 Control Signal Generation in the Analog Processing Block 80

7.2 Digital Pipeline 87

7.2.1 Coefficient and Pipeline Registers 88

7.2.2 Multiplexer 88

7.2.3 Ripple Carry Adders 90

7.3 Clock Generation 92

8 Top Level Design 95

8.1 Analog Processing Block 95

8.2 Digital Block 100

8.3 Top Level Layout 103

9 Test Chip Results 106

9.1 Clock Generation 106

9.2 Digital Pipeline 107

9.3 Analog Block 109

9.4 Comparator Operation 113

10 Conclusions 114

A Simulation Source Code 116

B Maple gain error derivation 125

C Test Chip and Board **129**

D Schematics **134**

List of Tables

4.1 Settling time for one half LSB resolution. 40

5.1 Simulated performance of test chip amplifier. 73

7.1 Switching of fully differential amplifier. 81

10.1 Summary of test chip results. 115

List of Figures

2.1 Typical flash converter. 4

2.2 Converter implemented with a folding block. 6

2.3 Folding circuit with a folding rate of four. 6

2.4 Pipeline Analog to Digital Converter Block Diagram 7

2.5 Origin of the Error Voltage 8

2.6 Single Pipeline Stage with Residue Plots 9

3.1 Closed loop gain effect on 3dB frequency 14

3.2 Residue exceeding full scale voltage in a 2 bit per stage converter 15

3.3 Missing Code errors caused by the output residue exceeding the full scale voltage in the first stage of a 2-b per stage converter. 17

3.4 Comparator threshold shift causing out of range value. 18

3.5 1.5 bit per stage analog block 19

3.6 Residue plot for 1.5 bit per stage architecture 21

3.7 Residue plot with possible errors. 22

3.8 1.5-bit per stage residue plot with comparator error limits. 23

3.9 2.5-bit per stage residue plot with comparator error limits 23

3.10 Origin of charge transfer equation. 24

3.11 Example of Differential and Integral Non-Linearity 29

3.12 C implementation of an uncalibrated pipeline converter 31

3.13 C code for the measurement of α 32

3.14 Formulation of output codes using decision bits and calibration results. 33

3.15 Uncalibrated converter simulation results 36

3.16 Calibrated converter simulation results 37

4.1 Phase Margin 42

4.2 Second order response. 43

4.3 Ideal DC sweep, actual, and error voltage between them. 45

4.4 Amplifier with closed loop feedback. 46

4.5 Plot of the closed loop gain error ΔA with changing open loop gain A_0 and output voltage. 48

4.6 Bias currents and slew rate. 49

4.7 Equivalent circuit for determining bias currents to achieve a required slew rate. . . 50

5.1 Current mirror amplifier. 55

5.2 MOS transistor acting as a current source. 56

5.3 Current mirror amplifier with parasitic capacitances C_p. 59

5.4 Simulation showing the effect of parasitic capacitances in a current mirror amplifier. 60

5.5 Cascoded current mirror. 61

5.6 Cascoded current mirror amplifier 62

5.7 Folded cascoded amplifier. 63

5.8 Small signal model of a folded cascode. 64

5.9 AC simulation of a folded cascode amplifier. 65

5.10 Active current source. 67

5.11 AC simulation of an actively cascoded current mirror amplifier. 67

5.12 Summary of Architectural Performance 68

5.13 Continuous Time Common Mode Feedback Circuit. 70

5.14 Switched Capacitor Common Mode Feedback. 72

5.15 Schematic of test chip amplifier. 74

5.16 Test chip Amplifier Layout 75

6.1 Latched Comparator 77

6.2 Flashback 78

6.3 Test chip comparator layout. 79

7.1 Fully differential switched capacitor amplifier. 82

7.2 Timing Diagram for digital control of analog processing block. 83

7.3 Switching Decoder. 84

7.4 Switch Decoder Layout. 85

7.5 Post layout simulation of digital controller for analog processing block. 86

7.6 Pipeline stage block diagram 87

7.7 D Flip Flop 88

7.8 D flip-flop layout. 89

7.9 3-to-1 Multiplexer 89

7.10 Layout of a 3-to-1 Multiplexer 90

7.11 Ripple Carry Adder 91

7.12 Simulation of a Ripple Carry Adder in a 0.6μm technology. 91

7.13 Pipeline Clocks 93

7.14 Block diagram of clock generator. 94

7.15 Post layout simulation of clock generation. 94

8.1 Analog block schematic 96

8.2 Analog block layout. 98

8.3 Single stage residue plot. 99

8.4 Digital block schematic 101

8.5 Digital block layout. 102

8.6 Top level schematic 104

8.7 Top level layout. 105

9.1 Test chip clock generator running at 30MHz. 107

9.2 Logic analyzer screen shot of worst case 14 bit addition in 10ns. 108

9.3 Test block output. 109

9.4 Analog block simulation with corrected patch. 110

9.5 Time averaged differential output of test block. 111

9.6 Comparator Operation 113

D.1 Calibration decoder schematic. 135

D.2 Calibration decoder layout. 136

D.3 Register loading decoder schematic. 137

D.4 Register loading decoder layout. 138

D.5 Clock buffer schematic. 139

D.6 Clock buffer layout. 140

D.7 Register loader schematic. 141

D.8 Register loader layout. 142

Chapter 1

Introduction

Variations on analog to digital converters are numerous, each tailored for specific performance parameters. More modern converters have implemented some form of parallel processing of the analog signal to increase bit resolution while still maintaining the same speed. Converters of this type include folding, multi-step, and pipeline [19][20][21].

Converter architectures are still a rapidly developing area. Normally analog to digital converters are not strictly limited to pure flash, folding, or pipeline. Implementations with pipelined folding stages or smaller pipelined stages with a large flash section at the end are not uncommon [17]. Most of these architectures are implemented as discrete converters for board level integration. With a movement toward system on a chip high performance converters are frequently implemented on the same chip with microcontrollers and other digital systems. This introduces new noise and process problems which are not as dominant in discrete converter implementations. Additionally, processes tailored for digital logic are not the best processes to make the linear circuits required for analog to digital conversion but are becoming more frequently the place where converters physically take shape. With this movement toward chip level integration it is desirable to have a converter architecture that is tolerant of matching and process errors as well as noise introduced by adjacent

devices. With this in mind, this project focuses on the pipeline architecture because of its high tolerance of process variations, low power consumption, and small area making it an ideal candidate for system level integration.

The thesis is organized into ten chapters. Chapter 2 will introduce a few of the more common high resolution analog to digital converter architectures. Some of the concepts that apply to converters in general such as resolution and the error voltage will also be introduced. Chapter 3 will focus on pipeline converter behavior which will emphasize some of the problems encountered in this design. Digital correction is proposed as a solution to some of the conversion problems. The remainder of the chapter tests the applicability of digital correction through behavioral simulations. Chapter 4 is the beginning of the design process which introduces a few of the requirements of the analog processing blocks. Derivations of the requirements are included where necessary. Chapter 5 discusses several amplifier architectures that have possible use in the analog processing block. Emphasis is placed on the most important design criteria of each type of amplifier as well as performance limitations. The chapter is concluded by a statement of the amplifier used in this project and a summary of its performance. Chapter 6 documents the comparator used in the test chip. Likewise, chapter 7 is a strateforward documentation of the digital components used in the test chip. A discussion of the more critical digital control block which produces the switching signals for the analog processing block and the clock generator is also included. Chapter 8 combines the components in chapters 5, 6, and 7 to create both the analog and digital blocks. The test chip results are included in chapter 9. Chapter 10 is a conclusion with an outline of future design considerations.

Chapter 2

Architecture

2.1 Flash

The most well know of all ADC architectures is the flash, which consists of a resistor divider network that generates a “ladder” of reference voltages, and comparators that compare the input to the reference voltages. Decoding logic is used to take the comparator decisions and generate a digital output code. A two bit flash converter is shown in Figure 2.1.

A flash converter needs 2^N comparators and $2^N + 1$ resistors for an N bit output code. This presents a power and silicon area problem with ADCs requirening large output codes. For this reason flash converters are generally used in applications requiring high speed and low resolution (<8 bits). To get higher resolution several smaller flash converters can be combined to process the analog input signal and form a larger output word. The methods used to combine several smaller converters can be quite complex. A few of those methods are now introduced.

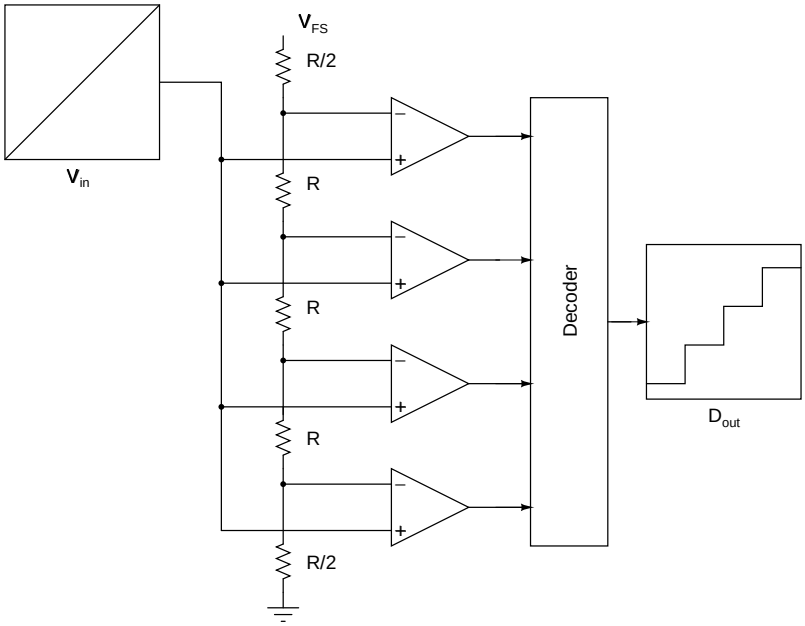


Figure 2.1: Typical flash converter.

2.2 Folding

To reduce the number of comparators as well as increasing the total resolution some form of folding is commonly implemented [22]. A folding converter has an extra analog preprocessing block that folds the input back into range so that a second flash converter can make a more precise estimate on the input signal. The folding converter in Figure 2.2 is shown with two single bit converters for simplicity, a coarse converter and a fine converter. With two single bit converters only two comparators are required versus the four required in the previous two bit flash converter. Generalizing this relationship, if the fine and the coarse converter resolve the same number of bits, the number of comparators required is $2^{\frac{N}{2}}$. The advantage gained by using folding circuitry versus some form of pipelining is that additional sample and holds are not necessary. The signal is concurrently processed by the coarse flash and the folding circuitry and fine flash converter.

The number of folds that the folding circuitry performs is referred to as the folding rate, which is one for the example in Figure 2.2. A limitation on performance is created by the folding circuitry because of the frequency multiplication that occurs. From the graphs in Figure 2.2 it is evident that a frequency increase equal to the folding rate will occur as the signal passes through the folding circuitry. In this aspect folding converters are not as fast as pure Flash converters because of the speed limited folding circuitry [15].

A possible implementation of a folding circuit with a folding rate of four is shown in Figure 2.3 [12]. Folding type converters are generally implemented in a bipolar process because of the more ideal folding behavior of bipolar transistors versus MOS transistors.

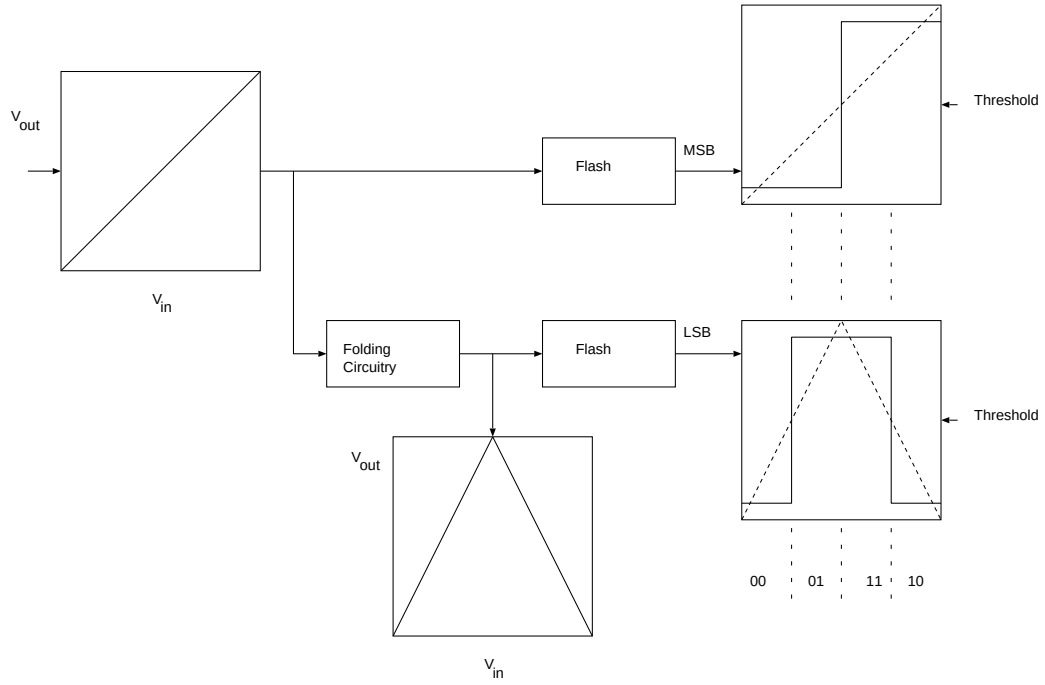


Figure 2.2: Converter implemented with a folding block.

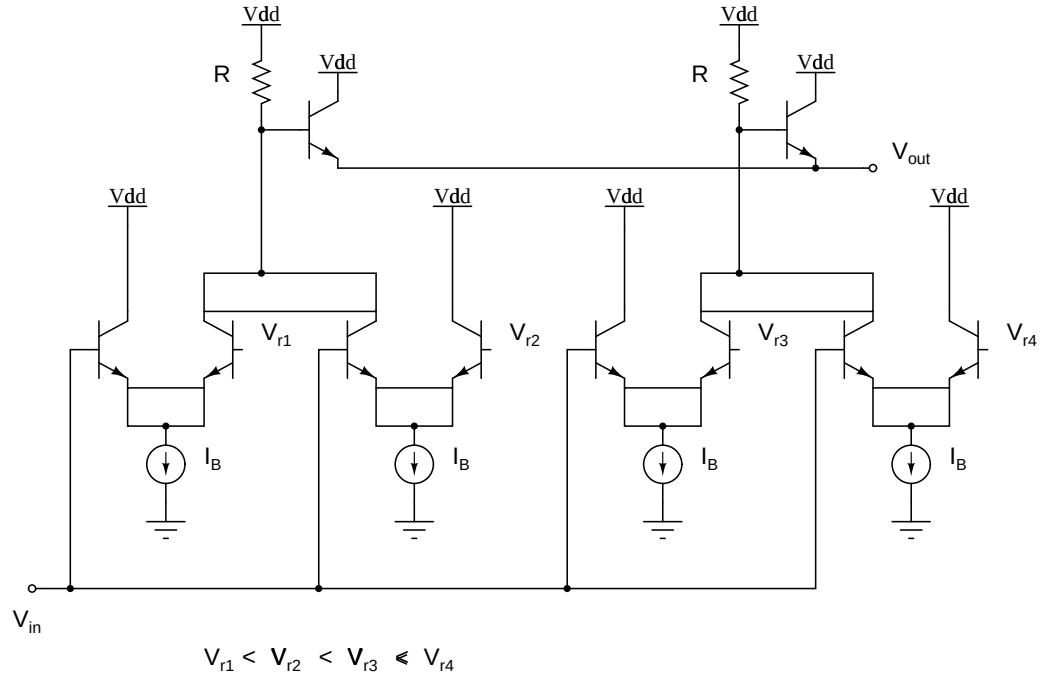


Figure 2.3: Folding circuit with a folding rate of four.

2.3 Pipeline and Multi Pass

Just as with folding converters, a pipeline ADC combines several smaller flash converters to create a larger converter. However, the smaller flash converters are combined by decimating time, that is each smaller converter has a specific time period in which it can convert part of the input signal before or after other stages have or will make their estimate. The outputs of all stages are combined to form the final output code. This is best explained by Figure 2.4.

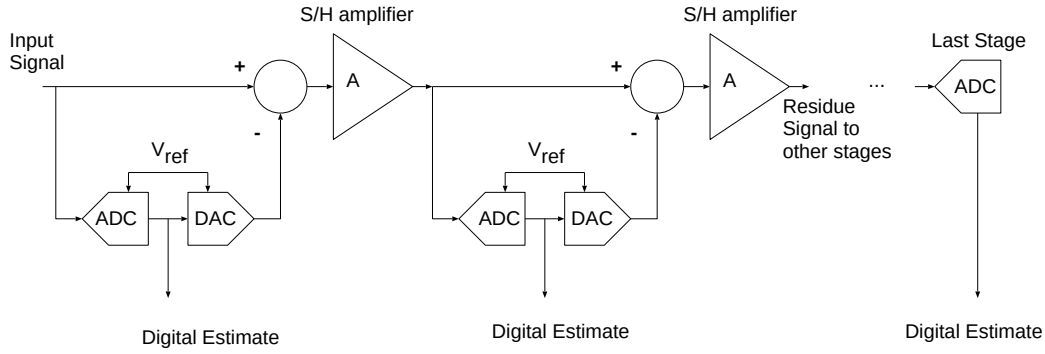


Figure 2.4: Pipeline Analog to Digital Converter Block Diagram

The signal propagation in Figure 2.4 begins at the node labeled input signal, where the ADC of the first stage makes a digital estimate of the input signal. The digital estimate is used immediately by the DAC of the first stage to create an analog equivalent to the digital estimate. This analog equivalent to the digital estimate is subtracted from the original input signal creating a residue that is used by the next stage. Essentially, each successive stage makes a finer estimate of the original input signal by looking at the residue of the previous stage. It can be thought of as a bitwise estimate of the input signal, MSB first. The amplifier is needed for hardware simplification. If each stage had to make an estimate of an ever shrinking residue, different voltage references would have to be used. Each amplifier effectively puts the residue in the same range as the original input signal so that hardware and voltage references can be reused.

The relationship between the input signal, digital output, and residue output voltage is depicted

better in Figure 2.5 where the error voltage for a two bit analog to digital converter is shown. Note that in this case the error voltage has not been amplified for use by later stages. For a more in depth explanation and probabilistic definition of the error voltage see [2].

Examining a single stage in more depth will help to explain these concepts better. Figure 2.6 shows the specific case for two bits per stage. On the left hand side the input output signal plot shows a ramp, the right hand side shows a residue plot that would be typical of a two bit per stage ADC. Essentially it is a plot of the error voltage that occurs in any analog to digital converter, which is typically discarded. However, in pipeline converters the error voltage of one stage is passed on to the next stage as a residue signal that following stages use to make a finer estimate of the input signal.

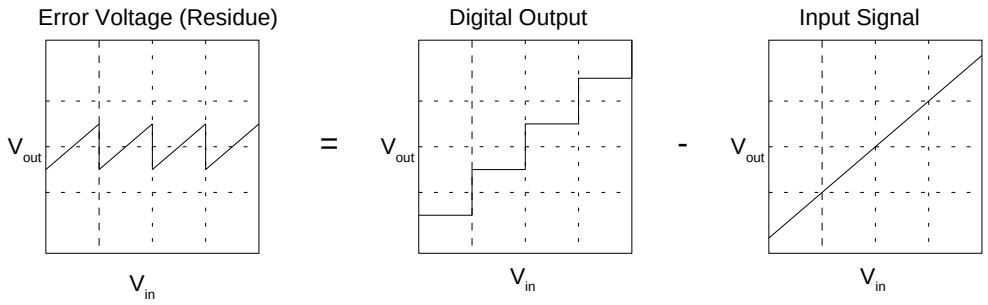


Figure 2.5: Origin of the Error Voltage

Pipelined A/D conversion has several advantages over more traditional flash converters. The main advantage is that pipeline converters have much lower power consumption because the number of comparators increases linearly with the bit resolution versus a power of two with typical flash architectures. However to minimize power and maximize speed in pipeline converters, several issues must be taken into consideration which will be discussed in more detail in later sections.

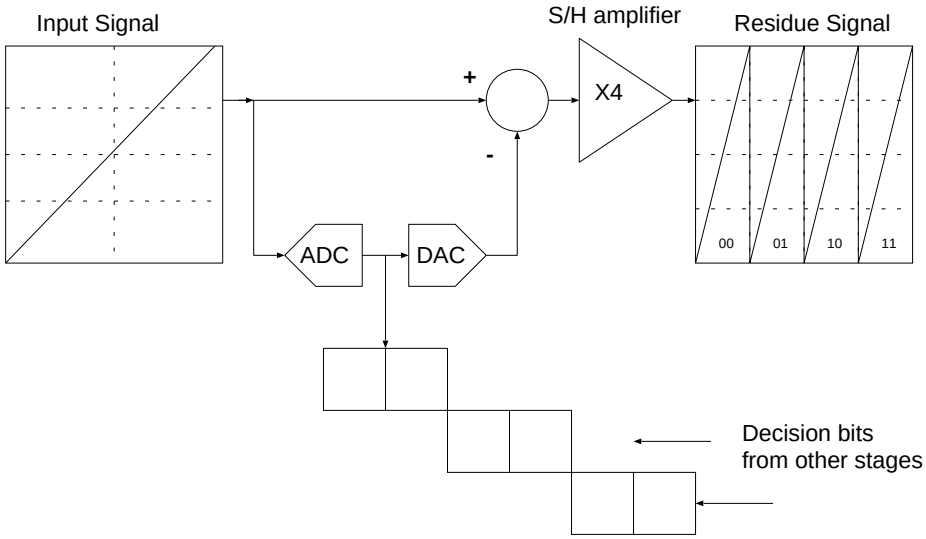


Figure 2.6: Single Pipeline Stage with Residue Plots

2.4 Summary

Of the three architectures discussed the pipeline structure has the most desirable characteristics for the problem at hand. It is best suited for CMOS processes, requires minimal area, and more hardware replication takes place keeping design time to a minimum. This conclusion will become more evident in the next chapter where the specifics of the structure are developed in more detail.

Chapter 3

Pipeline Architecture

As discussed in the previous section, a pipeline converter makes a rough estimate of the input signal in each stage and passes a precise version of the remainder onto the next stage. The remainder is calculated by subtracting the estimate from the original signal and amplifying by an appropriate gain factor. The gain factor depends on the number of bits estimated in each stage. The digital estimate is sent to a D/A converter to be converted to an analog signal that can be subtracted from the input signal. The digital estimate from each stage is sent to digital correction logic that combines the stage outputs into an output code. This basic pipeline concept is straightforward but has many problems that make the physical implementation difficult. To see how all of these problems interact together, a few concepts need to be explored further.

3.1 Bit Resolution Per Stage

The bit resolution per stage becomes an important aspect in pipeline ADC design because of its influences on gain bandwidth of the residue amplifiers, total capacitance, power consumption, error tolerance, and accuracy of required components, just to name a few. More recent pipeline ADC

converters have taken two different functional paths. Some designs are purely analog focusing primarily on component matching [3] and others design use less accurate analog designs and rely on digital correction logic to correct any random variations [1].

The goal is to design a 12-b ADC operating at 50MHz sampling rate accurate within +/- 0.5 LSB in DNL and INL. Another limiting constraint is that the ADC must be made in an entirely CMOS process. These requirements have been met numerous times in a BiCMOS process [3] but rarely in an entirely CMOS process, and most often using digital correction. The main limitation in an entirely analog and CMOS process is the bandwidth and current drive required to get the necessary gain and matching. Typically, fewer stages with higher bit resolution are used in an entirely analog process to reduce injection error that could normally be corrected with digital logic. In this design it is proposed to make an 12-b pipelined ADC in a $0.6\mu m$ CMOS process with digital correction.

3.1.1 Device Matching

In most analog integrated circuit designs device matching becomes more important than the actual physical value of the device whether it is a resistor, capacitor, or active device. Typically device values are designed within an absolute tolerance of 20% (depending on the process used), but the ability to match two components on an IC is usually much better. Smaller devices are harder to match than larger devices because of uncertanties in the circuit fabrication process. The spatial tolerance of a CMOS process is usually specified by the parameter λ, which is half the minimum gate length. The worst possible mask mismatch is typically 0.75 λ. This is not the only factor that determines component tolerances. Process variations such as etching, ion implantation, and oxide growth are all controlled to a certain point beyond which random variations determine the exact characteristic of a device. Assuming that these tolerances are known and the process is guaranteed accurate within certain physical dimensions, device sizes can be increased to a point where process variations become insignificant compared to the overall size of the device. With this in mind two

devices can be constructed with a ratio of values with specified precision.

Matching becomes the dominant factor in the performance of a pipeline ADC. Each stage of a pipelined converter makes a rough estimate of the input signal, subtracts that estimate from the signal, and amplifies the remainder to be passed on to the next stage. The amplification is dependent on the bit resolution per stage and must be accurate so that the next stage receives the proper signal. Typically the amplification is implemented with some sort of switched capacitor array and the capacitors must have a known ratio in order for the amplification to be accurate. Any deviation from the desired ratio results in improper results being passed on to the next stage and the ADC will not output the right answer.

Previous work has attempted to characterize the worst possible mismatch of components based on size [4]. The paper referred to in [4] uses a $0.8\mu m$ double-poly process that is similar to the $0.6\mu m$ process that is going to be used in this thesis. It is assumed that the newer $0.6\mu m$ process will have no worse matching than the $0.8\mu m$ process. From the analysis in [4], a $25\mu m$ X $25\mu m^2$ capacitor has a ratio standard deviation of 0.05-0.1.%. This size results in a large capacitance (greater than 1.25pF) that puts too many constraints on the analog design as well as consuming excessive silicon area.

Obviously a smaller capacitor that is just as well matched must be found. Another type of capacitor that can be constructed in the $0.6\mu m$ process is a metal-poly cap instead of the poly-silicon caps that are standard in the process. This capacitor construction method has an additional problem: the capacitor is not as closely matched for the same size area. The reason for this is that the metal layer is in different physical position than the poly-silicon. The first metal layer is typically constructed on top of a protective layer of silicon oxide that is not well controlled for its planar properties. For this reason any metal layer that is applied on top of the silicon oxide would not be planar. Since capacitance is proportional to the distance between two plates, it is evident that the non-planar metal will not result in a very predictable capacitance and for this reason should not be used in

any matching situations[5]. From this analysis it is clear that the project description cannot be met by matching alone. Later sections will discuss the implications of matching errors and introduce a method used for correction of these errors.

3.1.2 Bandwidth

As previously stated, the number of bits resolution has a direct effect on the required bandwidth of the analog circuitry. After one stage approximates the signal and creates an error voltage, the error voltage must be amplified to full scale range to create a residue voltage. The gain factor increases with increasing bit estimate in the first stage because the size of the error voltage is dependent on how many bits the first stage approximates (the larger the bit estimate the smaller the error voltage). Summarizing the effect on the amplifier, the larger the gain the larger the gain bandwidth product. This phenomenon is shown in Figure 3.1, where every additional bit of precision added to a stage increases the closed loop gain of the residue amplifier by 6dB. This lowers the time constant of the circuit. The effect of the time constant on the pipeline ADC performance will be developed more in the Analog Architectural Requirements chapter.

3.1.3 Latency

The time when the input signal enters the converter to the time that the digital output is available on the output pins is called the latency of the converter. Because a pipeline converter uses decimation in time to increase throughput, latency is a drawback to their use. For example, if a converter has twelve stages , and it determines one bit per stage, the minimum latency is twelve cycles for the signal to propagate through the entire pipeline. If the bit resolution per stage for the same converter is increased, fewer stages will be required for the same number of bits. For this reason latency must also be considered in the design of pipeline converters.

In this respect latency limits the applications of pipeline converters. The speed of a pipeline con-

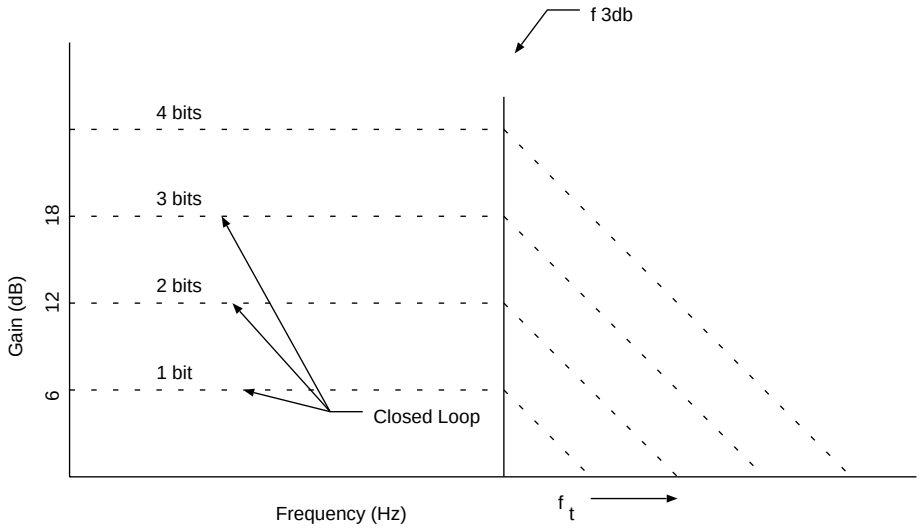


Figure 3.1: Closed loop gain effect on 3dB frequency

verter may be reasonable for certain applications but if latency is an issue, as in high speed control loops, another converter architecture might be preferable.

3.2 Pipeline Errors

3.2.1 Offset Errors

There are three main sources of error in pipeline A/D converters: comparator, offset, and interstage gain error all of which lead to nonlinearities in the output codes. The causes of both offset and gain error will be examined.

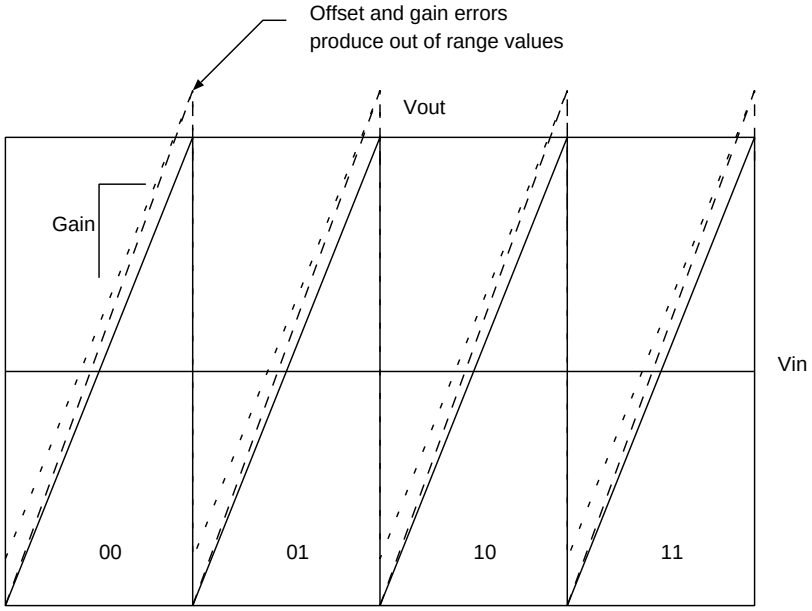


Figure 3.2: Residue exceeding full scale voltage in a 2 bit per stage converter

Offset error can be caused by charge injection during the sample and hold process, random offsets in gain stages created by manufacturing variations, and characteristics changing due to device aging. While these can be minimized through design and layout choices it usually is a tradeoff with power and speed. For example charge injection and noise can be minimized in the sample and hold by making a larger capacitor and a smaller sampling switch. This results in a slower sample and hold and increased power loss due to the larger charge required by the capacitor.

Figure 3.2 shows the output residue of the first stage of a two bit per stage ADC if the input were a ramp. As discussed earlier, the residue output is just the error voltage amplified to the full scale voltage. The peaks of the output plot lie exactly on the full scale voltage of the converter defined by the outer box. This presents a problem that is inherent in pipeline converters: any type of error that causes the residue to exceed the full scale voltage results in missing codes. Or more simply stated, all input values that cause the output residue to exceed the full scale voltage will result in the same output code. This is best illustrated by Figure 3.3 which shows a simulated output of a two bit per stage converter. Figure 3.3 (a) shows the Input versus. Binary Output for an 8-bit converter with stages that have two bits of resolution. An offset has been included in the first stage that makes the residue output appear exactly as shown in Figure 3.2. Notice the flat spots in the output that correspond to residue output exceeding the full scale voltage (the same digital output is occurring for many different input values). In this case, digital correction cannot be used because the same digital output corresponds to many different inputs, there is no way for correction circuitry to differentiate between the values.

3.2.2 Gain Errors

It can be shown that gain error has a similar effect on the residue output as an offset. Referring back to Figure 3.2 the slope of the residue output between decision levels is defined by the gain of the interstage amplifier. As the slope of the line increases the peak of the residue output again exceeds the full scale voltage and the same output shown in Figure 3.3 results. The main cause of gain error, assuming sufficient open loop gain of the op-amp, is mismatch between components of the interstage amplifier.

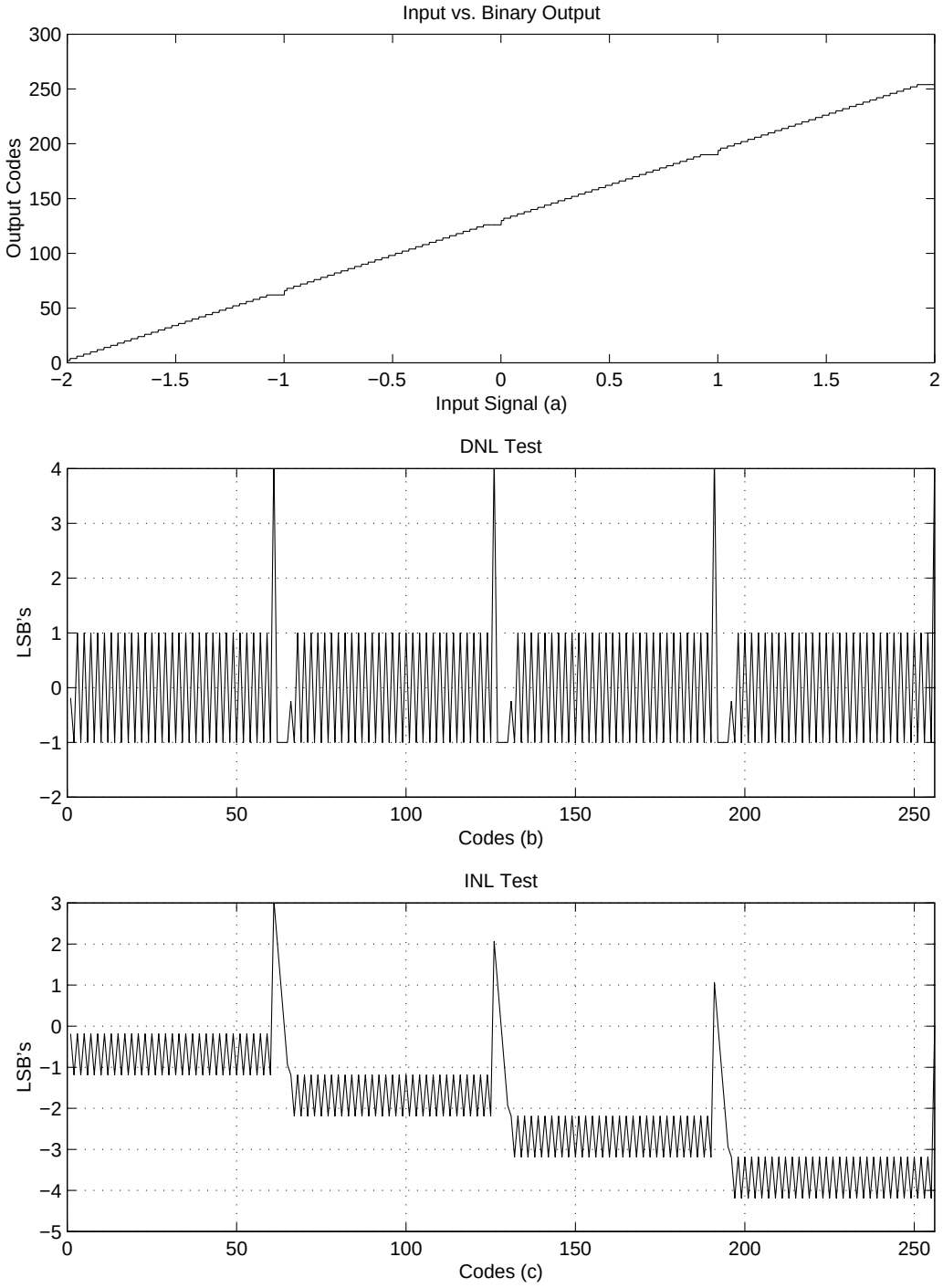


Figure 3.3: Missing Code errors caused by the output residue exceeding the full scale voltage in the first stage of a 2-b per stage converter.

3.2.3 Comparator Errors

By now it is evident that anything that causes the residue output to exceed the full scale voltage causes an error in the output codes. Comparator errors also do just that. In the two bit per stage structure a comparator error causes the vertical line, which represents the comparator threshold, to shift in either direction. This is show graphically in Figure 3.4.In this regard, comparator threshold can be a critical aspect of pipeline design.

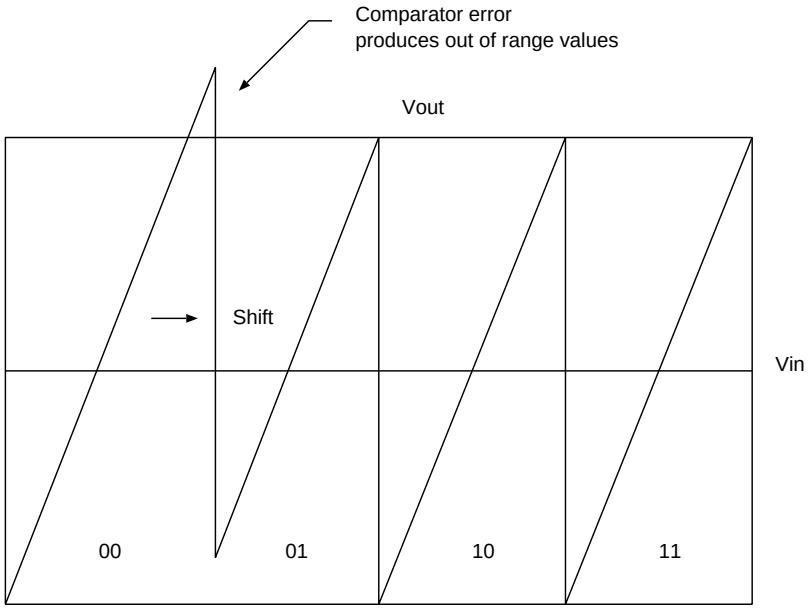


Figure 3.4: Comparator threshold shift causing out of range value.

3.3 1.5 bit Per Stage Architecture

The problem of missing codes discussed previously has several different solutions. The most common is the concept of overlap between stages. The result of overlap is best illustrated by a discussion of the 1.5 bit per stage architecture. Figure 3.5 shows the specific pipeline converter stage used for this design, which will introduce some specifics of this project and help understand overlap in pipeline converters in general.

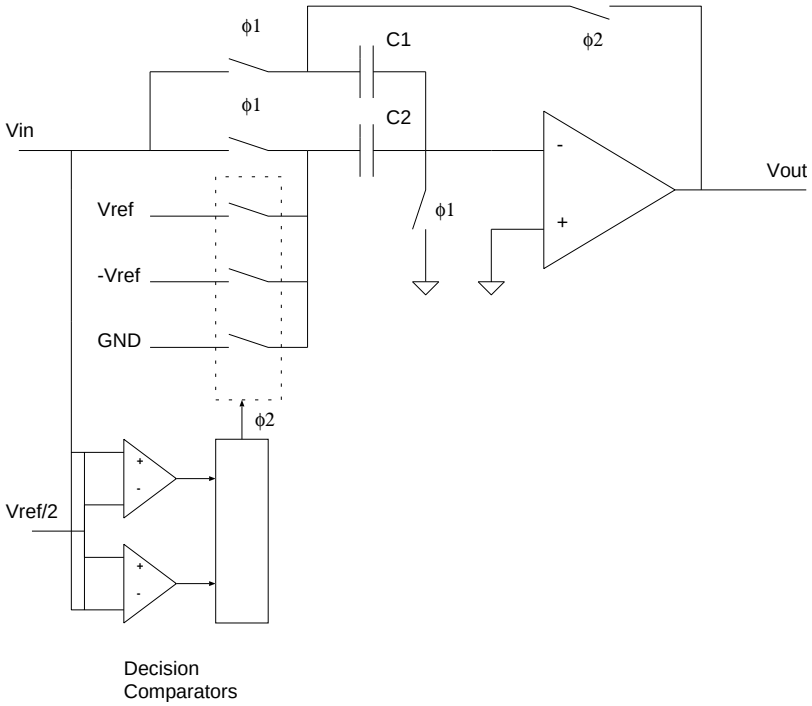


Figure 3.5: 1.5 bit per stage analog block

The typical radix 2, 1.5 bits per stage, pipeline converter has three decision levels. Those are 1, 0, and -1. These are used to decide the bit values for a particular stage and to calculate the residue that will be passed on to the next stage. The residue is calculated by

$$V_{out} = \frac{C_1 + C_2}{C_1} V_{in} - D\frac{C_2}{C_1} V_{ref} \tag{3.1}$$

substituting $C_1 = C_2$ results in

$$V_{out} = 2V_{in} - DV_{ref} \tag{3.2}$$

where V_{ref} is half of the full scale voltage. The first stage compares the input value to $\frac{1}{4}V_{ref}$. If the input is greater, V_{ref} is subtracted from the input and the result is sent through a gain=2 amplifier. Should a negative number greater than $-\frac{1}{4}V_{ref}$ result, this simply tells the next stage should add some back into the signal. The decision levels are described by equations:

$$-1 = -V_{ref} < V_{in} < -\frac{1}{4}V_{ref} \tag{3.3}$$

$$0 = -\frac{1}{4}V_{ref} < V_{in} < \frac{1}{4}V_{ref} \tag{3.4}$$

$$1 = \frac{1}{4}V_{ref} < V_{in} < V_{ref} \tag{3.5}$$

The decision levels of each stage are used to calculate the digital output in the following manner.

$$V_{out}[digital] = D[1]2^{N-2} + D[2]2^{N-3} + ... + D[N] \tag{3.6}$$

3.3.1 Overlap and Error Correction

In the previous non-overlapping scheme, gain and offset errors produced out of range values, resulting in non-unique output codes for many different input values. In overlapping schemes the residue amplifier gain can be divided by two as a result of binary arithmetic. The overlap of one bit from one stage to the next is a multiplication by two of the effect of the digital output of that stage.

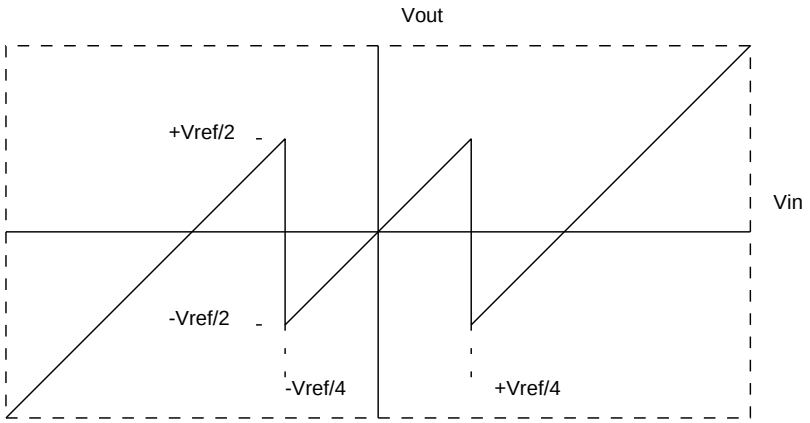


Figure 3.6: Residue plot for 1.5 bit per stage architecture

Because any given stage can have double the effect on the digital output code, the analog residue only needs to be multiplied by half as much when being passed to the next stage. In this manner the LSB of one stage can either be determined by itself or the following stage. Looking at the 1.5 bit per stage residue plot in Figure 3.7 it can be seen that the full scale voltage level, defined by the dotted outline, is well clear of the input output voltage curve, even with errors present. The input output voltage curve can move anywhere within the dotted outline and not produce any values out of range.

The bit overlap allows for some sort of residue to be passed on the next stage without a total loss of data. The digital number that the A/D reports includes the offset, gain and comparator errors of all stages. In this aspect the number is wrong but it is still a unique number versus non-overlapping schemes which have wrong and non-unique numbers. To correct wrong numbers in overlapping schemes some sort of algorithm must be implemented digitally to recognize each unique number and substitute in a correct value.

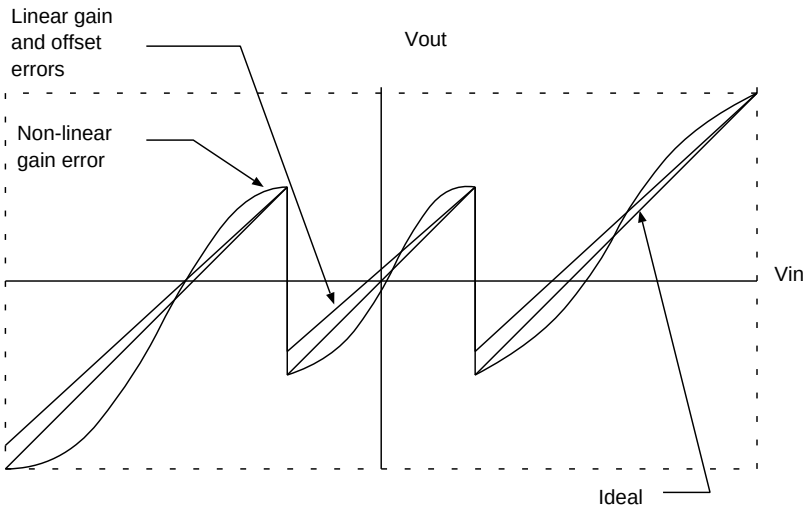


Figure 3.7: Residue plot with possible errors.

3.3.2 Overlap and Bit Resolution

The bit resolution per stage can be increased while maintaining one bit overlap to help decrease the throughput latency of a pipeline converter, but with a decrease in several other performance parameters. The bit resolution per stage not only affects bandwidth, matching, and latency, but the feed-through offset susceptibility of overlapping schemes as well. Two possible cases for a twelve bit converter are examined, twelve 2-bit converters and six 3-bit converters. Each of the 2-bit stages require two comparators to distinguish between each of the three different levels. The 3-bit converters require six comparators to distinguish between each of the seven levels.

The 1.5-bit converter structure has greater offset error correction. While the 1.5-bit converter can correct errors up to 1/4 of the reference voltage, shown graphically in Figure 3.8, the 2.5-bit converter can only correct up to 1/8 of the reference voltage, shown in Figure 3.9 . In general, for every added bit of resolution per stage, the offset error correction limit reduces by half[6].

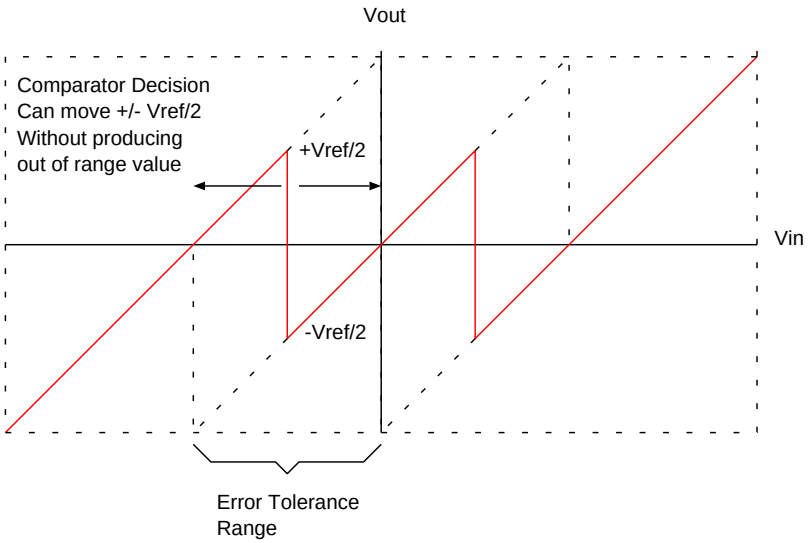


Figure 3.8: 1.5-bit per stage residue plot with comparator error limits.

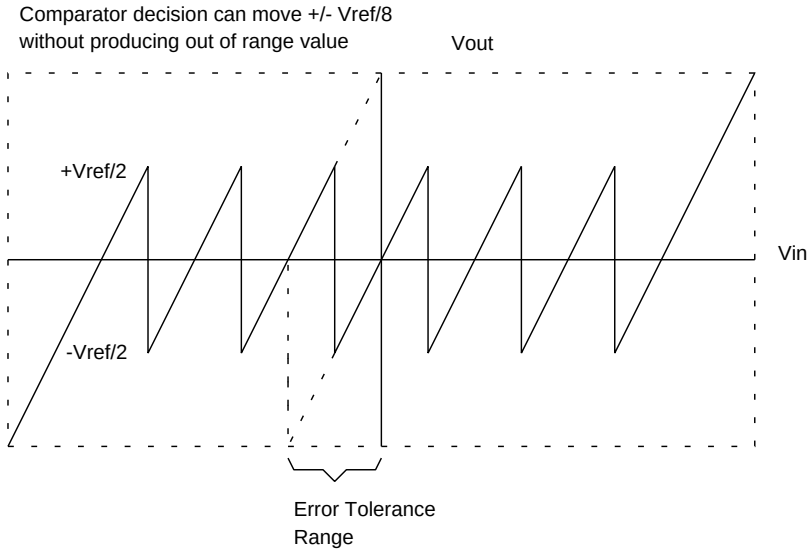


Figure 3.9: 2.5-bit per stage residue plot with comparator error limits

3.4 Calibration

3.4.1 Mathematical Characterization

The following section presents a method for correcting gain errors in a 1.5 bit per stage structure originally proposed by [1].

Implementing a radix=2 1.5-b per stage pipeline, and using a switched capacitor amplifier the transfer function for the residue amplifier in each stage is as follows

$$V_{out}[i] = \frac{C_1[i] + C_2[i]}{C_1[i]} V_{in}[i] - D[i] \frac{C_2[i]}{C_1[i]} V_{ref} \tag{3.7}$$

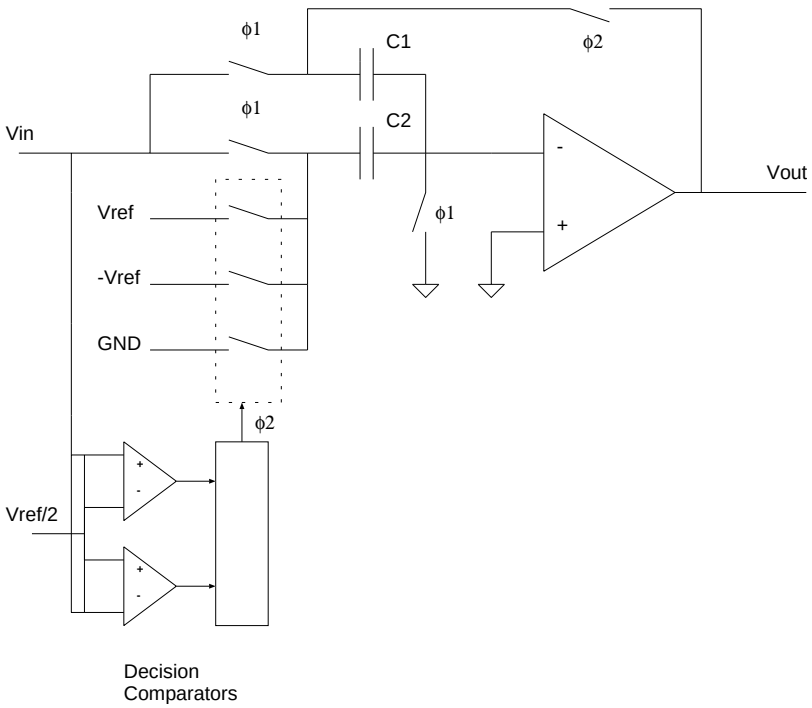


Figure 3.10: Origin of charge transfer equation.

Which is identical to equation 3.1 but the subscript [i] has been included to differentiate between stages.

For radix=2 1.5-bits per stage, the amplification must be exactly 2 and a known amount of V_{ref} must be subtracted out. The quantity $\frac{C_1[i]}{C_2[i]}V_{ref}$ is the analog equivalent of the digital value that that part of the stage estimated. If the ratio of $C_1[i]$ and $C_2[i]$ is not 1 then the digital estimate and the analog value subtracted will not match. In addition the residue passed on to the next stage will not be correct. The pipeline concept relies on the fact that the residue is amplified by a factor 2 so that essentially each stage determines the value of the next bit of the digital output code. Due to process variation discussed previously in the matching section, the ratio of $C_1[i]$ and $C_2[i]$ is not unity. However if the difference between $C_1[i]$ and $C_2[i]$ is known, digital values can be substituted into each stage so that the digital value now matches the analog value. For example, the first stage might be radix=1.91 and the second stage could be radix=2.11. It does not matter what the actual value is as long as it is known and close enough to 2 that out of range values are not produced.

To measure the gain and analog value subtracted out of each stage, the mismatch of the capacitors can be used. Making the following substitutions in equation 3.7: $V_{in}[i]$ is $V_{ref}[i]$, the charge on $C_2[i]$ is set to 0, and the decision $D[i]$ is forced to 1 which results in:

$$V_{out}[i] = \frac{C_1[i] - C_2[i]}{C_1[i]} V_{ref}$$

$$V_{out}[i] = 1 - \frac{C_2[i]}{C_1[i]} V_{ref}$$

After dividing out V_{ref} the mismatch between capacitors α is known. Substituting $C_2[i] = (1 + \alpha[i])C_1[i]$ into equation 3.7 results in

$$V_{out}[i] = 2\{1 + \frac{\alpha[i]}{2}\}V_{in}[i] - D[i]\{1 + \alpha[i]\}V_{ref}$$

$$V_{out}[i] = 2V_{in}[i] - D[i]V_{ref} + \alpha[i]V_{in}[i] - \alpha[i]D[i]V_{ref}$$

subtracting out the quantity $\alpha[i]V_{in}[i] - \alpha[i]D[i]V_{ref}]$ results in the desired transfer function of the residue amplifier.

$$V_{out}[i] = 2V_{in}[i] - D[i]V_{ref}$$

Subtracting the quantity $\alpha[i]V_{in}[i] - \alpha[i]D[i]V_{ref}]$ to correct the output of each stage is difficult to do and would result in a great amount of additional analog circuitry. If instead the analog value is subtracted out in the digital circuitry a much simpler design results. The digital output equation 3.6 replaced with the following.

$$= 2^{N-i}(d[i]\tfrac{\alpha_i}{2} + d[i+1]\tfrac{\alpha_i}{4} + d[i+2]\tfrac{\alpha_i}{8}...)$$

Simplified, the equations that determine the coefficient for each register of the pipeline are:

$$V_{out[1]}[digital] = 2^{N-2}D[1]\{1 - \frac{\alpha[1]}{2}\}$$

$$V_{out[2]}[digital] = 2^{N-3}D[2]\{1 + \frac{\alpha[1]}{2} - \frac{\alpha[2]}{2}\}$$

$$V_{out[3]}[digital] = 2^{N-3}D[3]\{1 + \frac{\alpha[1]}{2} + \frac{\alpha[2]}{2} - \frac{\alpha[3]}{2}\}$$

...

Notice that the quantity $D[i]\alpha[i]V_{ref}$ is immediately subtracted from the output code since that is a known quantity at the time of the decision. However the quantity $\alpha[i]V_{in}[i]$ is subtracted as the residue moves down the pipeline and $V_{in}[i]$ is known with more precision. The entire quantity could be subtracted at the end of the pipeline but this would require a large amount of memory. The method shown above only requires three different numbers to be stored for each stage. In addition the digital value of V_{ref} is actually divided by two moving down the pipeline so V_{ref} is not digitally the same for each stage.

With this calibration method it should be noted that later stages are used to correct previous stages. This presents a problem in that errors that occur in later stages will show up in the measurement of errors in the previous stage. It is thought that this effect is minimal because error measured is already small and the interaction of several errors when multiplied together will be small compared to a single error. In other words, $\alpha >> \alpha^2$. This effect was simulated behaviorally as a proof of concept and is included later in this chapter.

3.4.2 Offset

An offset term for each stage is also present in the measurement of the term α. The offset is actually an accumulation of several different factors in each stage and is treated as a superposition on the output of each stage. Thus, in the previous α measurement procedure the number that comes out of the pipeline is α plus some offset, where the offset is the sum of all of the offsets that occur after the stage being calibrated. To obtain the correct digital number for α all that needs to be done is subtract the offset. The digital number for the offset is measured in the same manner as α, using later stages to quantize the output of previous stages. The only difference is that the input to the stage being

calibrated is set to zero so that the digital number that comes out of the pipeline corresponds to later offsets superimposed upon one another.

3.5 Behavioral Simulations

3.5.1 Linearity Characterization

An important value that is frequently used to characterize analog to digital converters is linearity. The typical linearity figures used are differential nonlinearity and integral nonlinearity, both of which will now be briefly introduced.

Differential linearity is a measure of how often one code appears versus an adjacent code. Basically it is a normalized measurement of the digital step size. Theoretically the distance between two adjacent digital code centers should be one LSB.

Integral linearity can be though of as a measurement of the overall shape of the Input versus Binary output plot. Basically it is a running summation of the differential non-linearity errors. For example, consider eight different bits that occurred for some linear input test signal. If the distance between each step varies but the sum of the steps still equals eight LSB's, then the integral nonlinearity error at the end of test signal should be zero. An example of both differential and integral non-linearity is shown in Figure 3.11. For a more in depth analysis of integral and differential non-linearity see [8].

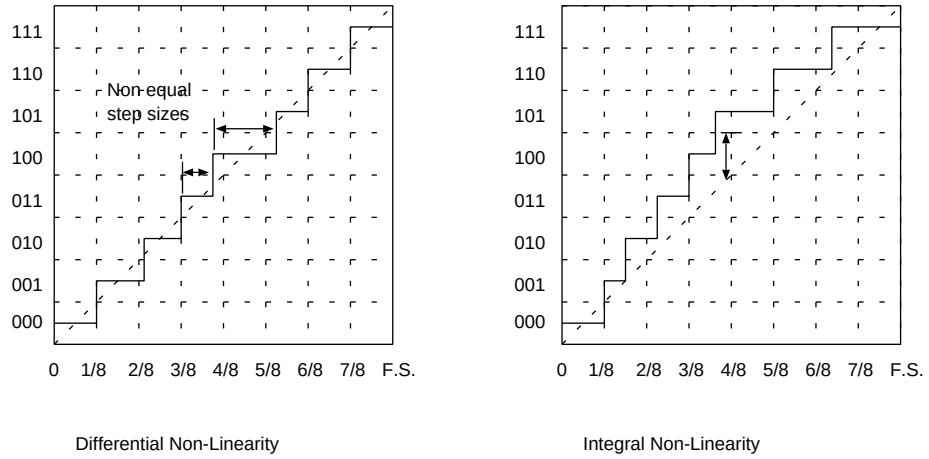


Figure 3.11: Example of Differential and Integral Non-Linearity

3.5.2 Simulation Procedure

An overview of the simulation procedure is included here. For complete simulation code written in C, see the Appendix. The effects of two of the three main sources of error have been included in the behavioral simulation: gain and offset. Additionally the correction scheme has also been included in the simulation to test its applicability to the architecture.

The goal of the simulations was to test the performance of the entire converter as close as possible without getting into too much complexity. With this in mind the first step was to describe the behavior of the analog processing block using equations 3.1, 3.3, 3.4, and 3.5. Equation 3.1 is described by charge. The reason for this was to implement the calibration later, which better depicted what was actually occurring. The code in Figure 3.12 shows the implementation of one stage, uncorrected. The actual formulation of the digital output is not done here. The decision bits are stored in the array d[12] and used later to calculate the output.

To simulate calibration a similar code structure as in Figure 3.12 was used. As discussed earlier, only one stage is calibrated at a time while later stages are used to quantify the calibration result. In the code shown in Figure 3.13, one stage is calibrated and later stages are created using a simple loop. Within the loop is the same structure as the regular uncalibrated converter. The stage that is being calibrated occurs first in the code, where the charge on C_2 (q2) is set to zero and the same output charge relating equation is used as in the uncalibrated converter. Note that the input is set to V_{ref} when the subroutine is called.

Finally, the decision bits are combined with the measured $\alpha[i]$ terms to form the output codes as shown in Figure 3.14. At first glance the math appears to be complex and difficult to implement in digital logic, but the coefficient for each register can be broken down into a few shift and add operations. The first line is used to calculate the number that is stored in the register of the first stage of the pipeline. It is calculated through the addition of 1 and $\alpha[0]$. Once the number is stored in the register of the first stage it is either not added into the output, half is added, or all of it is added

```
//uncorrected pipeline
float pipeline(double input,float cap1[], float cap2[], float offset[])
{
 float output=0;
 int output1=0;
 int i;
 double q1;
 double q2;
 double q3;
 int d[12]={0,0,0,0,0,0,0,0,0,0,0,0};
//comparators for all 7 stages
 for (i=0; i<=11; i++)
 {
  if (input <= -0.5)
  {
   d[i]=0;
   q3=cap2[i]*2;
  }
  else if (-0.5 < input && input < 0.5)
  {
   d[i]=1;
   q3=0;
  }
  else if (input >= 0.5)
  {
   d[i]=2;
   q3=-cap2[i]*2;
  }
//Residue amplifier
  q1=input*cap1[i];
  q2=input*cap2[i];
  input=((q1+q2+q3)/cap1[i])+offset[i];
 }
//Calculate outputs using decision bits
 for (i=0; i<=11; i++)
 {
  output = output + d[i]*pow(2,(6-i));
 }
 output1=output;
 return(output1);
```

Figure 3.12: C implementation of an uncalibrated pipeline converter

```
//Error calculator
float gain_error(double input,float cap1[], float cap2[], int x, float offset[])
{
 float output=0;
 int i;
 double q1;
 double q2;
 double q3;
 int d[12]={0,0,0,0,0,0,0,0,0,0,0,0};
//Use Lee's method on ith stage
 q1=input*cap1[x];
 q2=0;
 q3=-cap2[x];
 input=((q1+q2+q3)/cap1[x])+offset[x];
//Send output down remainder of pipeline
 for (i=x+1; i<=11; i++)
 {
//Comparators
  if (input < -0.5)
  {
   d[i]=0;
   q3=cap2[i]*2;
  }
  else if (-0.5 <= input && input <= 0.5)
  {
   d[i]=1;
   q3=0;
  }
  else if (input > 0.5)
  {
   d[i]=2;
   q3=-cap2[i]*2;
  }
//Residue amplifier
  q1=input*cap1[i];
  q2=input*cap2[i];
  input=((q1+q2+q3)/cap1[i])+offset[i];
 }
//Calculate ouput using decision bits
 for (i=0; i<=11; i++)
 {
  output = output + d[i]*pow(2,(6-i));
 }
 return(output);
}
```

Figure 3.13: C code for the measurement of α

depending whether D[0] is 0 1 or 2 respectively[1]. The second line corresponds to the coefficient stored in the register of the second stage of the pipeline. The coefficient can be calculated by adding two more numbers to the coefficient of the first register, $2\alpha[0] + \alpha[1]$. The coefficients for the remaining registers are calculated in the same manner, adding two more numbers to the coefficients of the previous register.

```
 }
  output = output + (d[0]*64)*(1-alpha[0]);
  output = output + (d[1]*32)*(1+alpha[0]-alpha[1]);
  output = output + (d[2]*16)*(1+alpha[0]+alpha[1]-alpha[2]);
  output = output + (d[3]*8)* (1+alpha[0]+alpha[1]+alpha[2]-alpha[3]);
  output = output + d[4]*4*   (1+alpha[0]+alpha[1]+alpha[2]+alpha[3]);
  output = output + d[5]*2*   (1+alpha[0]+alpha[1]+alpha[2]+alpha[3]);
  output = output + d[6] *    (1+alpha[0]+alpha[1]+alpha[2]+alpha[3]);
  output = output + d[7] *    (1+alpha[0]+alpha[1]+alpha[2]+alpha[3])/2;
  output = output + d[8] *    (1+alpha[0]+alpha[1]+alpha[2]+alpha[3])/4;
  output = output + d[9] *    (1+alpha[0]+alpha[1]+alpha[2]+alpha[3])/8;
  output = output + d[10]*    (1+alpha[0]+alpha[1]+alpha[2]+alpha[3])/16;
  output = output + d[11]*    (1+alpha[0]+alpha[1]+alpha[2]+alpha[3])/32;
  output1=output-m_offset[0];
 return(output1);
}
```

Figure 3.14: Formulation of output codes using decision bits and calibration results.

Before moving on to the actual results of the behavioral simulation results a few things should be noted. First, random gain and offset values within tolerable calibration range were chosen. The same values were used for both calibrated and uncalibrated simulations so that any measured performance increase correlates directly. Second, errors were included for all stages of the pipeline including the uncalibrated stages to better simulated the behavior of the converter. Essentially this will prove if $\alpha^2 << \alpha$ is correct.

3.5.3 Simulation Coverage

The previous simulation was intended to cover the most common and dominant errors that occur in pipeline converters. Unfortunately there are numerous other errors that can also be a dominant

[1]In previous sections the decision levels are -1, 0, and 1. In the simulation and physical implementation the decision levels have been shifted up by one so that only addition takes place in the digital pipeline.

factor in operation. However if designed properly, these errors will be negligible compared to gain, offset, and comparator errors.

One problem that occurs in many digital devices, not just pipeline converters, is digital rounding. Binary representation of numbers are convenient for implementing digital devices, but the most precision that can be carried out in any binary arithmetic operation is one LSB. With regard to the pipeline converters, this becomes most critical when the pipeline is used to perform calibration on itself. The most precision that can be carried out on the calibration routine is one LSB. This means that the measurement and calculation of the coefficients that are loaded back into the pipeline will be subject to binary rounding. Unfortunately this effect is not covered in the behavioral simulation and is minimized in the physical implementation by adding additional stages onto the pipeline.

Another effect that can result in poor operation of the pipeline is a non-linear gain error. This subject is expanded on in the Analog Architecture chapter but is mentioned briefly here to clarify its absence from the behavioral simulation. Ideally the residue that is passed between stages is amplified by two in the 1.5 bit per stage architecture, regardless of what voltage level it is. However, analog amplifiers typically have a non-linear voltage amplification relationship. The result of this effect on the residue output has been exaggerated in Figure 3.7 Page 22. From the figure it is evident that measuring the slope of that line, which is what is proposed by measuring the gain factor α, will not work since the slope of the line is not a constant. How this non-linear gain problem is minimized is covered in the Analog Architecture chapter.

3.5.4 Residue Plots

The previous simulation outputs large amounts of data into a text file, including calibrated and uncalibrated digital output codes. To speed up the manipulation and plotting of the data Matlab was used. The routines used for making the following plots have been included in the Appendix. Three plots of both the calibrated and uncalibrated data are included to measure the performance increase,

raw output codes, differential linearity plots, and integral linearity plots.

Figure 3.15 shows the simulation results of the uncalibrated converter. The differential linearity is ± 1.5 LSB and the integral linearity is within ± 5 LSB. These results are poor for even the worst converters. With the digital correction scheme, the differential linearity is within $\pm$ .25 LSB and the integral linearity is within $\pm$ .35 LSB. For the same analog pipeline with digital correction this is a dramatic improvement.

3.5.5 Simulation Results

Simulation of the calibration procedure indicates that it will work provided that second order effects such as binary rounding and non-linear gain are minimized. The results of the simulation are significant but can be misleading mainly because of the phenomenon not modeled in the simulation. However, the improvement is significant enough to warrant its use in an actual pipeline. In an actual pipeline the INL and DNL quoted in the calibrated converter could triple, but even $\pm$ .75 LSB in INL and DNL is a respectable performance.

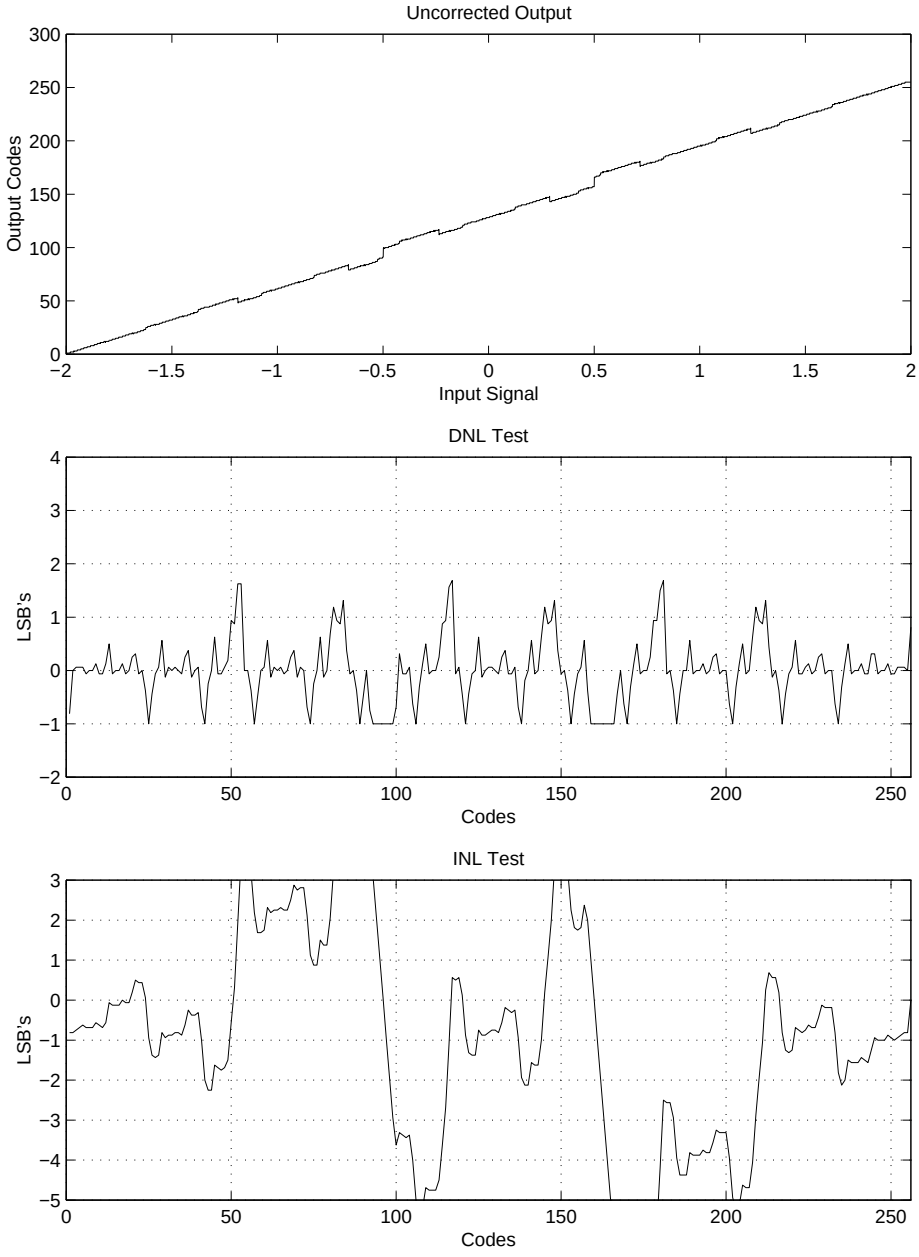


Figure 3.15: Uncalibrated converter simulation results

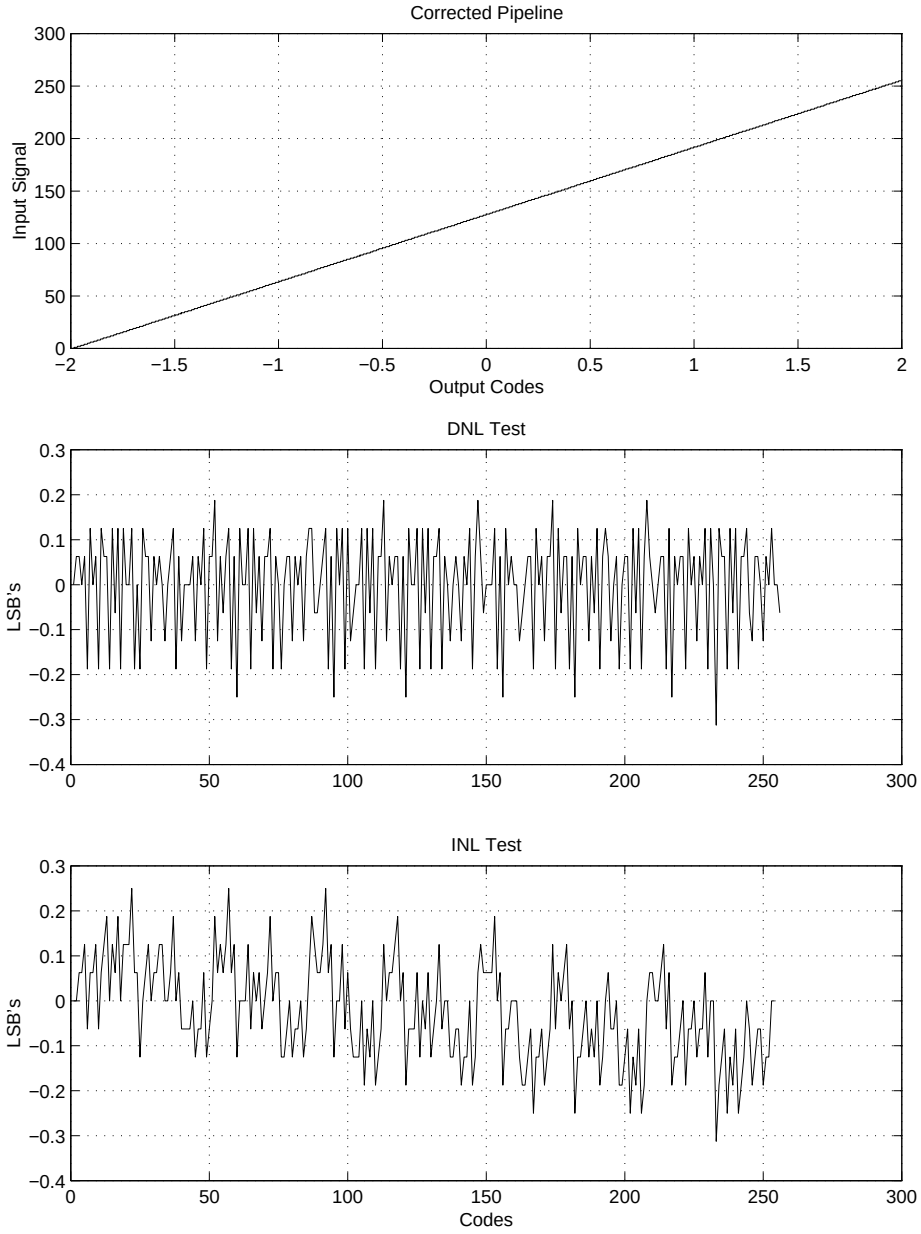


Figure 3.16: Calibrated converter simulation results

3.6 Summary of Requirements

Due to the lack of process matching specifications and limited experience in mixed signal design in the HP 0.6μm process, the radix=2 1.5-bit per stage architecture shows the most promise. Other variations within the overlapping scheme, such as increased bit resolution per stage, could be used but they would not permit as much error to occur within the pipeline. In later chips these variations could be used once process tolerances are known and the performance of mixed signal designs in the process are better understood. Another item in support of the radix=2 converter scheme is the required bandwidth of the analog amplifier, basically it requires the least amount of bandwidth. Since the goal of this project is an ADC and not a high speed op-amp, the design requirements on the op-amp need to be as minimal as possible. The incorporation of other projects, such as a high speed op-amp, might make it possible to go to another scheme in a later ADC project, but in the first run design time needs to be kept as short as possible. For these reasons a radix=2 1.5-bit per stage pipeline ADC with digital correction has been implemented in the first test chip.

Chapter 4

Analog Architectural Requirements

To meet the required performance of a pipelined ADC several performance criteria must be met by the analog processing circuitry. The criteria include gain, linearity, bandwidth, noise, stability, slew rate, and settling time. The following is a discussion of how to determine the criteria of the analog processing circuitry based on the required performace of the overall pipeline ADC.

4.1 Bandwidth

To ensure that the proper signal is passed to the next stage in a pipeline ADC the output of one gain stage must settle to a specified value before the next stage can sample it. Settling time is influenced by the 3dB bandwidth of the interstage amplifier. Most systems are typically concerned with the ten to ninety percent rise time which is usually approximated by 2.2 τ. τ is specified by:

$$f_{3dB} = \frac{1}{2\pi\tau}$$

However this relationship is only valid for first order systems, or higher order systems that are overdamped. Assuming that the response of the intersage amplifier is overdamped, the first order time

resonse is defined by:

$$V(t) = V_{max}e^{-\frac{t}{\tau}} \tag{4.1}$$

To apply this equation to the output of the interstage amplifier it is better defined in terms of LSB's. The level defined as one LSB in volts is simply the full scale voltage divided by the number of bits resolved by the ADC. Substituting this into equation 4.1results in:

$$\frac{V_{fs}}{2^N} = V_{fs}e^{-\frac{t}{\tau}}$$

$$\frac{1}{2^N} = e^{-\frac{t}{\tau}} \tag{4.2}$$

Equation 4.2 is the settling time for one LSB. With this approximation the time constant, and thus the bandwidth specified in LSBs of accuracy can be calculated. Table 4.1 shows the required settling time for one half LSB resolution for increasing converter bit resolution.

Bit Resolution	t(τ)
8	6.24τ
9	6.93τ
10	7.62τ
11	8.32τ
12	9.01τ
13	9.70τ
14	10.40τ
15	11.09τ

Table 4.1: Settling time for one half LSB resolution.

4.2 Stability

For the approximation made in the bandwidth section to be correct the AC closed loop response of the interstage amplifier must be over-damped. The interstage amplifier is of much higher order than a first order system but if the system is over-damped it can be approximated by a first order system. A typical measurement of the stability of a closed loop system is the phase margin. Depicted graphically in Figure 4.1, it is defined as the phase "distance" from the 180^o point of the feedback loop, when the gain of the amplifier has reached unity. Physically, the 180^o point is the frequency at which the negative feedback loop, typical of a stable system, has 180^o of phase shift making it positive feedback, or an oscillator. If at the point where the negative feedback becomes positive the net gain of the system is less than one, then the system is stable. How stable is defined by the phase margin: The larger the phase margin, the more stable the system [12].

The response of a second order systems, and many higher order systems with second order dominant poles, is described by a damped sinusoid. Figure 4.2 shows several typical responses from a second order systems with different damping factors. The underdamped response reaches the desired voltage faster but it oscillates around that point for a long time. This behavior is undesirable in the interstage amplifier because the next stage has to wait longer to get the correct residue. Typically in pipeline ADC the settling time is specified with LSBs. For a typical converter it is desirable to have the sample and hold settle within some specified amount of LSBs. The overdamped response has the most desirable characteristics and can be better modeled by simple approximations such as those in the bandwidth section.

From the previous discussion the over-damped second order response is the most desirable and can be approximated by the first order model in the bandwidth section. To put this response in terms of phase margin, 75^o or more ensures that the system will be dominant first order. Essentially this means that there is a single dominant pole that adds 90^o of phase shift in the feedback loop plus an additional 15^o from higher order poles before the unity gain frequency is reached. The fact that

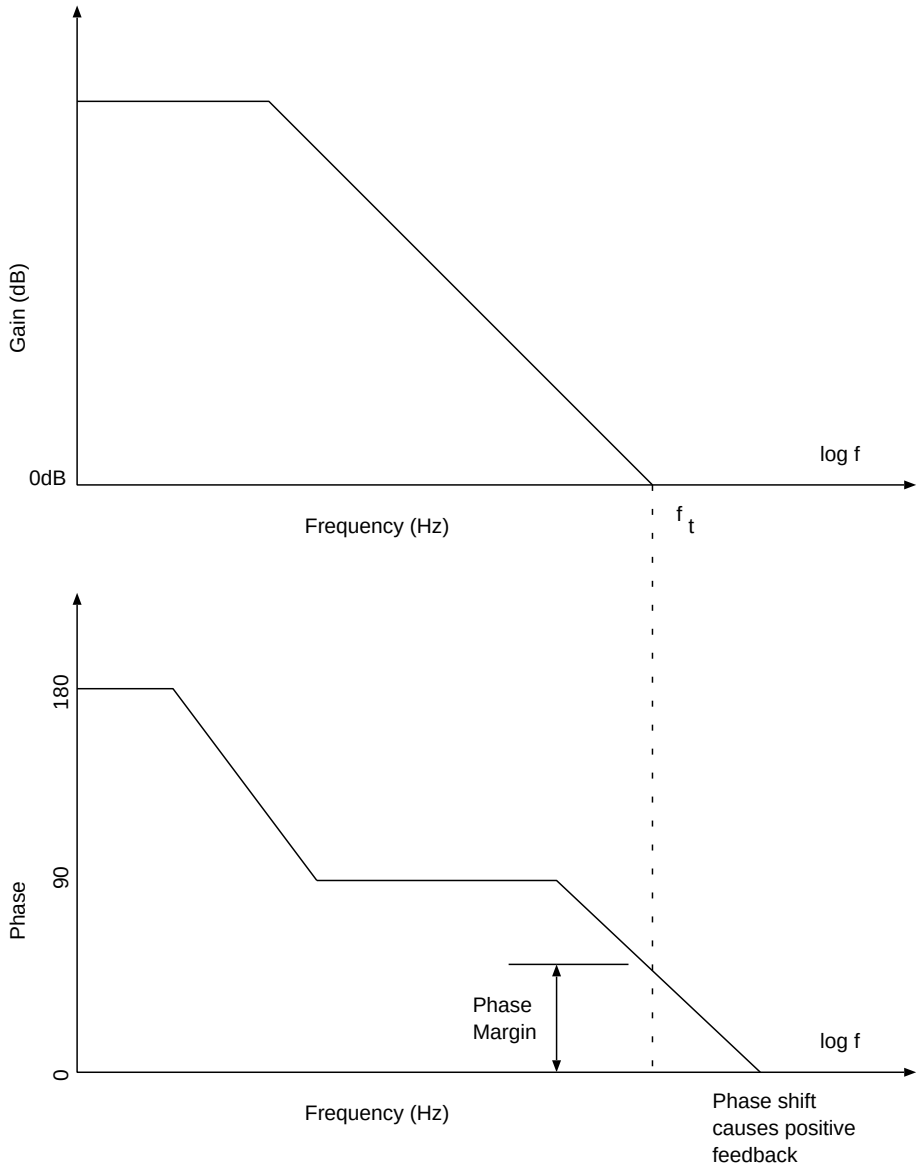


Figure 4.1: Phase Margin

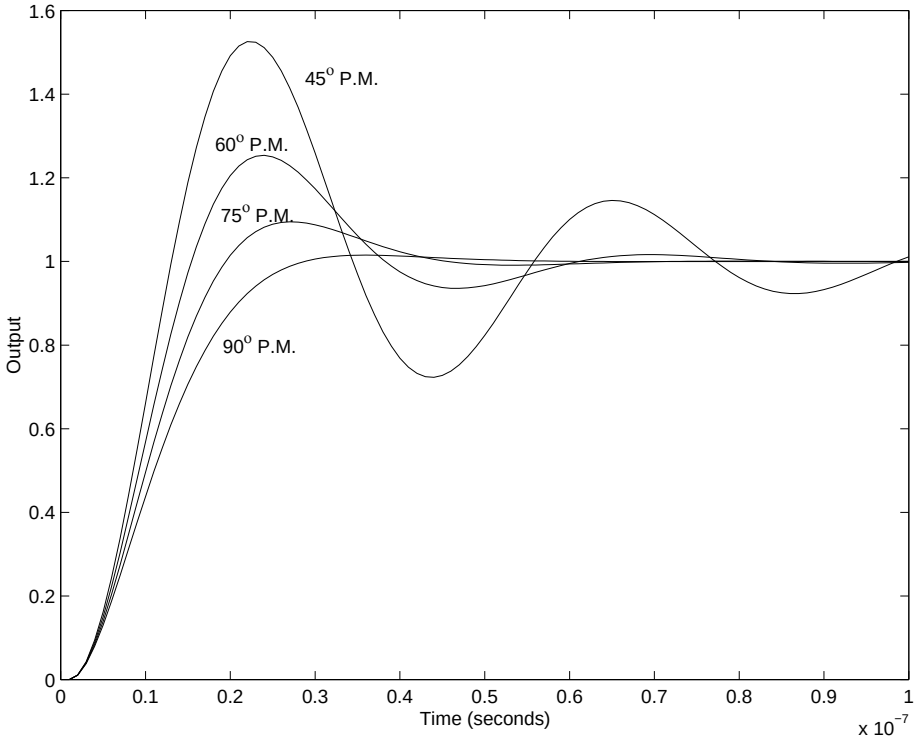


Figure 4.2: Second order response.

higher order poles only add 15^o implies that their location is actually beyond f_t. Additionally, the higher order poles have had no substantial effect on the magnitude plot yet.

4.3 Open Loop Gain

Open loop gain of an operational amplifier is typically quoted as a number on the AC response plot and little though is given to where the value comes from or where it is valid. AC models usually define the open loop gain as the derivative of the DC response evaluated at some specific operating point. However, when the AC operating signal grows to be to large, the AC small signal model can start to break down.

Figure 4.3(b) shows a typical DC sweep of a differential CMOS pair with a single ended output. It has two saturation regions and a two transition regions defined by the corners of the plot and a somewhat linear region in the middle. The linear region is typically the region that amplifiers are set to operate in because the gain from input to output can be assumed constant, where the gain quoted is the slope of the line in the linear region. When the output signal begins to enter the corners of the plot the slope is no longer constant and therefore the gain can no longer be assumed constant. This can result in a distorted output. When applied to pipeline converters this means that information is being lost.

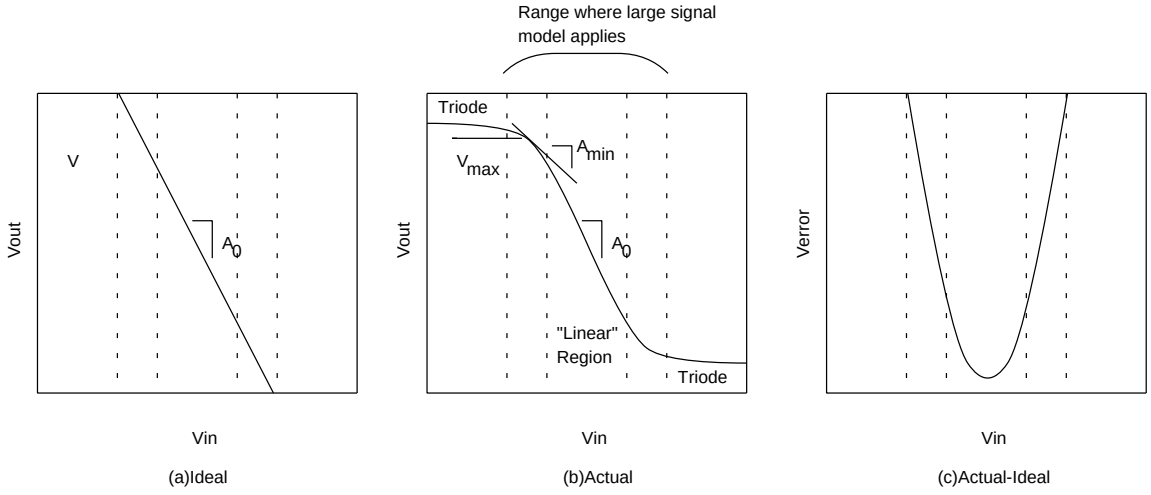


Figure 4.3: Ideal DC sweep, actual, and error voltage between them.

Looking closer at the linear region, it is in fact not linear and can be described by the large signal

DC model equation 4.3 [10], where V_{max} is the maximum output voltage and A_0 is the highest gain in the linear region usually denoted by $g_m R_o$. Equation 4.3 does not apply when the transistors enter the triode region.

$$V_{out} = A_0 V_{in} \sqrt{1 - \frac{A_0^2 V_{in}^2}{4V_{max}^2}} \tag{4.3}$$

As stated earlier, simple AC models appoximate the slope of the line as a constant and plug that constant into a closed loop model to get a closed loop gain. The simple closed loop model shown in Figure 4.4 is described by equation 4.4.

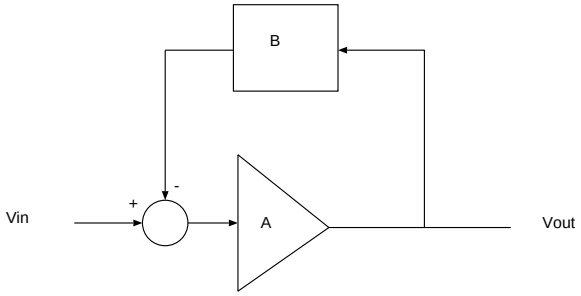


Figure 4.4: Amplifier with closed loop feedback.

$$V_{out} = [V_{in} - \beta V_{out}]A$$

$$\frac{V_{out}}{V_{in}} = \frac{A}{1+\beta A} = \frac{1}{\frac{1}{A}+\beta} \approx \frac{1}{\beta} \tag{4.4}$$

Typically the $\frac{1}{A}$ term is assumed to be negligable in amplifier design making the desired gain factor $\frac{1}{\beta}$ and the error factor $\frac{1}{\beta+A\beta^2}$ [11]. With digital correction this error would not matter unless the gain factor A changes with voltage, which it does. This is because digital correction will only measure the gain at the nominal voltage. This makes the difference in the closed loop gain from the nominal

voltage to the maximum voltage the critical design parameter that needs to be minimized.The change in closed loop gain is described by:

$$\Delta A = \frac{1}{\beta + A_{min}\beta^2} - \frac{1}{\beta + A_0\beta^2} = \frac{A_0 - A_{min}}{1 + \beta(A_0 + A_{min}) + A_0 A_{min}\beta^2}$$

$$\approx \frac{A_0 - A_{min}}{A_0 A_{min}\beta^2} \tag{4.5}$$

Where A_{min} and A_0 are the derivative of equation 4.3 evaluated at the nominal and maximum voltage levels. The appoximation made in equation 4.5 is allowed because dropping the extra terms makes the estimate of the error larger than the actual error resulting in a more conservative and mathematically easier approach. Substituting the derivative of equation 4.3 into 4.5 results in equation 4.6 which is the closed loop gain error as a function of the output. Only the results are shown here but a derivation using Maple is included in the appendix. Although the equation does not appear to be significant some important results can be concluded. The gain error is a group of functions of V_{out} denoted by Γ divided by the small signal open loop gain A_0. The implications of this are shown in Figure 4.5 where as the open loop gain gets larger the closed loop gain error as a result of changing output voltage gets smaller.

$$\Delta A = \frac{\sqrt{2 + 2\Gamma} - 2\Gamma}{2A_0\Gamma\beta^2} \tag{4.6}$$

$$\Gamma = \sqrt{\frac{V_{max}^2 - V_{out}^2}{V_{max}^2}}$$

Since the first stage will contribute the most error to the system then minimizing the gain error of the first stage is critical. With the results of equation 4.6 a good estimate on the required open loop gain can be made by putting the equation in terms of LSB's in the first stage. The only required

information is the maximum output voltage V_{max}, the full scale voltage $V_{out} = V_{fs}$, and the gain of the interstage amplifier $\frac{1}{\beta}$. It should be noted that these equations are rough estimates of the actual behavior but with some added safety margin the desired performace can be reached.

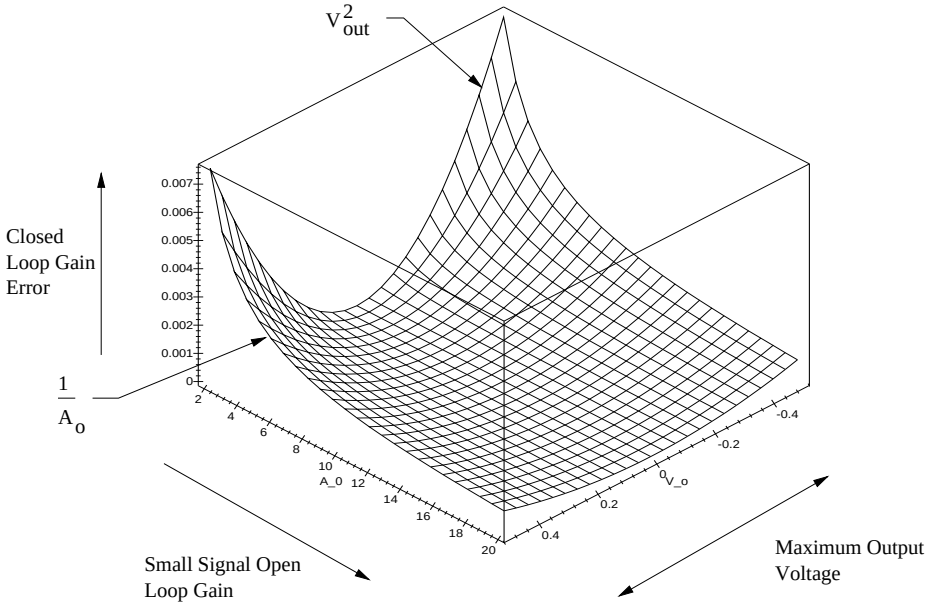


Figure 4.5: Plot of the closed loop gain error ΔA with changing open loop gain A_0 and output voltage.

4.4 Slew Rate

Another important design parameter for the interstage amplifier that can affect the output response is the slew rate of the op-amp. This is a quantity that describes the interaction of the bias currents of the op-amp and the capacitors that form the sample and hold . The transistors that form the output stage of an op-amp are (in a DC sense) current sources. The maximum current that they can supply is the DC bias current I_{bias} and the charge that they supply is $I_{bias}\Delta t$. Assuming that this charge is delivered to the sample and hold capacitor, the voltage change resulting from the charge will be $\frac{I_{bias}\Delta t}{C}$. With this approximation the bias current for an op-amp can be determined if it is known what voltage swing the output undergoes, in what amount of time, and what the load capacitance is as depicted in Figure 4.6.

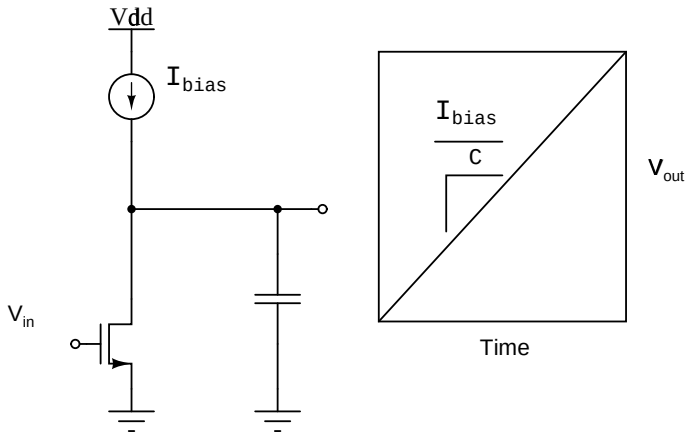


Figure 4.6: Bias currents and slew rate.

The simple approximation shown in Figure 4.6 can be used in more complex switched capacitor amplifiers such as the interstage amplifier of a pipeline ADC; the only difference is that the load is not a single capacitance but a group of capacitors. A simple method for estimating the necessary bias current from the required slew rate is to find the equivalent impedence and equate charges [12]. Figure 4.7 shows an interstage amplifier with the sampling capacitor connected in the feedback loop and driving the sampling capacitors of the next stage. Note that in the radix=2 1.5-bit per stage

architecture all capacitors are the same size. Looking at the schematic of the radix=2 1.5-bit per stage the equivalent impedence that must be met by $I_{bias}\Delta t$ is:

$$I_{bias}\Delta t = V_{fs}[C_{1B} + C_{2B} + \frac{C_{1A}C_{2A}}{C_{1A} + C_{2A}}] \tag{4.7}$$

The fact that all capacitors are the same size results in:

$$I_{bias}\Delta t = V_{fs}2.5C$$

This method is used to calculate the bias currents for a radix=2 converter but it can be applied to higher resolution converters as well. Additionally, if the input stage to the amplifier is large in comparason to the switching capacitors, the extra capacitance of the input stage should be included in the equivalent circuit.

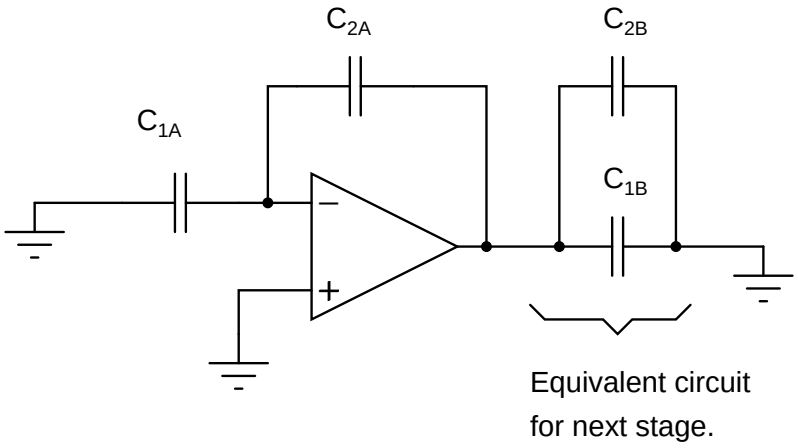


Figure 4.7: Equivalent circuit for determining bias currents to achieve a required slew rate.

4.5 Noise

The radix=2 1.5-bit per stage pipeline ADC with digital correction allows capacitor sizes in the sample and holds to be sized based on noise considerations rather than matching. This is because any matching errors that occur are corrected digitally. With noise being the lower limit on performance it is necessary to understand the effect of noise on the overall pipeline structure.

There is a sample and hold for each stage but each sample and hold does not contribute the same amount of noise. Considering the 12-bit converter again, the first stage would add KT/C noise to 12 bits and the second stage would add KT/C noise to 11 bits. Essentially the noise floor limitation is halved with each successive stage. The total noise of all sample and holds can be approximated by:

$$V_{rms} = \sqrt{\sum_{n=0}^{N-1} \frac{\frac{KT}{C_n}}{(2^n)^2}}$$

If it is desired that each stage contribute the same amount of noise to the system then the capacitor for each successive stage can be decreased by a factor of four. The advantage to this is that the bias currents can also be decreased by a factor of four. Another approach might be to keep every stage the same making the majority of the noise contributed by the first few stages. Comparing the two approaches, decreasing capacitor size by a factor of four saves power but results in the the most noise, and keeping the capacitor sizes the same results in the lowest noise but uses the most power. A compromise between the two is scaling down by a factor of two[2]. However, an added benefit of keeping all capacitors the same is that each stage is exactly the same, minimizing design time. This is the approach taken in this design. With this approach designing for one quarter LSB rms noise in the first stage ensures that one half LSB noise is contributed by the entire system.

To apply the previous result to the switched capacitor interstage amplifier it is necessary to know how noise is added in this structure. Referring to Figure 4.7 it is evident that the feedback capacitor C_{2A} and the input capacitor C_{1A} are both charged through separate sampling switches. This means

that the rms noise on each capacitor will be uncorrelated. Assuming that the effects add in rms fashion, this increases the total noise by a factor of $\sqrt{2}$.

4.6 Summary of Requirements

The previous discussion outlines some important design criteria for the interstage amplifier and sample and hold system in pipeline ADCs. When the design of the amplifier is attempted these criteria cannot be met on an individual basis because they are dependent on one another. For example: capacitor size in the sample and hold determines the slew rate, noise, and stability of the system. The results of this interaction is that the design of the interstage amplifier is guided by slew rate, noise, bandwidth, and stability. The optimal design for an interstage amplifier is reached by calculating the boundaries with the previous criteria and iterating within those boundaries until a suitable agreement between all is made. With this in mind a few op-amp architectures are discussed and how they fit within the ADC's operating boundaries.

Chapter 5

Op-Amp Architectures

5.1 Introduction

In all previous schematics the circuits have been shown as single ended implementations. The main reason for this is to aid in understanding pipeline concepts. For performance, those circuits were actually implemented differentially. The main advantage of differential over single ended signal processing blocks in integrated circuits is substrate noise immunity, wheras substrate noise in single ended circuits manifests itself as an offset, in differential circuits it results in a common mode signal that is for the most part rejected.

The most common integrated circuit differential amplifier is the operational transconductance or $g_m - C$ amplifier which is used to drive capacitive loads. It is characterized by its simplicity while still maintaining reasonably high performance qualities such as speed and gain. There are many different variations of fully differential $g_m - C$ amplifiers that are used depending on what performance parameters are most important. For this design a few basic op-amp architectures were considered as potential candidates for the interstage amplifier: the cascoded current mirror amplifier, the folded cascode, and the current mirror amplifier with active cascodes. The design aspects of each will now

be reviewed.

5.1.1 Current Mirror Amplifier

A current mirror amplifier is shown in Figure 5.1. Q9 and Q10 form a differential pair whose source load are the diode connected transistors Q2 and Q3. Because Q2 and Q3 are diode connected they form a low impedance node with an approximate value $\frac{1}{g_m}$, where g_m is the small signal transconductance [13]. Because they are diode connected the signal that is amplified in the differential pair is current, which is reflected to the output transistors Q1 and Q4 where the actual voltage amplification takes place. The current gain that occurs in the outer transistors Q1 and Q4 is equal to the current gain in Q2 and Q3 multiplied by the ratio of Q2 and Q1 or Q3 and Q4(The ratios are sometimes scaled to get more gain.) The current changing in Q1 and Q4 also changes in Q14 and Q16 which behave as non ideal current sources with high impedances. It is the impedance of Q14 and Q16 (as well as Q1 and Q4) that converts the changing current into a changing voltage. The specifics of these relationships will be developed more in the small signal gain section.

The range over which the cascode op-amp functions fairly linearly is defined by the common mode input and differential output range. It is described as fairly linear because none of the relationships are linear but can be approximated as linear in certain regions. Desired operation results when all transistors are in saturation, where they can be modeled as current sources with high impedances, versus the triode region where they behave as resistors with a linear voltage resistance relationship. Current sources are more desirable because higher voltage gain can be realized.

The boundary between the triode and the saturation region is determined by the drain-source voltage V_{ds}. Specifically, if $V_{ds} > V_{gs} - V_t = V_{eff}$ then the output transistors will behave like high impedance current sources. This limits the maximum differential output to $2(V_{DD} + V_{eff_p} - V_{eff_n})$. To achieve a more linear signal, differential output signals should be limited to something less than this. Since these amplifiers are configured in a feedback loop the differential input signal is often small and

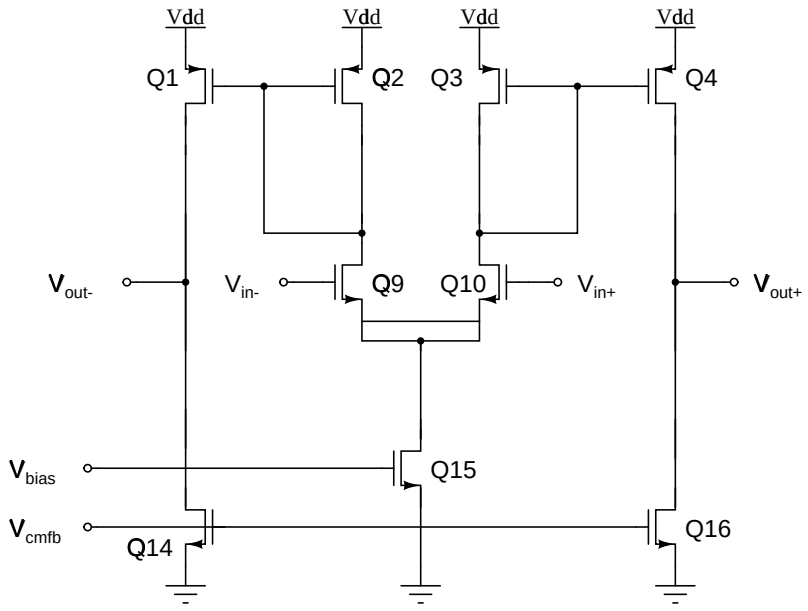


Figure 5.1: Current mirror amplifier.

it is not necessary to be concerned about different regions of operation. The common mode input voltage, however; is a different matter. Although it should stay around some nominal value it can vary because of noise and other limiting factors in the common mode feedback circuitry. For this reason it is desired to have a large common mode input range to minimize the effects of common mode signals. The maximum input signal is limited by the transistors Q2 and Q3 and is defined by $V_{DD} + V_{gs_p} - V_{th_n}$. The minimum input common mode voltage is limited by the differential input pair Q9 and Q10 and the current source Q15 and is defined by $V_{eff_n} + V_{gs_n}$.

5.1.2 Small Signal Operation

The voltage gain of the current mirror amplifier is all derived at the output since it is the only high impedance node in the circuit. This classifies the current mirror amplifier as a single stage amplifier, limiting its applications to driving capacitive loads only. The main reason for this is that a resistive load would lower the output impedance, potentially negating any voltage gain that occurred in the circuit. For this reason resistive loads are typically driven by an op-amp with two stages or more. Another point to note is the added complexity of the structure. The output stages are intended to increase the differential output swing while maintaining the same voltage gain. If a small differential output swing is desired then only a differential pair may be necessary. Figure 5.2 shows the small signal model for a transistor acting as a current source.

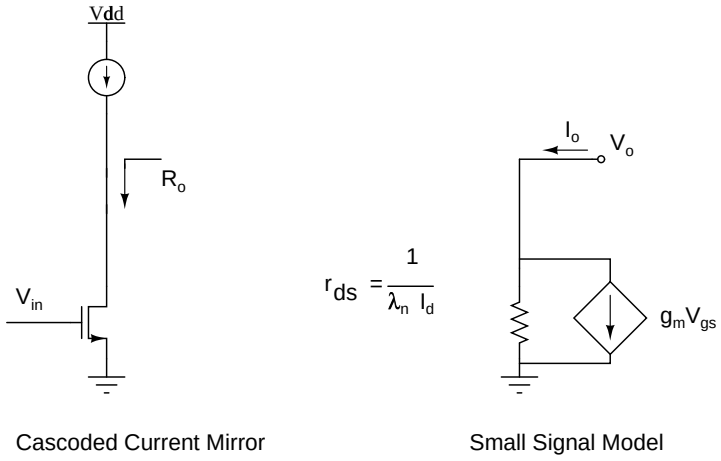


Figure 5.2: MOS transistor acting as a current source.

Using the small signal model of a MOS transistor, the open loop gain of the current mirror amplifier in Figure 5.1 is described by:

$$A_o = Kg_mR_o$$

where R_o is the parallel combination of the impedances of transistors Q1 and Q14 or Q4 and Q16.

The factor K is included to represent the current ratio from the differential pair to the output stage. Substituting expressions relating g_m and R_o to the bias current I_D results in[12]:

$$A_o = \frac{K\sqrt{2K_n\frac{w}{l}I_D}}{(\lambda_p + \lambda_n)I_D} = \frac{K\sqrt{2K_n\frac{w}{l}}}{(\lambda_p + \lambda_n)\sqrt{I_D}}$$

In other words, to maximize gain the bias current should be as small as possible, but the minimum current is dictated by the desired slew rate and noise generated by the output stage. The factor K could be increased indefinitely but is usually kept below five for stability reasons. This will be developed more in the next section.

5.1.3 AC Response

The most frequenty used method for op-amp compensation is to capacitively load one of the high impedance nodes in the circuit if possible incorporating the miller effect. The reason for this is to realize the most compensation for the smallest compensation capacitance (capacitance is area). For a simple RC circuit $f_{3db} = \frac{1}{2\pi RC}$. Although this relationship is modified somewhat in an active circuit the principle is still the same, to make f_{3db} small R and C must be large. Since the only high impedance node of current mirror amplifier is at the output, the load serves as the compensation making $f_{3db} = \frac{1}{2\pi R_o C_L}$ and the gain bandwidth product :

$$f_t = \frac{g_m}{2\pi C_L} \tag{5.1}$$

The first order principle of gain bandwidth product only applies if higher order poles do not interfere with the magnitude response until beyond f_t and as stated in the stability section of chapter 4, higher order poles can dramatically effect the phase response and thus the stability of the closed loop system. For this reason it is necessary to understand where higher order poles occur in the current mirror amplifier to minimize there interaction with both the phase and magnitude responses. Figure 5.3 shows a current mirror amplifier with the load capacitance C_L and the parasitic capacitances C_p at the gates of transistors Q1 and Q4. C_p is a combination of capacitances that occur at those nodes. The effect of the parasitic capacitances at the base of transistors Q2 and Q3 can add enough phase shift in the output to go beyond 180^o before any change in the magnitude response is seen. This is shown in the simulation of Figure 5.4. At 600kHz the dominant pole from the load starts to affect the magnitude response and by 6MHz, 90^o of phase shift has occurred. Also at about 6MHz the two secondary poles that occur because of the parasitic capacitance begin to interfere with phase response. The simulation shows the results of different capacitances at the base of transistors Q2 and Q3 with the minimum adding the least amount of phase shift. The best way to make a stable design is to minimize the capacitances that occur at those nodes through device size and layout.

As mentioned earlier, the current mirroring factor K also determines where the parasitic poles occur. As K increases, the current in the differential pair decreases, raising the impedance at the bases of Q2 and Q3. This has the effect of moving the parasitic poles closer to f_t (Assuming that g_m is kept the same).

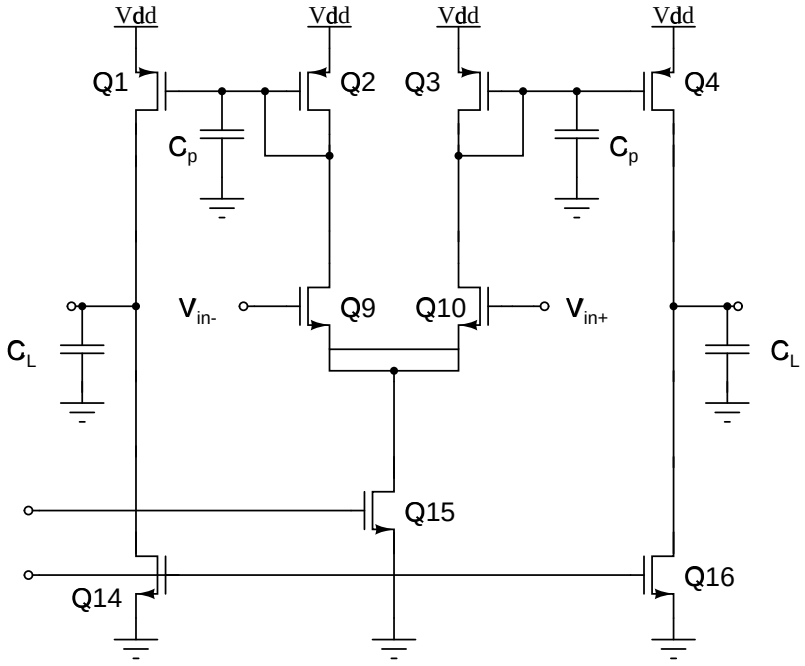


Figure 5.3: Current mirror amplifier with parasitic capacitances C_p.

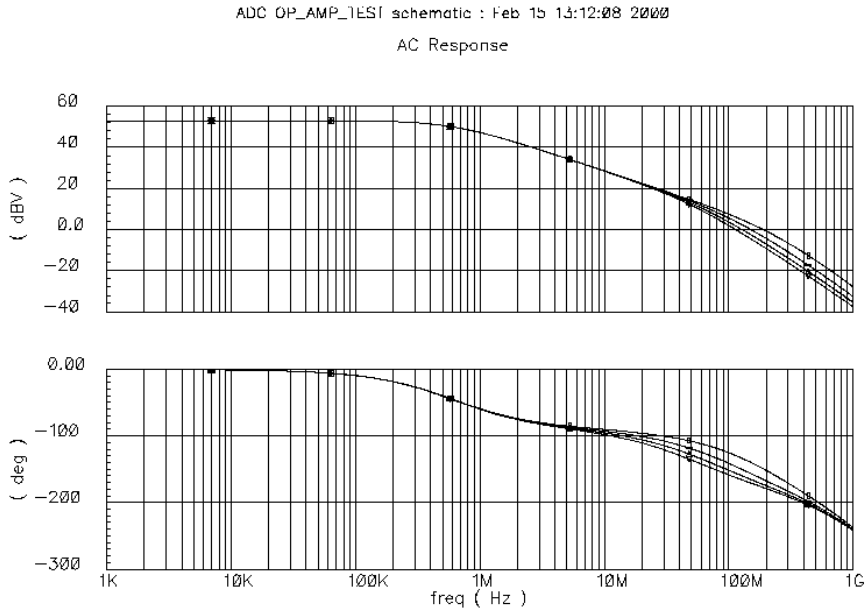


Figure 5.4: Simulation showing the effect of parasitic capacitances in a current mirror amplifier.

5.1.4 Cascoded Current Mirror Amplifier

With the ever growing need for faster transistors, channel lengths have been steadily deceasing and as a result the drain to source impedances decrease. Without getting into short channel and saturation velocity effects, r_{ds} is directly proportional to the channel length L. To overcome this limitation it is frequently necessary to use cascoded current mirrors such as the simple one shown in Figure 5.5. To calculate the small signal input impedance the simple relationship $R_o = \frac{V_o}{I_o}$ is used, resulting in:

$$R_o = [1 + r_{ds2} g_m + \frac{r_{ds2}}{r_{ds1}}] r_{ds_1} \approx r_{ds1} r_{ds2} g_m$$

The region in which this relationship is valid is $V_{in} > V_{eff1} + V_{eff2}$[1] [12].

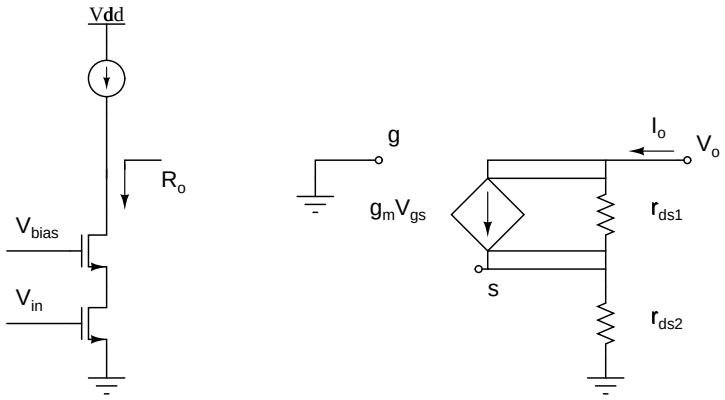


Figure 5.5: Cascoded current mirror.

Combining all of the elements previously mentioned results in the schematic shown in Figure 5.6 which differs only in the extra cascode transistors. Following the basic outline previously mentioned an op-amp with sufficient gain and stability can be obtained. The cascode transistors Q5, Q6, Q7, and Q8 allow the transistors Q1, Q2, Q3, and Q4 to be minimum channel length while still having

[1] V_{bias} must be chosen so that $V_{bias} > V_{gs} + V_{eff}$.

reasonable gain. As well as being faster, minimum size transistors have lower parasitic capacitance making the secondary poles as ineffective as possible. Additionally, the DC gain can be controlled by the cascode transistors Q5 and Q8. For symmetry the transistors Q11, Q12 and Q13 can be included but cascoded N-channel devices are not necessary. Usually the limiting factor are the P-channel devices. For speed Q11, Q12 and Q13 can be made with minimum length.

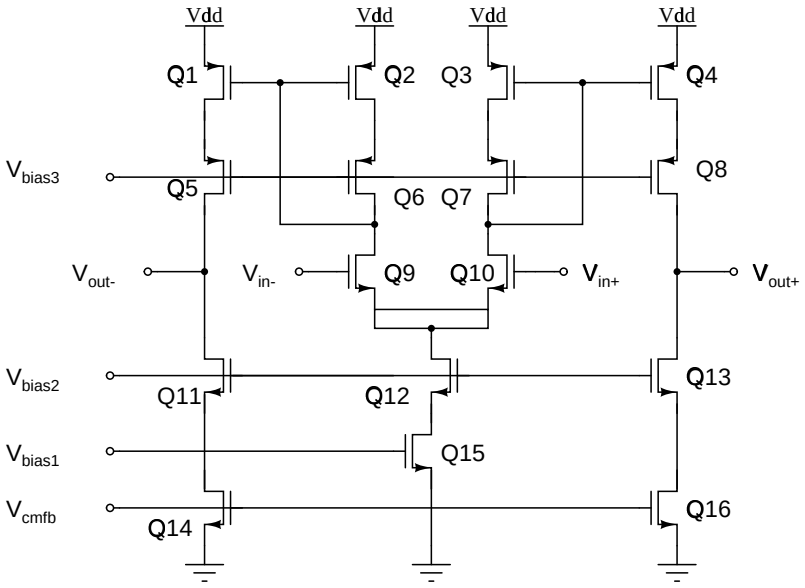


Figure 5.6: Cascoded current mirror amplifier

5.2 Folded Cascode

A variation of the cascoded current mirror amplifier is the folded cascode. The main difference between the simple cascoded amplifier and folded cascode amplifier is the current sharing between the output stage and the differential pair that takes place in the folded cascode. Figure 5.7 shows a typical schematic of a folded cascode amplifier. The currents in the differential pair and the output stage are both supplied by the current sources Q1 and Q2. Any current that changes in the differential pair Q5 and Q6 because of a differential input signal is subtracted or added to the output stage directly. Active devices are no longer used to mirror the current from the differential pair to the output stage.

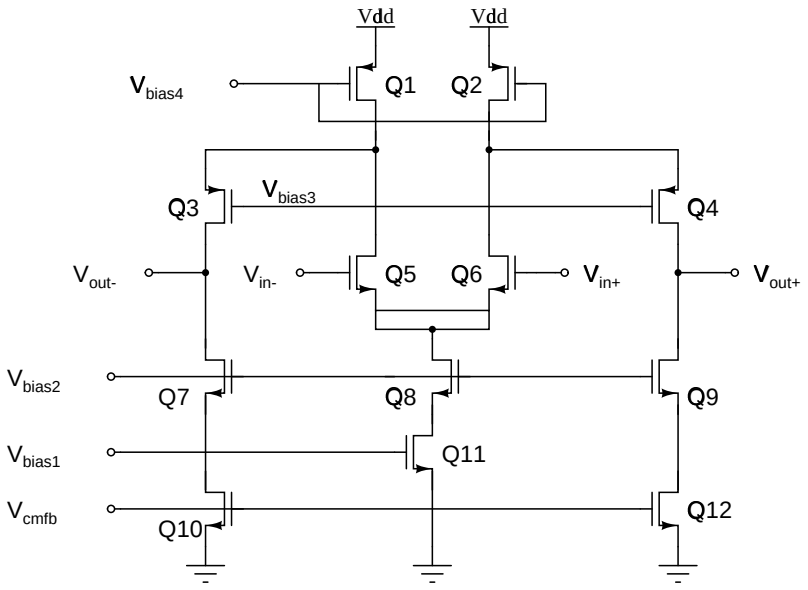


Figure 5.7: Folded cascoded amplifier.

5.2.1 Small Signal Operation

Just as with the cascoded current mirror amplifier, the voltage gain that occurs at the output is a result of the transconductance of the differential pair and the output impedance, or $A_o = g_m R_o$.

However, the output impedance R_o is different because of the structure. Figure 5.8 shows the small signal model for one side of the folded cascode. Using $R_o = \frac{V_o}{I_o}$ as before, the input impedance is found to be:

$$R_o = [1 + (r_{ds2}//r_{ds1})g_m + \frac{r_{ds2}//r_{ds1}}{r_{ds3}}]r_{ds_3} \approx r_{ds3(}r_{ds2}//r_{ds1})g_m$$

For similar devices this is a slight decrease in the output impedance from a regular cascode to a folded cascode. This has a direct effect on the open loop gain but the unity gain frequency stays approximately the same for the same load capacitance and g_m. Note that for greater gain the current source I_b should be cascoded as well.

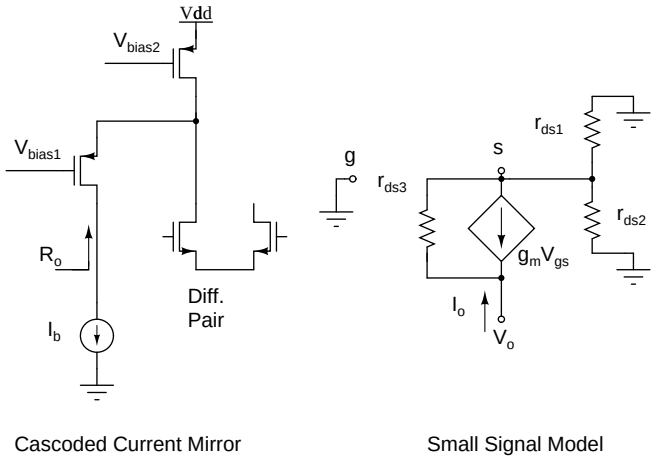


Figure 5.8: Small signal model of a folded cascode.

5.2.2 AC Response

The advantage of a folded cascode over a regular cascode is stability. Greater stability is obtained by the elimination of the parasitic poles at the base of transistors Q2 and Q3 in Figure 5.6 (the regular cascoded current mirror amplifier). This leaves the dominant pole from the load capacitance. Additional poles do occur but they lie at much higher frequencies on the magnitude and phase

spectrum. In this regard the folded cascode is somewhat more stable than a typical cascoded current mirror amplifier. Figure 5.9 shows a simulation of a folded cascode were the only thing that appears in the magnitude spectrum is a single pole roll off and in the phase secondary poles have little effect. The simulation shown has a phase margin of 80^o and a open loop gain of 48dB.

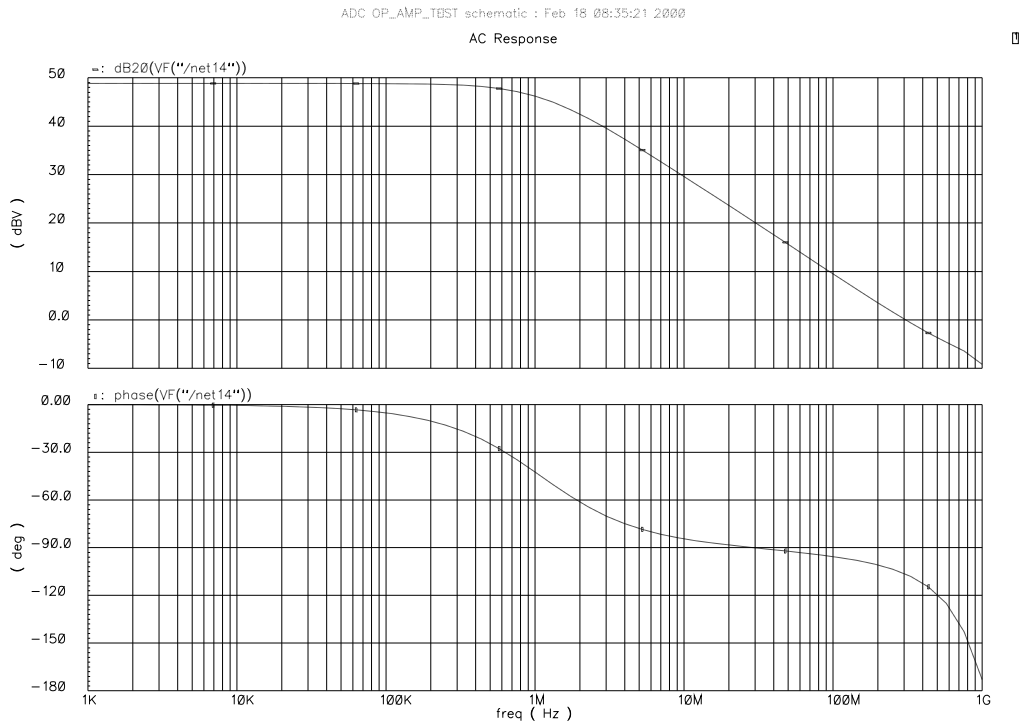


Figure 5.9: AC simulation of a folded cascode amplifier.

5.3 Cascode with Active Current Sources

Another variation of the cascoded current mirror amplifier is an active cascode amplifier. Typically the gate to source voltage in a cascode transistor changes with the output voltage because the gate is connected to a bias voltage and the source voltage changes as a result of r_{ds} of the other cascode transistor. With an active bias a feedback amplifier is used to keep the gate to source voltage a constant as possible. In this way the interaction of the r_{ds} of one cascode transistor has little effect on the gate to source bias voltage of the other cascode transistor.

Figure 5.10 shows an actively biased current mirror. Using the small signal models in the previous analysis the impedance is described by:

$$R_o = r_{ds1} r_{ds2} g_m (1 + A)$$

As an example, a device with $\frac{W}{L} = \frac{40}{0.6}$ in the HP 0.6μm process has an r_{ds} of 9K Ohms. Assuming a reasonable value of 500μA/V and A=5, the input impedence is 250K Ohms. As a comparason a device with $\frac{W}{L} = \frac{120}{3}$ only has an impedence of 200K Ohms and is significantly slower.

With active current sources high gains can be achieved but with many additional amplifiers. To make one current mirror amplifier with active current sources four additional amplifiers would be necessary. A simple MOS implementation is also shown in Figure 5.10. The feedback amplifier is made using a common source amplifier. Note that with this configuration the output is limited to within $V_{gs} + V_{eff}$ of the top and bottom rail. “Wide swing” active current mirror configurations can be used with added complexity [12].

Figure 5.11 shows a simulation of an amplifier with active sources. The phase margin is 45^o with 85dB of open loop gain. Note that f_t is about the same as the other configurations because f_t is only dependent on g_m of the differential pair as stated in equation 5.1.

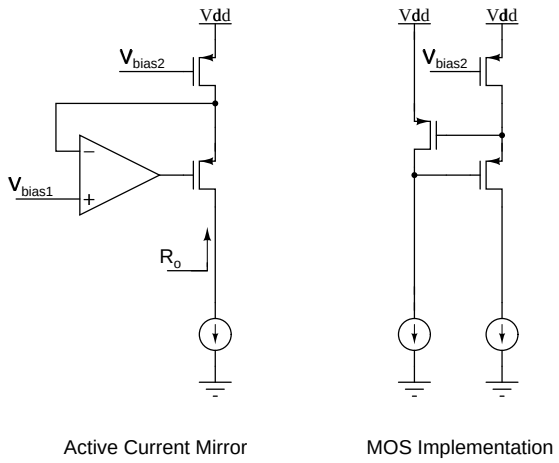


Figure 5.10: Active current source.

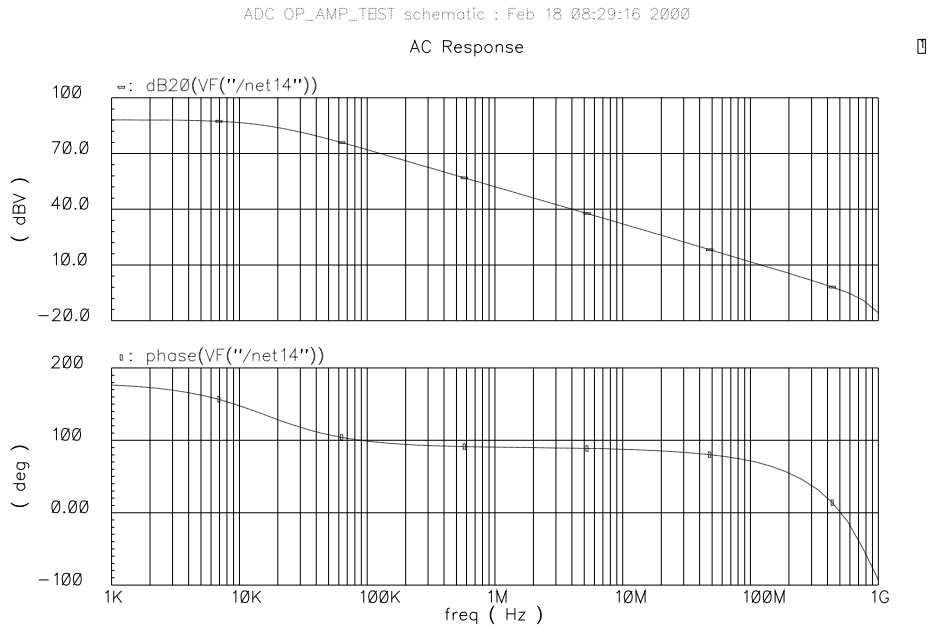


Figure 5.11: AC simulation of an actively cascoded current mirror amplifier.

5.4 Summary of Architectures

Figure 5.12 summarizes the performance of the different architectures discussed. The Folded cascode is in general not desirable because of its lower gain but if stability is an issue, it requires less compensation capacitance (thus area) to achieve stable operation. On the opposite end of the spectrum is the cascode amplifier with active current sources. With approximately the same speed as the folded cascode a large open loop gain can be obtained. In situations where a linear response and small error voltages are desired an active current source amplifier would be a good choice. The main drawback is a great deal of active circuitry is required, consuming much more area and power. A good compromise between the two is a cascoded $g_m - C$ amplifier. It is characterized by average gain, reasonable stability, and good speed. Additionally, it requires minimal extra circuitry.

Regardless of what scheme is chosen the design procedure is the same. Set g_m based on the desired bandwidth and design current sources based on the necessary open loop gain and slew rate. Device sizes are selected based on bias currents and the required output range. Finally, a simulator is used to refine any problems that occur because of secondary poles.

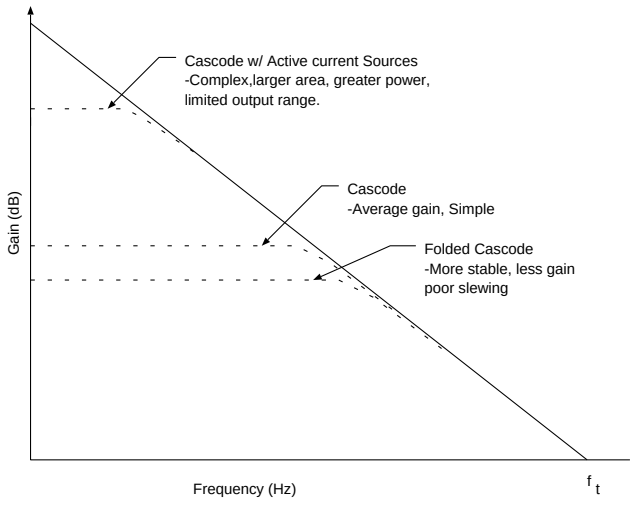


Figure 5.12: Summary of Architectural Performance

5.5 Common Mode Feedback

The design of a $g_m - C$ amplifier is straightforward as shown in the previous sections, but the design of a common mode feedback circuit to control fully differential amplifiers is usually the most difficult part. The purpose of the common mode feedback is to sense the common mode voltage from the differential output and control the bias levels in the amplifier to get the desired common mode output voltage. The difficult part is rejecting large differential output voltages (sometimes rail to rail) while still sensing small common mode changes. If the common mode voltage on the output is to high or low it may cause differential signals to clip. Generally the desired common mode voltage is half way between the supply rails to get the largest swing out of the differential circuits. Usually the differential signal rejection range in the common mode feedback circuit sets the limit on the largest differential output voltage, not the bias levels in the amplifier itself.

5.5.1 Continuous Time Common Mode Feedback

Several different schemes have been used to control the common mode voltage, the most common continuous time circuit is shown in Figure 5.13. It uses two differential pair, one for each output of a fully differential amplifier, to compare the outputs to some common mode voltage. If the voltages are equal at the inputs of each differential pair, then the current I_b from each source will split exactly in half and then recombine in transistors Q7 and Q8. In the presence of a differential input signal, the current in Q3 will be $\frac{I_b}{2} + \Delta i$ and the current in Q6 will be $\frac{I_b}{2} - \Delta i$, which sum in transistor Q7 to be I_b. In a similar fashion, the current in Q4 will be $\frac{I_b}{2} - \Delta i$ and the current in Q5 will be $\frac{I_b}{2} + \Delta i$, which also sums to I_b. If the common mode voltages are not the same the current will not divide in transistors Q7 and Q8 evenly and the common mode feedback voltage will change, changing the bias currents in the amplifier.

The sharing of current in the feedback circuit is dependent on the transistors Q3, Q4, Q5, and Q6 all operating in a linear fashion, which is not always the case. The AC linear model for a MOSFET

is the derivative with respect to a small input voltage, but with large differential input signals, it is difficult to have all transistors operating in the same region. In fact, the small signal model does not apply and the behavior of the two differential pairs is governed more by the large signal model in equation 5.2.

$$I_d = \mu_p C_{ox} \frac{W}{L} (V_{GS} - V_t)^2 = \mu_p C_{ox} \frac{W}{L} V_{eff}^2 \tag{5.2}$$

For all transistors to be operating in the same region the gate to source voltage V_{GS} must be greater than the threshold voltage V_t, as well as being in the active region. With this restriction the maximum differential input voltage can be found. Assuming that V_{in-} is as low as possible and V_{in+} is a high as possible, $I_{Q3} = I_B$, $I_{Q4} = 0$, $I_{Q5} = I_B$, $I_{Q6} = 0$. Noting that the gate to source voltage at $I_{ds} = 0$ is V_t, the maximum differential input is $2V_{eff}$. In other words, the effective voltages of the differential pairs should be maximized to increase the differential output range. The upper limit on the effective voltages is set by transistors Q1, Q2, Q7, and Q8. The effective voltages of the differential pairs must be made small enough to fit within $V_{dd} - V_{effQ1,Q2}$ and $V_{GS,Q7,Q8}$.

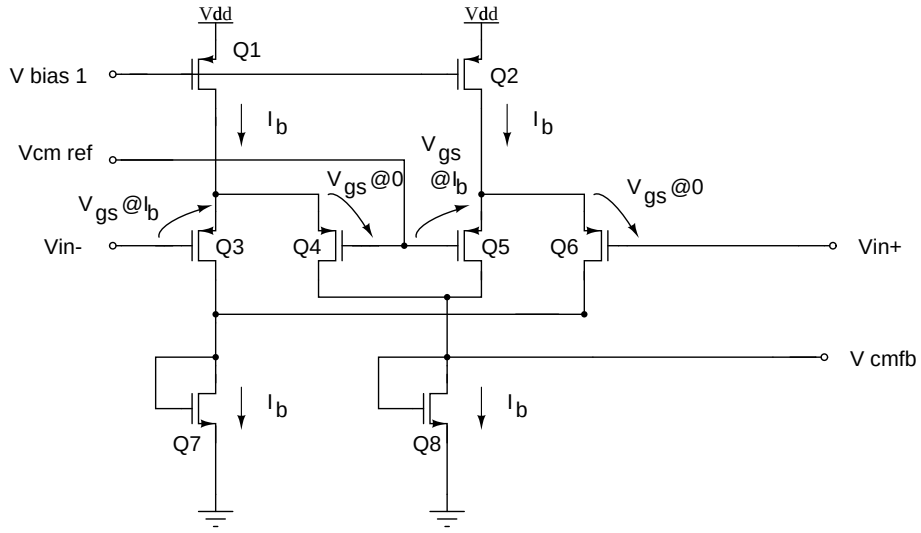


Figure 5.13: Continuous Time Common Mode Feedback Circuit.

The stability of this type of feedback structure is another limiting factor that must be considered in the design. In Figure 5.13, transistors Q7 and Q8 are not connected in a standard single ended output. Instead they are both diode connected significantly lowering the gain. This is necessary because of the instability of the common mode feedback network. Because of the complexity of the common mode feedback and amplifier working together, it is usually necessary to optimize the design using a simulator to achieve stability.

5.5.2 Switched Capacitor CMFB

More recent developments in common mode feedback circuits have incorporated the use of switched capacitor circuits for improved linear range. Variations on the overall topology are numerous but the basic concept is to sample the two differential outputs on separate capacitors, and in the second phase average the charge to get the common mode voltage. With a little additional circuitry the common mode voltage can be used to control the bias levels in the amplifier. The advantage gained through switched capacitor circuits is a linear relationship over the entire operating range.

A switched capacitor common mode feedback circuit is shown in Figure 5.14. This circuit is only useful when the amplifier itself is being used in a switched capacitor application because the outputs are disabled during the sampling phase θ_1. During this time the charges on the output capacitors are averaged and fed into an amplifier for comparison to the desired common mode voltage. Any desired correction of the error is put back on the output capacitors and the circuit goes into the next cycle of operation [16].

A problem with switched capacitor common mode feedback implementations is that pattern noise at the sampling frequency is injected into continuous time signals. For this reason, a high precision amplifier would not be appropriate for switched capacitor common mode feedback implementations. However, in some applications where mixed mode clocking signals already exist on chip, the advantages gained (such as increased linear output) might be worth the trade off.

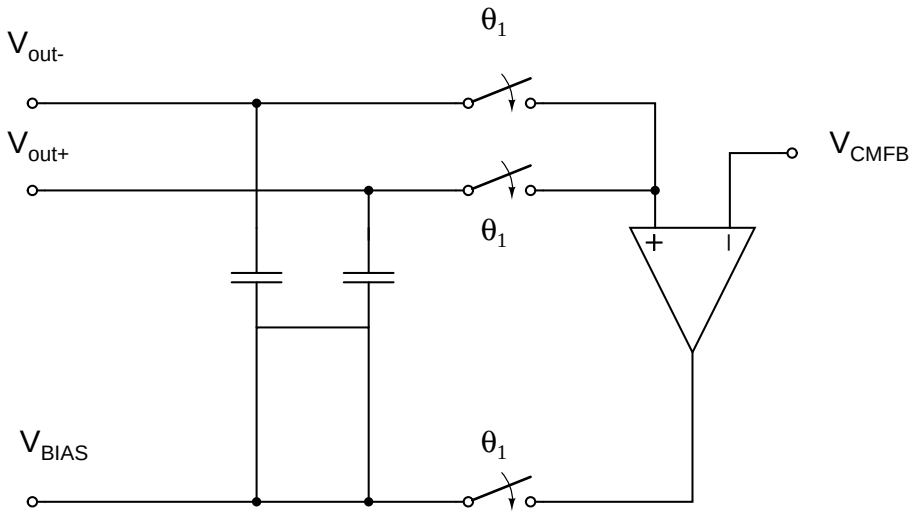


Figure 5.14: Switched Capacitor Common Mode Feedback.

5.6 Test Chip Amplifier

The architecture used in the test chip was the cascoded current mirror amplifier with continuous time common mode feedback. This choice was made because of the small area required while still maintaining sufficient performance. The final schematic and layout used are shown in figures 5.15 and 5.16. Specific performance criteria met by this design were obtained through simulations and are summarized in Table 5.1.

Criteria	Value	Remarks
Open Loop Gain	58dB	Fully differential
f_t	400MHz	1pF Load
f_{3dB}	600kHz	1pF Load
Linear Output Range	$\pm$1 V	For error < 1/4 LSB
Slew Rate	500V/μS	1pF Load
Supply	+5V	
Bias Current	1.9mA	
Phase Margin	45^o	1pF Load

Table 5.1: Simulated performance of test chip amplifier.

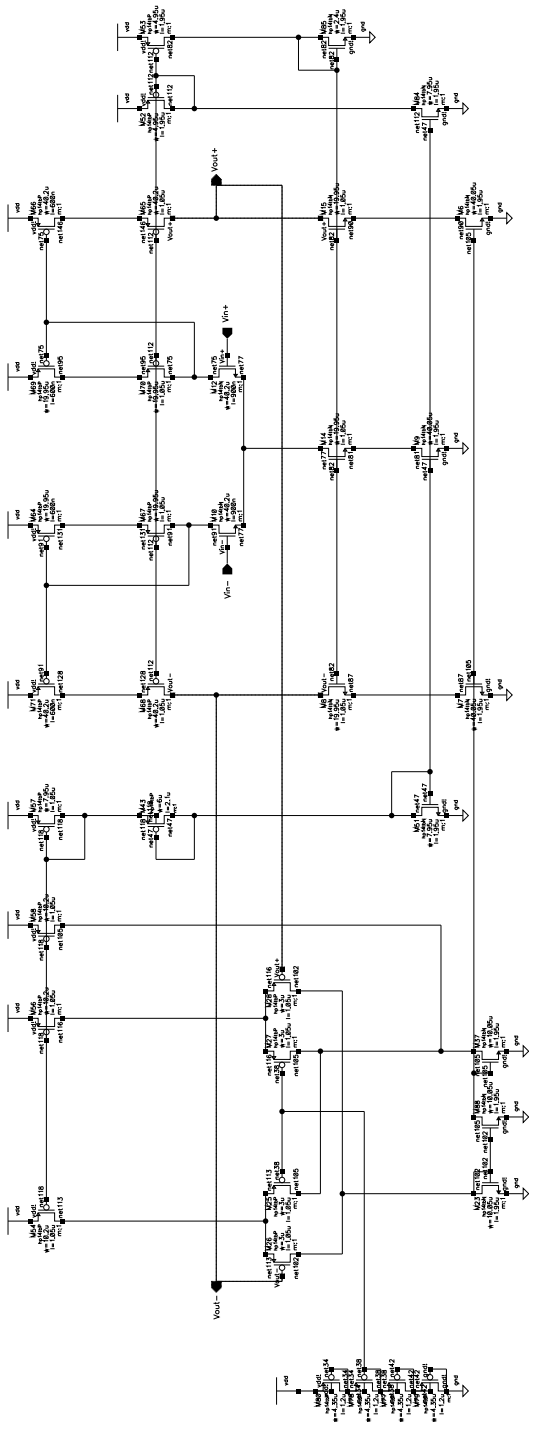


Figure 5.15: Schematic of test chip amplifier.

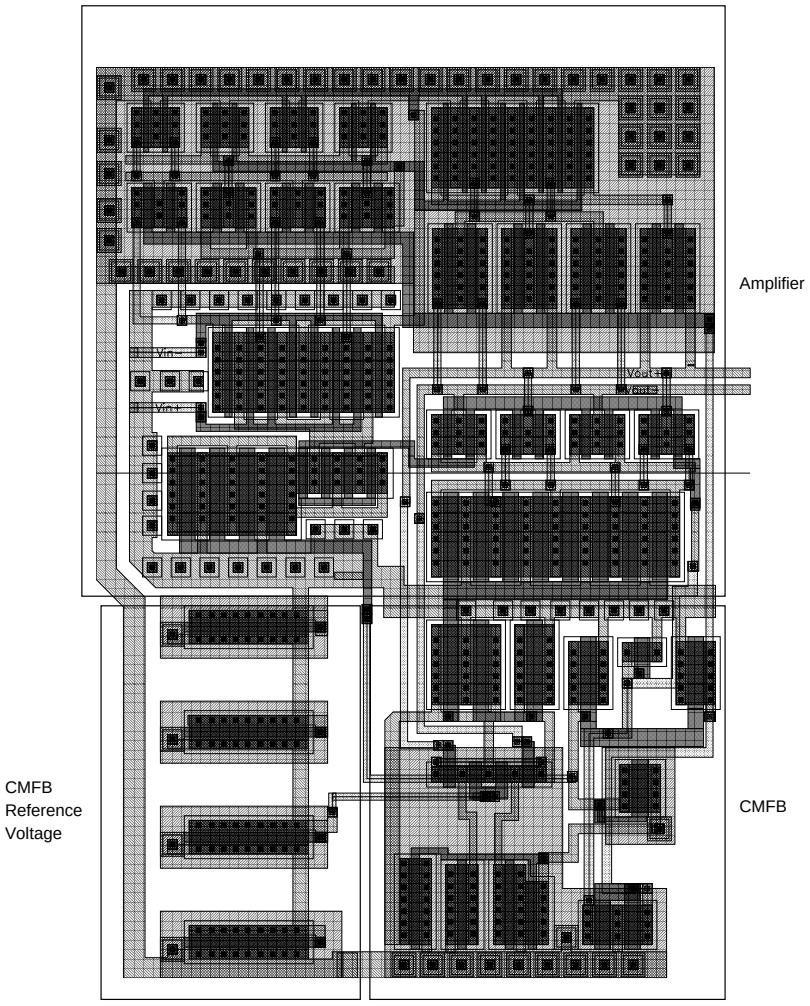


Figure 5.16: Test chip Amplifier Layout

Chapter 6

Comparator Architecture

To make the rough decisions for each stage of the pipeline converter two comparators are necessary. Due to the inherent correction in the pipeline with digital overlapping scheme, the threshold accuracy requirements are minimal. As discussed earlier an error up to $\frac{V_{ref}}{4}$ can be made in the 1.5 bit per stage architecture without any consequences on the output codes. The two most important performance requirements are speed and flashback both of which are now discussed.

6.1 Flashback

Most converters require some sort of latching of the output to take place so that the value at a particular time can be read. Another thing gained from latching is that it allows the comparator to be reset completely before going on to the next value. In this way residue charges from one value to the next do not interfere with one another. At the moment the latching takes place, nodes in the circuit can move as much as the supply rail in very little time. Because of the nature of the the structure parasitic charges can get injected from the circuit itself to the input at the moment of latching, adding error to the signal being processed. This is commonly referred to as flashback.

The explanation of flashback can best be seen in the circuit in Figure 6.1 which is a latched comparator. The center part of the comparator forms the latching circuitry. It has speed limited only by the process. The transistors Q4 and Q5 form source followers to buffer the input signal before it goes into the high gain setup stage. The buffering helps to minimize flashback during latching. Any charge that is to be injected into the input signal during latching must first transmit through the parasitics of the gates of Q6 and Q4 as well as Q7 and Q5. Since Q4 and Q5 form source followers they can be small (assuming that they have sufficient matching) minimizing flashback even further. A simulation which shows flashback is included in Figure 6.2. In this case the flashback is of the order of 500μV peak.

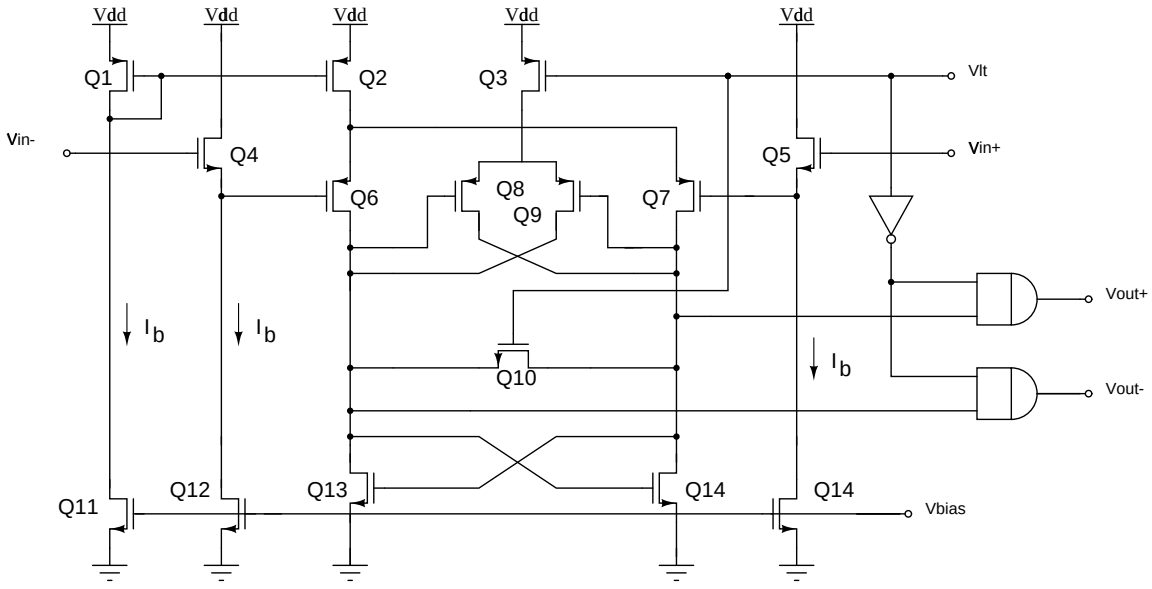


Figure 6.1: Latched Comparator

6.2 Reference Voltage

The reference voltage is generated through an intentional mismatch in the input circuitry of the comparator. The inaccuracies associated with this are high but can be tolerated because of the

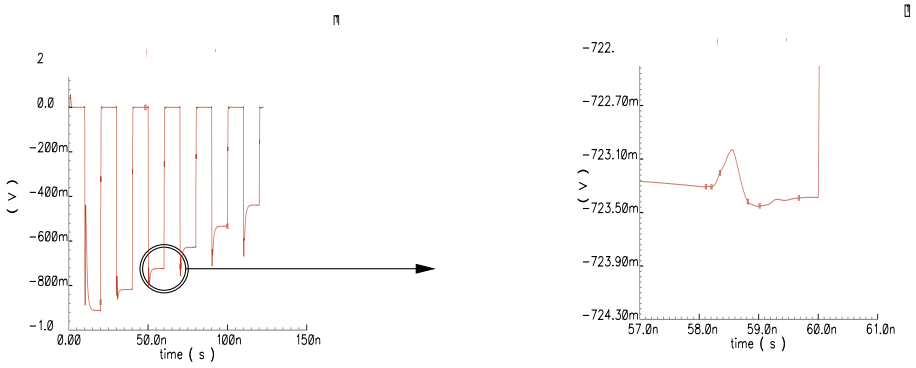


Figure 6.2: Flashback

decision offset correct-ability inherent in the overlapping scheme. In the physical implementation the offset occurs in the source followers Q4 and Q5, where one transistor has an effective voltage $\frac{V_{ref}}{4}$ greater than the other. Obviously the reference voltage will change from comparator to comparator as well as being process dependent. Another drawback is that the reference voltage will not track with other voltages changing in the circuit. In spite of all of these negatives it is still believed that because of the robustness of the digital correction, internally generated offset voltages will be sufficient.

6.3 Test Chip Comparator

The schematic used in the test chip is the same as that shown in Figure 6.1. The test chip comparator layout is shown in Figure 6.3. The critical components such as the latching stage (transistors Q6 Q7 and Q8 Q9 in Figure 6.1) have been place as close as possible for the best matching. The transistor sizes are also kept as small as possible to keep flashback to a minimum.

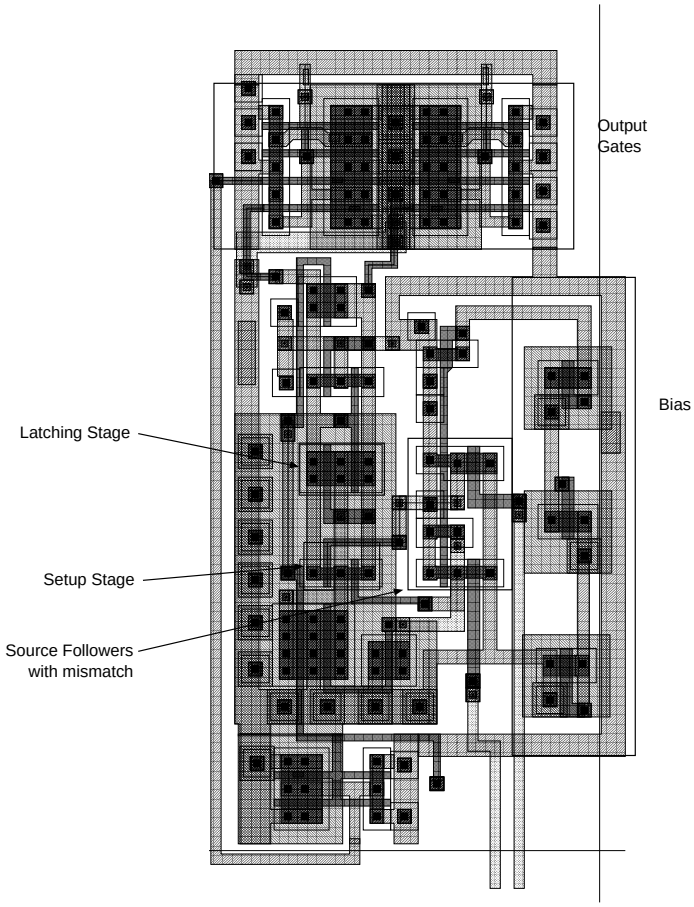


Figure 6.3: Test chip comparator layout.

Chapter 7

Digital Architectural Requirements

Most of the previous discussion has focused on the requirements of the analog processing blocks. This is because it is difficult to make devices that have high bandwidth and linear response. The requirements of the digital components are much less stringent because the process being used is optimized for digital circuitry. With gate delays as low as 200ps most of the digital processing required for each stage can be completed easily in one half a clock cycle. In this chapter the overall block diagram will be discussed and the function of each component will be explained. The minimal speed requirements allowed each digital component to be constructed from widely used circuits with little or no extra modification.

7.1 Control Signal Generation in the Analog Processing Block

To implement each pipeline stage, several additional control signals are required to determine the operating mode. Those modes are: offset measurement, gain measurement, and normal operation. Normal operation includes three sub-states for the decisions of the two comparators (00, 01 10).These control signals are decoded and sent to switch drivers that do the actual switching inside

the analog block. The switching structure used in the test chip is shown in Figure 7.1 and Table 7.1 helps to clarify the groupings of switches and how they are used in each operating mode.

All modes of operation are depicted in the timing diagram in Figure 7.2. In normal operation all capacitors sample the input in the first phase. In the second phase the capacitors C_{1A} and C_{1B} are switched in feedback while C_{2A} and C_{2B} are connected to $\pm V_{ref}$ or virtual ground depending on the decisions of the comparators which is analogous to the single ended implementation. In the first phase, α is measured by sampling V_{ref} on capacitors C_{1A} and C_{1B} while C_{2A} and C_{2B} are completely discharged. During the second phase C_{1A} and C_{1B} are again connected in feedback while C_{2A} and C_{2B} are connected to $-V_{ref}$. And finally, in the first phase for offset measurement all capacitors are discharged, and in the second phase C_{1A} and C_{1B} are connected in feedback, and C_{2A} and C_{2B} are connected to virtual ground.

Switch Group	Switches	Remarks
S_{GND}	θ_{GND}	Occurs in the first half of all phases
S_{FB}	θ_{FB}	Occurs in the second half of all phases
S_{C2_GND1}	θ_{C2_GND1}	01, Offset
S_{C1_GND}	θ_{C1_GND}	Offset
S_{Vin}	$\theta_{C2\_Vin}, \theta_{C1\_Vin}$	Sample input
$S_{C2+Vref}$	$\theta_{C2+Vref}$	Add V_{ref}
$S_{C2-Vref}$	$\theta_{C2-Vref}$	Subtract V_{ref}
$S_{C1+Vref}$	$\theta_{C1+Vref}$	α
S_{C2_GND2}	θ_{C1_GND2}	Offset, α

Table 7.1: Switching of fully differential amplifier.

The schematic used to create the switching signals from the four inputs is shown in Figure 7.3. To simplify as well as optimize the speed all cases were not considered in the decoder (For example, alpha and offset should never be asserted at the same time and could not in the physical design) One item to note about the control block is that all signals are processed first and then ANDed with the clock or /clock signals. This assures that all transmission gates are switched at the same time as well as helps to eliminate glitches. Additionally, Comparator decisions are made 2.5ns before the edges of the clock signals to leave enough time for the decision logic to set up.

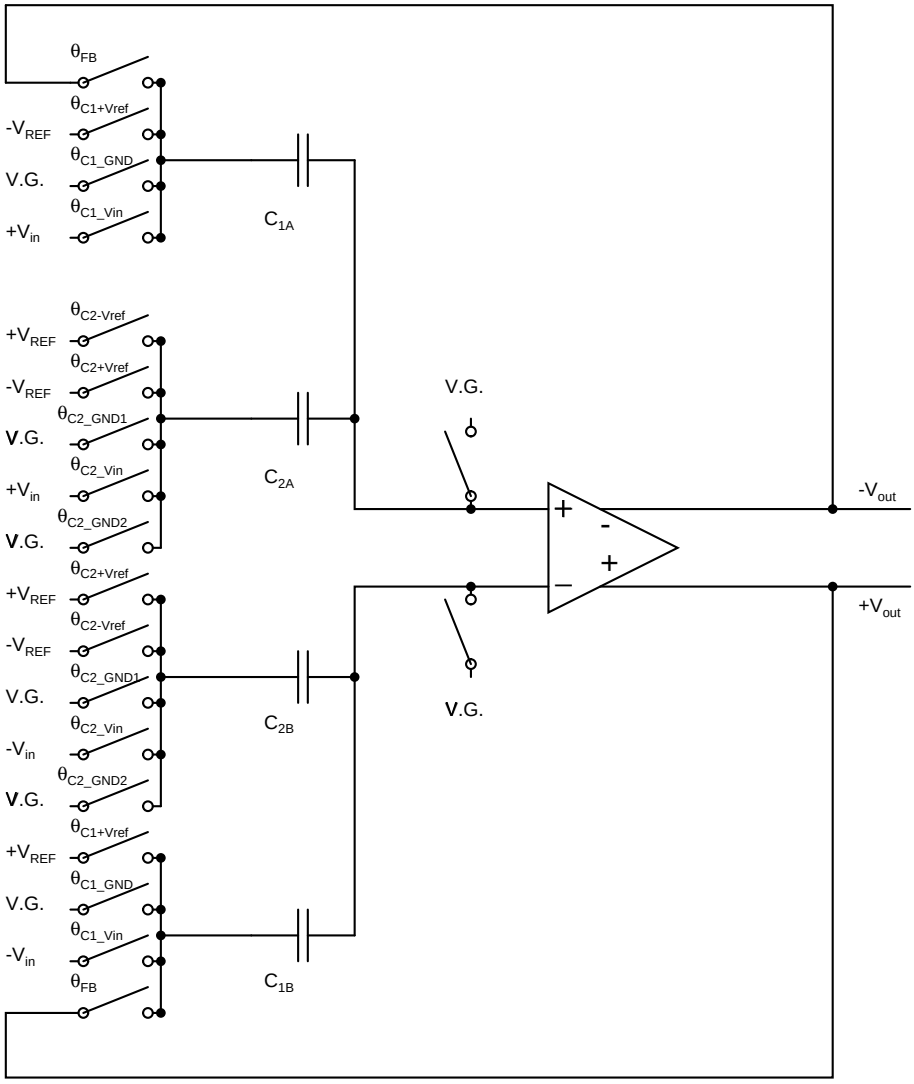


Figure 7.1: Fully differential switched capacitor amplifier.

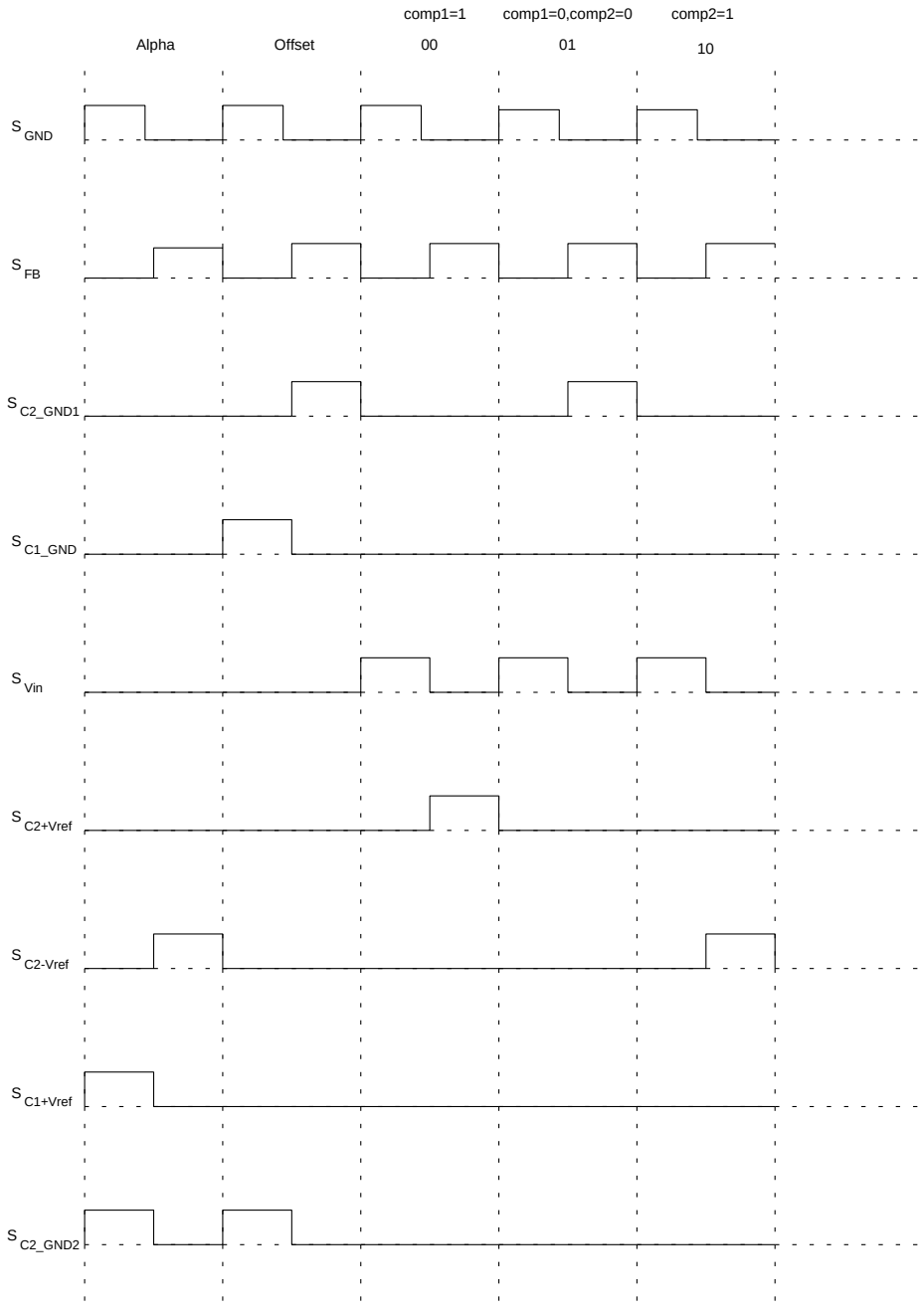


Figure 7.2: Timing Diagram for digital control of analog processing block.

In certain cases (α and offset measurement) it is necessary for the capacitors C_{1A} and C_{1B} to be tied to virtual ground during both halves of the clock cycle. In order to use the same transmission gates, additional logic involving both clock signals would be required, causing switching of those gates to occur some time after the clock edges (not synchronized with the other gates). To solve this problem additional transmission gates were included that connect capacitors C_{1A} and C_{1B} to ground during the first half of the operating cycle. An added benefit of using separate transmission gates is that the switching behavior is closer to ideal. During the offset measurement mode, capacitors C_{1A} and C_{1B} are connected to virtual ground during both phases. If only one set of gates was used, no switching would take place from the first to second phase, and the charge injection that would take place in normal operation would not be measured. With two sets of gates, switching between phases would inject charge that would occur during normal operation yielding a more accurate reading.

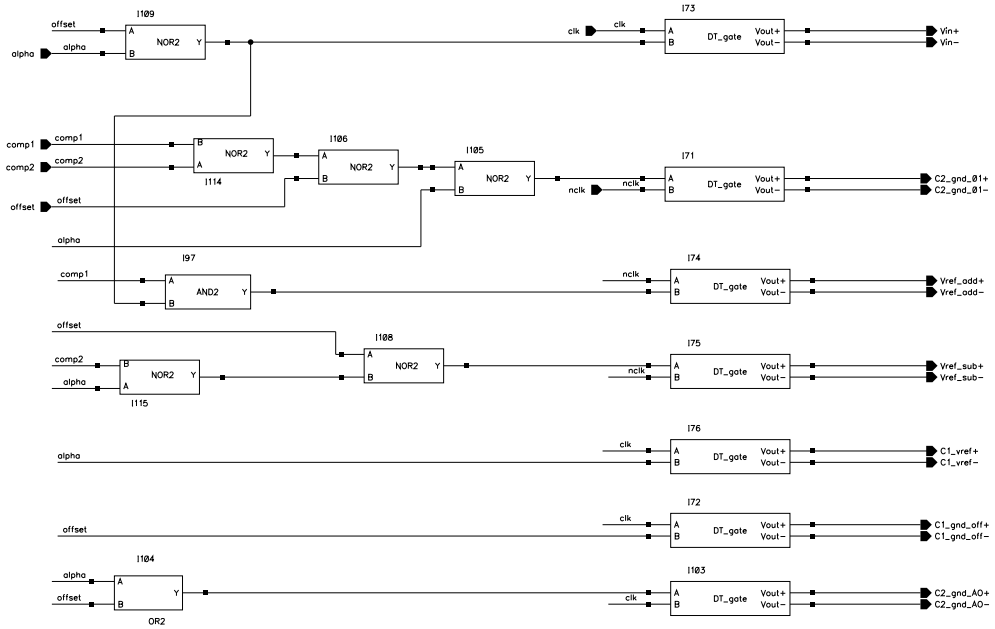


Figure 7.3: Switching Decoder.

The layout of the control block is shown in Figure 7.4. The shape is result of the space the control block fits in. A post layout simulation is included in Figure 7.5 as a verification of operation.

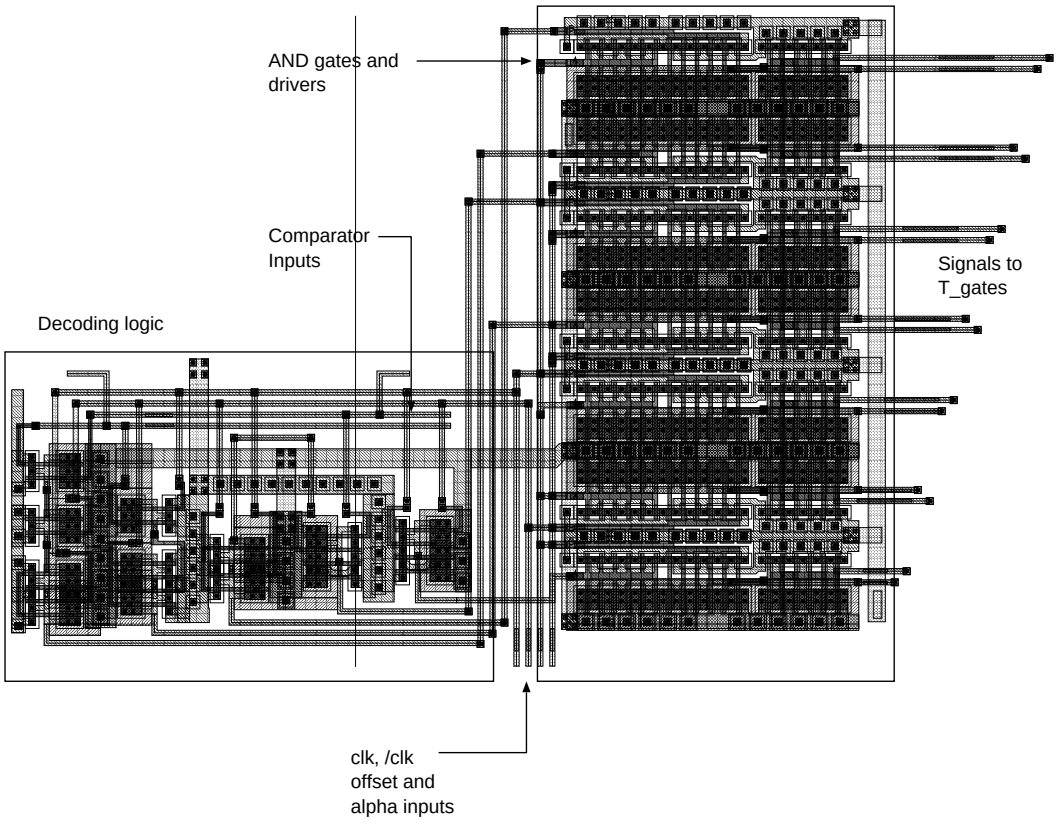


Figure 7.4: Switch Decoder Layout.

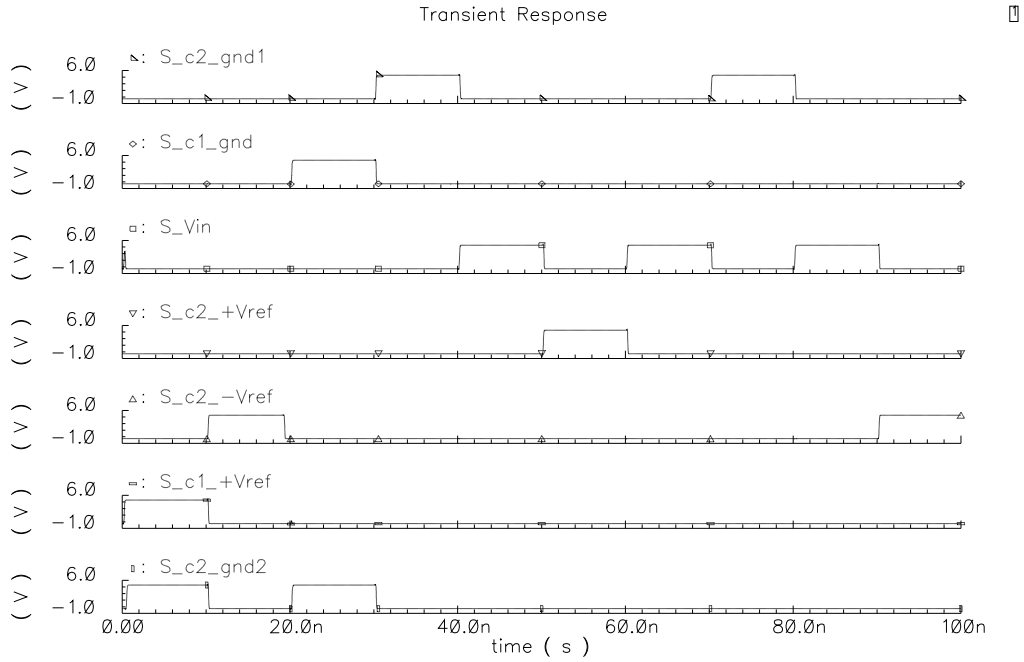


Figure 7.5: Post layout simulation of digital controller for analog processing block.

7.2 Digital Pipeline

The remaining components of the digital pipeline, shown in Figure 7.6, are: coefficient storage register, multiplexer, pipeline storage register, and adder. The coefficient storage register $A[i]$ stores the calculated value for each stage and can be loaded independent of the pipeline. The multiplexer takes the two bit decision level from the comparators in the analog processing block and either sends all or the value in the coefficient register to the adder, half, or none. The adder is required to add the value from the mux into the pipeline. And finally the pipeline register latches the value from the previous adder in the pipeline. With this brief introduction the timing, structure, and simulations of each component are presented.

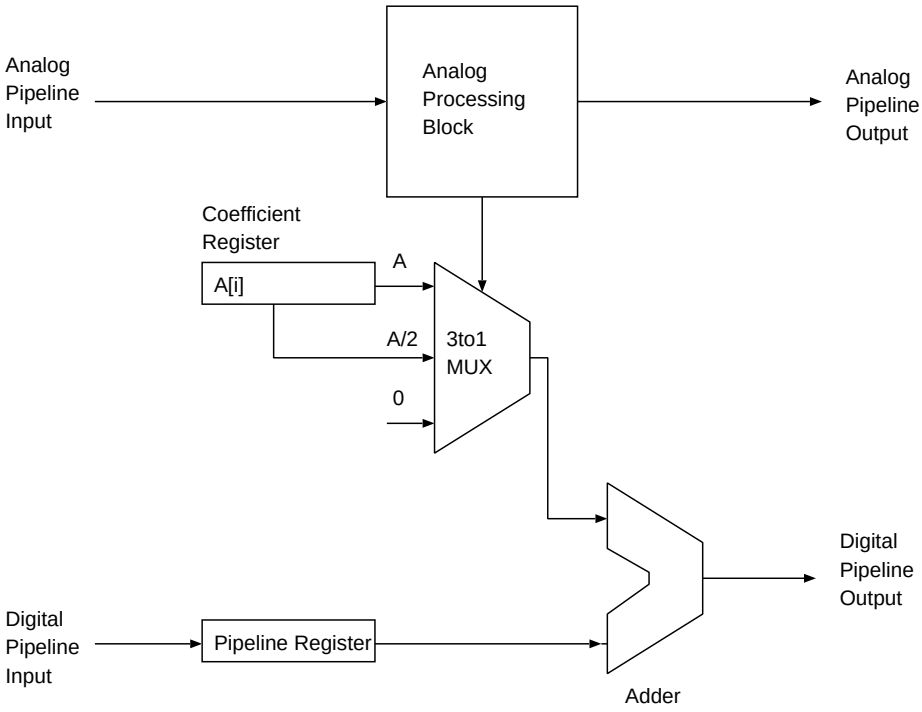


Figure 7.6: Pipeline stage block diagram

7.2.1 Coefficient and Pipeline Registers

All data storage in the ADC was implemented with D flip flops. Although flip flops require more area to store large amounts of data, the total amount of memory in the ADC is small making flip flops more economical than complex structures that might require refresh circuitry. Another added benefit to using D flip flops for the storage registers is that they can be reused for the pipeline registers shortening layout time. The schematic for and edge triggered D flip flop is shown in Figure 7.7 [14]. It is constructed out of two D latches making it edge triggered.

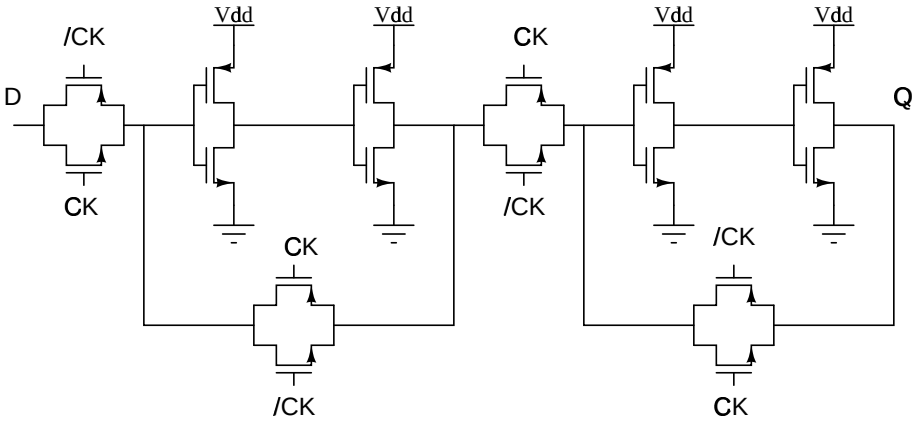


Figure 7.7: D Flip Flop

The layout for a D flip-flop is shown in Figure 7.8

7.2.2 Multiplexer

The multiplexer is constructed out of 16 smaller 3-to-1 multiplexers added together to make a 48-to-16 mux. The three inputs to the mux are: the input taken directly from the coefficient storage register, the input from the coefficient register shifted by one bit (giving the division by two), or 0. The schematic for the 3-to-1 mux is shown in Figure 7.9 and the layout in Figure 7.10.

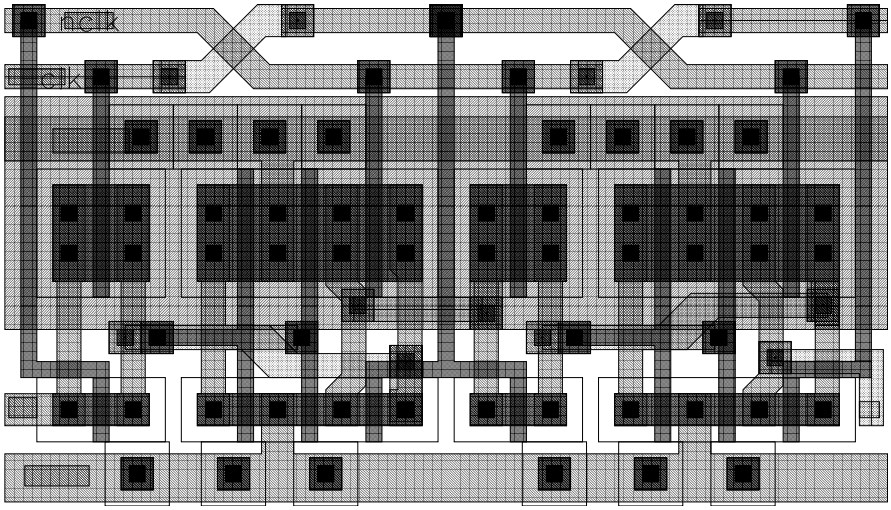


Figure 7.8: D flip-flop layout.

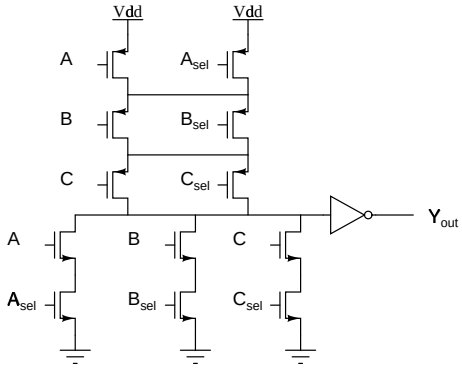


Figure 7.9: 3-to-1 Multiplexer

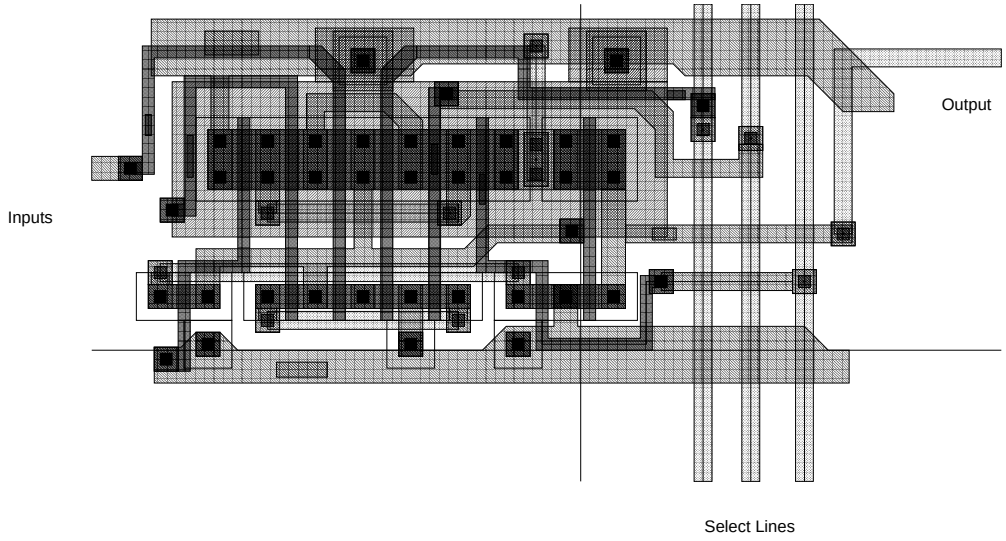


Figure 7.10: Layout of a 3-to-1 Multiplexer

7.2.3 Ripple Carry Adders

The most commonly known and easiest adder to implement is the ripple carry adder. The circuit diagram for a single full bit adder is shown in Figure 7.11 [14]. Fourteen bits of addition has take place in each stage of the pipeline in less than a half of a clock cycle(10ns). This would require 13 full adders and one half adder. The worst case for the addition of sixteen bits would require one sum out t_{so} and 13 carry outs t_{co} . The sum out and carry out times are for 50% rise times because that is when the inputs to the next gate begin to switch. From the simulations shown in Figure 7.12, the total time required for 14 bits of addition is about 3.5ns which is much less than 10ns (half a clock cycle at 50MHz). With this safety margin a sufficiently fast adder should result, even with a change in the switching threshold due to process variations.

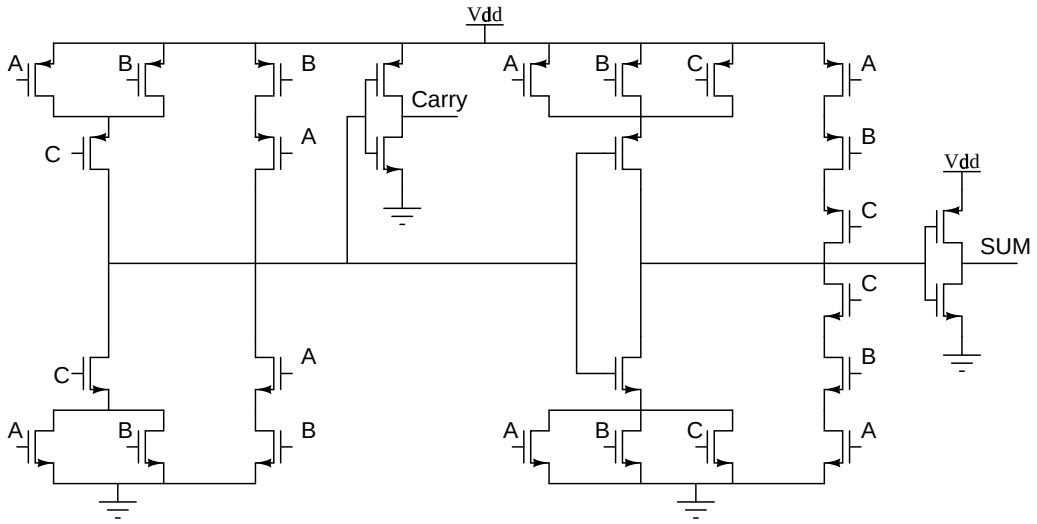


Figure 7.11: Ripple Carry Adder

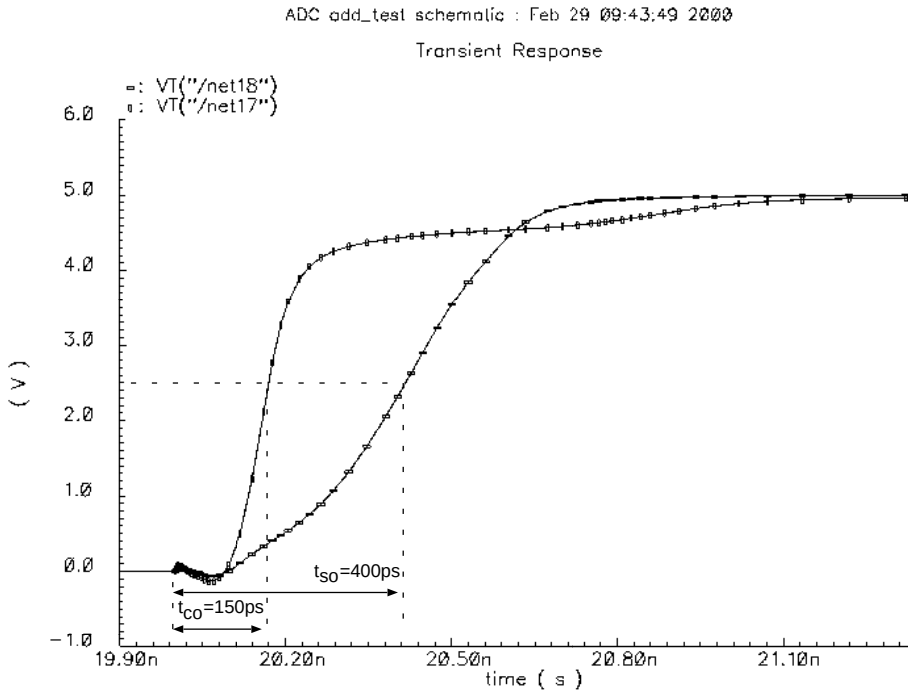


Figure 7.12: Simulation of a Ripple Carry Adder in a 0.6μm technology.

7.3 Clock Generation

A drawback to pipelined ADCs with switched capacitor circuits is that they require many non-overlapping clock phases. Several clock phases are needed to ensure the proper switching between stages as well as a separate clocks for the comparators to latch. Additionally, the main clock needs to be as close to a 50% duty cycle as possible to avoid pattern errors caused by different settling times between stages. However with clock speeds approaching 50MHz, an off chip system clock with an accurate duty cycle is an impractical requirement. The division of a faster clock is a common method for fixing all of these problems which is what was used in this project. The details of which is now presented.

The different clocks required are shown in Figure 7.13. Two out of phase clocks are required by the analog processing block so that while one stage is sampling the previous stage is creating the output. Likewise, two out of phase latching signals are required for the comparators to make their decisions in each stage. The rising edge of the latch signals occur at the same time as the rising edge in the clock signal, however the falling edge of the latch occurs slightly before the edge of the clock. This is necessary for the comparators to make their decisions before the stage actually changes state, giving the decoding circuitry in each state enough time to set up.

Also included in Figure 7.13 are the three clock signals used to derive the latch and the stage clock signals. Clock B is clock A divided by two, and clock C is clock B divided by two. Since clock A (200MHz) was available in the testing environment it was not necessary to bring a slower clock on chip and use PLL multiplication. This saves design time and complexity.

Figure 7.14 shows the block diagram of the clock generator. There are two D flip-flops used as division elements to create signals B and C. Each signal path has additional delay elements to put all signals back in phase after the division, followed by decoding logic to create the final clock signals. A post layout simulation was performed to verify operation and is shown in Figure 7.15.

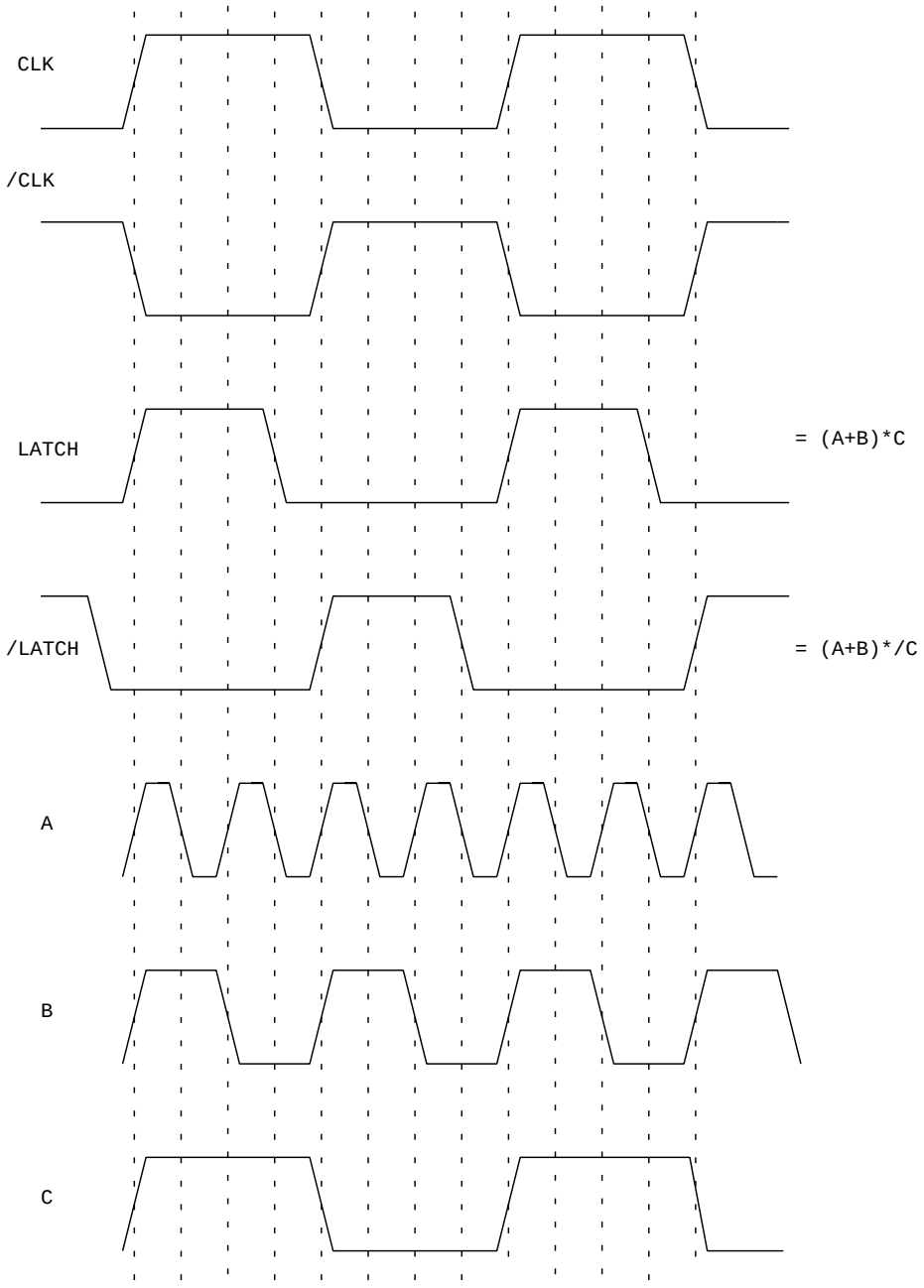


Figure 7.13: Pipeline Clocks

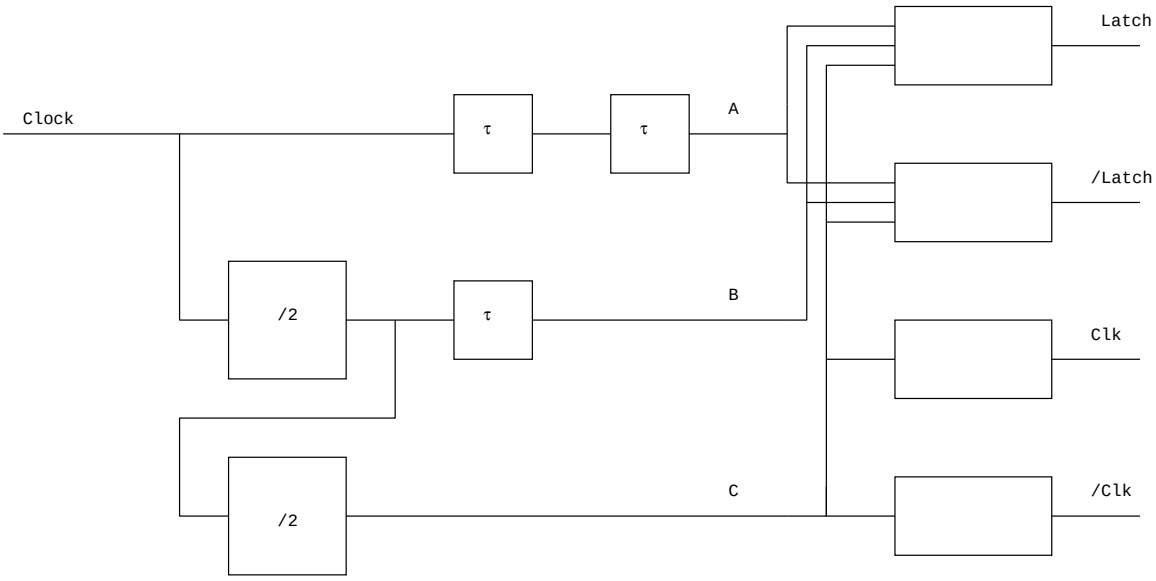


Figure 7.14: Block diagram of clock generator.

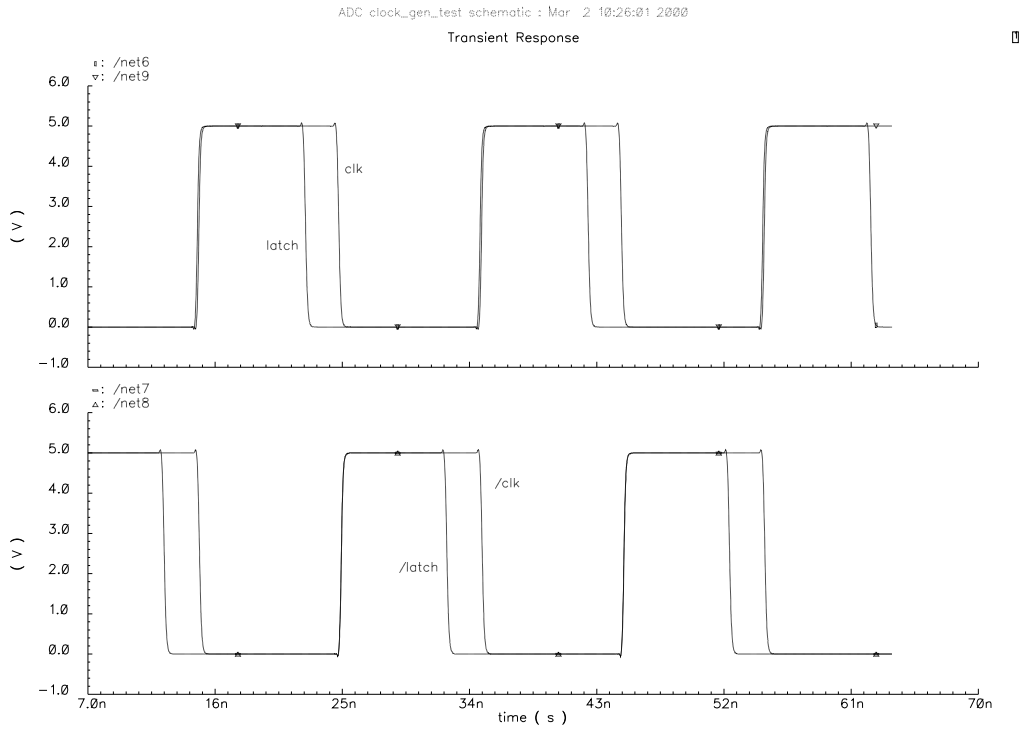


Figure 7.15: Post layout simulation of clock generation.

Chapter 8

Top Level Design

This chapter talks about how all of the parts in the digital and analog architecture sections work together as well as how they fit together in the top level layout. Simulations are included where possible.

8.1 Analog Processing Block

Figure 8.1 is the complete schematic for the analog processing block which includes the amplifier, digital control block, and comparators. The column of boxes labeled T_gate are complementary transmission gates driven by the digital decoder. The remaining transmission gates around the amplifier are made of NMOS transistors only because the voltages are well below the supply voltage. The block has seven input signals and two output signals. They are: clk, /clk, latch, /latch, Vin+, Vin-, alpha, offset, Vout+, and Vout-.

The layout of the analog block is shown in Figure 8.2. All supply and reference voltages are located at the top of the layout. All substrate noise producing sources such as the digital control block and the comparators are situated at the bottom of the layout and a guard band of substrate contacts

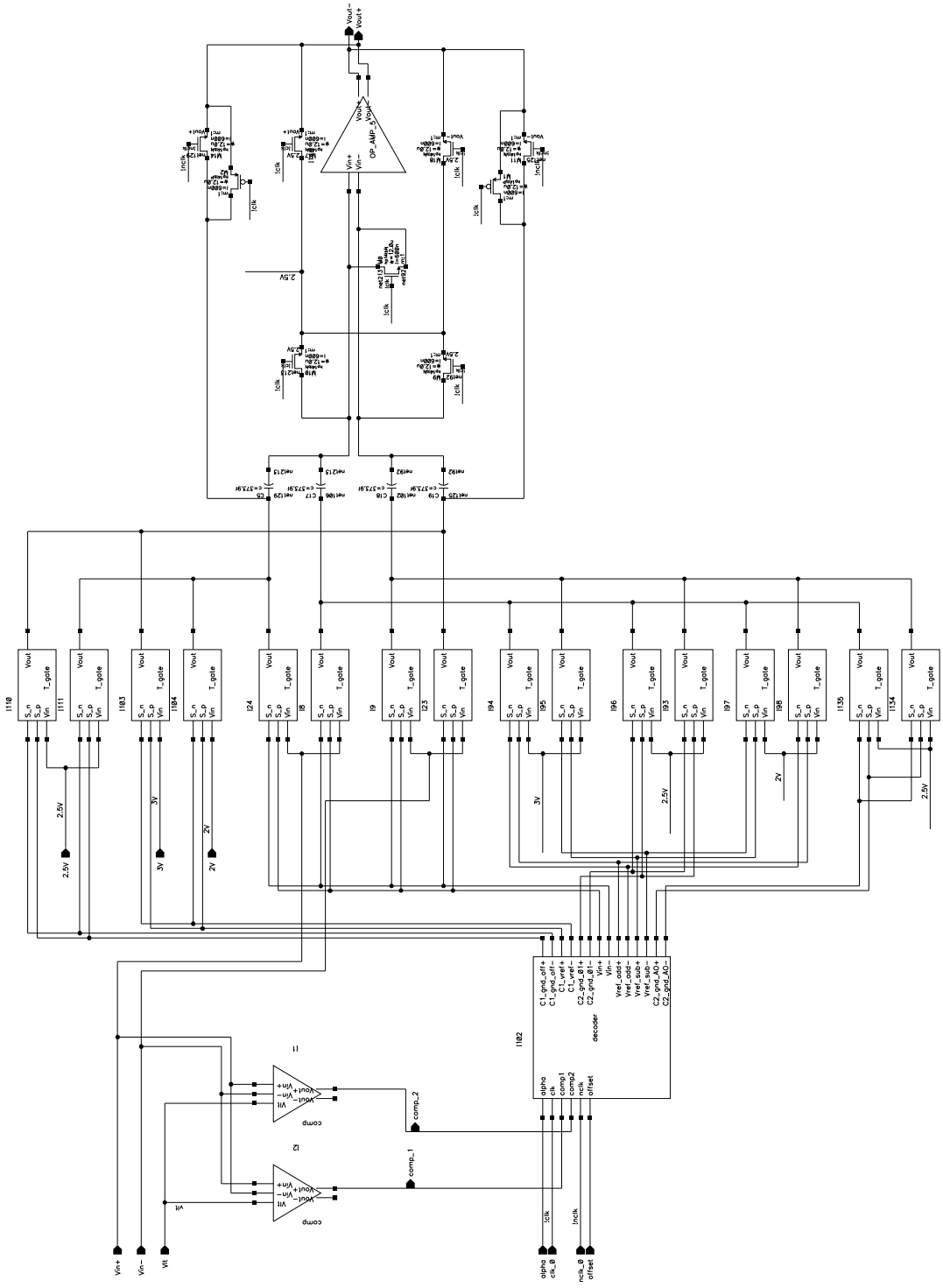


Figure 8.1: Analog block schematic

surrounds the amplifier and capacitors. An additional heavy guard band is located at the bottom of the layout to shield the analog block from the digital pipeline which is located directly beneath it (shown later in the top level layout). After the guard band is a bus for control signals.

A post layout simulation of the analog block is included in Figure 8.3. It is equivalent to the residue plots that have been shown in previous chapters except that the input is represented by the time axis because of the switched capacitor nature of the output. The input is a ramp from -1.0V to 1.0V with the time axis scale representing 0.25V per 100ns. As expected the comparators change at 300ns and 500ns (-0.25V and 0.25V).

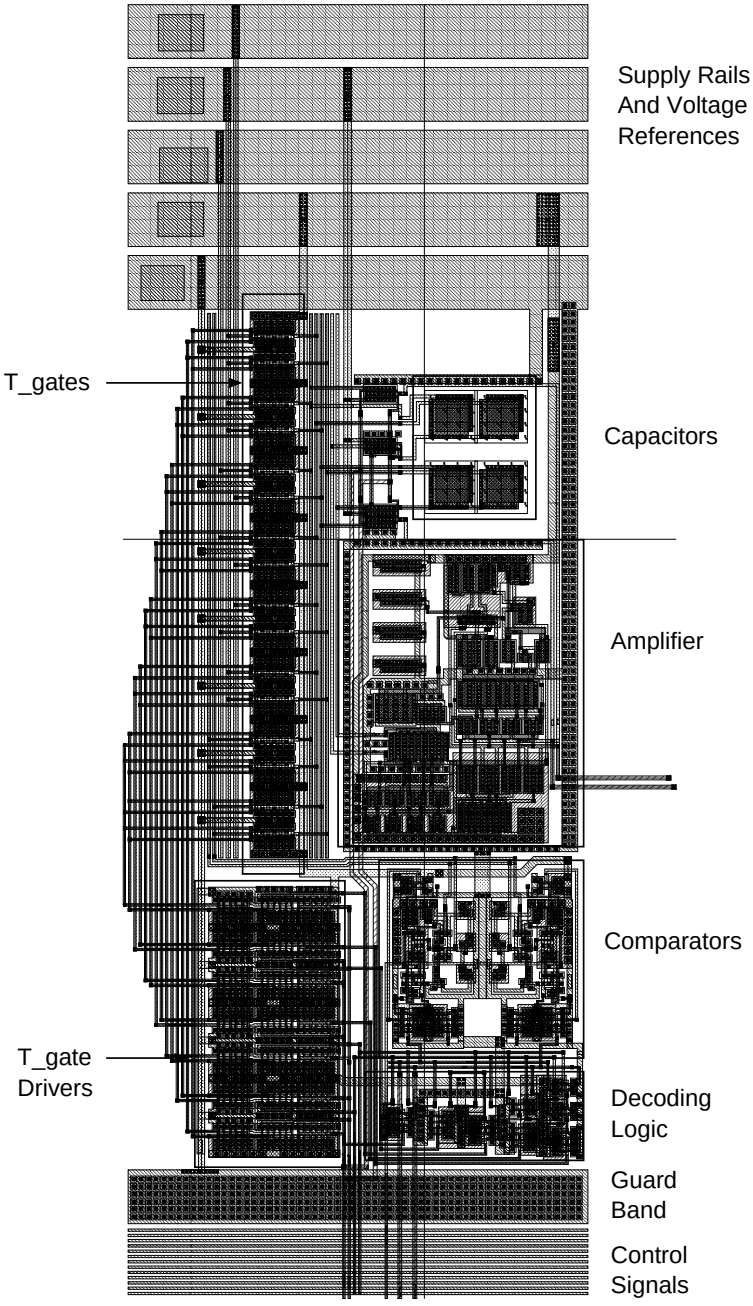


Figure 8.2: Analog block layout.

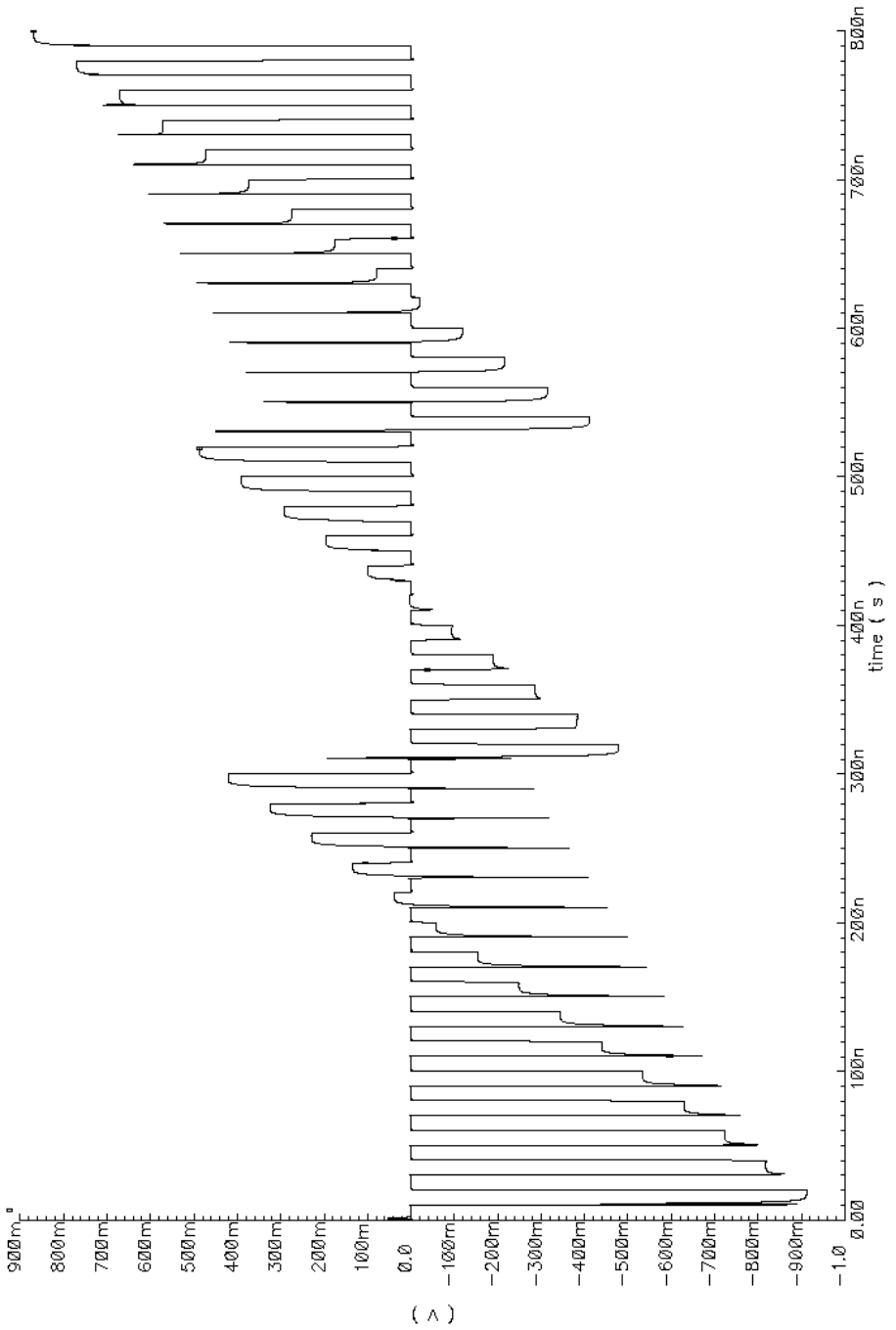


Figure 8.3: Single stage residue plot.

8.2 Digital Block

The full schematic for the digital block is shown in Figure 8.4. There are two registers to latch the comparator decisions from the analog block. The outputs are NORed together to form the three decisions levels and then sent through buffers to drive the multiplexers. Two additional groupings of registers store the coefficients for each stage and latch the outputs from previous stages. The registers that form the pipeline feed into the ripple carry adders directly. The other group of registers feed the 48 to 16 mux and then the mux feeds the other input to the ripple carry adders. Two signals write_p and write_n feed the clock signals of the registers to form a latching mechanism to load coefficients into the stage (shown on the bottom of the schematic.)

The layout of the digital block , shown in Figure 8.5, is almost exactly as depicted in the schematic. There are two columns of registers, a column of multiplexers, and finally the ripple carry adders on the output. The comparator latches and multiplexer drives are located on the top of the layout.

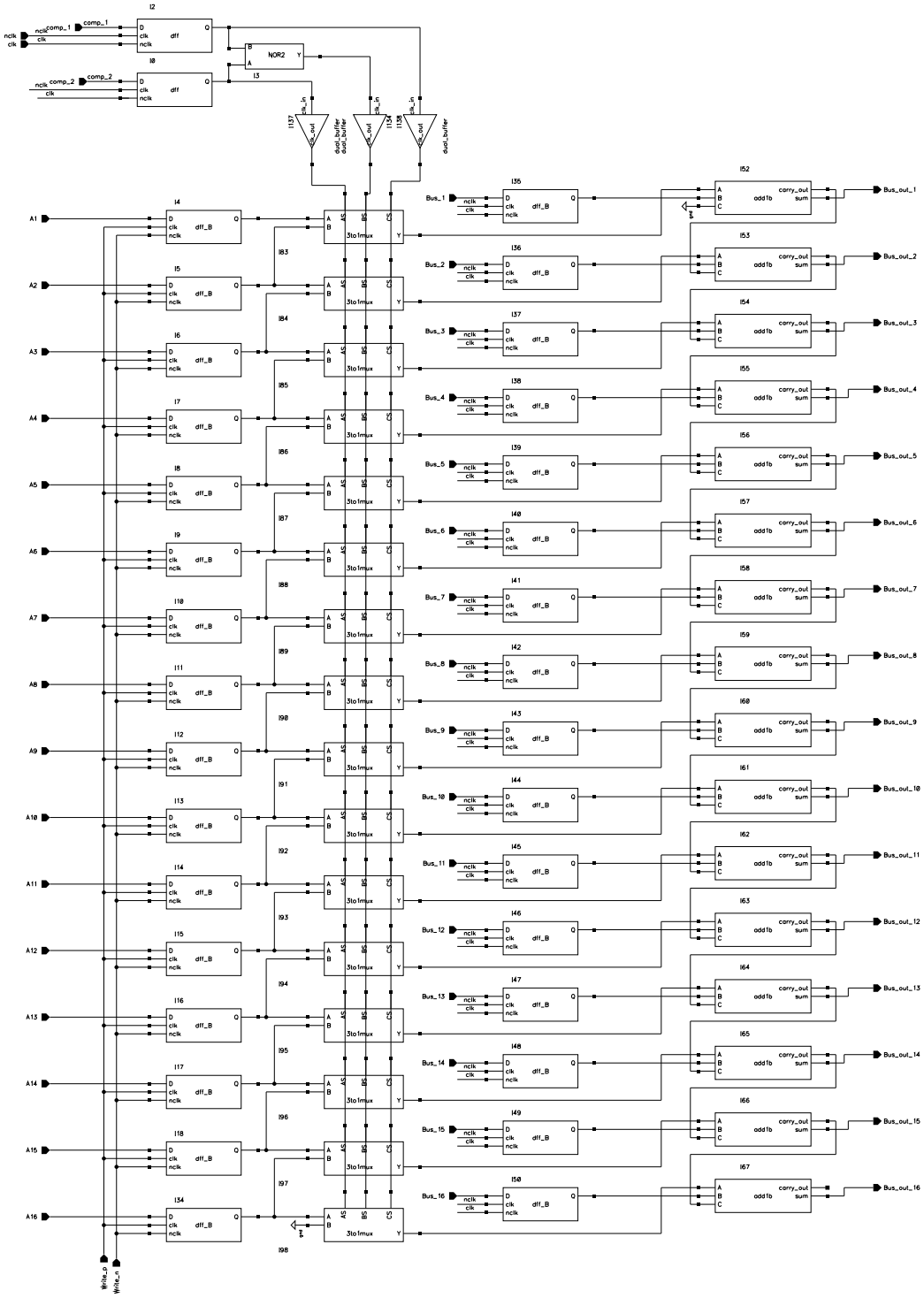


Figure 8.4: Digital block schematic

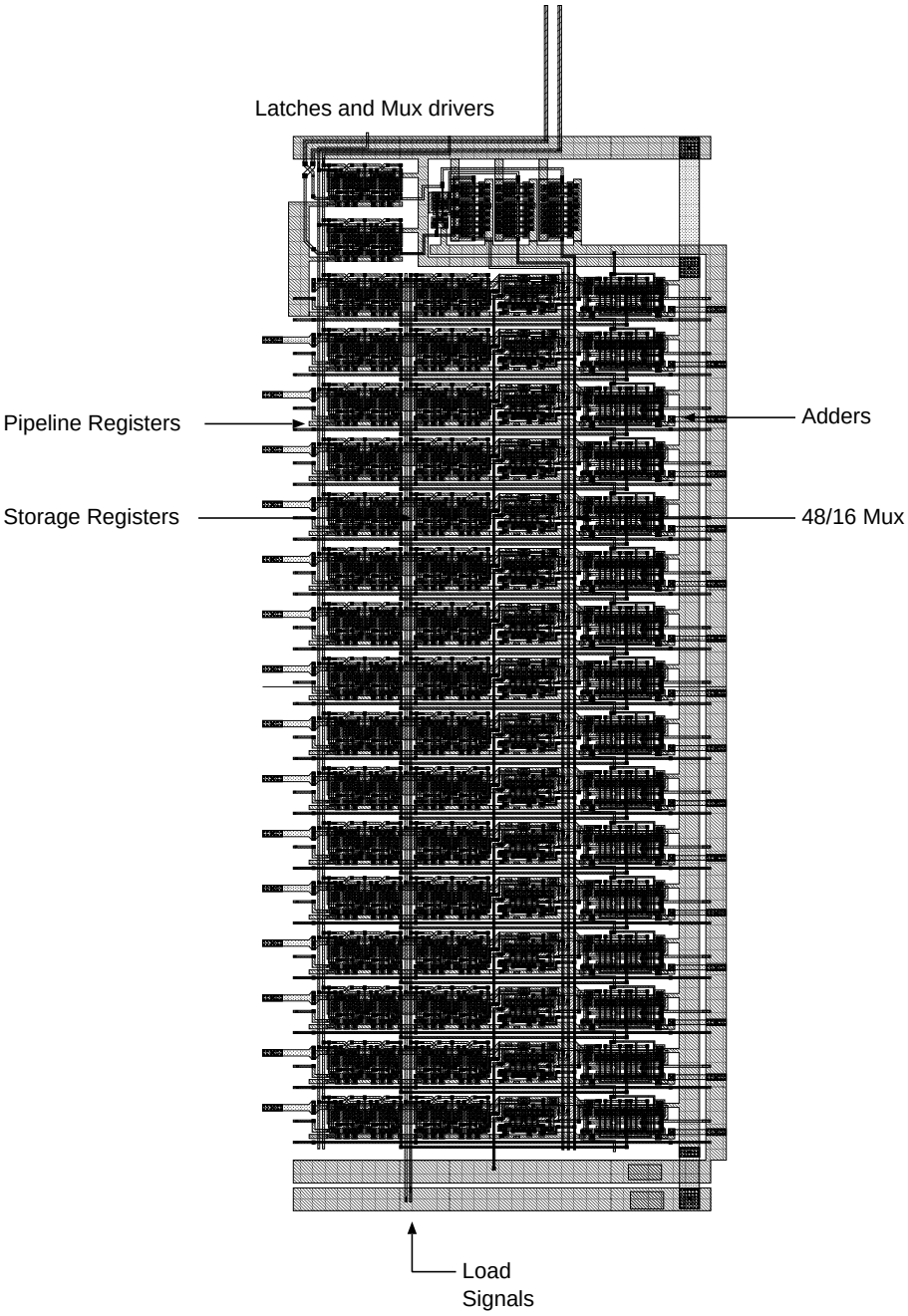


Figure 8.5: Digital block layout.

8.3 Top Level Layout

Combining 14 analog and digital stages together forms the test converter shown in the top level schematic in Figure 8.6. Additional peripheral circuits not discussed include: clock buffers for each stage, two 4 to 16 multiplexers to derive the calibration and coefficient load signals, and other miscellaneous digital buffers. The schematics and corresponding layouts for these peripheral circuits have been included in Appendix D.

At the end of the pipeline a test block has been included that allows access to the residue output and comparator test signals. The test block derives its clocking signals from the same input clock but has an independent clock generator.

Data is loaded into the pipeline through a 16-b parallel input bus that is connected to the inputs of all stages. Which stage actually latches the data is determined by a 4-b input vector. The input data and address vector are allowed one cycle to set up before a write enable signal is strobed, latching the input values.

Data is taken directly off chip for processing by a 16-b parallel output bus. This bus is taken from the last pipeline stage buffered once and then buffered through pad drivers before it is seen on the output pins.

Offset and gain calibration can be performed on the first six stages. A 4-b input vector determines the calibration mode.

The overall size of the test chip shown in Figure 8.7 is 1.8mm by 4.2mm. The signal flow is also indicated on the top level layout. Because the top level layout contains more than 30k transistors a full test chip simulation was not possible. A chip pin-out is included in Appendix C which describes the input vectors needed to control the converter.

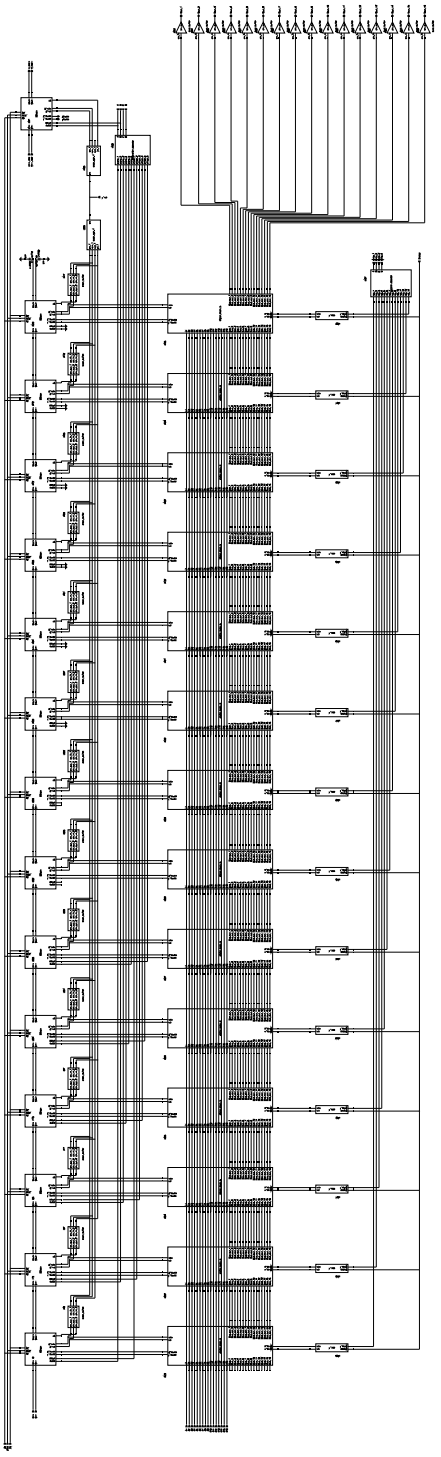


Figure 8.6: Top level schematic

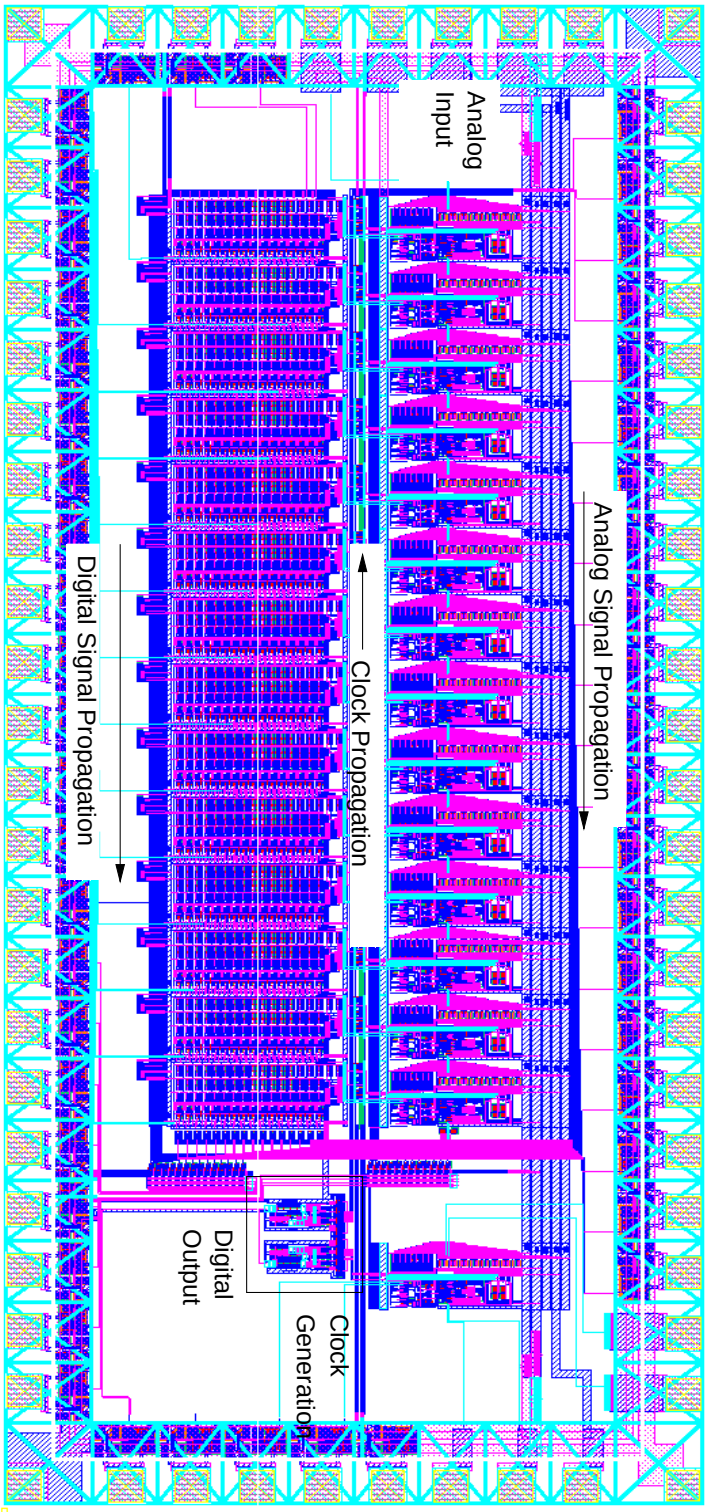


Figure 8.7: Top level layout.

Chapter 9

Test Chip Results

The intent of the test chip was to verify and judge the performance of as many of the ADC structures as possible as well as make a first attempt at a fully operational converter. This chapter will cover the verification of functional components and attempt to explain any non-ideal behavior.

9.1 Clock Generation

The clock was the beginning of testing of the chip because its operation is mandatory for anything else to work. As discussed in the clock generation section, phasing as well as duty cycle of the on chip clocks is very important for the overall operation of the analog processing blocks. Figure 9.1 shows the four clocking signals (latch, clk, /latch, /clk) taken directly from the chip running at 30MHz. The reason the generator is not running at the full 50MHz is because of the limited driving capability of the standard pad drivers used in the chip. A 50MHz clock can be seen but its edges are not as distinct as those shown in Figure 9.1. From the 30MHz test it is evident that the clock phases are correct and the duty cycle very close to 50%. The first signal has the fastest rise and fall times because it was measured using a low capacitance probe.

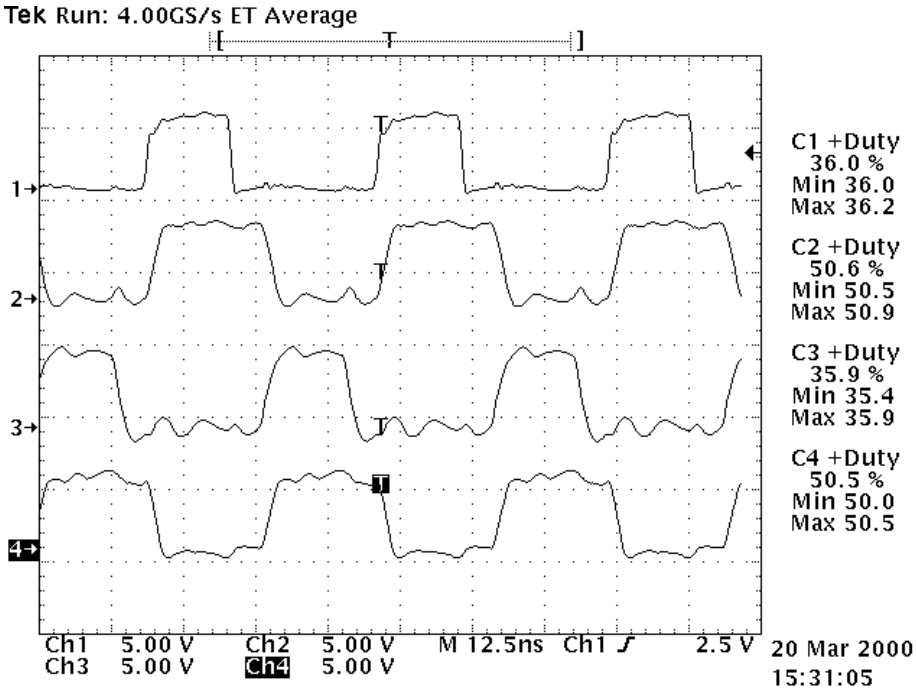


Figure 9.1: Test chip clock generator running at 30MHz.

9.2 Digital Pipeline

One of the more critical aspects of the pipeline design was the ability of the multiplexers and adders to work in conjunction to add 14 bits in one half of the clock cycle (10ns). To test this the hex number h1FFF was loaded in one register and h0001 in a second register. Figure 9.2 shows the result with the test chip clocked at 50MHz. In one case h0000 is added to h1FFF leaving h1FFF at the output of the pipeline. In the second case h1FFF and h0001 are added leaving h2000 on the output of the pipeline. This verifies that for the worst case (13 carry outs and 1 sum out) the adders are still able to perform in the alloted half clock cycle.

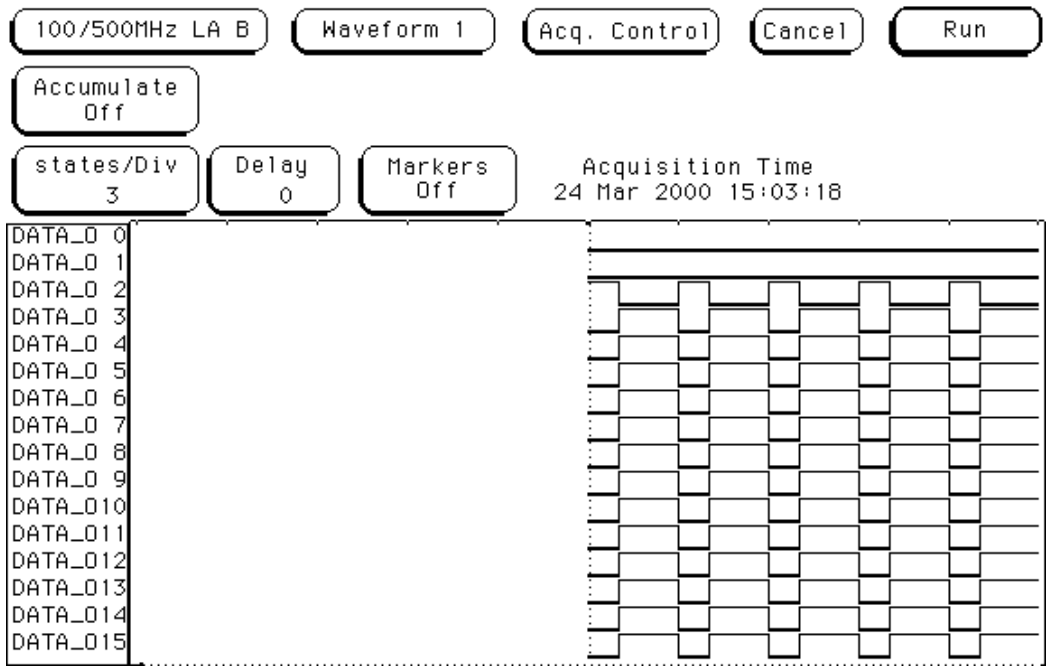


Figure 9.2: Logic analyzer screen shot of worst case 14 bit addition in 10ns.

9.3 Analog Block

To understand any problems that might occur in the operation of the larger converter, the operation of the single stage was verified. One item to note is that the test block cannot operate as fast as the blocks in the larger converter because the switched capacitor amplifier is driving the output pads directly. The capacitance of the pad and external measuring equipment is a factor of ten greater than what the amplifier was designed for. Slowing the test block down allows the amplifier to drive the pads while still obtaining sufficient information about the block's operation.

With the input as a $2V_{pp}$ 1KHz ramp and an operation speed of 10MHz, the output shown in Figure 9.3 results. The output was not as expected and indicated a problem with the layout. Upon further investigation it was discovered that an error occurred in the layout that was reinforced by the extraction software.

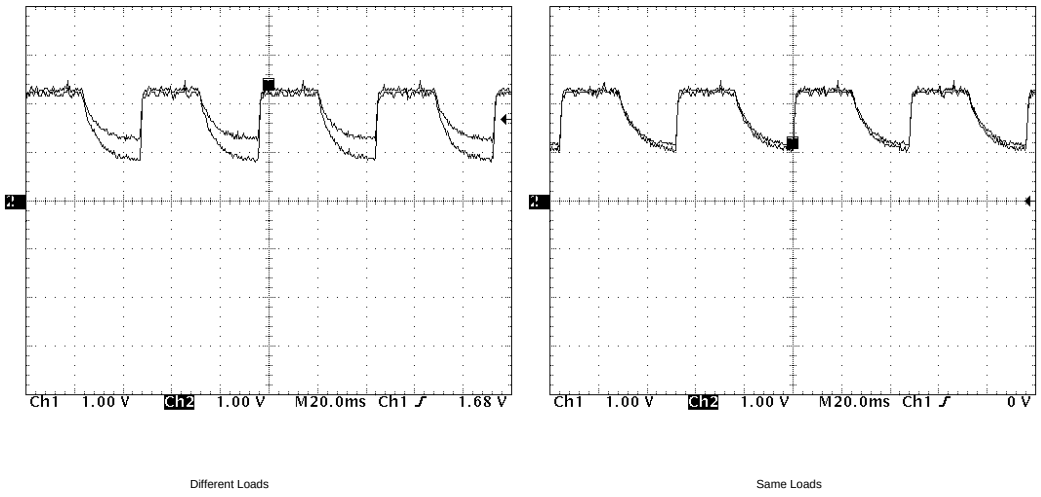


Figure 9.3: Test block output.

The error occurred in the layout of the capacitors. Because of the increased capacitance per unit area, linear capacitors were used. In the HP AMOS14LC process, linear capacitors are made by heavily implanting the substrate to form a highly conductive well, growing a thin oxide layer, and

then growing a polysilicon layer on top of that. In this manner the well region forms one plate and the polysilicon forms the other plate of a capacitor. In addition to the heavily doped well region a lightly doped region of the same dopant type is installed around the entire area. If within this lightly doped layer other capacitors are included, a conductive region between the wells will be formed. This is what was done in the layout of the capacitors for the analog block, all wells were put in the same lightly doped region. The extraction software did not connect the wells together and a false simulation resulted.

Upon notification that an error occurs during layout extraction the writers of the extraction software sent a corrected patch. Installing the patch and re-simulating the extracted view results in Figure 9.4, which is the same as observed on the test chip. With this discovery it is known that the test chip converter will not work so not further testing as a whole converter was performed.

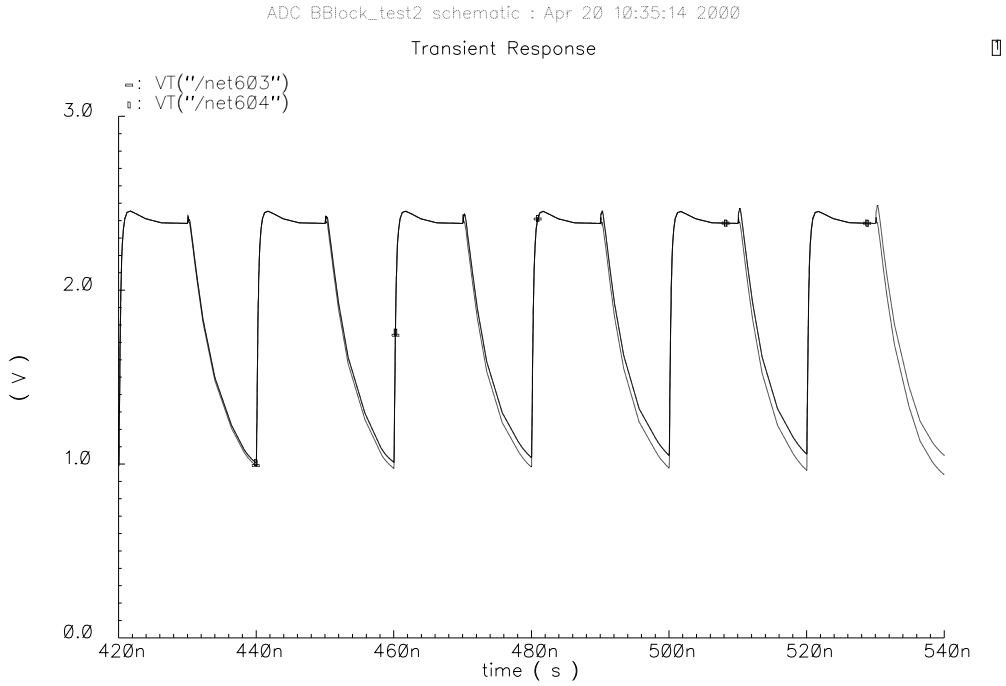


Figure 9.4: Analog block simulation with corrected patch.

An attempt was made do verify the operation of digital control logic for the analog block. Since those signals were not taken directly off chip the only means to verify their operation was to attempt to analyze the output of the test block. As previously stated, the wells of the capacitors are shorted and additional dummy capacitors located in the same well are shorted to ground. This results in the differential amplifier being turned off because the well side of the capacitors are connected to the inputs of the amplifier. With the amplifier being turned off the only output seen is the charge of the feedback capacitors being placed on the test probes. This behavior is believed to be valid because different capacitive loads results in different output amplitudes as shown in Figure 9.3.

Looking at the differential component of the two outputs results in an attenuated version of the input as shown in Figure 9.5. The signal has been filtered to remove the switched capacitor output components.

The output when the test block is put in calibration mode is also shown in Figure 9.5. The input signal is bypassed and the reference voltages are sampled on the inputs. The output is some DC value that represents the error in the test block.

.

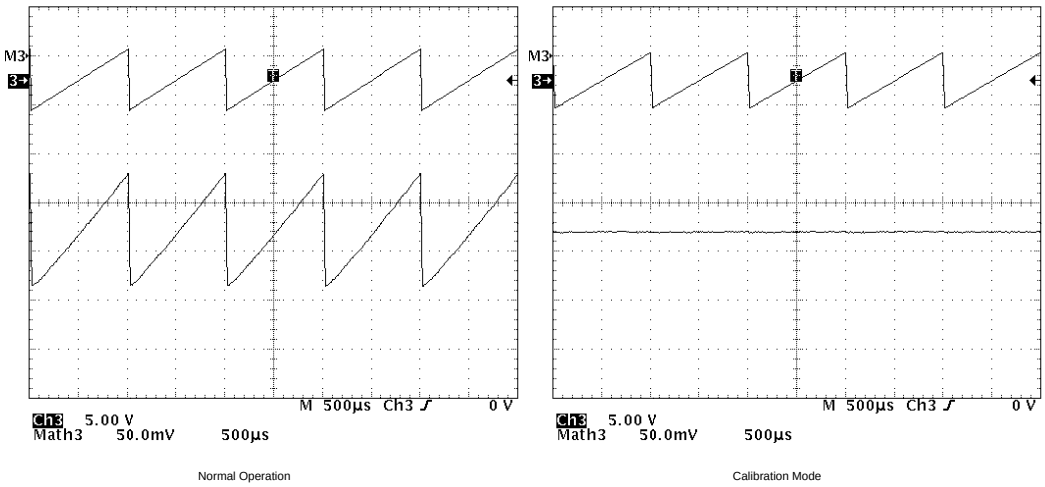


Figure 9.5: Time averaged differential output of test block.

In spite of the fact that a completely operational test block was not achieved, it is believed that from the results obtained that switching within the test block is taking place properly.

9.4 Comparator Operation

The test chip comparator results are shown in Figure 9.6. Both comparator outputs for the test block are shown as well as the ramp input form -2.0V to 2.0V. As expected one comparator is triggered on the negative end of the input and the other on the positive end. The comparator thresholds are at 0.50V and 0.75V which is farther than the designed 0.25V value. All test chips show the same unequal offsets in the comparator thresholds indicating some sort of systematic error. One possible explanation is that the inputs are connected to the shorted capacitors causing some sort of reflection into the input signal. The exact cause for the threshold errors could not be found and should be verified in later test chips.

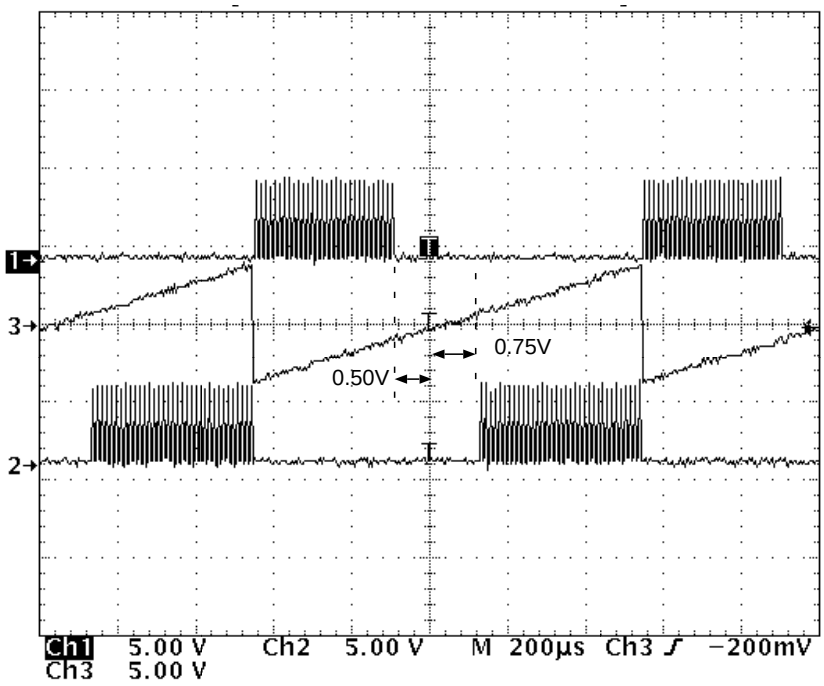


Figure 9.6: Comparator Operation

Chapter 10

Conclusions

A detailed performance analysis of pipeline ADCs has been presented. Design considerations such as bit resolution per stage, speed, and matching were discussed. The specifics of the 1.5-bit per stage pipelined ADC with digital correction were extensively examined and a behavioral simulation was included as a proof of concept. With an understanding of pipeline ADCs with digital correction developed, a design procedure was outlined. The design procedure was used to make an initial test chip to gather information for future designs, the results of which are summarized in Table 10.1.

The intent of this work was to implement a capacitor measurement and calibration technique originally proposed by [1]. This method has advantages over the predominant method of measuring transition heights in that the interaction of errors in the pipeline are believed to be smaller thus making the calibration readings more accurate. In behavioral simulations done in this work this assumption is theoretically verified. System level aspects that are important to the implementation of the calibration technique have also been verified on the first test chip.

An improvement to this relatively unused calibration technique is the order in which the pipeline is calibrated. The original author implemented the calibration technique with a cyclic converter, dictating that the higher order stages are calibrated using lower order uncalibrated stages. In this

proposed implementation the possibility exists for the reordering of calibration, calibrating lower order stages before the higher order ones, further improving performance. Any future work should be directed in this area whether it is an actual test chip or behavioral simulations which encompass more pipeline behavior.

Device	Functionality	Remarks
Clock generator with 200MHz off chip clock	Functions	Results indicate it is possible to send 400MHz clock on chip.
Pipeline loading mechanisms	Functions	
Pipeline w/ mux, adders and latches	Functions	Addition in only two stages verified but values propagate down pipeline correctly. 100MHz operation possible.
Comparators	Functions but not within limits	Thresholds off. Stand-alone operation should be verified in future test chips.
Amplifier	Not Verified	Testing limited by extraction errors. Stand-alone verification needed.
Stage Control Block	Believed to be functional.	Results obtained from limited operation of test chip.
Behavioral and theoretical operation	Not Verified	Limited by test chip operation

Table 10.1: Summary of test chip results.

Future Design Considerations

One Time Calibration

A full production version of this converter could use one time calibration at the end of the production line with some sort of electrical fuses on chip. Recent full production designs have used on time calibration with success [17]. This design uses a low gain amplifier to decrease area and increase speed and depends on the calibration to fix any gain errors. Because of the low gain, different calibration coefficients might be necessary for different operating temperatures. An amplifier with higher gain should be used to fix this problem if one time calibration is to be considered [16].

Appendix A

Simulation Source Code

The following section is a listing of the source code written in C used for the behavioral modeling of a Pipeline ADC. The code is written as an 8 bit converter with a radix=2 1.5-bit per stage structure. The code includes a section which implements digital correction for the first four stages. All results are included in chapter 3. This code was written using reference [19].

```
#include <stdio.h>
#include <math.h>

void main()
{
//Fixed pipeline
        float fixed_pipeline(double input,float cap1[], float cap2[], float offs
et[], float m_offset[], float alpha[]);
//Function to calculate offset
        float off_calc(double input,float cap1[], float cap2[], int x, float off
set[]);
//Function for Plotting
        void res_plots(double input,float cap1[], float cap2[], int x, FILE*out,
 float offset[]);
//function that calculates alpha*Vref
        float gain_error(double input,float cap1[], float cap2[], int x, float o
ffset[]);
//function to implement uncompensated output
        float pipeline(double input, float cap1[], float cap2[], float offset[])
;
        /*
//Capacitors for each stage
        float c1[12] = {1,  1,  1,  1,  1,  1,  1,  1,1,1,1,1};
        float c2[12] = {1,  1,  1,  1,  1,  1,  1,  1,1,1,1,1};
//Offset for each stage as given
        float offset[12]   = {0,0,.1,0,0,0,0,0,0,0,0,0};
        */
//Capacitors for each stage
        float c1[12] = {1.1,  1,  1.1,  1,  1.02,  1,  1.01,  1,1,1.03,1,1};
        float c2[12] = {1,  1.1,  1,  1.05,  1,  1.04,  1,  1,1.05,1,1.02,1.001}
;
//Offset for each stage as given
        float offset[12]   = {-.05,.1,.1,.02,.03,-.03,-.01,.01,-.01,.05,.02,-.04
};

//Offset for each stage as measured
        float m_offset[12] = {0,0,0,0,0,0,0,0,0,0,0,0};
//Array to store input waveform
        double s1[4096];
//Array to store output codes
        double r1[4096];
        double r3[4096]; //fixed output, not quite working
//Array to store alpha
        float alpha[12]={0,0,0,0,0,0,0,0,0,0,0,0};
        float gain[12] ={0,0,0,0,0,0,0,0,0,0,0,0};
//increament integers
        int i;
        float j;
//open output file and create pointer to file
        FILE*out;
        out=fopen("output.m","w+");


//Create input ramp
//from -2 to 2 Volts
//with 4096 points
        for (i=0; i<4096; i++)
        {
                j=i;
                s1[i]=((j-2048)/1024);
        }
//calculate output offset error by setting the
//input to each stage to 0 and letting the
//remainder of the pipeline calculate the offset
//print the measured offsets to a file for reference
        for (i=0; i<=5; i++)
        {
                m_offset[i]=off_calc(s1[i],c1,c2,i,offset);
                fprintf(out,"offset(%i)=%f;\n",i+1,m_offset[i]);
        }
```

```
//First attempt at calculation gain errors
//using the equation
//alpha*Vref=(Vref*c1-Vref*c2)/c1 for each stage
//the result is then divided by Vref
//and printed to the output file
//note alpha is actually alpha divided by 2
        for (i=0; i<=5; i++)
        {
                alpha[i]=(((gain_error(1,c1,c2,i,offset)-m_offset[i])/(pow(2,6-i
)))-1);
                fprintf(out,"alpha(%i)=%f;\n",i+1,alpha[i]);

        }
//calculate uncorrected ramp from output
//and print output vector to file
        for (i=0; i<4096; i++)
        {
                r1[i]=pipeline(s1[i],c1,c2, offset);
                fprintf(out,"r1(%i)=%f;\n",i+1,r1[i]);
        }
//calculate corrected ramp from output
//and print output vector to file
//this function is in progress and does not work
        for (i=0; i<4096; i++)
        {
                r3[i]=fixed_pipeline(s1[i],c1,c2, offset, m_offset, alpha);
                fprintf(out,"r3(%i)=%f;\n",i+1,r3[i]);
        }
//Print input ramp to output file
//so it can be used in plotting routines
        for (i=0; i<4096; i++)
        {
                fprintf(out,"s1(%i)=%f;\n",i+1,s1[i]);
        }
//Residue plotting routine. The output file pointer
//is passed to the function so that an output value
//for each stage can be printed.
        for (i=0; i<4096; i++)
                res_plots(s1[i],c1,c2,i, out, offset);
//Close file
        fclose(out);
}
//
//
//
//
//
//
//
//
//
//uncorrected pipeline
float pipeline(double input,float cap1[], float cap2[], float offset[])
{
        float output=0;
        int output1=0;
        int i;
        double q1;
        double q2;
        double q3;

        int d[12]={0,0,0,0,0,0,0,0,0,0,0,0};
//comparators for all 7 stages
        for (i=0; i<=11; i++)
        {
                if (input <= -0.5)
                {
                        d[i]=0;
                        q3=cap2[i]*2;
                }
```

```
                else if (-0.5 < input && input < 0.5)
                {
                        d[i]=1;
                        q3=0;
                }
                else if (input >= 0.5)
                {
                        d[i]=2;
                        q3=-cap2[i]*2;
                }
//Residue amplifier
                q1=input*cap1[i];
                q2=input*cap2[i];
                input=((q1+q2+q3)/cap1[i])+offset[i];
        }
//Calculate outputs using decision bits
        for (i=0; i<=11; i++)
        {
                output = output + d[i]*pow(2,(6-i));
        }
        output1=output;
        return(output1);

}
//
//
//
//
//
//
//
//
//
//Error calculator
float gain_error(double input,float cap1[], float cap2[], int x, float offset[])
{
        float output=0;
        int i;
        double q1;
        double q2;
        double q3;

        int d[12]={0,0,0,0,0,0,0,0,0,0,0,0};
//Use Lee's method on ith stage
        q1=input*cap1[x];
        q2=0;
        q3=-cap2[x];
        input=((q1+q2+q3)/cap1[x])+offset[x];
//Send output down remainder of pipeline
        for (i=x+1; i<=11; i++)
        {
//Comparators
                if (input < -0.5)
                {
                        d[i]=0;
                        q3=cap2[i]*2;
                }
                else if (-0.5 <= input && input <= 0.5)
                {
                        d[i]=1;
                        q3=0;
                }
                else if (input > 0.5)
                {
                        d[i]=2;
                        q3=-cap2[i]*2;
                }
//Residue amplifier
                q1=input*cap1[i];
                q2=input*cap2[i];
```

```
                input=((q1+q2+q3)/cap1[i])+offset[i];
        }
//Calculate output using decision bits
        for (i=0; i<=11; i++)
        {
                output = output + d[i]*pow(2,(6-i));
        }
        return(output);
}
//
//
//
//
//
//
//
//
//
//plot residues
void res_plots(double input,float cap1[], float cap2[], int x, FILE*out, float o
ffset[])
{
        int i;
        double q1;
        double q2;
        double q3;

//comparators for all 8 stages
//decision bits are not used in this
//routine because we are only concerned
//with the analog output value of each stage
        for (i=0; i<=6; i++)
        {
                if (input <= -0.5)
                {
                        q3=cap2[i]*2;
                }
                else if (-0.5 < input && input < 0.5)
                {
                        q3=0;
                }
                else if (input >= 0.5)
                {
                        q3=-cap2[i]*2;
                }
//Residue amplifier
                q1=input*cap1[i];
                q2=input*cap2[i];
                input=((q1+q2+q3)/cap1[i])+offset[i];
//print an output for each progression through the loop
                fprintf(out,"r2(%i,%i)=%f;\n",x+1,i+1,input);
        }

}
//
//
//
//
//
//
//
//
//
//Offset calculator
float off_calc(double input,float cap1[], float cap2[], int x, float offset[])
{
        float output=0;
        int i;
        double q1;
        double q2;
```

```
        double q3;

        int d[12]={0,0,0,0,0,0,0,0,0,0,0,0};
//This routine is only called once for each stage
//of the pipeline. The index x is passed to the routine
//so that one stage is removed from the pipeline
//each time the function is called.
        for (i=x; i<=11; i++)
        {
//Comparators
                if (input < -0.5)
                {
                        d[i]=0;
                        q3=cap2[i]*2;
                }
                else if (-0.5 <= input && input <= 0.5)
                {
                        d[i]=1;
                        q3=0;
                }
                else if (input > 0.5)
                {
                        d[i]=2;
                        q3=-cap2[i]*2;
                }
//Residue amplifier
                q1=input*cap1[i];
                q2=input*cap2[i];
                input=((q1+q2+q3)/cap1[i])+offset[i];
        }
//Calculate output using decision bits
        for (i=0; i<=11; i++)
        {
                output = output + d[i]*pow(2,(6-i));
        }
        return(output);
}
//
//
//
//
//
//
//
//
//
//Corrected pipeline (does not work yet)
float fixed_pipeline(double input,float cap1[], float cap2[], float offset[], fl
oat m_offset[], float alpha[])
{
        int output1=0;
        float output=0;
        float beta;
        int i;
        double q1;
        double q2;
        double q3;

        int d[12]={0,0,0,0,0,0,0,0,0,0,0,0};
//comparators for all 7 stages
        for (i=0; i<=11; i++)
        {
                if (input < -0.5)
                {
                        d[i]=0;
                        q3=cap2[i]*2;
                }
                else if (-0.5 <= input && input <= 0.5)
                {
                        d[i]=1;
```

```
                        q3=0;
                }
                else if (input > 0.5)
                {
                        d[i]=2;
                        q3=-cap2[i]*2;
                }
//Residue amplifier
                q1=input*cap1[i];
                q2=input*cap2[i];
                input=((q1+q2+q3)/cap1[i])+offset[i];
        }


                output = output + (d[0]*64)*(1-alpha[0]);

                output = output + (d[1]*32)*(1+alpha[0]-alpha[1]);

                output = output + (d[2]*16)*(1+alpha[0]+alpha[1]-alpha[2]);

                output = output + (d[3]*8)* (1+alpha[0]+alpha[1]+alpha[2]-alpha[
3]);

                output = output + d[4]*4*   (1+alpha[0]+alpha[1]+alpha[2]+alpha[
3]);

                output = output + d[5]*2*   (1+alpha[0]+alpha[1]+alpha[2]+alpha[
3]);

                output = output + d[6] *    (1+alpha[0]+alpha[1]+alpha[2]+alpha[
3]);

                output = output + d[7] *    (1+alpha[0]+alpha[1]+alpha[2]+alpha[
3])/2;

                output = output + d[8] *    (1+alpha[0]+alpha[1]+alpha[2]+alpha[
3])/4;

                output = output + d[9] *    (1+alpha[0]+alpha[1]+alpha[2]+alpha[
3])/8;

                output = output + d[10]*    (1+alpha[0]+alpha[1]+alpha[2]+alpha[
3])/16;

                output = output + d[11]*    (1+alpha[0]+alpha[1]+alpha[2]+alpha[
3])/32;

                output = output + d[12]/64;

                output1=output-m_offset[0];

        return(output1);
}
```

```
%Plotting routines
Ni=hist(r3,256);
for i=1:256,
   Ni(i)=(Ni(i)-16)/16;
end;
inl=0;
for i=3:254,
   inl(i)=sum(Ni(3:i));
end;


subplot(3,1,1),
plot(s1,r3);
grid;
title('Corrected Pipeline');
ylabel('Input Signal');
xlabel('Output Codes');


subplot(3,1,2),plot(Ni);
title('DNL Test');
ylabel('LSB''s');
%xlabel('Codes');
grid;


subplot(3,1,3),plot(inl);
title('INL Test');
ylabel('LSB''s');
xlabel('Codes');
grid;
%print -djpeg c:\fig1.jpg
%axlimdlg;
```

```
%Plotting routines
N=hist(r1,256);
for i=1:256,
   N(i)=(N(i)-16)/16;
end;
inl=0;
for i=1:256,
   inl(i)=sum(N(1:i));
end;

i=1:4096;
subplot(3,1,1),
plot((i-2048)/1024,r1);
title('Uncorrected Output');
ylabel('Output Codes');
xlabel('Input Signal');

subplot(3,1,2),plot(N);
axis([0 256 -2 4])
title('DNL Test');
ylabel('LSB''s');
xlabel('Codes');
grid;


subplot(3,1,3),plot(inl);
axis([0 256 -5 3])
title('INL Test');
ylabel('LSB''s');
xlabel('Codes');
grid;
%print -djpeg c:\fig2.jpg

%axlimdlg;
```

Appendix B

Maple gain error derivation

This section is a listing of the Maple code used to derive the required open loop gain of an amplifier. These results are derived for a source coupled pair which describes the structure of a single stage $g_m - C$ amplifier. For amplifiers with more than one stage further analysis is required. This derivation is a combination of material contained within references [9][10][11].The results are mentioned in chapter 4.

```
> V_out:=(A_0)*(V_e)*sqrt(1-(((A_0)*(V_e)/(V_max))^2)/4);
```

$$V\_out := \frac{1}{2} A\_0\, V\_e \sqrt{4 - \frac{A\_0^2\, V\_e^2}{V\_max^2}}$$

```
> gain:=diff(V_out,V_e);
```

$$gain := \frac{1}{2} A\_0 \sqrt{4 - \frac{A\_0^2\, V\_e^2}{V\_max^2}} - \frac{1}{2} \frac{A\_0^3\, V\_e^2}{\sqrt{4 - \frac{A\_0^2\, V\_e^2}{V\_max^2}}\, V\_max^2}$$

```
> error:=(A_nom-A_max)/(A_max*A_nom*f^2);
```

$$error := \frac{A\_nom - A\_max}{A\_max\, A\_nom\, f^2}$$

```
> x:=subs(A_nom=A_0,A_max=gain,error);
```

$$x := \frac{A\_0 - \frac{1}{2} A\_0 \sqrt{4 - \frac{A\_0^2\, V\_e^2}{V\_max^2}} + \frac{1}{2} \frac{A\_0^3\, V\_e^2}{\sqrt{4 - \frac{A\_0^2\, V\_e^2}{V\_max^2}}\, V\_max^2}}{\left(\frac{1}{2} A\_0 \sqrt{4 - \frac{A\_0^2\, V\_e^2}{V\_max^2}} - \frac{1}{2} \frac{A\_0^3\, V\_e^2}{\sqrt{4 - \frac{A\_0^2\, V\_e^2}{V\_max^2}}\, V\_max^2} \right) A\_0\, f^2}$$

```
> x:=simplify(x);
```

$$x := -\frac{\sqrt{-\frac{-4\, V\_max^2 + A\_0^2\, V\_e^2}{V\_max^2}}\, V\_max^2 - 2\, V\_max^2 + A\_0^2\, V\_e^2}{A\_0 \left(-2\, V\_max^2 + A\_0^2\, V\_e^2\right) f^2}$$

```
> delta_error:=x;
```

$$delta\_error := -\frac{\sqrt{-\frac{-4\, V\_max^2 + A\_0^2\, V\_e^2}{V\_max^2}}\, V\_max^2 - 2\, V\_max^2 + A\_0^2\, V\_e^2}{A\_0 \left(-2\, V\_max^2 + A\_0^2\, V\_e^2\right) f^2}$$

```
> V_error:=(V_max*sqrt(2)/A_0)*sqrt(1-sqrt(1-(V_o/V_max)^2));
```

$$V\_error := \frac{V\_max \sqrt{2} \sqrt{1 - \sqrt{1 - \frac{V\_o^2}{V\_max^2}}}}{A\_0}$$

```
> V_error:=subs(V_max=5,V_error);
> y:=subs(V_max=5,f=0.5,V_e=V_error,x);
```

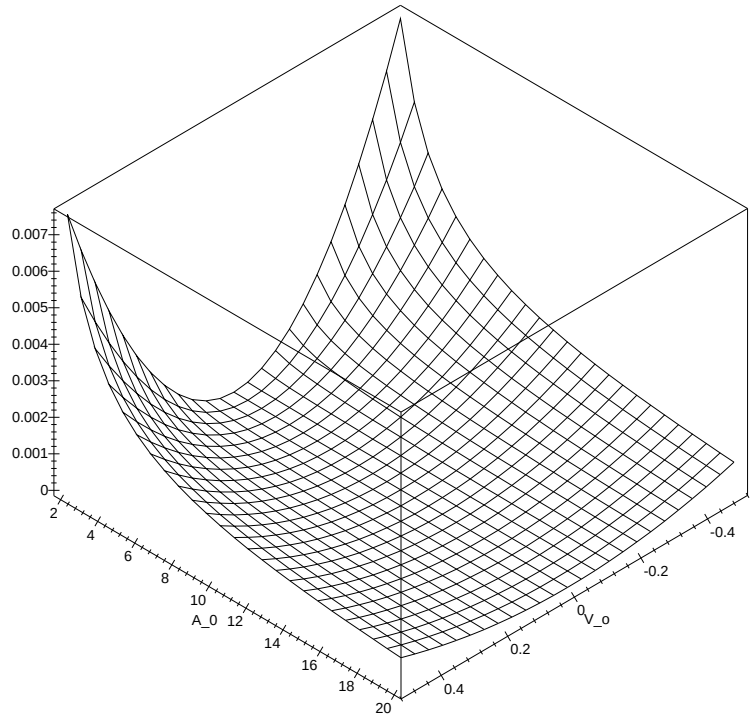
$$V\_error := 5\,\frac{\sqrt{2}\,\sqrt{1-\sqrt{1-\frac{1}{25}\,V\_o^2}}}{A\_0}$$

$$y := .08000000000\,\frac{25\,\sqrt{2+2\,\sqrt{1-\frac{1}{25}\,V\_o^2}}-50\,\sqrt{1-\frac{1}{25}\,V\_o^2}}{A\_0\,\sqrt{1-\frac{1}{25}\,V\_o^2}}$$

```
> z:=simplify(y);
```

$$z := -2.\,\frac{-1.\,\sqrt{50.+10.\,\sqrt{25.-1.\,V\_o^2}}+2.\,\sqrt{25.-1.\,V\_o^2}}{A\_0\,\sqrt{25.-1.\,V\_o^2}}$$

```
> 
> 
> plot3d(z,V_o=-.5..0.5,A_0=2..20);
```




```
> y:=subs(V_e=V_error,delta_error);
```

$$y := -\left(\sqrt{-\frac{-4\,V\_max^2 + 2\,V\_max^2\left(1-\sqrt{1-\frac{V\_o^2}{V\_max^2}}\right)}{V\_max^2}}\,V\_max^2 - 2\,V\_max^2\right.$$

$$\left. + 2\,V\_max^2\left(1-\sqrt{1-\frac{V\_o^2}{V\_max^2}}\right)\right) \Big/ \left(A\_0\right.$$

$$\left(-2\,V\_max^2 + 2\,V\_max^2\left(1 - \sqrt{1 - \frac{V\_o^2}{V\_max^2}}\right)\right) f^2$$

```
> simplify(y);
```

$$-\frac{1}{2}\,\frac{-\sqrt{2 + 2\sqrt{\dfrac{V\_max^2 - V\_o^2}{V\_max^2}}} + 2\sqrt{\dfrac{V\_max^2 - V\_o^2}{V\_max^2}}}{A\_0\sqrt{\dfrac{V\_max^2 - V\_o^2}{V\_max^2}}\,f^2}$$

```
>
```

Appendix C

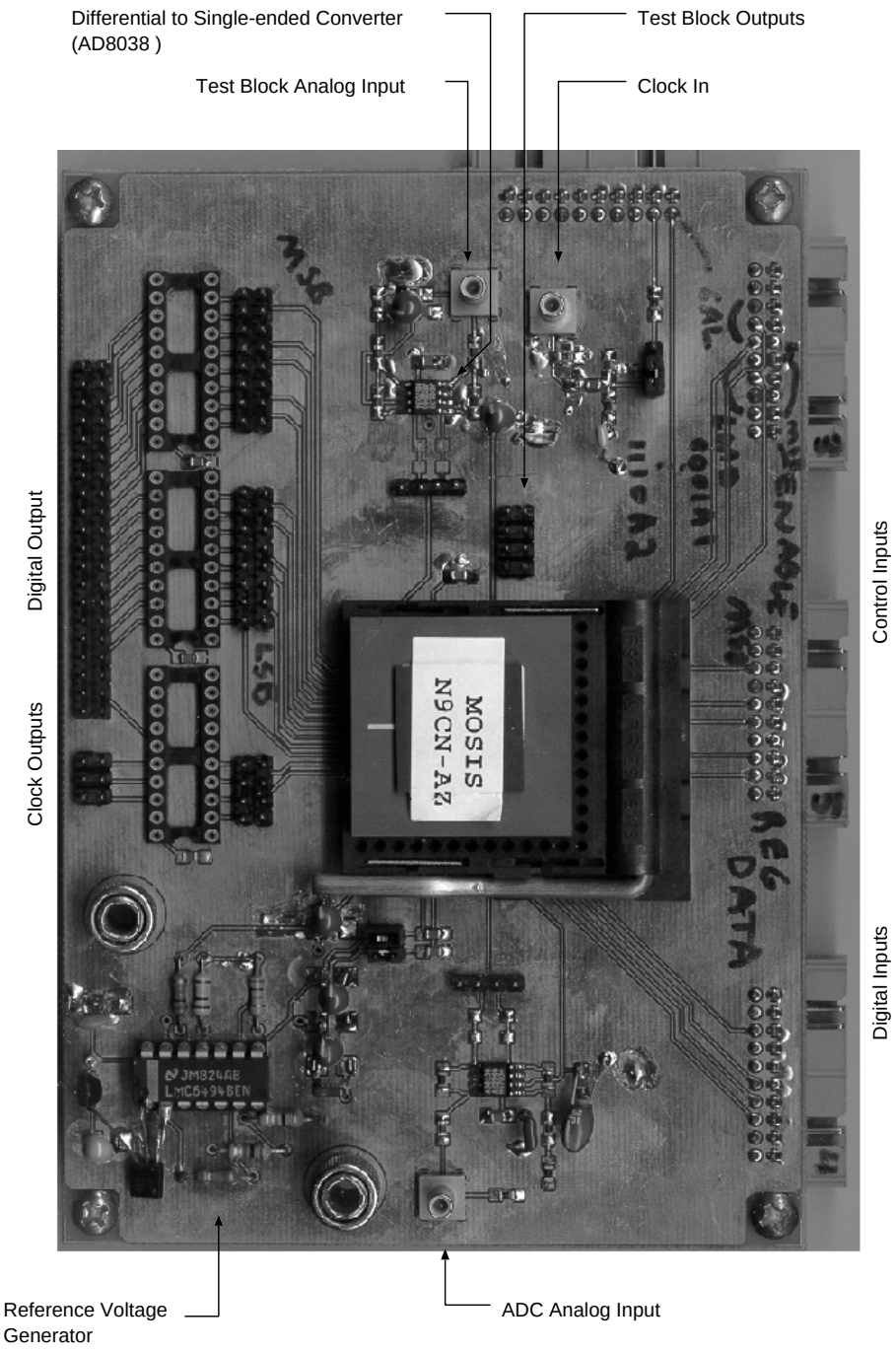
Test Chip and Board

Pin name	Pin number	Description
VDD	2	Power
VIN+	3	Test block input
VIN-	4	Test block input
D16	5	Data out LSB
D15	6	
D14	7	
D13	8	
D12	9	
D11	10	
D10	11	
D9	12	
D8	13	
D7	14	
D6	15	

Pin name	Pin number	Description
D5	16	
D4	17	
D3	18	
D2	19	
D1	20	Data out MSB
CLK	21	output clock
/CLK	22	output /clock
LATCH	23	output latch
/LATCH	24	output /latch
GND	25	Ground
3V	37	3V reference input
2.5V	38	2.5V reference input
2V	39	2V reference input
VIN+	40	Full pipeline input
VIN-	41	Full pipeline input
A1	42	Data in MSB
A2	43	
A3	44	
VDD	57	Power
A4	58	
A5	59	
A6	60	
A7	61	
A8	62	

Pine name	Pin number	Description
A9	63	
A10	64	
A11	65	
A12	66	
A13	67	
A14	68	
A15	69	
A16	70	Data in LSB
ENABLE	71	Data in Enable
AD_1	72	Data in address LSB
AD_2	73	
AD_3	74	
AD_4	75	Data in address MSB
CAL_1	76	Calibration MSB
CAL_2	77	
CAL_3	78	
CAL_4	79	Calibration LSB
GND	91	Ground
CLK_IN	92	Clock input
D1	94	Test block comparator 1 output
D2	95	Test block comparator 2 output
VOUT+	97	Test block residue output
VOUT-	98	Test block residue output

Test Board


Differential to Single-ended Converter
(AD8038)
Test Block Outputs
Test Block Analog Input
Clock In
Digital Output
Clock Outputs
Control Inputs
Digital Inputs
Reference Voltage
Generator
ADC Analog Input
MOSIS
N9CN-AZ
DATA

Appendix D

Schematics

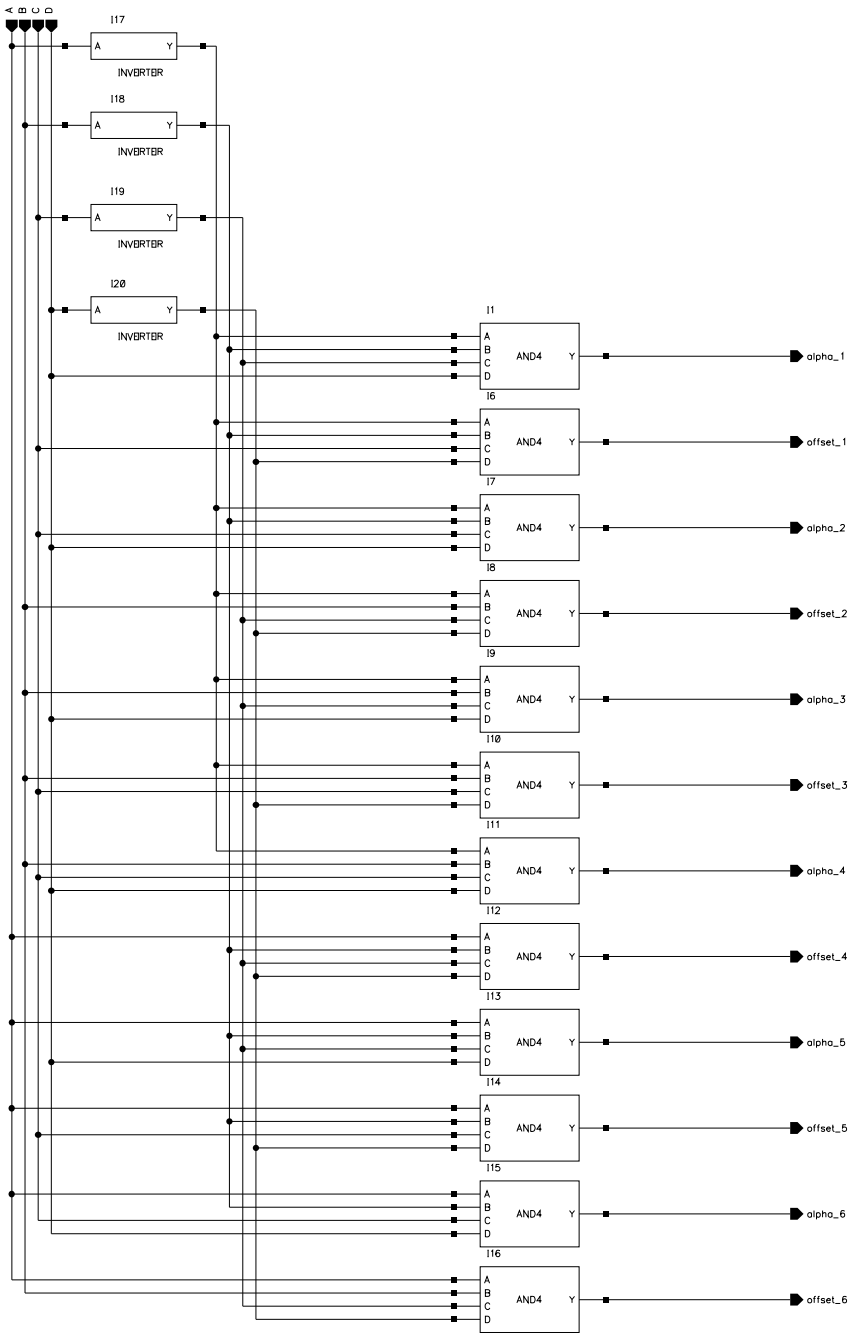


Figure D.1: Calibration decoder schematic.

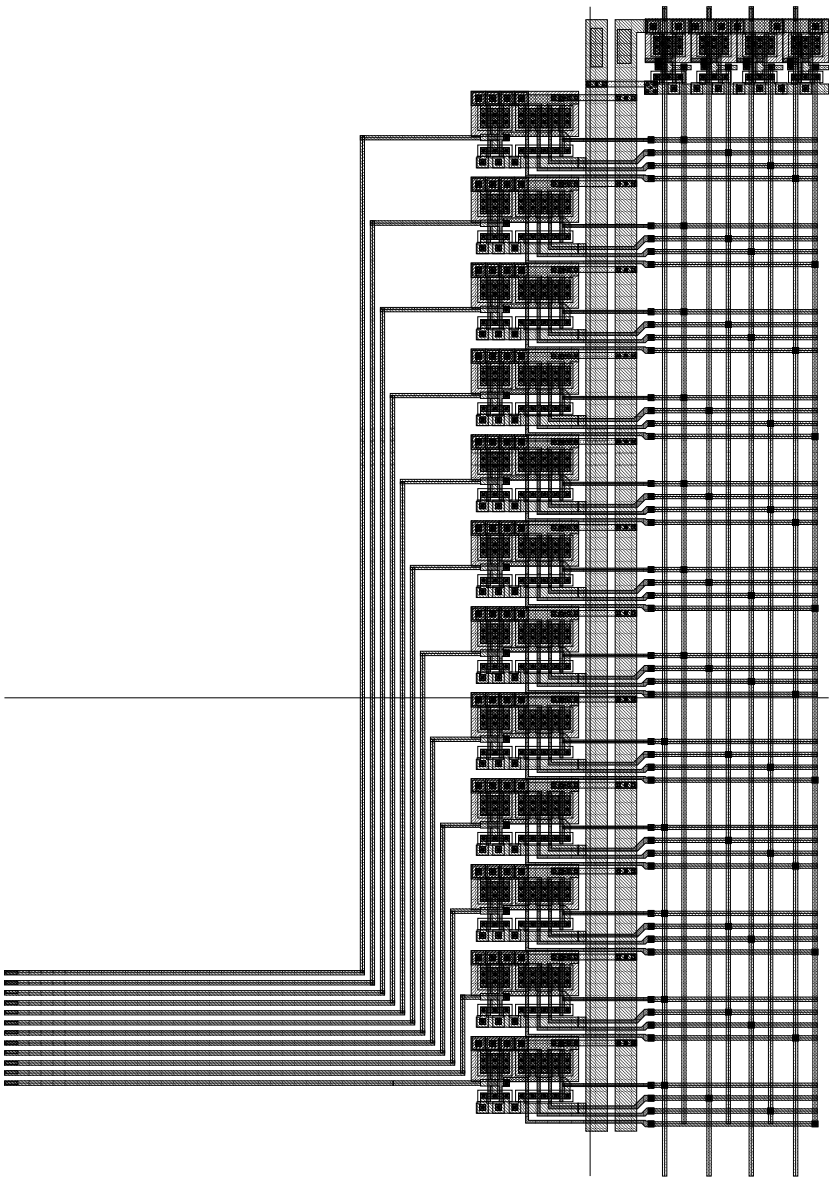


Figure D.2: Calibration decoder layout.

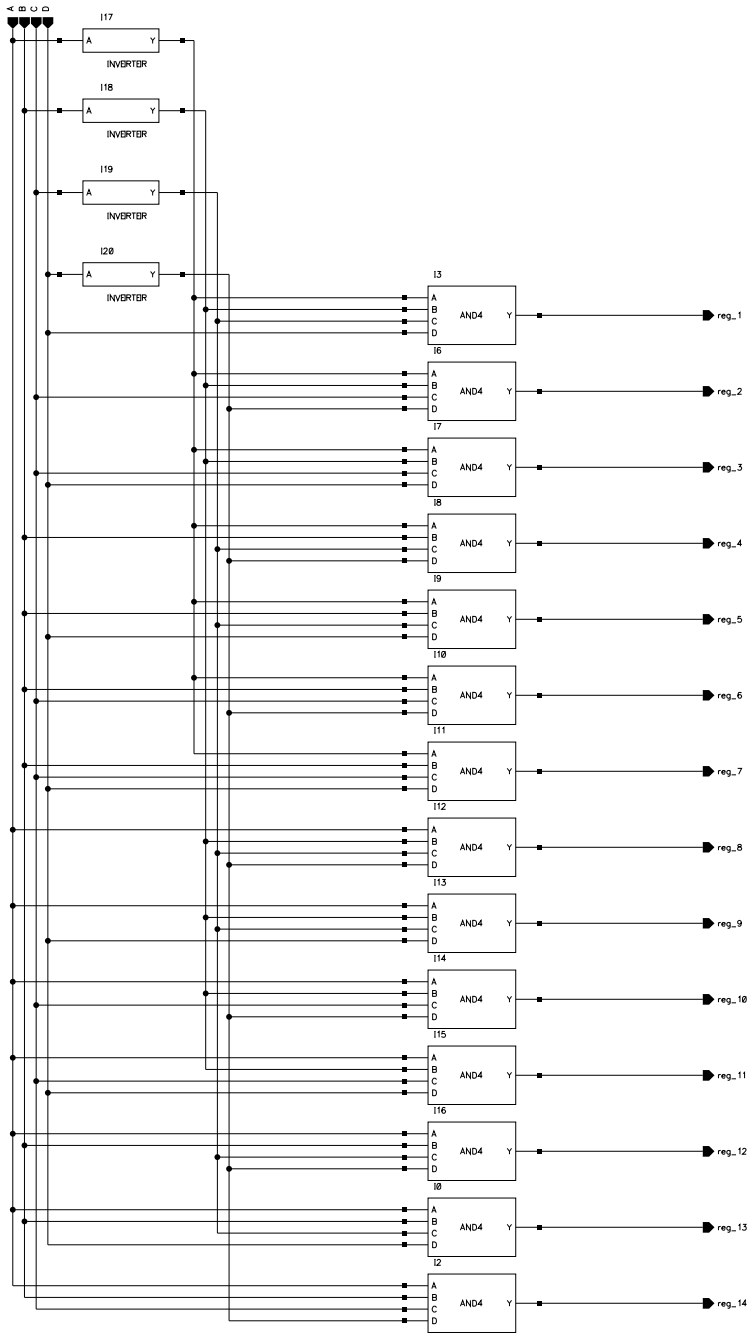


Figure D.3: Register loading decoder schematic.

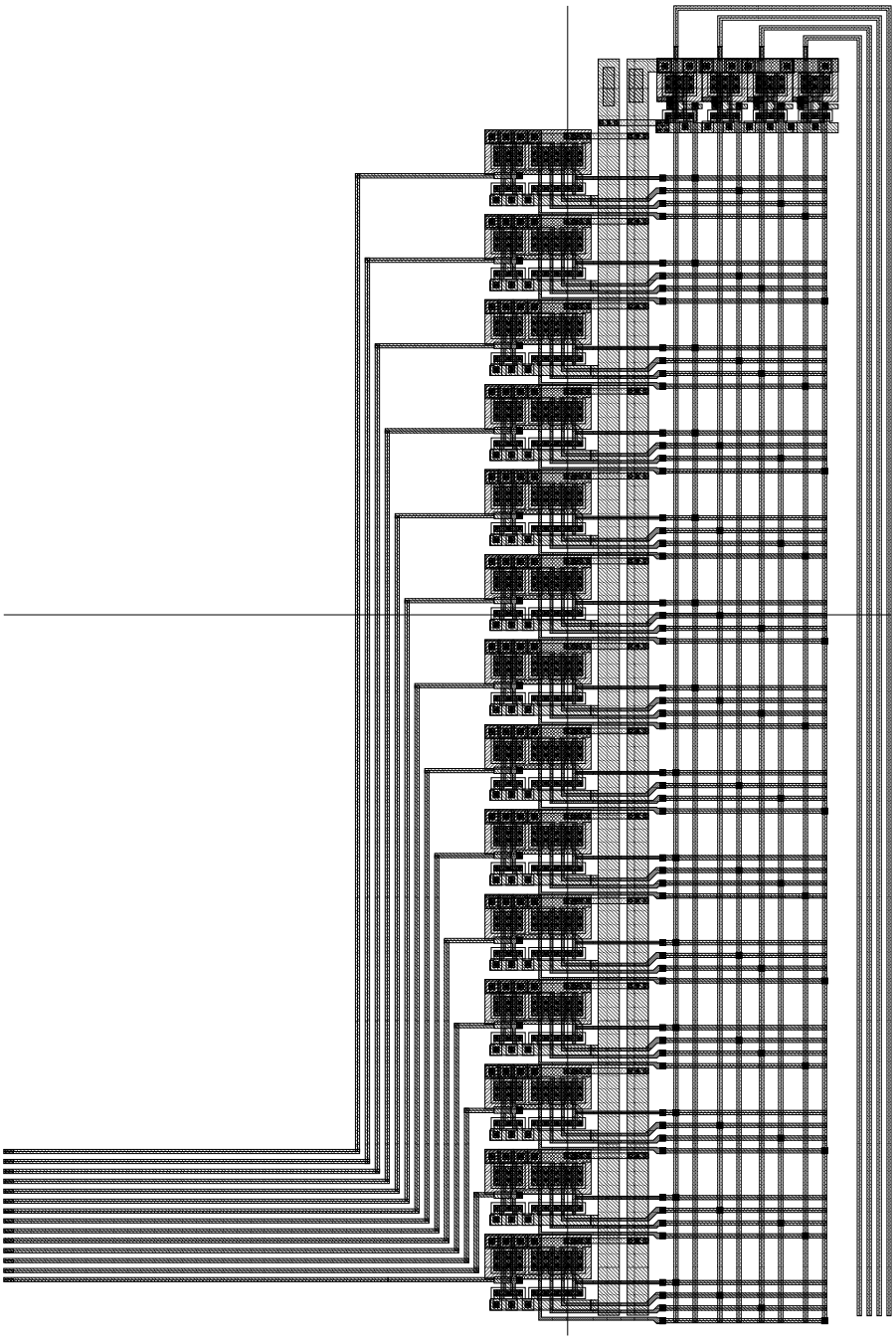


Figure D.4: Register loading decoder layout.

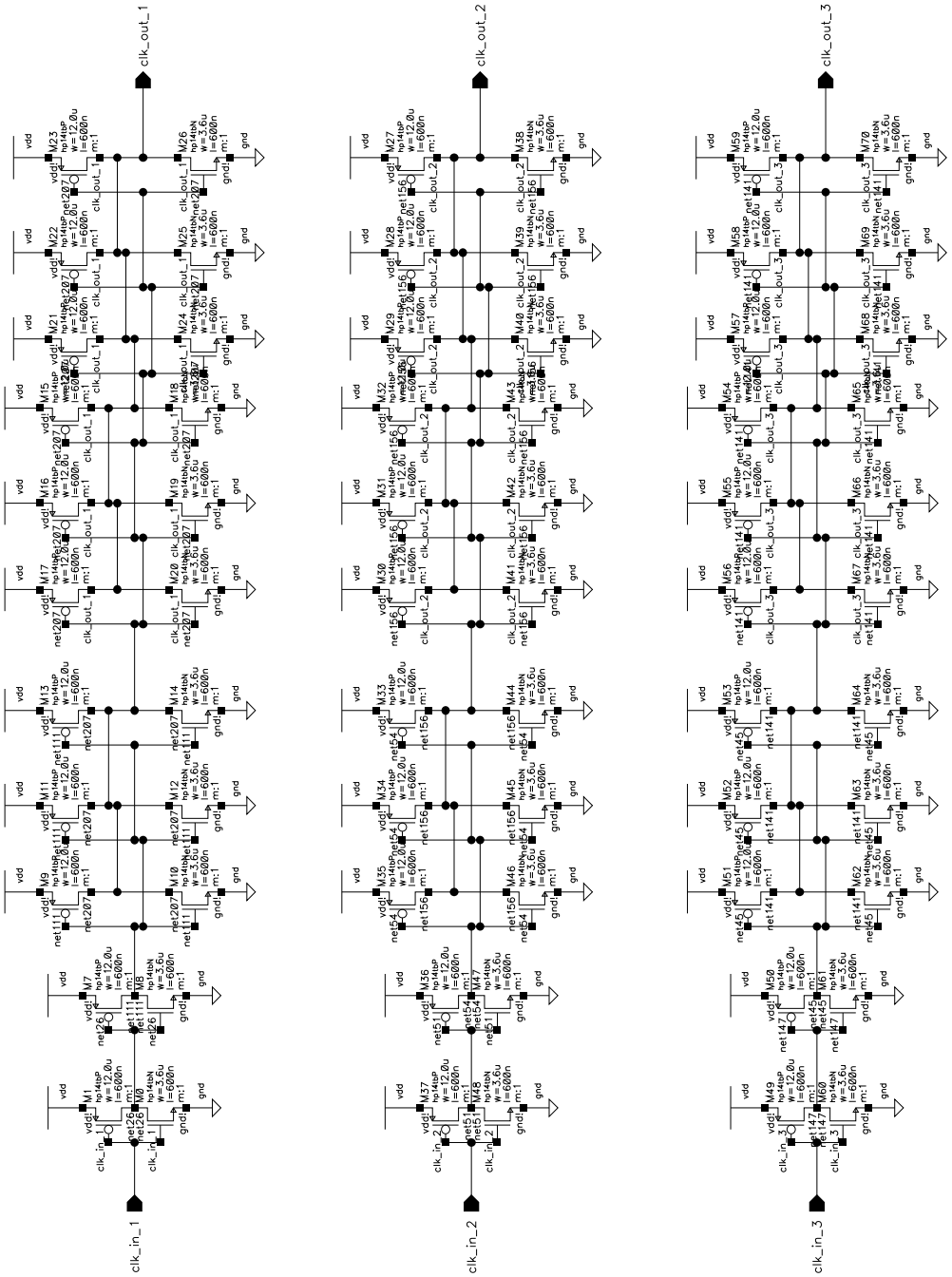


Figure D.5: Clock buffer schematic.

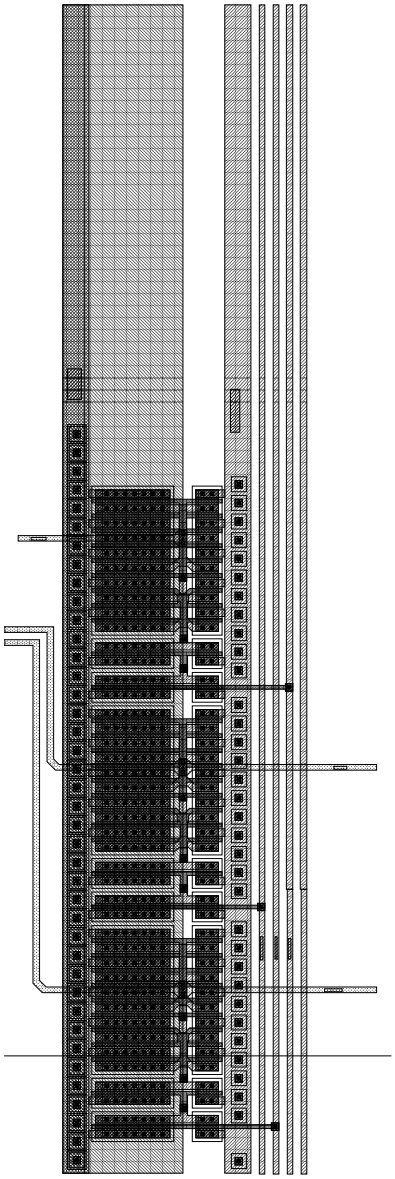


Figure D.6: Clock buffer layout.

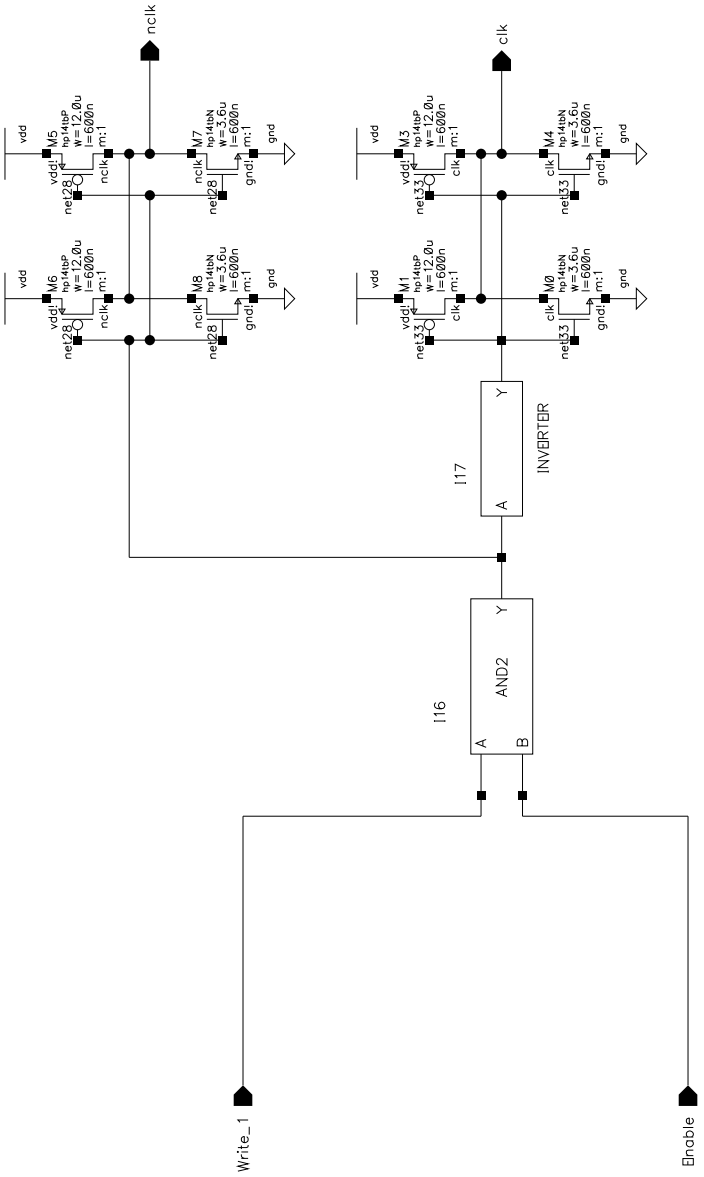


Figure D.7: Register loader schematic.

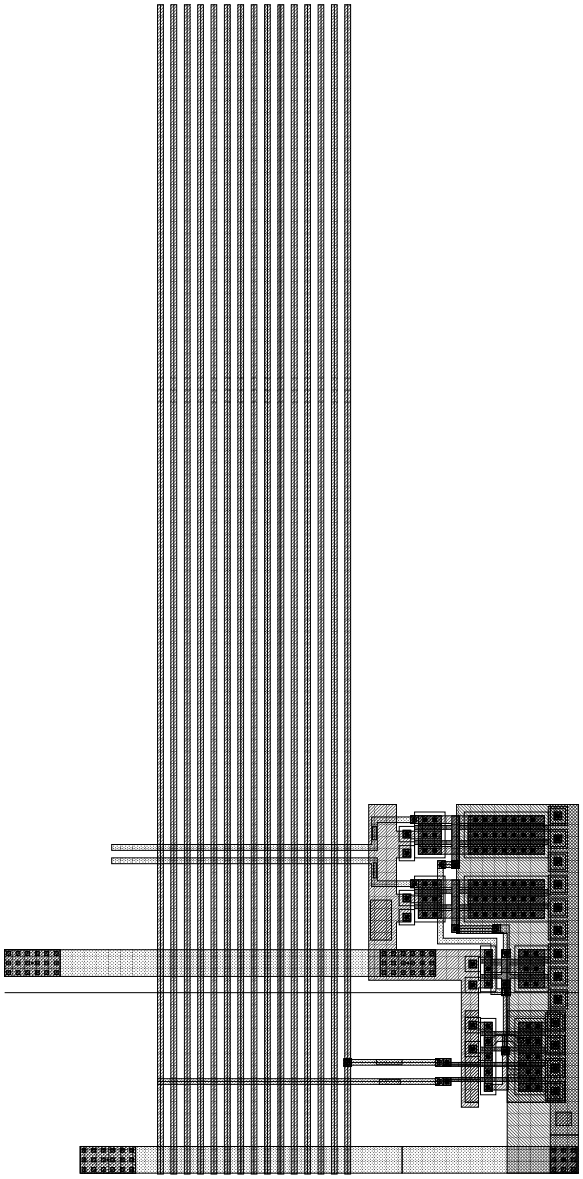


Figure D.8: Register loader layout.

Bibliography

[1] Hae-Sung Lee, " A 12-b 600ks/s Digitally Self Calibrated Pipelined Algorithmic ADC," IEEE J. Solid State Circuits, Vol. 29, No. 4, pp 509-515, April 1994.

[2] Alan Oppenheim, Ronald Schafer, “ Discrete-Time Signal Processing,” 2nd ed, Prentice Hall, Upper Sadle River, New Jersey, 1999.

[3] David Reynolds, Stacy Ho, "An Integrated 12 Bit Analog Front End For CCD Based Image Processing Applications," Symposium on VLSI Circuits Digest of Technical Papers, Vol. 41, pp 96-97, 1997.

[4] Hans P. Tuinhout, Heinze Elzinga, J.T. Brugman, Fokke Postma," The Floating Gate Measurement Technique for Characterization or Capacitor Matching," IEEE Transactions on Semiconductor Manufacturing, Vol. 9, No. 1, pp 2-8, February 1996.

[5] Personal communication with Hans P. Tuinhout

[6] T. Cho, "Low Power Low Voltage Analog To Digital Conversion Techniques Using Pipelined Architectures," PhD Thesis, Berkely, 1995.

[7] B. A. Wooley, "Design of Low-Power, Low-voltage A/D Converters," Electronics Laboratories Advanced Engineering Course on Architectural and Circuit Design for Portable Electronics Systems, Lausanne Switzerland, June 1997.

[8] Analog Devices, Inc., ”Analog-Digital Conversion Handbook,” Prentice Hall, Upper Saddle River, New Jersey, 1986.

[9] David W. Cline, Paul R. Gray, “A Power Optimized 13-b 5Msamples/s Pipeline Analog-To-Digital Converter in 1.2um CMOS.” IEEE J. Solid State Circuits, Vol 31, No 3, March 1996.

[10] Phillip E. Allen, Douglas R. Holdberg, “CMOS Analog Circuit Design,” Harcourt Brace Jovanovich, Inc., Orlando, Florida, 1987.

[11] Paul R. Grey, Robert G. Meyer, “Analysis and Design of Analog Integrated Circuits,” 3rd ed, John Wiley and Sons, Inc., New York, 1993.

[12] David Johns, Ken Martin, “Analog Integrated Circuit Design,” John Wiley and Sons, Inc., New York, 1997.

[13] Yannis Tsividis, “Operation and Modeling of The MOS Transistor,” 2nd ed, WCB/McGraw-Hill, Boston, Massachutsetts, 1999.

[14] Sung-Mo Kang, Yusuf Leblebici, “ CMOS Digital Integrated Circuits,“ 2nd ed, WCB/McGraw-Hill, Boston, Massachutsetts, 1999.

[15] Rudy van de Plassche, “Integrated Analog-to-Digital and Digital-to-Analog Converters,” Kluwer Academic Publishers, Boston, Massachutsetts, 1994.

[16] Ion E. Opris, Laurence D. Lewicki, Bill C. Wong, “A Single-Ended 12-bit 20 Msample/s Self-Calibrating Pipeline A/D Converter,” IEEE J. Solid State Circuits, Vol. 33, NO. 12, December 1998.

[17] Larry Singer, Stacy Ho, Mike Timko, dan Kelly, “A 12b 65Msample/s CMOS ADC with 82dB SFDR at 120MHz,” ISSCC 2000, Session 2, Paper MP 2.3, 2000.

[18] Stephen G. Kockram, “Programming in ANSI C,” Revised ed, Sams Publishing, Indianapolis, Indiana, 1994.

[19] Krishnaswamy Nagaraj, Scott Fetterman, Joseph Anidjar, Stephen Lewis, Robert Renninger, "A 250-mW, 8-b, 52-Msamples/s Parrallel-Pipelined A/D Converter with Reduced number of Amplifiers," IEEE J. Solid State Circuits, Vol. 32, NO. 3, March 1997.

[20] Thomas Cho, Paul Gray, "A 10-b, 20 Msample/s, 35 mW Pipeline A/D Converter," IEEE J. Solid State Circuits, Vol. 30, NO. 3, March 1995.

[21] Seung-Hoon Lee, Bang-Sup Song, "Digital Domain Calibration of Multistep Analog-to-Digital Converters," IEEE J. Solid State Circuits, Vol. 27, NO. 12, December 1992.

[22] Myung-Jun Choe, Bang-Sup Song, Kantilal Bacrania, "A 13-b 40MSample/s CMOS Pipelined Folding ADC with Background Offset Trimming," ISSCC 2000, Session 2, Paper MP 2.2, 2000.