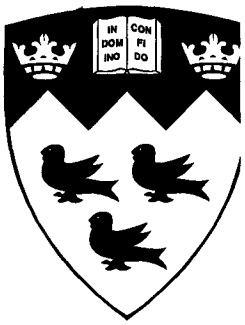
Nyquist-Rate Analog-to-Digital Conversion with Calibration

by

Bardia Pishdad, B. Eng. 1997

Department of Electrical and Computer Engineering

McGill University, Montréal, Canada



July 2002

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfilment of the requirements for the degree of Master of Engineering

© Bardia Pishdad, 2002

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services
Acquisisitons et services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-85895-2
Our file Notre référence
ISBN: 0-612-85895-2

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

Canada

Abstract

Analog-to-Digital Converter (ADC) microcircuits are required to meet stringent accuracy specifications in spite of their analog components' inherent nonidealities as well as the accuracy limitations due to fabrication technology. In this thesis, a 3-step Nyquist rate ADC is presented which makes use of bitstream processing to calibrate the digital-to-analog converter (DAC) and the residue amplifier, while using the same hardware to calibrate the sub-ADC. The system is designed to provide programmability and calibrate undesired circuit characteristics such as offset, gain error, and nonlinearity. Thus, the DAC can tolerate gain errors much higher than the standard amount, and has the potential to completely cancel nonlinearity and offsets in its transfer function. The offset of the residue amplifier can also be calibrated with this system. Moreover, the system eliminates the need for a reference ladder in the sub-ADC, and calibrates comparator offsets. Simulation and experimental results of the circuits fabricated in a 0.18μm CMOS process are presented.

Résumé

Les convertisseurs analogique-numérique (CAN) microélectroniques doivent satisfaire des exigences élevées en terme de précision, et ce en dépit des non-idéalitées de leurs composants analogiques et des limitations de précision inhérents au processus technolologique de fabrication. Dans cette thèse, on propose un CAN à 3 étapes qui utilise le traîtement de signaux numériques afin de calibrer le convertisseur numérique-analogique (CNA) et l'amplificateur. Le système est conçu de façon programmable, pour permettre de calibrer les défaults du circuit tels que des décalages d'amplitude, des erreurs de gain, et des non-linéarités. Ainsi, le CNA peut tolérer des erreurs de gain beaucoup plus élevées que la quantité habituelle, et peut complètement annuler les non-linéarités et les décalages d'amplitude dans sa fonction de transfert. Le décalage d'amplitude dans l'amplificateur peut également être calibré avec ce système. De plus, le système élimine le besoin pour une chaîne de résistance à la base du CAN, et parvient a calibrer le décalage d'amplitude de ses comparateurs. Les résultats de simulation et d'expérience du prototype de ce circuit fabriqué dans une technologie CMOS 0.18μm sont également présentes.

Acknowledgments

First and foremost, I would like to thank my research advisor, Professor Gordon W. Roberts, who has been an exceptional academic role model, and has consistently granted me with the insights, guidance, and encouragement required to pursue this work.

I have benefitted from interacting with many good folks at the MACS lab. My thanks to Antonio and Clarence for the good spirit in MC 544C, and to all the others who have made my stay here memorable: Ahmed, Arshan, Boris, Christian, Geoff, Henry, Lige, Melita, Mohamed, Mona, Mourad, Naveen, Nazmy, Rola, Sebastien, Tommy, Wei-wen, Yang, and Ye. I would also like to thank our system administrators who have kept the network running smoothly, Ben, Carl, Ehab, and Hugo.

This work would not have been possible without the help of the Canadian Micro-electronics Corporation for overseeing and funding the IC fabrication process, and Micro-net for providing research assistantship. I would like to acknowledge both of these organizations for their support.

I am grateful to all my friends, to the ones close by for sharing memorable times in Montreal, and to the far away ones for staying in touch.

Last, and most importantly, I extend a special thank you to my family and relatives for their generous love and support throughout my life. I am greatly indebted to them for being where I am today.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analog-to-Digital Converter
DAC	Digital-to-Analog Converter
IC	Integrated Circuit
LSB	Least Significant Bit
MSB	Most Significant Bit
INL	Integral Nonlinearity
DNL	Differential Nonlinearity
FSR	Full-Scale Range
PDM	Pulse-Density Modulation/Modulated
RMS	Root-Mean-Square
LPF	Low-Pass Filter
BPF	Band-Pass Filter
FIFO	First-In-First-Out
ATE	Automated Test Equipment
I/O	Input/Output (pin)

Chapter 1 - Introduction

1.1 - Motivation of Research

The trend towards shifting larger shares of signal processing functions from the analog domain to the digital realm has been prominent in the microelectronics industry over recent years. Despite this now well established paradigm, there remains the constant need to perform the conversion of a naturally occurring analog electrical signal into digital form. Moreover, to keep the analog signal processing to a minimum, the analog-to-digital converter (ADC) needs to be placed as early as possible in the circuit chain, thus not benefitting from prior signal processing or conditioning. This places great demands on these circuits in terms of linearity, noise rejection, and speed. One subclass of ADCs, known as delta-sigma (Δ–Σ)-based ADCs, alleviate some of these problems by relying on oversampling and feedback to correct the quantization error. Nevertheless, all ADC circuits being partially analog in nature, are subject to many inherent circuit non-idealities, such as the non-infinite gain of operational

amplifiers, their limited bandwidth, or the finite resolution of comparators. In addition, their implementation in today's microelectronics fabrication technology is faced with issues such as the mismatch and tolerance of components, variations due to temperature, and gradients across silicon wafers. These issues are especially important in another class of ADCs known as Nyquist-rate ADCs. As opposed to oversampling converters which can accept a limited signal range, traditionally in the speech and audio band, Nyquist-rate ADCs can operate up to half of the sampling frequency known as the Nyquist frequency. In light of the design constraints confronting this class of circuits, a need has emerged for techniques which can somehow alleviate these detrimental conditions. Error cancellation or correction techniques aim to measure an error signal and to subtract it from the final output in digital or analog domain. Another method is that of calibration, whereby on-chip circuits can be modified to operate at certain desired conditions. Although potentially more complex than error correction, they do offer the benefit of having the circuit become fully accurate after the initial calibration routine, and avoid the need for further correction during normal operation. In the realm of ADCs, the use of adjustable passive elements or laser trimming, which are analog in nature, has been in practice.

In this thesis, it is sought to use digital modulation as the basis for generating accurate analog signals which will then be used for the calibration of an ADC. The advantages and drawbacks of the modulation scheme in the context of the Nyquist-rate ADC will need to be explored. The calibration must aim to eliminate the undesired characteristics of the circuit such as offset, nonlinearity, and gain error. It is intended to obtain a system which has enough versatility to address these issues, and is able to exploit the simplicity of

a digital interface, all without compromising the needs of a modern-day Nyquist-rate ADC.

1.2 - Outline of Thesis

This thesis begins by providing an overview of the three most popular submicron ADC architectures in Chapter 2. The focus is on identifying error sources, and introducing some of the methods used for correcting them. The architectures of interest are the flash, folding/interpolating, and multistep/pipeline ADCs. Chapter 3 introduces the concept of pulse-density modulation, and how it can be used to generate DC voltages. Its application to the design and calibration of a Nyquist-rate ADC will be discussed through the presentation of system-level diagrams and calibration algorithms. It will be seen how some of the error mechanisms detrimental to the ADC can be corrected for with the proposed topology and algorithms. In Chapter 4, the circuit-level details of the ADC will be described including relevant design challenges. Chapter 5 will describe the work done related to the experimental prototype including system and circuit level simulations, IC design and layout, and testing and troubleshooting. Finally Chapter 6 will close the thesis by summarizing the thesis and outlining future improvements which can be made to the system.

Chapter 2 - ADC Overview

In this chapter, an overview of some popular ADCs is presented. After defining some of the basic terms used to quantify ADC behavior and performance, three ADC architectures will be presented. For each one, advantages and drawbacks will be outlined, with an emphasis on the error sources limiting conversion accuracy. Some of the more interesting methods that have been in place for compensating accuracy limitations and improving resolution are described for each architecture.

2.1 - ADC Metrics

2.1.1 - Offset

The offset can be defined in a number of ways, depending on the context. A general definition states that the offset of the ADC transfer curve is the least amount of deviation from the ideal value, common to all the output codes. This is seen in Figure 2.1(a). In this case, we can say that the whole transfer curve is 'offset' by x number of LSBs. A more current defini-

tion is that the offset is the deviation from the ideal value of the output code corresponding to a zero input level. For testing and characterization, this second definition, illustrated in Figure 2.1(b) could be misleading as a larger deviation in the code corresponding to the zero input level than the other codes would exaggerate the offset rating. As a result, a more accurate method is employed, which consists of first drawing a best-fit line through all the output codes first, and then measuring the offset as the deviation of the best-fit line's value corresponding to a zero input. This is shown in Figure 2.1(c). This third definition of offset is the one which will be referred to in this thesis.

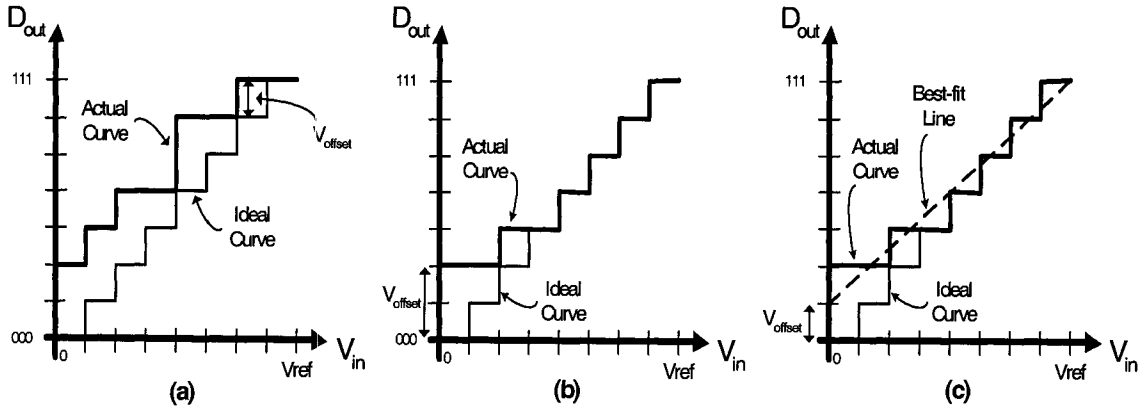


Figure 2.1. Different Representations of Offset

2.1.2 - Gain Error

Gain error refers to the deviation from an ideal gain. It can be quantified with the following formula, often expressed as a percentage:

$$GainError = \frac{ActualGain - IdealGain}{IdealGain} \quad (2.1)$$

2.1.3 - Differential and Integral Nonlinearity

The differential nonlinearity (DNL) for a particular output code of an ADC refers to the deviation in its width from the nominal value of 1 LSB. The integral nonlinearity (INL) at each code is the sum of all the DNLs up to and including that code. INL and DNL are illustrated in Figure 2.2.

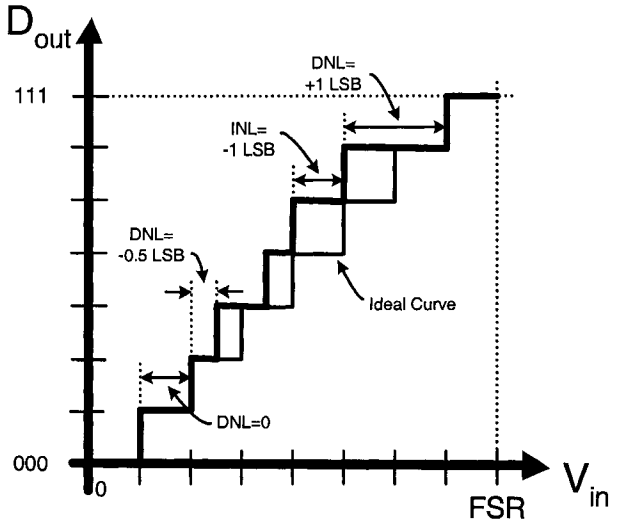


Figure 2.2. DNL and INL on the ADC Transfer Curve

2.1.4 - Full-scale Range

Full-scale range (FSR) refers to the range of voltage allowed for the input signal, as in a FSR of 0.5-2.5 V. It can also refer to the difference between the maximum and minimum allowable input voltages, for example FSR=2 V.

2.2 - The Flash ADC

2.2.1 - Architecture

The simplest architecture for high-speed ADCs is the flash ADC. A diagram for a flash ADC is shown in Figure 2.3. For an N-bit ADC, 2^N equally spaced reference voltages, usually generated by a resistor divider chain, are connected to the negative inputs of 2^N comparators. The outputs of the comparators form a sequence of binary values which is commonly referred to as a thermometer code. This thermometer code is then converted to a standard N-bit binary code via standared digital decoding. A single front-end track-and-hold or sample-and-hold circuit may be needed to lower the required dynamic performance of the comparators [12], and to control the sampling uncertainty associated with a large number of comparators [14]. Its design is subject to distortion, bandwidth, and noise requirements at the desired frequency [12].

2.2.2 - Drawbacks and Error Sources

This architecture has the potential of achieving very high conversion speeds as it maximizes parallelism by producing all N bits in one stage. Thus, recently reported ADCs with sampling rates above 1 GSample/s in a CMOS technology make use of this architecture [7] [8]. However, due to the exponential number of resistors and comparators required, it has the drawbacks of excessive die area and power consumption, making it an expensive choice for over 8 bits of resolution. As well, the resulting large input capacitance can create bandwidth limitations adverse to the intended speed benefit of this architecture. As for the resolution, non-idealities in the fabrication process limit it to a

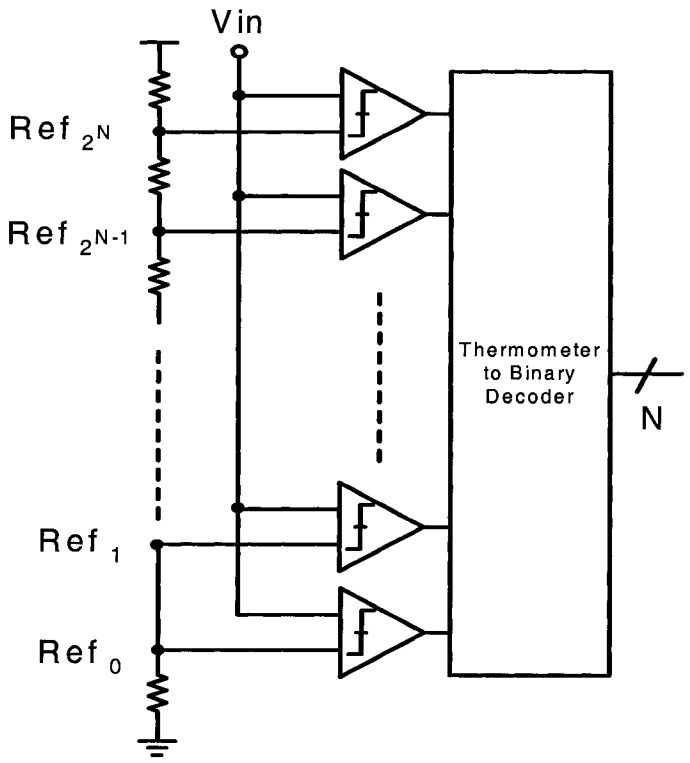


Figure 2.3. Architecture of a Flash ADC

maximum of 10 bits [6]. The relative mismatch in resistor values and random offsets in the input differential pairs of comparators have been two such non-idealities plaguing the design of flash ADCs. Therefore, this architecture is ideally suited to low-resolution, high-speed applications, an example of which is the read channel of data storage systems [7] [11]. At high speeds, designing a single input track-and-hold with good linearity is difficult [12], and therefore a distributed track-and-hold architecture has been implemented,

where each track-and-hold requires good linearity over the part of the input range that it handles [14].

The offset problem is often addressed by auto-zeroing techniques, which involve the pre-storing of the offset during one half of the clock cycle, within some kind of switched-capacitor configuration. However, high-speed applications for which flash ADCs are best suited cannot tolerate this idle time. To reduce the offset problem without resorting to auto-zeroing techniques, the method of averaging, originally developed in [16] has been successfully applied to the preamplifier stages of a number of flash and folding ADCs [7] [8] [9] [10] [13] [17]. The original implementation of averaging is as shown in Figure 2.4(a). The outputs of a bank of differential pairs typically used in a flash ADC are coupled through resistors. If the linear range of the amplifiers are overlapping, as shown in Figure 2.4(b), then the voltage at the output voltage of amplifier *i* will be set by the current in its own differential pair *as well as* currents from neighboring differential pairs. Thus, for a given offset or noise voltage on differential pair *i*, its output voltage will be affected by the combination of its own noise and the noise in neighboring cells. Since this noise distribution is usually random with an average tending towards zero, the net effect at the output of differential pair *i* will be that the noise currents will also tend towards zero. Thus, the output voltage of differential pair *i* will be affected much less than if all of its current flowed only through the load resistor R_L. Of course, the exact amount of this averaging effect will vary, and depends on amplifier gain, the ratio of R_{AVG} to R_L, and the number of amplifiers that are in the linear region [16].

The explanation of averaging given in [16] is as follows: Considering Figure 2.4(a) again, suppose at first that there is perfect matching between the tail currents and that no amplifier offsets exist. With the input signal equal to Ref_2, the middle amplifier is perfectly balanced. In this condition, the other amplifiers inject DC current into the averaging resistors R_{AVG} but the middle amplifier does not [16]. Thus, its AC gain is only determined by the load resistor R_L. As this amplifier leaves its balance point, the averaging resistor begins to draw some current, reducing the effective load impedance to a combination of R_L and R_{AVG}. Thus, the gain is reduced, and a filtering of the noise components has taken place.

A quantitative analysis was carried out to better understand the dynamics of averaging. It goes as follows: Suppose that the linear ranges of the five amplifiers overlap, as shown in Figure 2.4(b). Now, suppose that an offset voltage on the middle (3^{rd}) differential exists such that a noise current Δi_3 appears on M_3. A portion of it, $a_3 \bullet \Delta i_3$, will flow through R_{AVG} while the remainder will flow through R_L. Similarly, if another noise current Δi_2 appears through M_2, then it will inject a similar current $a_2 \bullet \Delta i_2$ through R_{AVG}. In general, the net current through R_{AVG} will be of the form,

$$\Delta i_{net_3} = a_1 \cdot \Delta i_1 + a_2 \cdot \Delta i_2 + a_3 \cdot \Delta i_3 + a_4 \cdot \Delta i_4 + a_5 \cdot \Delta i_5 \qquad (2.2)$$

Thus, Δi_{net_3} is the sum or average of the individual noise currents. From that, one can calculte the *change* of voltage at V_{out3} as,

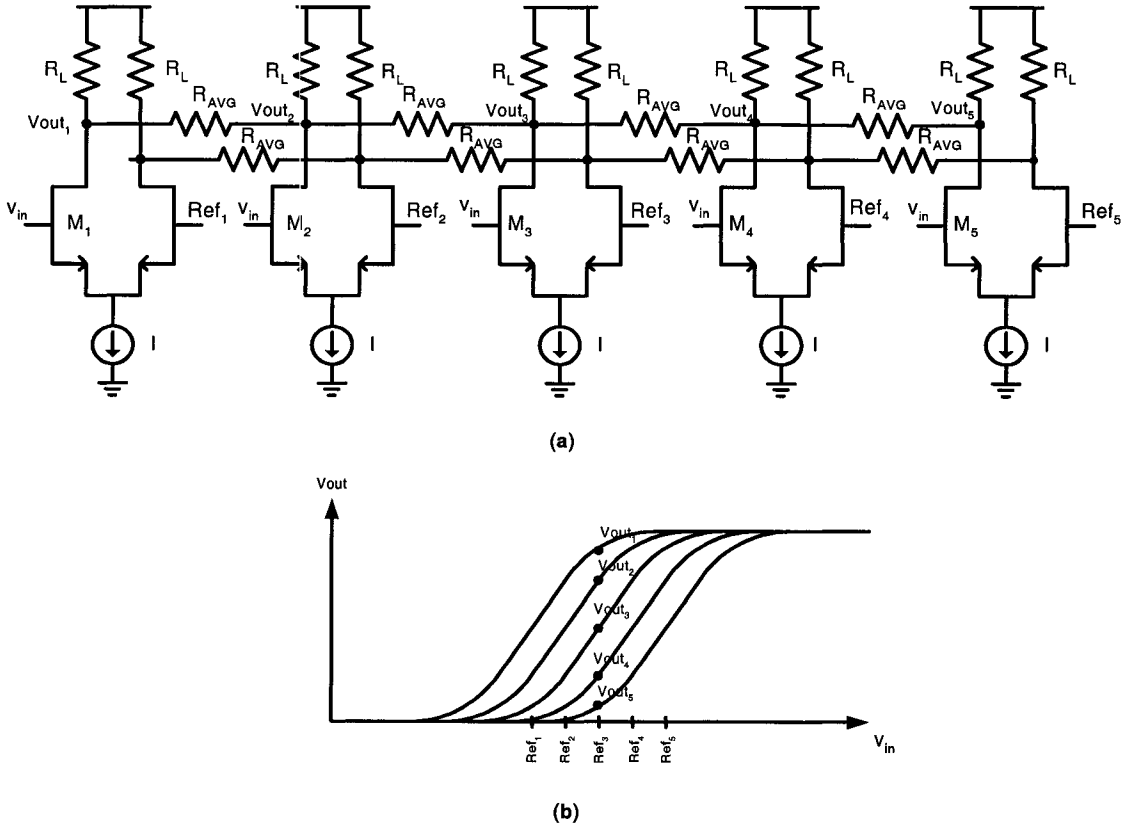


Figure 2.4. a) Averaging for ADC Differential Pairs b) Transfer Curve of Adjacent Differential Pairs

$$\Delta v_3 = \Delta i_{net_3} \bullet R_{AVG} \tag{2.3}$$

Since the offsets, and therefore the Δi noise currents , have a random distribution, Δi_{net_3} will be an averaged value smaller than the voltage change at the output of a single amplifier without averaging resistors.

The integrating resistors in this example can be seen as implementing a convolution of the noise currents with the coefficients a_1-a_5. This is equivalent to a discrete filter function with the exception that the discrete samples are the physical output nodes in a specific spatial arrangement rather than as a function of time. This type of filtering is referred to as spatial filtering. The smaller R_{AVG} is with respect to R_L, the more averaging takes place.

2.3 - The Folding/Interpolating ADC

2.3.1 - Folding

Resolving the problem of the large number of comparators needed for flash ADCs is the motivation behind the folding ADC. The idea of this architecture is to precede the comparator stage by a preprocessing stage which will allow the use of one comparator to output the decision corresponding to many reference levels. Let's first denote the event of the input signal crossing a reference level by the term 'reference crossing'. The function of the folding block is to combine many reference crossings into just one pair of complementary folding signals. That way, the folding signals can feed a single comparator, which then outputs the decision for those reference levels. The diagram of the folding block is shown in Figure 2.5.

Figure 2.6 shows the circuit implementation of a folding block. An odd number of reference levels (9 in total, 5 of which are shown on the diagram) are connected to the inputs of differential pairs, with the other input of each differential pair connected to the input signal V_{in}. The output currents are summed together with the connections reversed

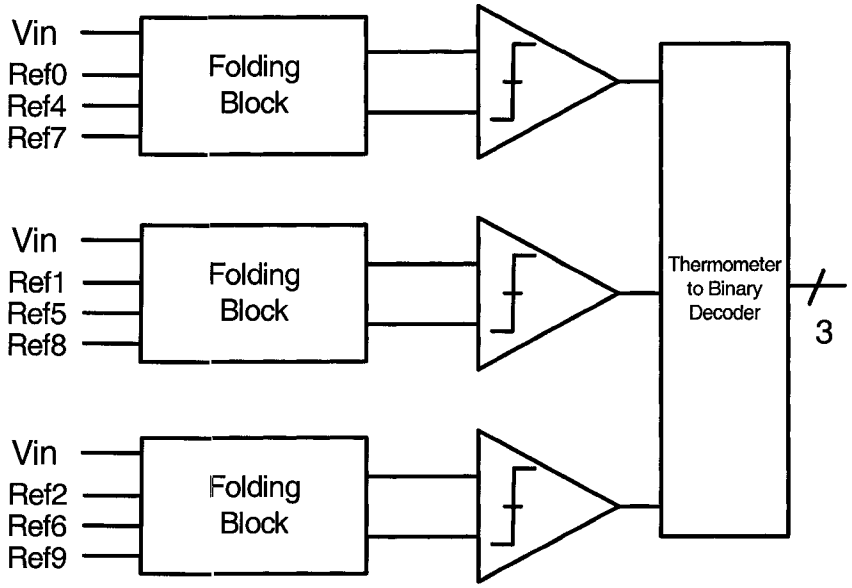


Figure 2.5. Incorporation of Folding Blocks in ADC

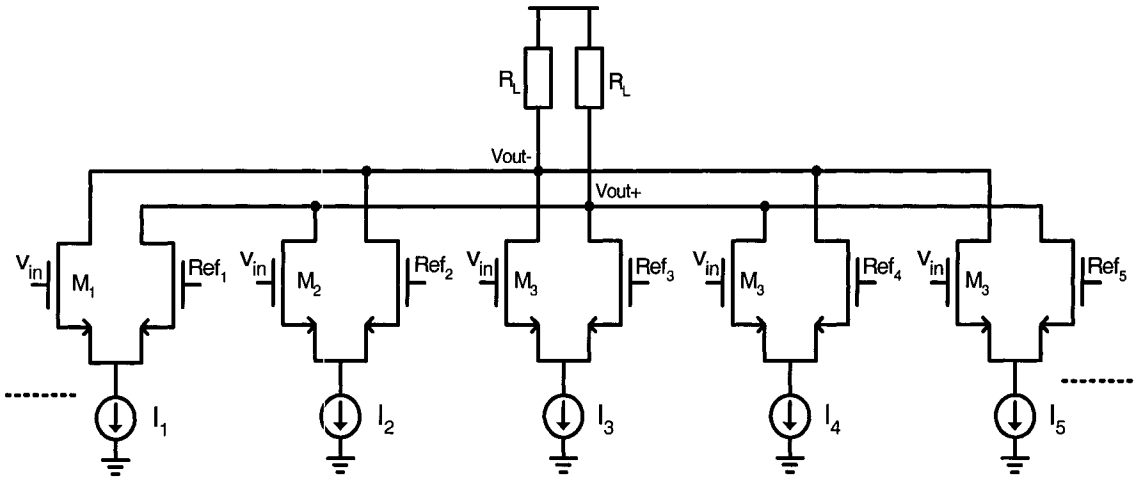


Figure 2.6. Circuit Implementation of Folding

from one differential pair output to the next. This specific configuration is what creates the folding effect. Figure 2.7 shows the resulting folding signal as the input signal is ramped up from 0 through 8 reference levels to which the folding amplifier is connected. A 9^{th} reference level is added in order to have an odd number of reference levels, which creates the required initial imbalance in the currents. At first when V_{in}=0, an imbalance exists in the load currents due to the odd number of differential pairs, with 5I through R_1 and 4I through R_2. Thus, Vout+ =0 and Vout- =1. As V_{in} crosses Ref_1, the current balance in the corresponding differential pair tips, which results in 4I through R_1 and 5I through R_2. This will cause the output signals and the output of the corresponding comparator to toggle, in turn causing the thermometer code to increment by one step. The term folding factor *F* is often used to refer to the number of references combined in one folding block.

Which references are combined together into the single folding pair of signals is the key here. In the example of Figure 2.7, we note that the digital code of the ADC has a repeating pattern of the 5 lower bits as the input is ramped from 0 to the full-scale range. In fact, this recurring pattern will appear $2^{(8-5)}$ times. Therefore, the references corresponding to the same 3-bit code in each of the 2^3 intervals throughout the input range are combined together. This will make each comparator output correspond to the same transition in the M-bit code. A zero crossing detection system, as implemented with folding, is more robust than a standard reference crossing system [9] since the processing is nonlinear and only the zero crossing position is of interest. In addition, the modest gain of the folding circuitry (optimized mostly for bandwidth) helps the accuracy of the comparison.

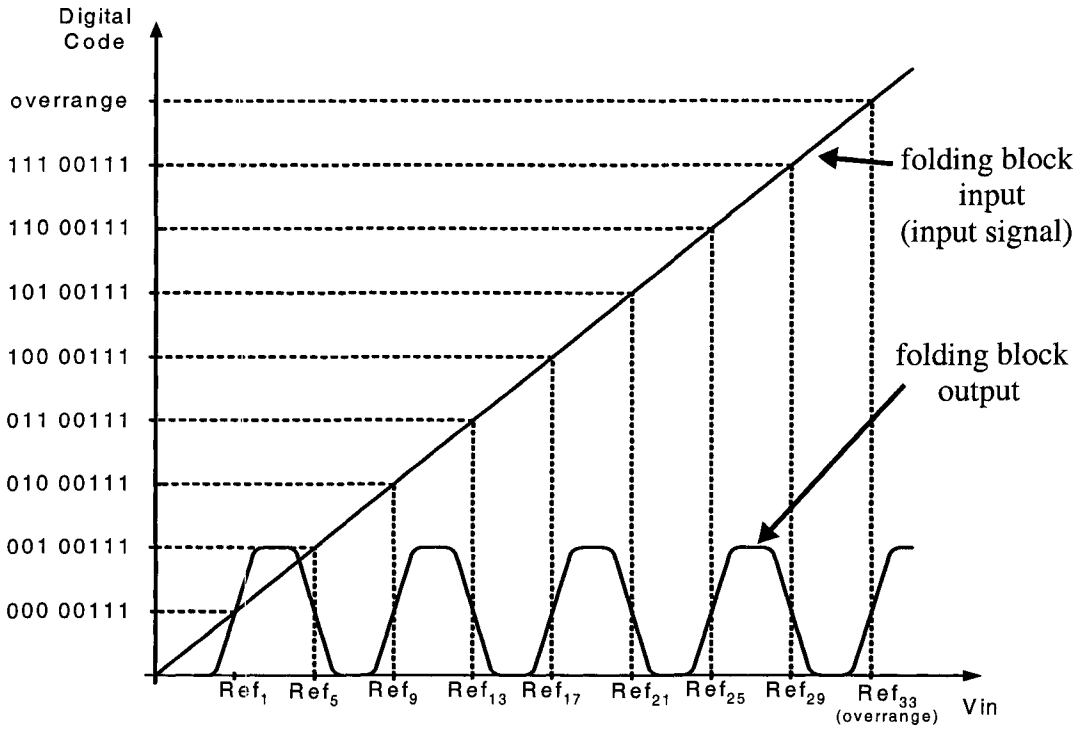


Figure 2.7. Transfer Curve of Folding Block

2.3.2 - Interpolation

Although folding reduces the number of comparators by the folding factor, it does create the need for 2^N differential pairs. As a result, interpolation is often used in combination with folding to reduce the number of differential pairs. The implementation of interpolation and the resulting interpolated signal are shown in Figure 2.8. The idea is to substitute some of the folding blocks by a resistor divider between their bordering folding block outputs. This theoretically provides a number of equally spaced zero crossings

between the folding block outputs. It is important that the latter be close enough to each other so that their linear regions overlap almost as much as possible, making resistive interpolation possible. The accuracy of the interpolated signals is only critical at the zero crossings, and therefore the only requirement is that the zero crossings not be offset from their corresponding reference crossings. The actual shape of the interpolated signals at points other than the zero crossing is not important [9].

Interpolation provides a significant savings in area. A total of n folding blocks each comprised of F differential pairs are replaced by $2 \bullet (n+1)$ resistors. In addition, power consumption is considerably reduced since the interpolating resistors consume no additional power, unless the tail current of the existing folding blocks needs to be adjusted to increase the gain. The interpolating resistors also inherently serve as averaging resistors. Thus, Folding/interpolation ADCs which also take advantage of averaging have been implemented in various publications [9] [10] [13] [17].

2.3.3 - Overall Architecture

The overall architecture for an 8-bit folding/interpolating ADC is shown in Figure 2.9. The 5-bit fine ADC is preceded by a preprocessing circuit which implements folding and interpolation, while the 3-bit coarse ADC is performed directly. From this figure, the role of folding is clearly seen: The 5-bit fine quantizer is not only used for the first 32 of the 256 fine codes, but also for each of the subsequent regions of the 3-bit coarse quantizer. Therefore, the 3-bit quantizer is only a region selector, which provides the MSB (or region) information which is not captured through the folding operation. Note that both the 3-bit and 5-bit quantizers need to have full 8-bit accuracy. This is because

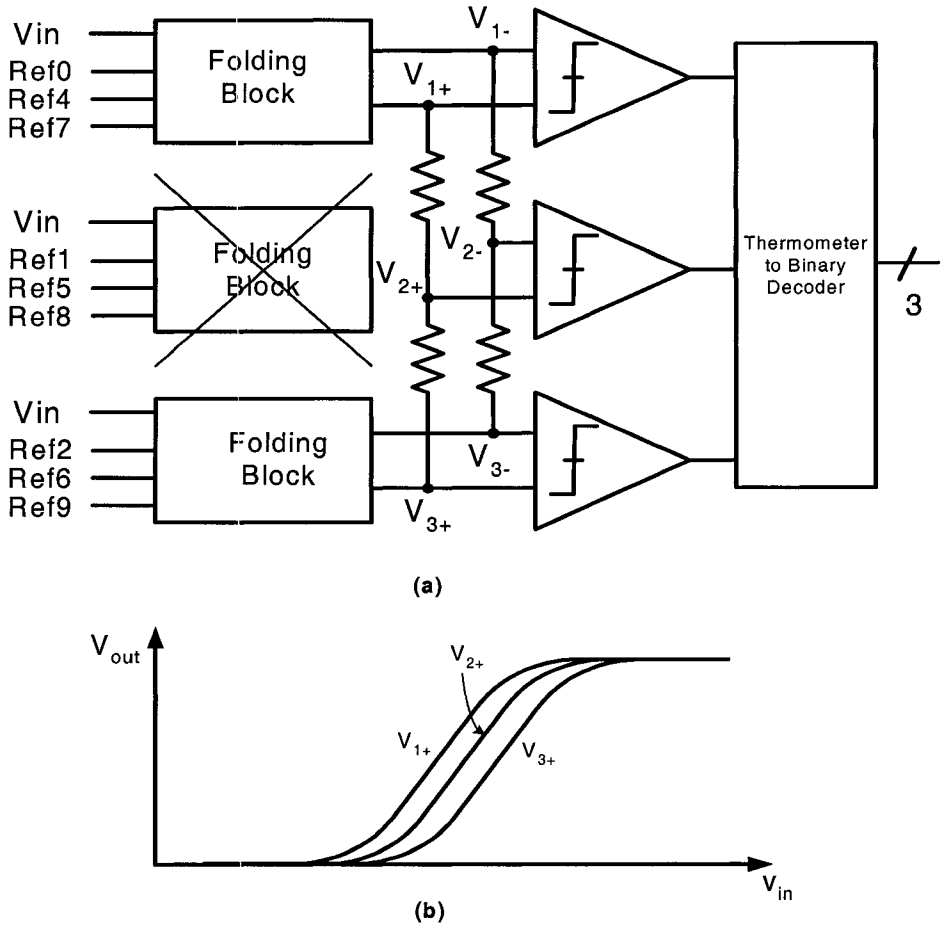


Figure 2.8. a) Implementation of Interpolation b) Resulting Signals

although the conversion is split in 2 parts, they are done in parallel just like a flash ADC. The inputs to the fine comparators benefit from the gain of the folder circuit, which although made small for fast settling, do make the fine comparators less offset-sensitive.

Decoding and synchronization are straightforward in the digital domain, and similar to the case of the flash ADC.

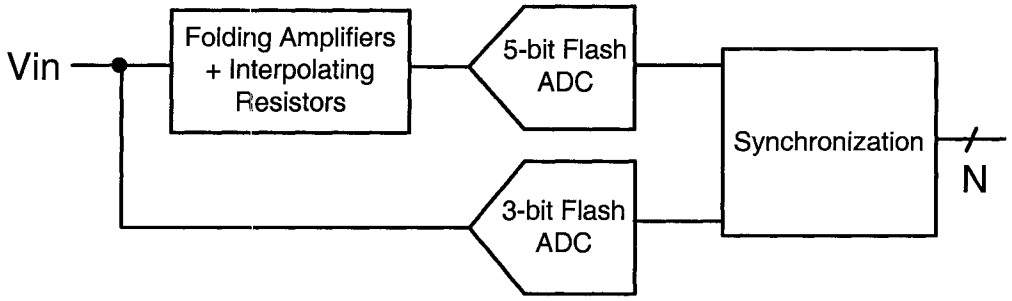


Figure 2.9. Overall Architecture of Folding/Interpolating ADC

The folding factor F in this example is 9, and the interpolation factor I is 4. In order to generate the required 2^N zero crossings, a total of n folding blocks must be used such that $F \bullet I \bullet n \geq 2^N$. Therefore n=8 in this case, which is also the number of comparators needed. The coarse quantizer requires $2^3 = 8$ comparators. The reference chain needs only $(F \bullet n) - 1$ resistors. This does not diminish the critical N-bit accuracy requirement of that circuit, but does have the benefit of reducing the total number of contacts, which in practice do have a certain spread in their resistance value [10]. Table 2.1 com-

pares the number of comparators and resistors required for a N-bit flash ADC and its equivalent folding/interpolating implementation.

TABLE 2.1. Hardware Requirements for Flash and Folding/Interpolating ADCs

	Flash ADC	Folding/Interpolating ADC
Number of Comparators	2^N	$\sim 2^N/F$
Number of resistors	2^N	$\sim (2^N/I) + (n \bullet (I+1))$

2.3.4 - Error Sources

The folding/interpolating architecture has been widely recognized for alleviating the exponential area, power, and input capacitance dependance of the flash architecture [9]- [15]. Nevertheless, there are known design problems, two of which are described here.

The interpolation should ideally generate equally fine crossings between the coarse zero crossings. However, since the DC transfer curve of the differential pairs is not linear [15] in their rise and fall regions, some of the interpolated crossing are 'pulled' towards the folding signals. This is shown in Figure 2.10, where the output signals corresponding to a 4-time interpolation are shown. This results in interpolation errors of 0.1 LSB and 1.3 LSB for reported 8-bit and 12-bit ADCs in [15] and [14]. However, a higher number of folding amplifiers over the input range would bring the folding signals closer to each other and make the interpolation occur over the more linear part of the slope. Thus, a

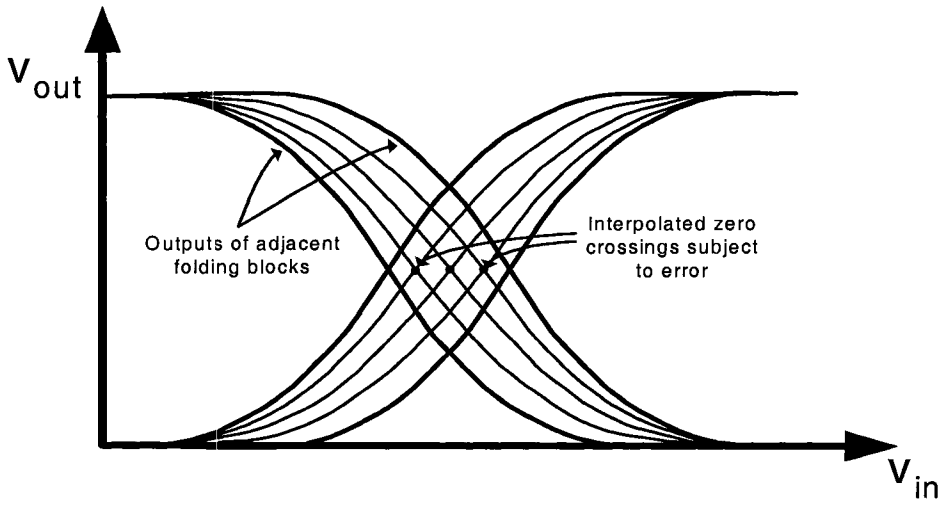


Figure 2.10. Interpolation and Associated Error

trade-off exists between the interpolation nonlinearity and the number of folding blocks [11]. Solutions to the interpolation problem have been proposed in active interpolation [9], where a differential pair is used in place of each resistor to obtain one interpolated crossing. This implementation does alleviate the aforementioned problem, although it is rather area and power inefficient [25]. A clever alternative was proposed in [10] where, say for a 3-time interpolation, instead of using the error-prone interpolation signals at the 1/4 and 3/4 taps, the intersection of the folding signals with the middle interpolated signal (the 1/2 tap) at a *different common-mode level* is taken. This works fine since for the detection of a crossing, the common-mode level is not relevant, and the middle interpolated signal does not suffer from the aforementioned nonlinearity problem. However,

beside the area savings obtained from omitting the error-prone reference taps, no mention of the linearity of the zero crossings obtained through this implementation is made.

As for the folding circuit, it is susceptible to tail current mismatches of the differential pairs within one folding block [11]. To see this, consider Figure 2.11. When Ref_1 is being crossed, the slope of the current determines the position of the zero crossing. Now, if I_1 is mismatched from the nominal current, that differential pair's gain and therefore the slope of its output in the linear range will change, which will in turn shift the position of its zero crossing. This deviation also affects the interpolated zero crossings to a lesser degree. In general, gain mismatches among folding blocks, caused by mismatches in load value, g_m, or any other factor, will appear as a shift in the zero crossing. The reader is referred to [11] for a quantitative analysis of this problem and the calibration algorithm. One more pertinent issue for folding amplifiers is that since the folding operation is equivalent to a frequency multiplication, the folding circuitry requires high bandwidth for Nyquist-rate operation.

2.4 - Multistep and Pipeline ADCs

2.4.1 - Overall Architecture

Instead of resolving all N bits in one stage, one can choose to partition the conversion operation in steps of incrementally finer sub-conversions. This is the method of multi-step analog-digital conversion. The first stage performs a coarse conversion, and passes on the error, or the difference between the input voltage and the analog equivalent of the digital output, to the next stage for finer quantization. This continues in the same

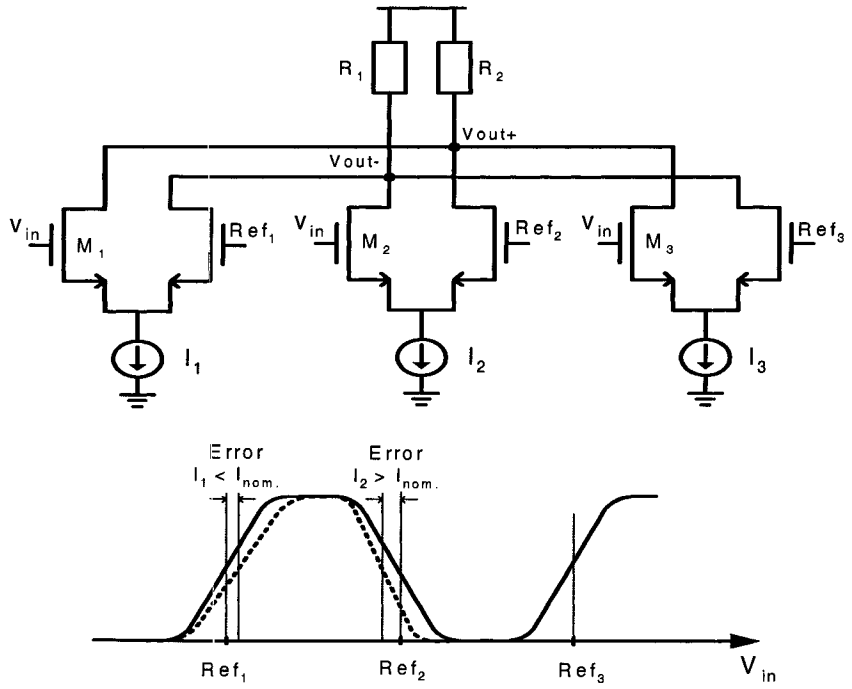


Figure 2.11. Current Source Mismatch in Folding Amplifiers

fashion for the subsequent stages, until the N bits are generated. This is called a subranging ADC, and its architecture is shown in Figure 2.12. In Figure 2.12(a), each ADC needs to be N-bit accurate. The later ADC stages exercise smaller and smaller parts of the input range, and therefore only part of the input reference voltages are used in each stage. In order to be able to use the same ADC design for each stage, and operate over the entire input range, the residue can be amplified to the FSR before being fed to the next stage. With the first stage converting M bits, this requires a 2^M-time amplification in the input range of the second stage ADC. Since the signal at the input of the second stage is amplified by an effective M bits, that stage now only has to have $N-M$ bits of linearity. Put in

other words, the nonlinearity of that stage, when referred back to the input, is reduced by the gain that precedes it. This reduced accuracy requirement provides the option of scaling down some of the components in the later stages. Figure 2.12(b) shows the completed topology of a multi-step ADC with bit requirements for each circuit. Note that only the first stage ADC and DAC need to have full N-bit linearity.

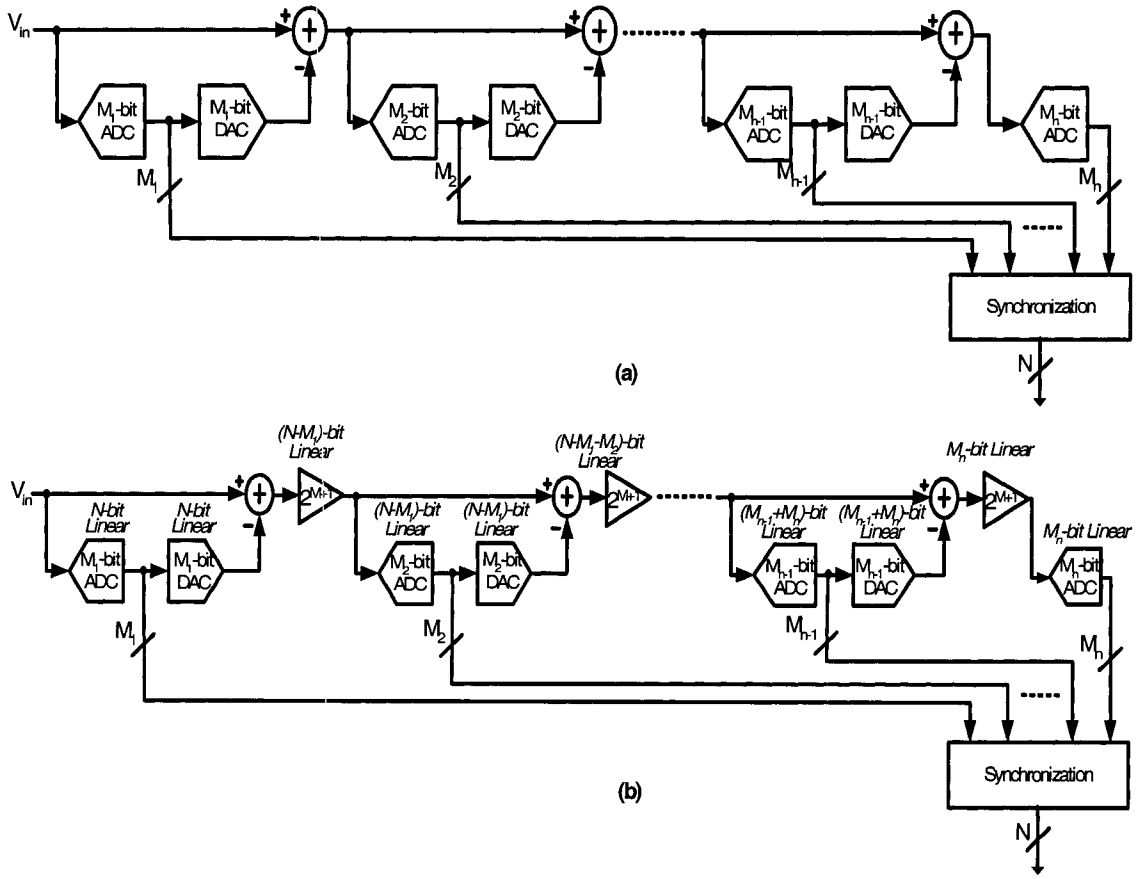


Figure 2.12. (a) Subranging ADC (b) Multistep/Pipelined ADC

Due to the serial nature of this converter architecture, a latency is required at the beginning until the first set of bits are available. After that, the data operates in a pipeline fashion with the throughput rate equal to the rate of conversion of one stage. Whether the latency is an issue or not depends on the application. In the extreme case, a 1-bit/stage implementation will have a latency in clock cycles equal to the total number of bits. Such fully 'serialized' ADCs are more often referred to as pipelined ADCs, while 'multi-step' typically refers to a small number of stages (typically 2-4).

The ADC in each stage is usually of the flash type, as described in Section 2.2. The accuracy of the DAC is usually limited by the matching of passive components and the opamp finite DC gain. Therefore, capacitor-based binary-weighted DACs have been favored in recent years due to their better matching available in IC fabrication technology. The residue generation circuit usually consists of an amplifier in a switched-capacitor network which implements the subtract-and-multiply function. A multi-phase clocking scheme (2 or 3 phases) is used to control the switching of the capacitors. The amplifier is operated in a feedback configuration, and hence the residue block is the slowest block in the ADC stage. If a capacitor-based DAC is used, it can be enhanced with a subtract-and-multiply function through additional switches and capacitors, thus combining both the DAC and the residue generation into a single multiplying DAC (MDAC).

2.4.2 - Error Sources

It is informative to view the error sources as they appear on the transfer function of one ADC stage and how they project on the overall ADC transfer function. Since the stage output is referred to as the residue, the stage transfer function is also commonly

called the residue curve. Figure 2.13 shows the main instances of such errors. The bottom curve in each case represents the transfer curve of the residue amplifier, while the top curve is the ADC's overall transfer curve. Note that in Figure 2.13(c), due to the higher gain, missing decision levels occur over the range of the input signal. This is seen a horizontal jump in the ADC transfer function. That is, k samples of the input signal (with $k>1$) map to the same clipped residue value V_{ref}, whereas the mapping should ideally be one-to-one. Thus, the number of decision levels are reduced by this number of input samples minus one. In contrast, the pattern displayed in Figure 2.13(d) suffers from missing codes which can be seen by the fact that codes situated within the vertical jump on the curve are not exercised. Table 2.2 lists the circuits non-idealities which are at the origin of those errors.

TABLE 2.2. Components Responsible for ADC Nonlinearity

Error in Transfer Curve	Corresponding Circuit Nonideality
ADC Offset	comparator offset, resistor string nonlinearity
DAC Offset	amplifier offset, Capacitor/resistor mismatch
Residue block gain error	capacitor mismatch, charge injection, amplifier offset

2.4.3 - Error Correction Schemes

In this section, two of the most cited error correction procedures for multi-step and pipeline ADCs will be described.

1. Missing Codes Calibration

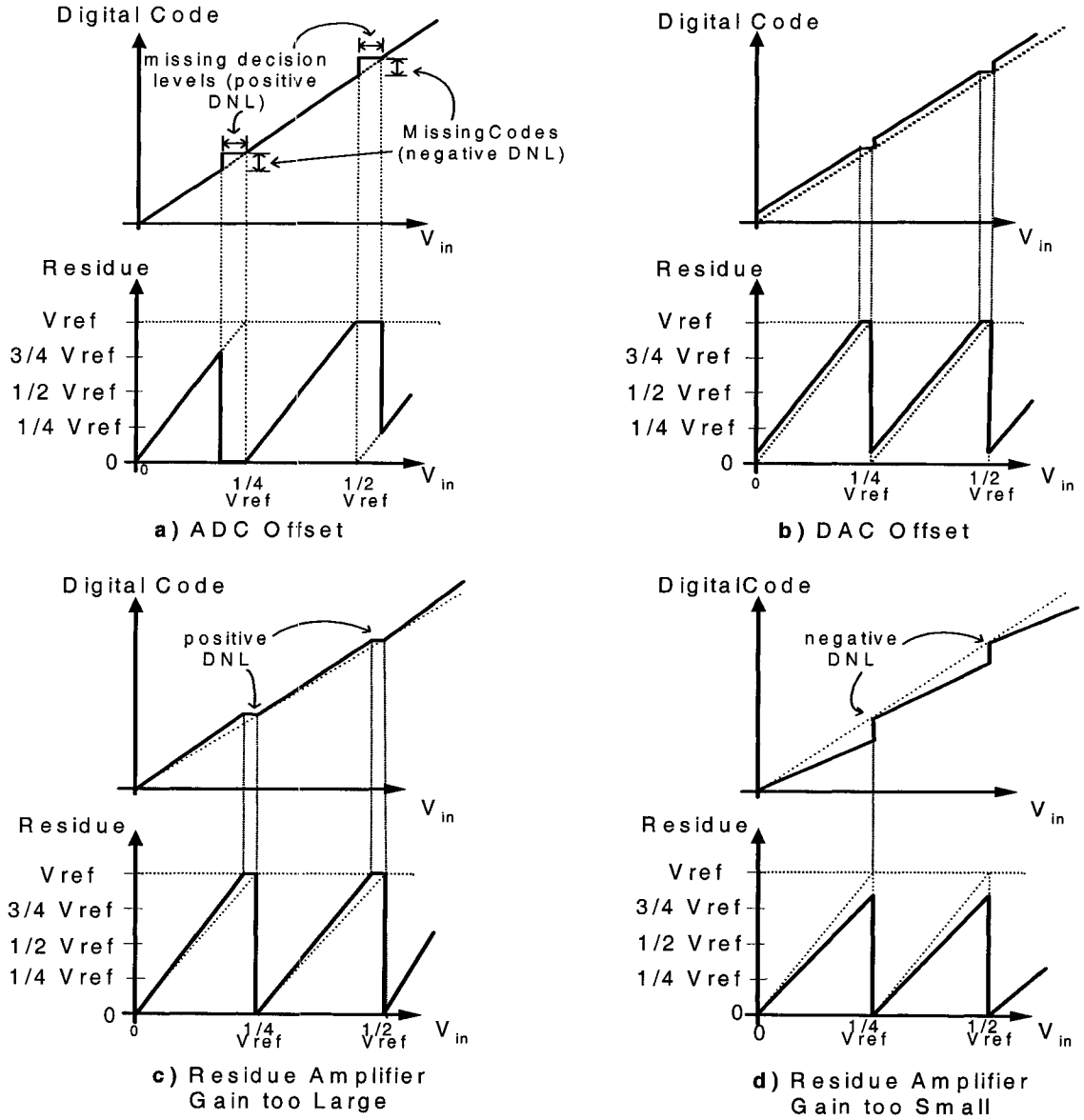


Figure 2.13. Error Sources in Multistep ADC

This method was initially reported in [18], and describes a calibration algorithm that corrects for missing codes in the ADC transfer curve, regardless of the particular circuit block that causes it. The premise for this calibration scheme is that missing codes, which correspond to vertical jumps in the ADC transfer function, can be recovered with digital correction alone. However, missing decision levels, which are horizontal jumps in the tranfer function, cannot be corrected digitally. Therefore, the prerequisite for this method is that nonlinearities be limited to missing codes only, and not contain missing decision levels. For a 1-bit/stage pipeline ADC, the interstage amplifier gain is 2. A gain higher than 2 results in missing decision levels, while a gain of smaller than 2 causes missing codes, as was just seen in Figure 2.13. As a result, the prerequisite translates into avoiding a gain higher than 2, which is easily done by sizing the capacitors for a ratio slightly smaller than 2.

The transfer curve of a residue stage with negative gain error is shown in Figure 2.14(a). The section of the overall ADC transfer function affected by this gain error is shown in Figure 2.14(b), along with the calibration operations. The calibration principle is very simple, and consists of pre-storing calibration constants S_1 and S_2 in digital memory, and subtracting them from the uncalibrated output X during normal operation to yield the calibrated output Y:

$$Y = X \qquad \text{if } X<S_2 \tag{2.4}$$

$$Y = X-(S_1-S_2) \quad \text{if } X>S_1 \tag{2.5}$$

This essentially has the effect of joining the joining the points S1 and S2 such that the curve becomes linear. Due to the reduced gain of each stage, a few additional stages

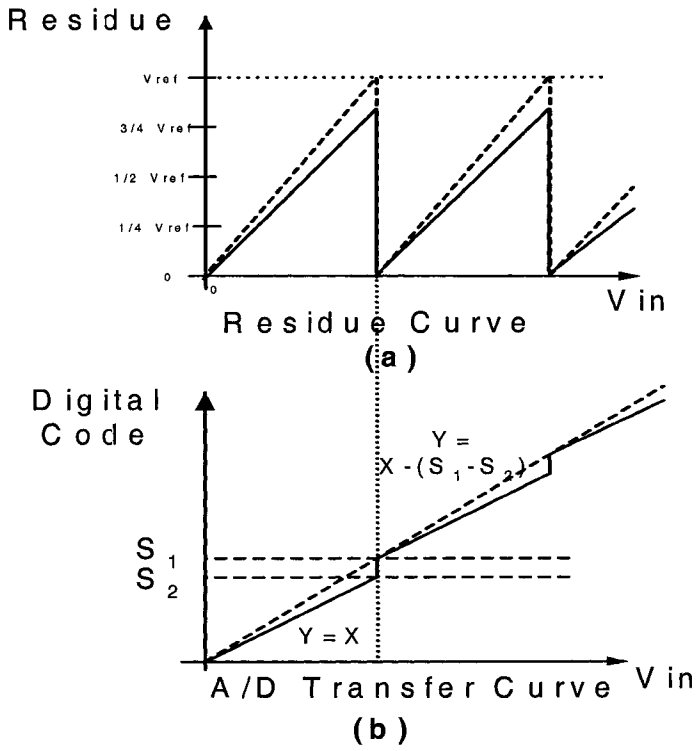


Figure 2.14. a) Missing Codes in Residue Transfer Function b) Missing Codes and its Calibration in Overall Transfer Function

are needed to carry out the conversion to the desired resolution. In [18], calibration is first performed on the lowest resolution stage, and then proceeds upward until the MSB stage. The calibration can correct for the cumulative effects of capacitor mismatch, opamp finite DC gain, charge injection, and comparator offsets, up to an amount which does not cause the residue output to exceed the FSR.

2. Digital Error Correction

This algorithm, originally proposed in [19], has been widely used in different varieties [20] - [24], and is a well accepted method of correcting the nonlinearity errors due to the stage ADC.

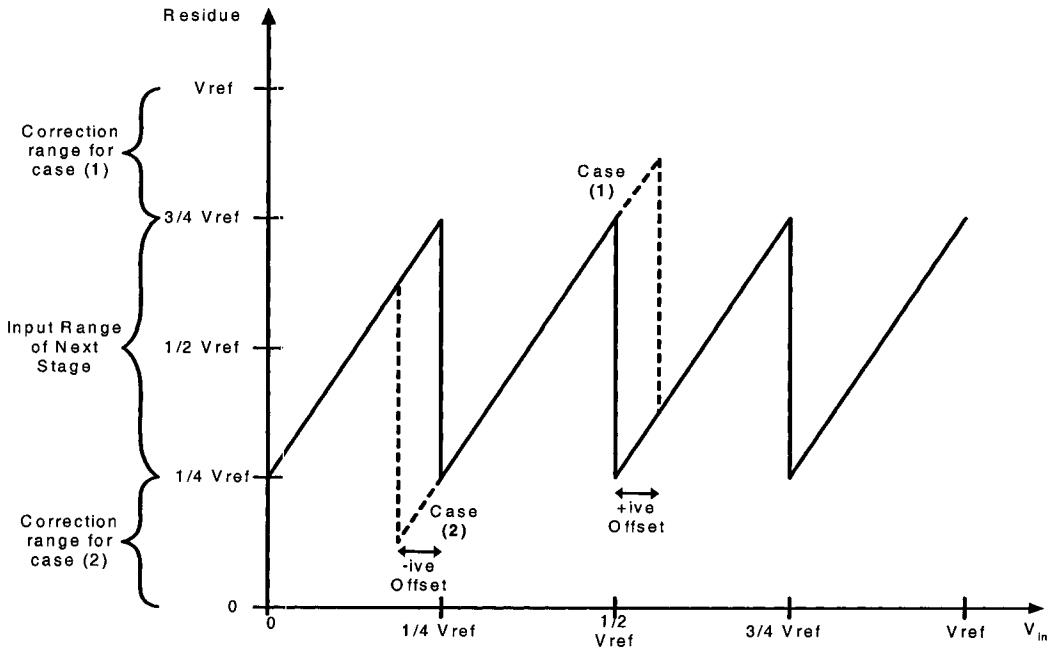


Figure 2.15. Transfer Characteristic of ADC Stage with Gain Reduced by 2

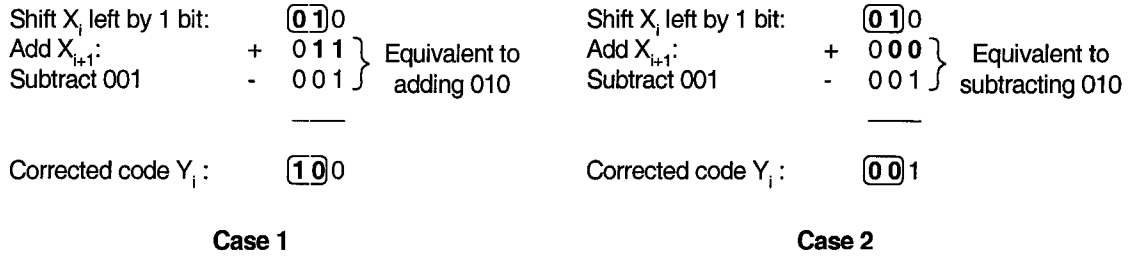
Digital error correction can best be described through an example. It was seen that an offset on the ADC threshold causes the residue, or the stage output, to exceed the FSR, thereby causing nonlinearity. This was illustrated in Figure 2.13(a). As a first step, one can reduce the amplifier gain so that offsets can occur without causing the residue to clip. This is shown in Figure 2.15 for a 2-bit first stage, with a residue gain of 2 instead of 4. It

can be seen on the figure that the offset can be as large as $\pm V_{ref}/4$ at each end without causing saturation. For instance, in case (1) shown on Figure 2.15, the false conversion yields a digital code of 01 (instead of 10) and a residue higher than $3/4 \cdot V_{ref}$, which corresponds to a digital code of 11 for the next stage (assuming a correct conversion in the next stage). A similar thing occurs for case (2). In both cases, a false conversion causes the digital ouput of the next stage to be different than it would be for a correct conversion. This implies, therefore, that the output of the next stage, assuming it is error-free, contains information about the correctness of the output of the current stage. Thus it can be used to detect and correct an error in the output of the current stage. In fact, one bit of the next stage output consists the regular output, while the other bit is used for error detection. To see how the correction is actually performed, let us consider the two cases in Figure 2.15 again. Suppose that case (1) corresponds to an input slightly higher than $1/2 \cdot V_{ref}$, and case (2) to an input slightly lower than $1/4 \cdot V_{ref}$: We have the codes shown in Table 2.3 below:

TABLE 2.3. Theoretical and Actual Output Codes For Offset Cases in Figure 2.15

Vin	Theoretical Code	Actual Code Obtained, X_i	Next Stage Code X_{i+1}
$1/2^{+}.V_{ref}$ (Case 1)	10	01	11
$1/4^{-}.V_{ref}$ (Case 2)	00	01	00

The digital correction algorithm is shown below for the two cases.



It should be noted again that the correctness of this algorithm depends on the next stage digital code being error-free. Given the 2-time gain in the residue stage, if the input to the current stage is x Volts away from the threshold, the input to the next stage will be $2x$ Volts away from the threshold, thus lessening the chances of a wrong conversion in the next stage. To attempt to increase the robustness of this algorithm, an experiment has been performed where not only the next stage, but also the one following that, are used to correct the current stage. Results of the comparison between this algorithms and the conventional one will be presented in Chapter 4.

An improvement of the standard digital correction is proposed in [24]. While the aforementioned issue of sensitivity to the next stage offset is not mentioned as the motivation for this improved scheme, it does turn out to resolve it nevetheless. With reference to the previous example of the 2-bit stage, this new algorithm alters the transfer characteristic as shown in Figure 2.16. This is done by shifting the first two thresholds to the right by $1/8 \cdot V_{ref}$, removing the last threshold, and shifting the DAC transfer curve up by $1/4 \cdot V_{ref}$. The rationale is that since the thresholds are shifted forward, the correction logic never needs to perform any subtraction, and only performs addition. Previously, an

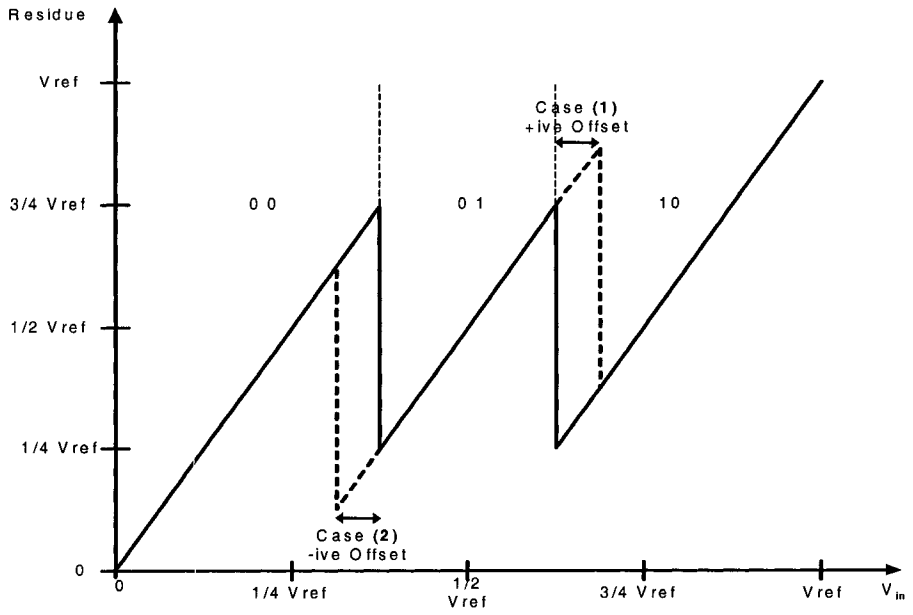


Figure 2.16. Transfer Characteristic of ADC Stage with Modified Error Correction

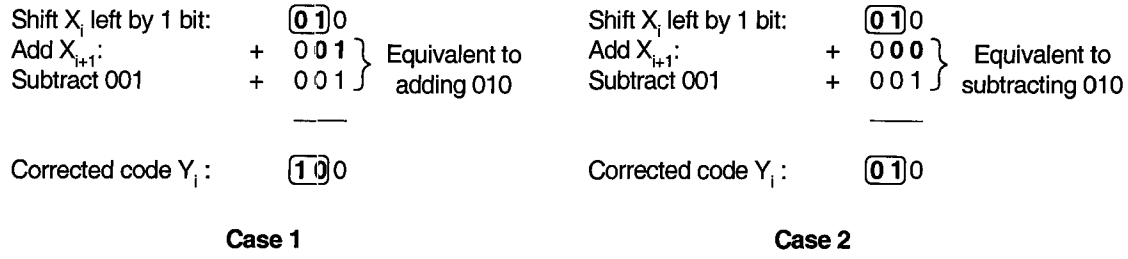
error in the subtraction logic could introduce unwanted errors or fail to correct existing errors. The new algorithm makes the design less sensitive to errors in the correction circuitry and provides more testing reliability. In addition, one less comparator is used [24]. More importantly, in the case of an error, this transfer characteristic places the residue corresponding to the erroneous conversion far from the thresholds, and in this way almost

eliminates the possibility of an error in the next stage output. To show the correction algorithm, consider the two cases again, and their corresponding codes in Table 2.4:

TABLE 2.4. Theoretical and Actual Output Codes For Offset Cases in Figure 2.16

Vin	Theoretical Code	Actual Code Obtained, X_i	Next Stage Code X_{i+1}
$5/8^{+}.V_{ref}$ (Case 1)	10	01	10
$3/8^{-}.V_{ref}$ (Case 2)	01	01	00

The digital correction algorithm is shown below for the two cases. A 2-bit/stage pipeline



ADC with this error correction scheme is often referred to as a 1.5-b/stage architecture, due to the redundant bit in between the two stages.

2.5 - Conclusions

In this chapter, three of the most modern ADC architectures have been presented. These are the flash, folding/interpolating, and multistep/pipeline architectures. A description of each architecture and some of the most important error mechanisms have been given in each case. In addition, some error-reduction and error correction schemes includ-

ing averaging, calibration, and digital error correction have been mentioned and their underlying principles described.

Chapter 3 - Multistep ADC Using Bitstream Processing

The existence of various technological limits and circuit non-idealities in the design of ADCs seriously compromises the robustness and success of analog circuit blocks. This calls for the use of calibration strategies with a two-fold advantage. The first advantage is the obvious one, that of eliminating the inherent and process-induced errors in the circuit blocks, while the second one is the reduced design effort required for the analog circuit blocks. As an example of the second benefit, the calibration of an amplifier's nonlinearity allows the designer to ease some of the engineering effort associated with designing a highly linear amplifier, while focusing more on the implementation of the the calibration circuitry. In this chapter, the application of single-bit processing to the design and calibration of a multistep ADC will be presented. First, the method of DC voltage generation by PDM bitstreams will be introduced, and then it will be shown how this technique can be applied to the design and calibration of a DAC within one stage of a multistep ADC. Lastly, the calibration of a sub-ADC will be demonstrated.

3.1 - DC Voltage Generation by PDM Bitstreams

Accurate DC voltages are required at the input of the ADC as well as throughout their data path. At the input, they are usually reference voltages, while throughout the ADC data path they can be the outputs of a DAC or a residue generation block.

Pulse-density modulation (PDM) consists of encoding the DC value of an infinite precision analog signal into a 2-level digital bitstream. With this technique, the running average of the digital bits represents a DC or average value, as shown in Figure 3.1. This

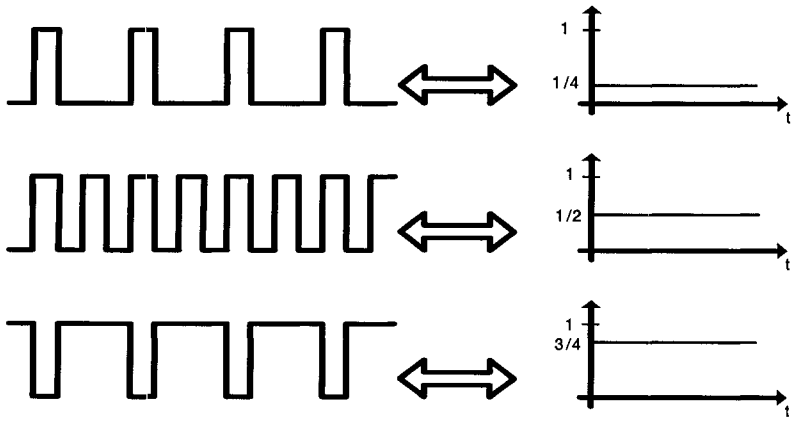


Figure 3.1. PDM Bitstreams and Corresponding DC Values

DC signal can be extracted from the digital bitstream using an appropriate low-pass filter, which is not shown here. Figure 3.1 is a simple example, showing 2^2=4 levels, or 2 bits of resolution in the output DC value. Extending this concept, we can obtain N bits of resolution with a bitstream of length 2^N. The advantage of this modulation scheme is its digital nature, which provides robustness and ease of control. In other words, by simply varying

the bit pattern of an N-bit digital stream, up to 2^N distinct DC values can be represented over the full-scale range.

The low-pass filter which enables the extraction of this DC component must suppress the harmonic energy of the bitstream to a level where the ripple on the DC signal is within the desired precision. At the same time, it is desirable for it to have higher bandwidth for faster operation, but this goes against the attenuation requirements contributing to the resolution of the DC signal. Therefore the filter design exercise becomes a tradeoff between speed of operation and resolution of the DC signal. In order to see the attenuation requirements, one can consider the case of the lowest harmonic frequency as follows: For a repeating bitstream pattern of length N_b where each bit lasts T_b seconds, the longest possible period occurs for the case of one '1' and N_b-1 '0's (or vice-versa), and equals $T_b \bullet N_b$ seconds. In other words, with F_s being the propagation frequency of the bitstream, the lowest possible harmonic will be at the frequency [2],

$$f_0 = \frac{F_s}{N_b} \tag{3.1}$$

with harmonics at multiples of this frequency. This implies that in order to increase the filter bandwidth, one can either settle for a smaller resolution by decreasing N_b, or find a means of increasing F_s. The latter option will be explored in Section 3.2 next.

3.2 - High-Speed Asynchronous FIFO

One way of allowing for a higher filter bandwidth is to increase the propagation rate, F_b, of the digital bitstream. With a synchronous implementation, one faces the difficulty of generating that high-frequency clock, achieving proper buffering, and maintaining a low level of clock skew among the large number of elements in the scan chain. Alternatively, a feasible solution is an asynchronous FIFO proposed in [3], which can be used as the basis for a high-speed bitstream generator. The architecture of this asynchronous FIFO is displayed in Figure 3.2.

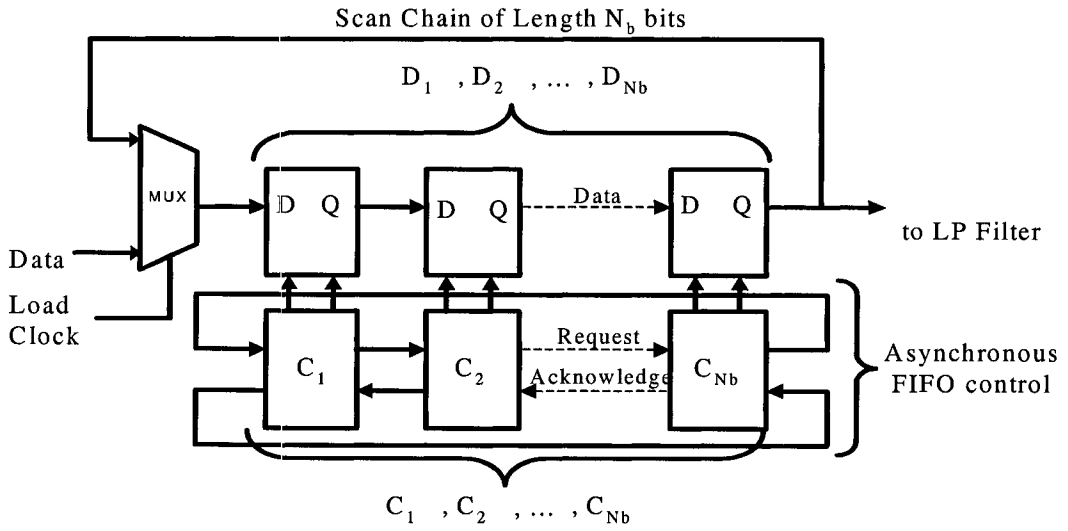


Figure 3.2. Architecture of Asynchronous FIFO

This implementation is a variation of an asynchronous FIFO originally proposed in [4] . Each stage contains a data and a control block, designated by D_i and C_i, respectively. The

data block is functionally equivalent to a flip-flop. The control block implements the handshaking protocol between stages of the FIFO, and consists of a request/acknowledge pair of signals. The request signal raised by the sender indicates data validity at its output, and the acknowledge signal raised by the receiver means that it has read and latched in the data. The receiver can only read the data if its previous stage has raised a request signal *and* the next stage has sent back an acknowledge signal. The maximum speed of the asynchronous FIFO is given by the delay in the control cell elements rather than a clock, which can therefore be much higher than its synchronous counterpart. In [3], the FIFO is optimized for speed by applying the principles of 'logical effort', which provide guidelines for MOS transistor sizing in order to achieve the best tradeoff between capacitive loading and current drive.

3.3 - Bitstream-Based DAC With Calibration

3.3.1 - DAC Architecture

Given that the desired signal at the output of a DAC is a DC voltage, the method of DC voltage generation by PDM bitstreams described in Chapter 3.1 can be applied to the design of a multi-bit DAC. A diagram of the proposed bitstream-based DAC is shown in Figure 3.3.

For a M-bit DAC with N-bit precision at the output, 2^M rows of memory are required, with each row containing 2^N 1-bit memory elements. Thus, each row contains the bits to encode one of the DAC outputs. The asynchronous FIFO described in Section 3.2 serves as a bitstream generator, which feeds a low-pass filter. Taking a 4-bit DAC with

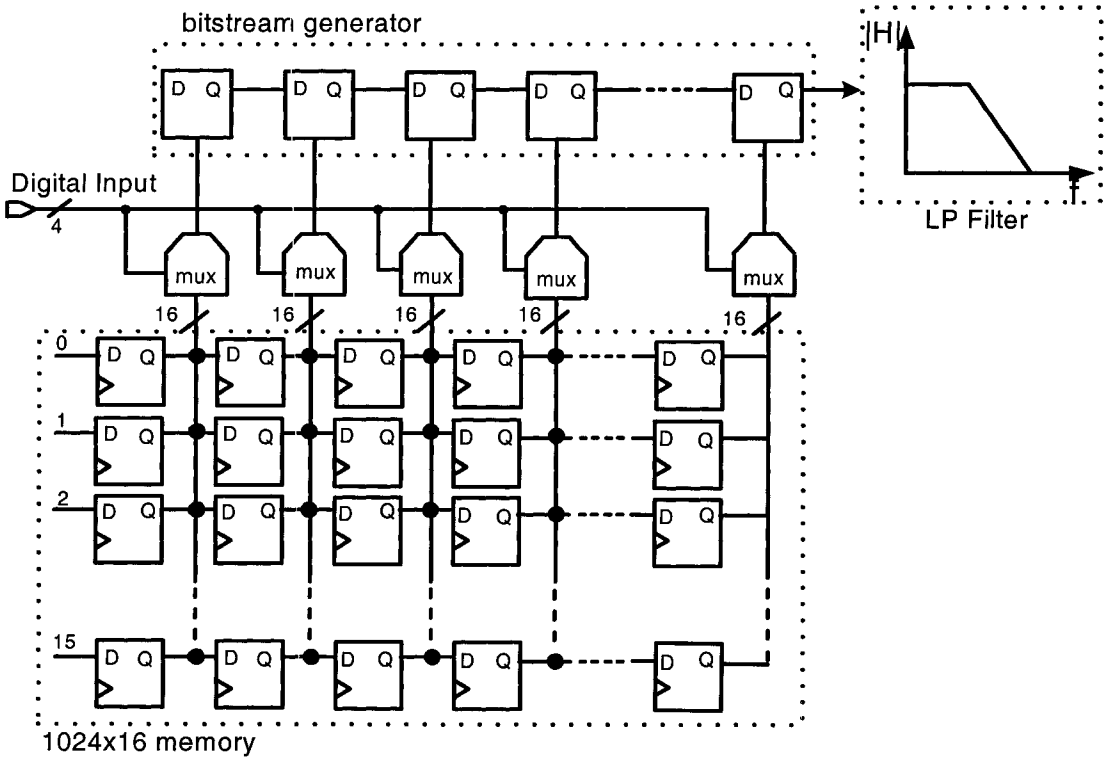


Figure 3.3. Architecture of Bitstream-Based DAC

a required resolution of 10 bits as an example (ie: M=4 and N=10), a cycle of its operation would proceed as follows: From the 4-bit digital input, a multiplexer selects one of the sixteen memory lines, and feeds its 1024 bits of data in parallel to the bitstream generator. The latter then enters the free-running mode, where the bits are fed serially and at high speed to the filter, which removes the harmonic content of the bitstream and settles to the encoded DC value. The filter settles at a speed determined by its internal time constants, which are set according to the attenuation requirements on the harmonics.

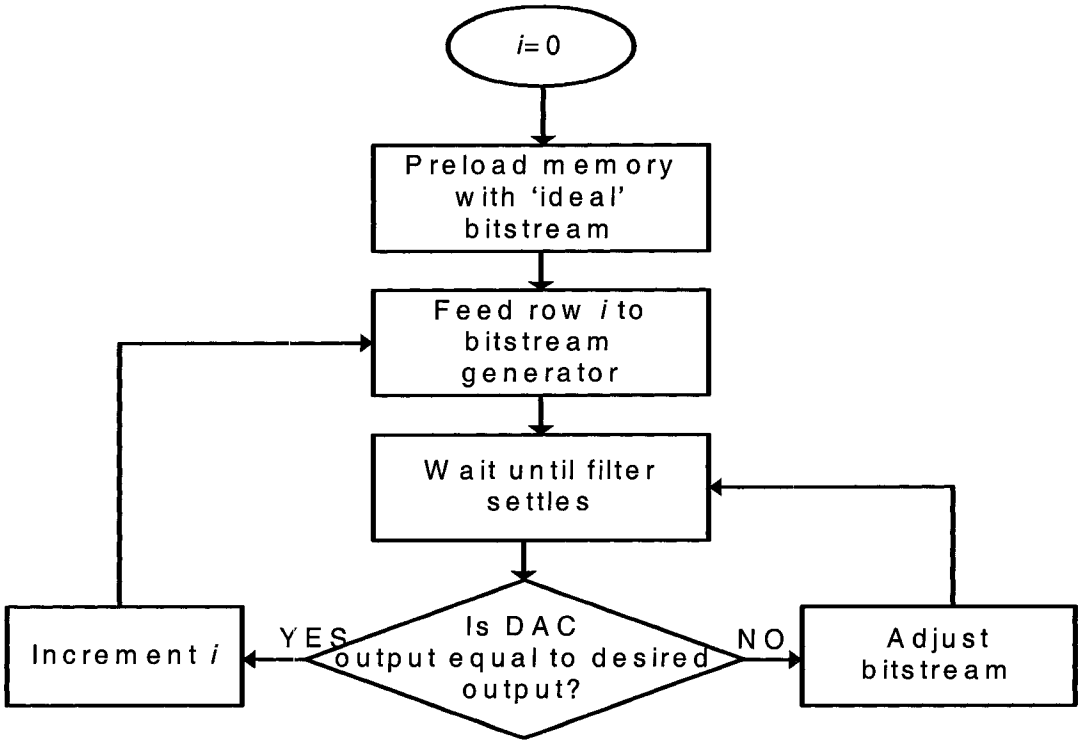


Figure 3.4. Calibration of Bitstream-Based DAC

3.3.1 - Calibration Algorithm

The bitstream-based calibration routine for the DAC is shown in Figure 3.4. Each row of the memory is filled with the ideal bitstream corresponding to the ouput. This bitstream will yield the uncalibrated DAC output. If this DAC output is not at the desired value, the bitstream is adjusted in an incremental fashion in the desired direction, with the smallest increment achieved by the changing of 1 single bit in the bitstream. In theory,

this increment moves the DAC output closer to the desired value by $(FSR)/(2^N - 1)$, but in practice this increment can vary due to imperfections of the circuits such as offset, gain error, or nonlinearity. The task of the calibration process is to repeat this process until the bitstream combination corresponding to a DAC output closest to its theoretical output is found. Ideally, the DAC output would be $(1/2) \bullet (FSR)/(2^N - 1)$ or $1/2$ LSB away from the theoretical output. The search loop can be implemented within the software of a digital testing system, and is repeated for each DAC value. The bitstreams corresponding to each of the correct DAC values are stored in the memory bank, and will be used during normal operation.

The advantage of the above system is two-fold. First, the inherent architecture of the DAC avoids the need for matching of passive devices or currents present in other architectures such as the R-2R topology, or DACs with binary-weighted capacitors or currents. Second and most importantly, the bitstream patterns stored in the memory bank can be made to compensate for the non-idealities in the remaining parts of the DAC, namely the bitstream generator and the filter. One instance of such a non-ideality occurs if the rise and fall times in the bitstream generator are not equal, as is often the case in high-speed digital circuits. Since the DC extraction operation is equivalent to integrating the area under the curve, a mismatch in the rise and fall times results in a different area and therefore a different average or DC value. In the case of the filter, if an active configuration is used, for each amplifier used, the finite DC gain A of the amplifier will result in a reduction of the filter DC gain by a factor on the order of $1/(A + 1)$, depending on the exact filter configuration. This gain error translates into a deviation of the slope of the DAC

transfer characteristic from its ideal value. This will in turn affect the outcome of the next stage: A negative DAC gain error will cause missing decision levels, while a positive gain error will cause missing codes. Moreover, this gain error can fluctuate across the input range if the DC gain A of the opamp also varies across the input range. This translates into nonlinearity in the DAC's transfer characteristic, which affects the conversion of the next stage. Although this gain error can be controlled by designing an amplifier with high enough DC gain, it still represents a design challenge for a high-resolution ADC. The next 2 sections describe how the calibration procedure can improve these unwanted signal effects.

3.3.1 - Gain Error Reduction

The calibration feature of the DAC can provide a higher tolerance to the gain errors in the system. To see this, let's consider the case of a 3-bit DAC in the first stage of a 10-bit multistep ADC. Figure 3.5 shows the ideal output values for such a DAC (left axis), as well as their modified values after the effects of gain error (middle axis) and finally after calibration (right axis). The full-scale range (FSR) is $2 \bullet FSR$ and the gain error through the DAC is constant at $-1/8 \bullet G$ V/bit, where G is its ideal gain. Taking the example of the input code '111', the theoretical output value is $-7/8 \bullet FSR$ as seen on the left vertical axis. With a DAC gain error of $-1/8 \bullet FSR$, the output FSR is reduced to $2 \bullet 7/8 \bullet FSR$ as shown in the middle axis, and the output corresponding to code '111' is also reduced to $7/8^{th}$ of the previous value. However, the calibration can restore the output back to its original value of $7/8 \bullet FSR$. This is done by setting the bitstream to all 1's,

which yields its highest equivalent DC value of $7/8 \bullet FSR$ to the required precision. Actually, since the output range is reduced by $1/8$ but the number of codes remains constant at 1024, the voltage step between adjacent DC values is reduced by $1/8$, which is equivalent to increasing the resolution by $\log_2(8/7)$ bits. This example shows that the gain error of $-1/8 \bullet G$ is the largest gain error that the DAC can still correct for.

If the gain error is positive, however, it will have an adverse effect. There will be clipping of the output voltage at each end for a number of bitstream combinations. These outputs will most likely not be needed, since only every $2^{M\text{th}}$ of the 2^N outputs are needed. However, the spacing between those 2^M useful values would be increased, which would reduce the DAC resolution. It must be said, however, that the filter's gain error can only be negative, since its transfer function contains a $1/(A+1)$ term. As for the bitstream generator, the lowest and highest DC values that it can encode are Vdd and Vss. Therefore, its gain error, in the strict sense of the definition, cannot possibly be positive either.

Thus, the calibration routine can tolerate, and correct for, any gain error down to $-1/2^M \bullet G$. This is in contrast to a normal DAC system which can tolerate a negative gain error down to $-1/2^N \bullet G$. In other words, the stage gain error needs to be commensurate with the resolution of that particular stage, M, rather than the resolution of the overall ADC, N. Assuming, for the sake of argument, that the bitstream generator does not intro-

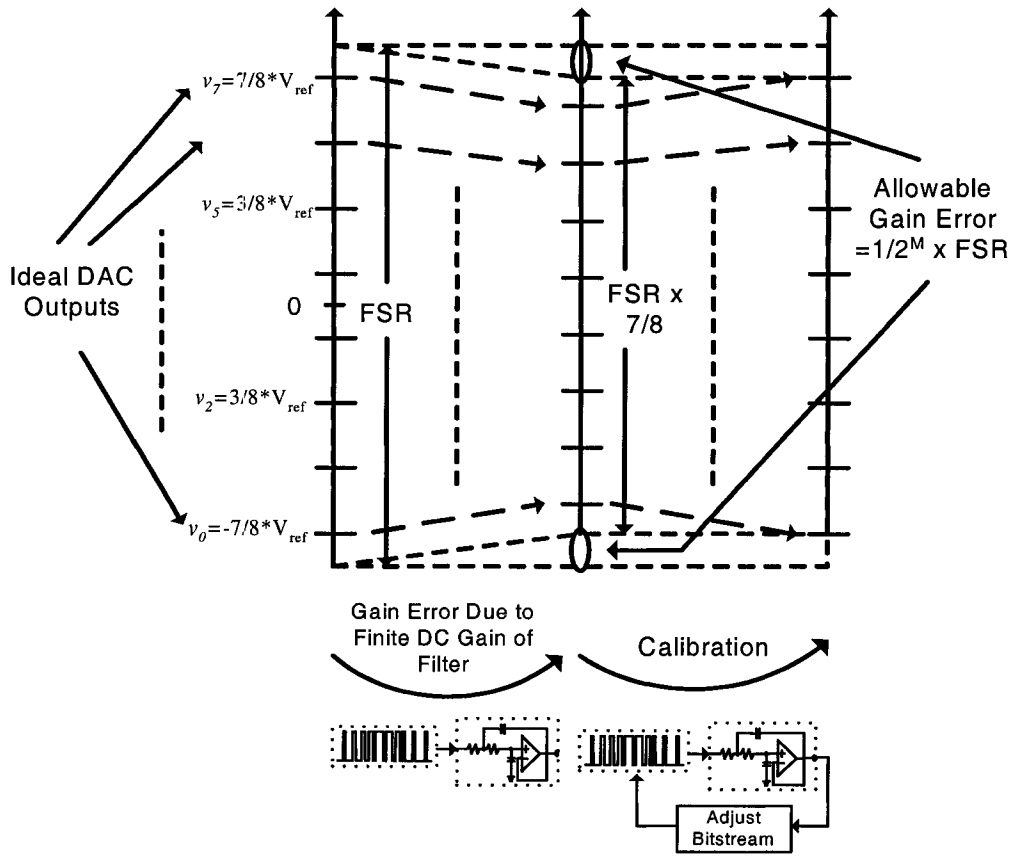


Figure 3.5. Allowable Gain Error for 3-Bit DAC with 10-bit Linearity

duce any gain error and that all the gain error is generated by the filter alone, a lower limit for the overall DC gain A of the filter's amplifier(s) can be obtained as follows:

$$\frac{1}{2^M} \geq \frac{1}{A+1} \tag{3.2}$$

$$A \geq 2^M - 1 \tag{3.3}$$

In reality, since the high-speed bitstream pattern out of the asynchronous FIFO is likely to contribute gain error due to process effects which are difficult to predict, the filter's gain error must be over-designed by a safe margin.

3.3.1 - Linearity and Offset

For the M-bit DAC to be N-bit linear, it is not required that the 2^N increments of the bitstream generator be regularly spaced, or that they have any particular degree of linearity. What is important is that *one* single bitstream be able to yield the desired DAC output within $LSB/2$ at N bits. In other words, N-bit precision is only important around the 2^M outputs of interest, and not for the entire 2^N outputs that can be generated. Thus, a nonlinear calibrator can still yield a perfectly calibrated DAC. This is shown in Figure 3.6, where the DAC's outputs for a number of bitstream patterns around output y_1 are shown. It can be seen that the nonlinearity of the points corresponding to different bitstreams is not important as long as one of them (bitstream 197 in this case) can provide the required output. However, there is also the possibility that no bitstream will be able to reach the theoretical output value to the desired precision. Thus, N-bit linearity over the entire range is a sufficient, but not a necessary condition for ensuring N-bit linearity at the 2^M points of interest.

With regards to the DAC offset, if it is defined as the deviation of the first code from its ideal value, then it is easy to see that it performed as part of the algorithm

described. However, such a definition of the offset is not accurate and a better definition is obtained based on a best-fit line drawn through the DAC outputs [1]. In that case, the final offset figure will depend on the best-fit gain, which in turn depends on all the output values, rather than just the first one. Thus, it is futile to separate the DAC offset calibration from the rest of the process, as it is closely intertwined with it.

3.3.1 - Residue block Offset Calibration

Since this calibration scheme must necessarily apply to a multi-bit DAC (a 1-bit DAC is inherently linear), it is well suited to a multistep ADC where each stage consists of an ADC, a DAC, and a residue amplifier. In that context, the calibration algorithm can be extended to include the residue block and its offset. To see this, we can model the offset of the residue block as a DC source which appears before the summation, in series with the DAC output, as shown in Figure 3.7. A flowchart of this extended offset calibration for the residue block is shown in Figure 3.8. As can be seen, the calibration of the DAC is done with the targetted output values modified to include the offset of the residue block as seen in Figure 3.8(b). Note that the gain error of the residue block cannot be calibrated out since its net effect is *not* a *constant* shift on the output.

3.4 - Bitstream-Based ADC With Calibration

A multistep ADC is composed of a cascade of smaller ADCs, where each ADC resolves M of the N total bits of resolution. Each smaller ADC is of the flash type, with a

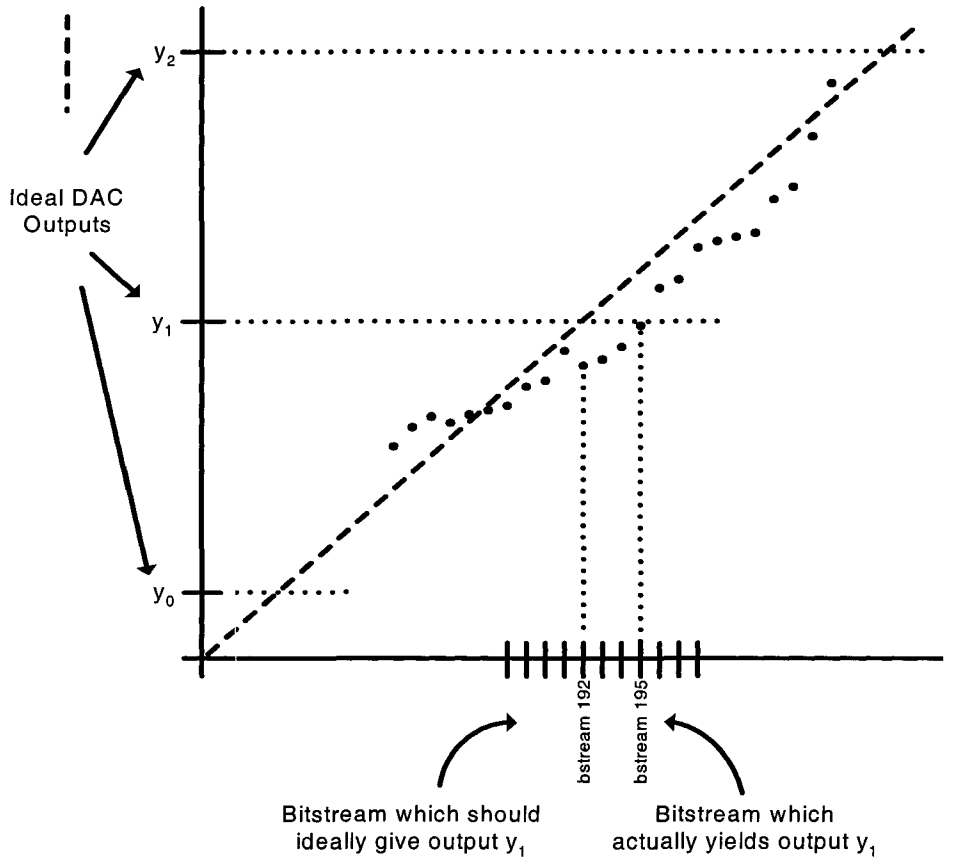


Figure 3.6. Example of DAC Outputs for Various Adjacent Bitstream Inputs

reference chain defining 2^M reference voltages and 2^M comparators performing comparisons of the input voltage with these reference voltages. The design of a first stage, high-resolution sub-ADC is faced with 2 main circuit challenges, namely the reference circuit and the comparator. The circuit which generates the reference voltages is usually made of a resistive ladder, which entails the ever present problem of resistor mismatch in IC fabrication technology. This mismatch is equivalent to a different DC offset on each reference

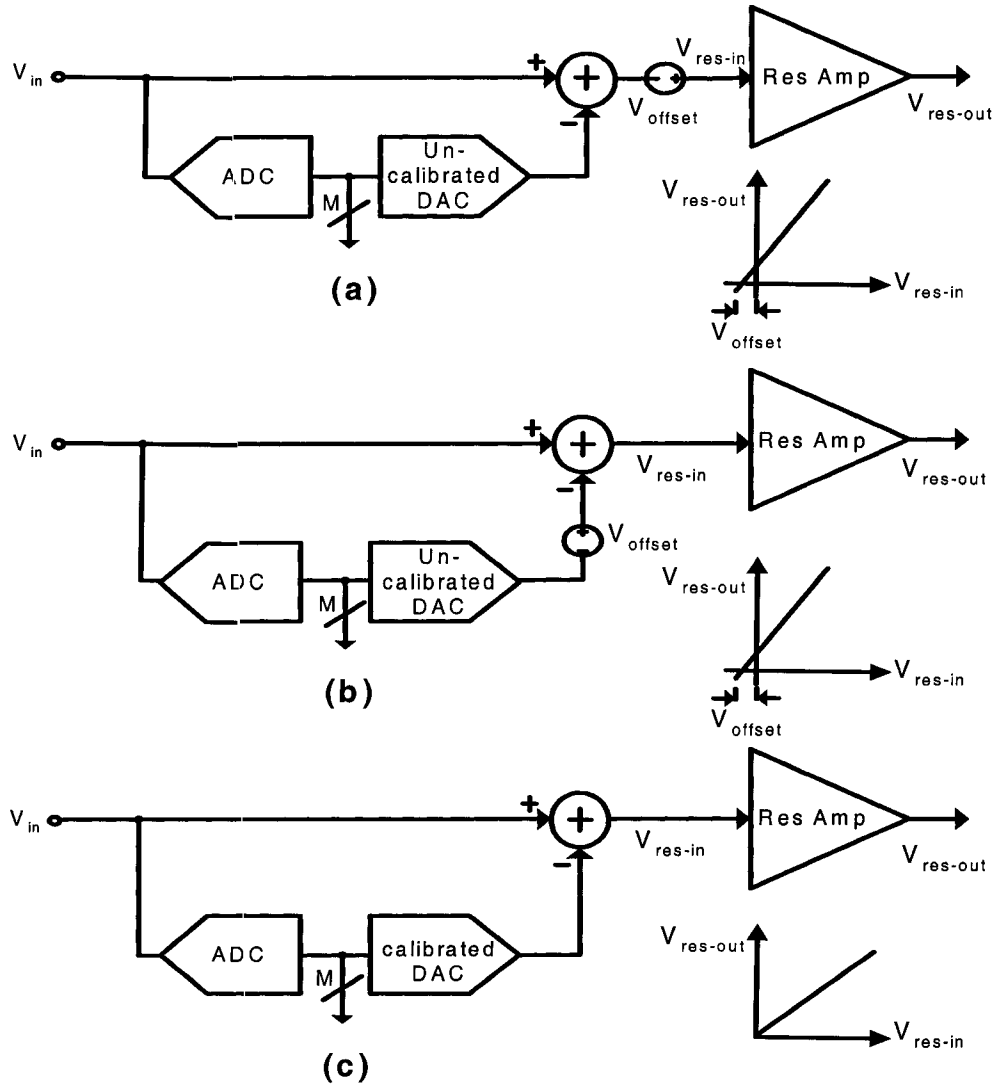


Figure 3.7. (a) ADC Stage with Residue Amplifier Offset (b) Offset Shifted to DAC Output (c) Residue Amplifier Offset Calibrated as part of DAC Calibration

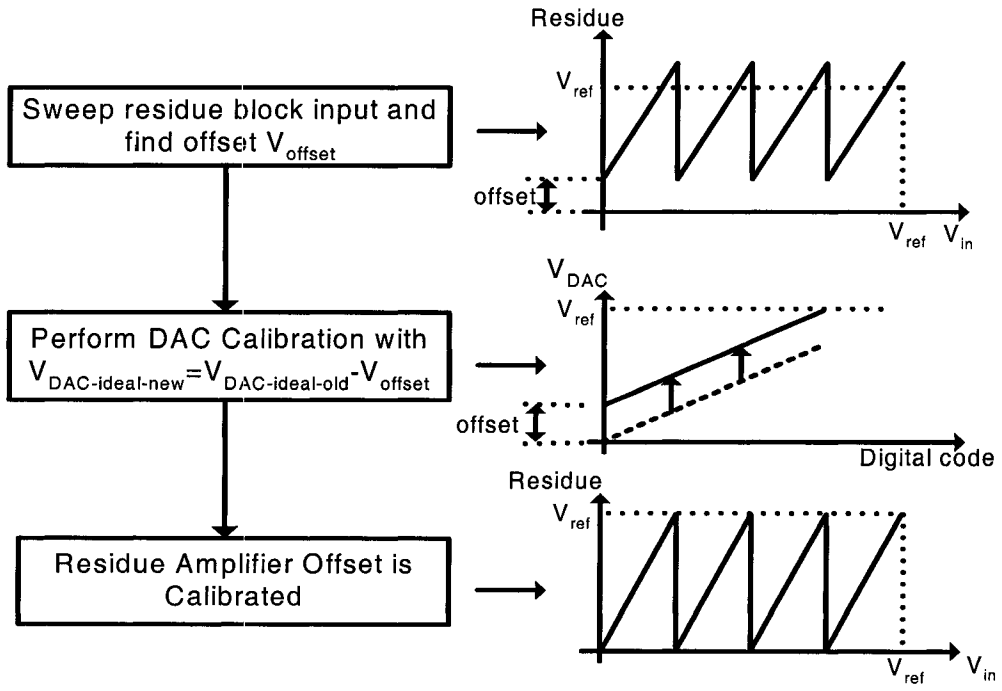


Figure 3.8. Calibration of Residue Amplifier Offset as part of DAC Calibration

voltage. Resistor mismatch is mostly controlled by applying a set of guidelines in layout. Another problem with the reference ladder is the coupling of the input signal onto the reference ladder through the gate-source parasitic capacitances of the comparator's input differential pair [9], as shown in Figure 3.9. This problem manifests itself as a temporary glitch on the reference voltage, and therefore only becomes critical at high speeds. The comparator also presents certain similar challenges. First, it also suffers from a random static input offset due to mismatches between input differential transistors introduced at various steps in the fabrication process. Secondly, it has a finite resolution, which means that it cannot distinguish a difference in the input signals below a certain minimum value.

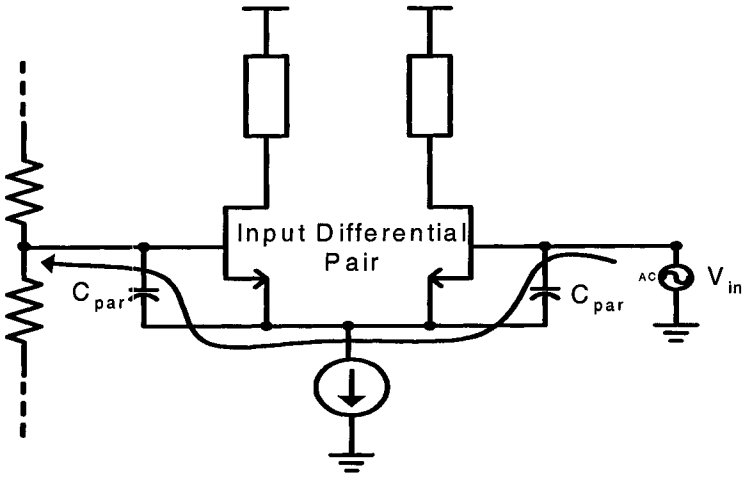


Figure 3.9. Parasitic Coupling From Input to the Reference Voltage

The offset problem is the one that is addressed here. It should be mentioned that offset cancellation schemes have been devised which calibrate out the offset on the first half of each conversion cycle (auto-zeroing), and perform the comparison on the next cycle. However, the calibrated quantity is lost at the end of each cycle, and each new cycle needs to measure it again, which is redundant and reduces conversion bandwidth. Given that the static offset is a time-invariant quantity, it only needs to be measured once, and therefore measuring and storing the offset once at the beginning can eliminate that idle time during normal operation.

The calibration technique proposed here is based on [5], and consists of storing the comparator offset before the start of normal operation. The algorithm for this calibration is shown in Figure 3.10. The comparator offset voltage is modeled as a voltage source in

series with the positive input. The calibration then proceeds as follows: The positive input is set to Analog Ground, and the negative input is ramped up in very small increments until the comparator toggles. The input value at which the comparator toggles is very close to the offset, and is stored on a capacitor at the negative input. This effectively cancels the comparator offset.

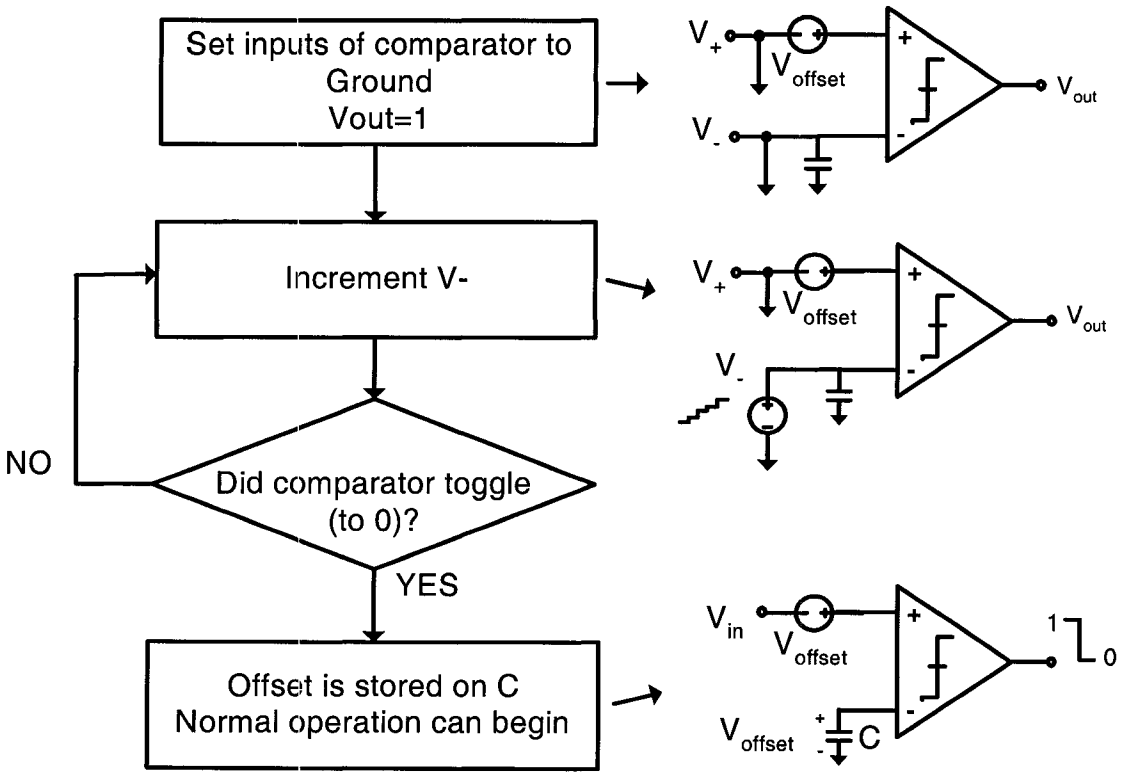


Figure 3.10. Comparator Offset Calibration Algorithm

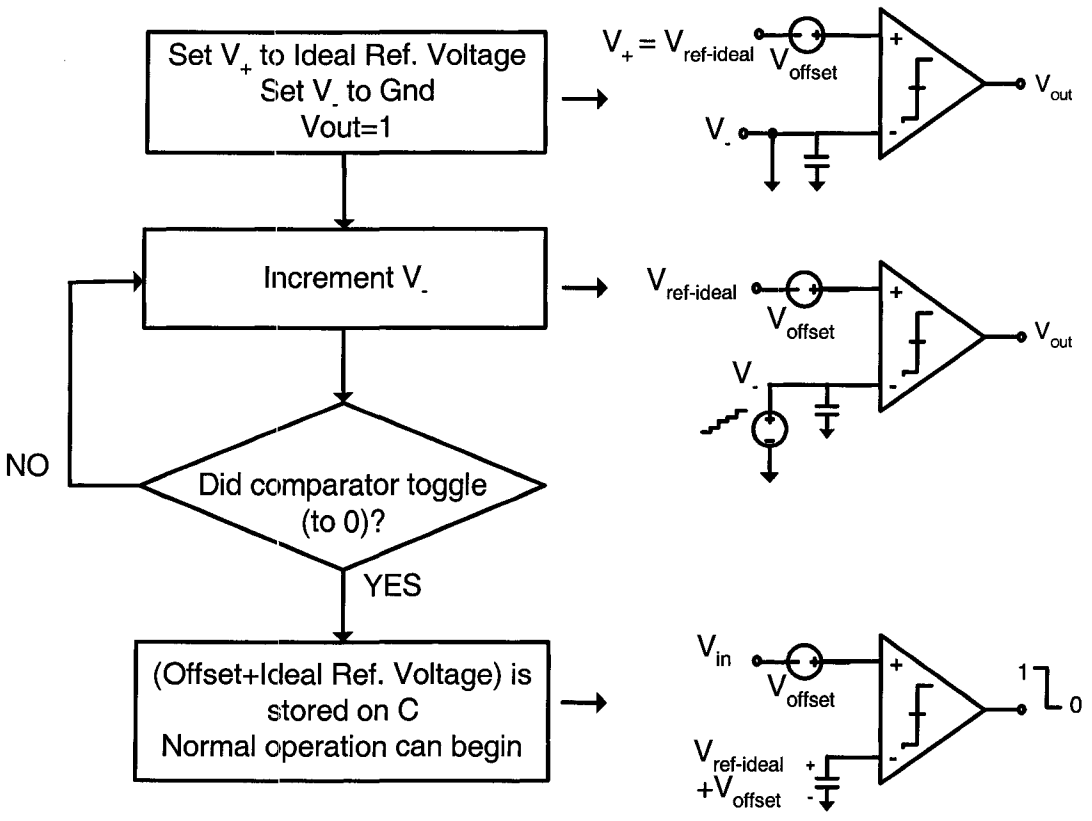


Figure 3.11. Comparator Offset and Reference Voltage Calibration Algorithm

Since the resistor-generated reference voltages in the ADC are also subject to innaccuracies, the calibration method just described can be made to include the reference voltages as well. This is shown in Figure 3.12. Thus, instead of Ground, we set the + input of the comparator to the ideal reference voltage. This way, the calibration will end

with the voltage stored on the negative input being the sum of the ideal reference voltage and the offset voltage. The comparator is then ready for normal operation, without the need for a reference ladder or further offset cancellation. It should be noted that in order to generate the DC voltages required at the inputs, the method of DC voltage generation by PDM bitstreams can also be applied to this part of the system, while making use of the same hardware available from the DAC calibration. In [5], one passive filter was used for each output value required. However, here, all that is used are 2 bitstream generators and 2 filters, which can be simultaneously connected to the comparator inputs through an analog demultiplexer. The filters and bitstream generators are already available from the DACs in the first two stages of a multistep ADC, and therefore the only additonal component is the analog demultiplexer, which is only a small overhead. This configuration is depicted in Figure 3.12.

3.5 - Summary

Table 1 summarizes the unwanted signal effects calibrated in one stage of a multistep ADC. The non-calibrated quantities have to be dealt with by suitable circuit design and layout. For instance, the dynamic offset of the comparator will require careful matching as will be seen in Chapter 4.

TABLE 1. Summary of Calibration for Multistep ADC Stage

Component	Calibrated Quantities	Non-Calibrated Quantities
ADC	Comparator Offset Reference Nonlinearity	Dynamic Offset (latch stage)

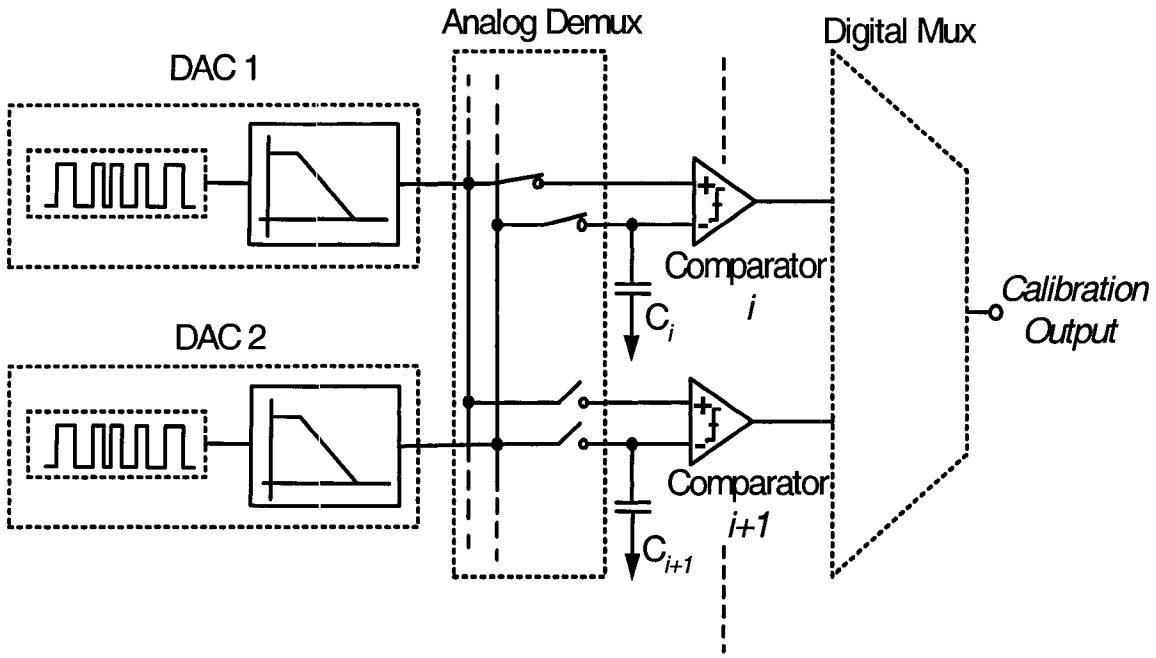


Figure 3.12. ADC Calibration Using an Analog Demultiplexer

TABLE 1. Summary of Calibration for Multistep ADC Stage

Component	Calibrated Quantities	Non-Calibrated Quantities
DAC	Offset, Gain Error, Nonlinearity	
Res Amp	Offset	Gain Error, Non-Linearity

For multistep ADCs, the first stage ADC has the highest linearity requirements, equal to the total ADC resolution. The linearity requirement of each subsequent stage diminishes by the total number of bits resolved in the previous stages, as shown in Chapter 2. Thus, the methods presented in this chapter are most valuable for the first stage ADC and DAC.

One important point is that in order for the calibration process to be truly accurate, it must be done under noise levels which closely resemble the normal operation of the ADC. Although this can never be perfectly achieved due to the many factors contributing to the overall noise, the best procedure is to perform the calibration of each block in parallel with the rest of the system running in normal mode.

3.6 - Conclusions

This chapter has presented design and calibration algorithms for multistep ADC circuit blocks using PDM bitstream processing. A bitstream-based DAC design has been described, consisting of a memory bank, an asynchronous FIFO, and a low-pass filter. Its calibration algorithm has been presented, including its extension to the offset-calibration of the residue amplifier. Finally, the calibration of the comparator and reference ladder of an ADC, also taking advantage of PDM bitstream processing, has been presented.

Chapter 4 - Multistep ADC Implementation

4.1 - System-level Design

The calibration scheme described in Chapter 3 is best suited to a multistep ADC where each stage resolves more than 1 bit of data. In this work, 4-bit DACs are chosen in order to have 16 output values for each DAC and thus sufficiently exercise the calibration feature.

It was seen in Chapter 3 that the DAC imposes a speed/resolution tradeoff due to the fact that the fundamental frequency of its output decreases with the resolution, as was previously quantified in Equation 3.1. Using Fs=2.2 GHz (obtained from simulation), which is the speed of the asynchronous FIFO, the fundamental frequency of the bitstream as a function of various numbers of bits was calculated, and is shown in Table 4.1. A reasonable harmonic

frequency is obtained at a resolution of 10 bits, and that is therefore chosen as the resolution for the ADC. A diagram of the implementation is shown in Figure 4.1.

TABLE 4.1. Harmonic Frequency as a Function of Bit Resolution

Resolution (bits)	Fundamental Frequency (MHz)
8	8.59
10	2.15
12	0.537

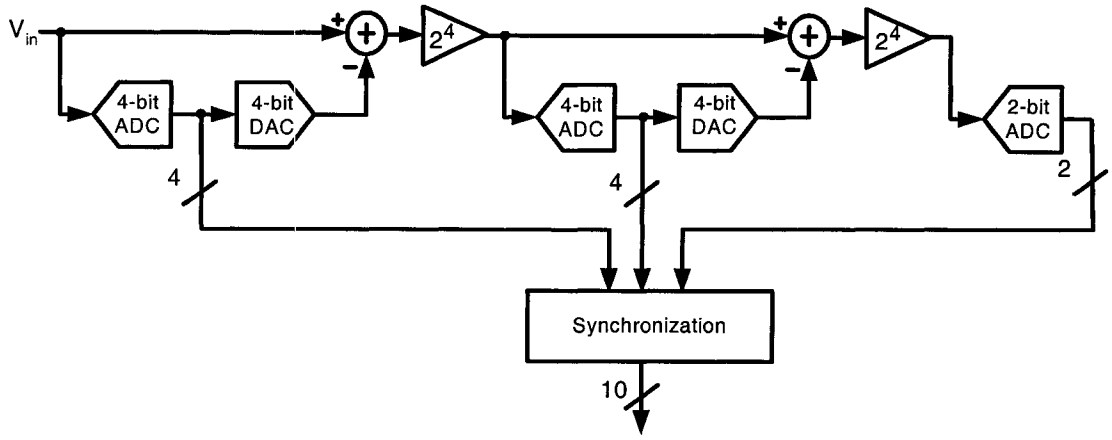


Figure 4.1. 10-bit Multistep ADC Architecture

One pertinent issue is that of the signal range. Since the binary signal encoding the DC values can take one of the two rail voltages, the extracted DC signal can extend from rail to rail. In order to accomodate the analog components, which are not inherently rail-to-rail, one can set the rail values for the bitstream portion of the circuit to the extremes of the analog input range. Another option consists of building the analog components to

operate over the entire input range. The latter option is chosen here since it provides a more interesting, although perhaps more complicated, approach.

The design of the last stage ADC is not critical as any inaccuracies there become relatively insignificant when referred to the input. That ADC is implemented as a standard 2-bit flash ADC, with a resistive ladder, 4 comparators, and simple decoding logic. It will not be elaborated in further detail here.

4.2 - ADC Design

4.2.1 - Rail-to-rail comparator

A common topology for comparators consists of one or more preamplifier stages followed by a latch stage. The preamplifer is there for two reasons. First and most importantly, it amplifies a small difference at the input to a voltage large enough for the latch to be able to distinguish and convert to digital levels, without remaining metastable in the available time. Secondly, the latching process is characterized by large signal swings, which can parasitically couple back to the input nodes. This phenomenon is referred to as kickback. The presence of the preamplifiers has the advantage of reducing the kickback effect by the gain of the preamplifier stages [27].

The rail-to-rail comparator is shown in Figure 4.2, and Table 4.2 shows the aspect ratios used. It requires a rail-to-rail stage at the input, and a digital output. A very commonly used input stage consists of two differential pair amplifiers, one being N-type, and

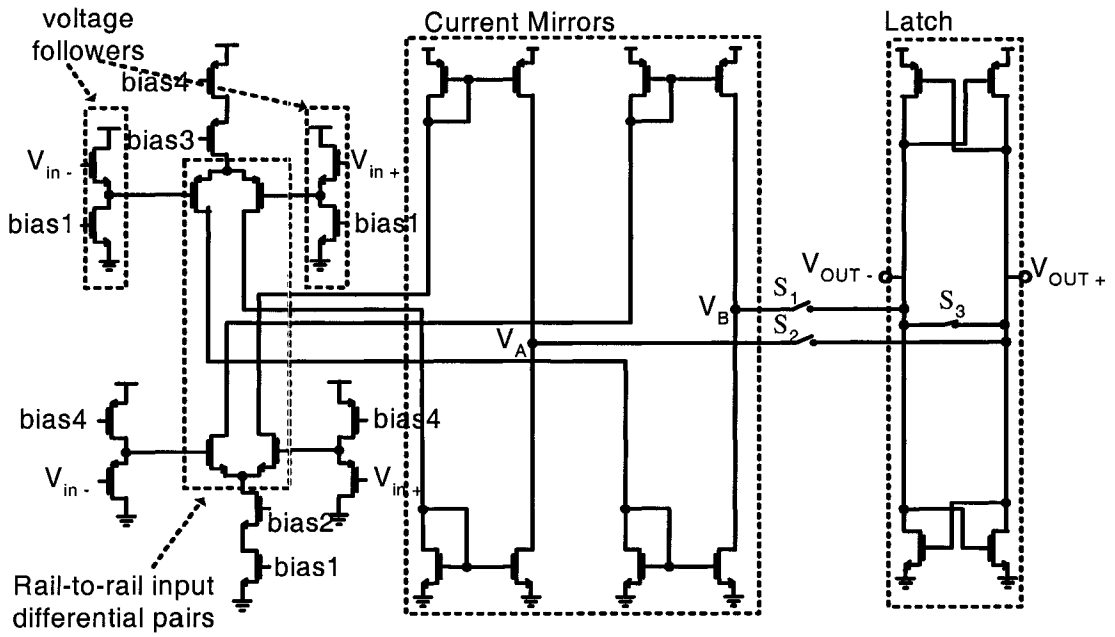


Figure 4.2. Rail-to-Rail Comparator

the other P-type. Usually, when the input signal is near V_{ss}, the P-type pair conducts, and

TABLE 4.2. Aspect Ratios (in μm) for Transistors of Comparator Circuit

Transistor	*Size*	*Transistor*	*Size*
N-type voltage follower	6.6/0.36	P-type differential pair	45/0.36
N-type voltage follower current source	9.5/0.36	P-type differential pair current source	21/0.36
P-type input voltage follower	24/0.36	N-type current mirror	2/0.36
P-type voltage follower current source	25/0.36	P-type current mirror	5/0.36
N-type differential pair	15/0.36	Latch: N-type transistors	10/0.36
N-type differential pair current source	8/0.36	Latch: P-type transistors	3/0.36

when it is around V_{dd}, the N-type pair conducts. In the middle of the input range, both amplifiers conduct. In this design, the input differential pairs are preceded by a set of volt-

age followers [26]. The positive level shift provided by the voltage followers preceding the N-pair allows the latter to conduct even when the input voltage is near V_{ss} [26]. The dual phenomenon occurs for the P-pair. The advantage of this topology is that neither differential pair turns off over the input range. $|V_{GS}| - |V_t|$ is always a positive quantity, and therefore the combined transconductance, g_m, of the two complementary differential pairs remains relatively constant over the input range. The resulting fluctuation in the first stage gain does create distortion in its output. However, since the output of the comparator is digital, the nonlinearity of the first stage is only relevant if it leads to errors in the latching action. Thus, first stage distortion is not as severe as for the case of a linear device such as an amplifier. The objective is to keep the lowest gain over the input range to a level which will satisfy the bandwidth requirements of the first stage, provide enough gain to avoid metastability and decision errors, and overcome any dynamic offset in the latch. The issue of dynamic offset will be discussed a little later in this section.

The current mirrors have the function of summing the P and N currents corresponding to each input of the comparator on the parasitic capacitances of their output nodes. Thus, the P and N currents corresponding to V_{inp} and V_{inn} are summed at V_A and V_B. The latch is a simple pair of cross-coupled inverters, and has the function of driving the preamplified signal at the output of the first stage to full digital value. Its time-domain function is given by $V_{out} = V_{preamp} \cdot (1 - e^{t/\tau})$ [27], where the positive exponent indicates the exponential rise due to positive feedback.

The comparator operates from a 2-phase non-overlapping clock. On the first phase ϕ_1, the voltages at the input terminal are preamplified and stored on V_A and V_B. Also, the outputs of the latch inverters are shorted together such that they settle at a midpoint value. This makes the subsequent transition to digital values faster. On the second phase ϕ_2, the outputs of the preamplifier are fed to the latch through a pair of switches, and the latch amplifies them to digital values through positive feedback.

One crucial aspect of the comparator design is the symmetry of the positive and negative signal paths. Mismatches in their parasitic resistance and capacitance can lead to a reversal of the polarity of the signal at the output of the first stage and therefore a wrong comparison. In addition, since V_A and V_B are high-impedance nodes, the charge injection and capacitive feedthrough from the turning off of switch S_3 is likely to create a common-mode jump, or step, at those nodes. Now, if their impedances are mismatched due to parasitics, there will be a potentially significant mismatch in these steps. This phenomenon is generally referred to as dynamic offset. For these reasons, much care was taken to equalize the geometries of the layouts for the two signal paths.

One other issue which requires attention is that of unwanted charge sharing at the beginning of the latch phase. At the end of ϕ_1, the capacitors at nodes V_A and V_B have been amplified away from each other, while the outputs of the latch, V_{outp} and V_{outn} have been shorted. At the start of ϕ_2, V_A is shorted to V_{outp}, while V_B is shorted to V_{outn}. A charge sharing operation occurs at that time, whereby capacitors at each pair of nodes will share charges and tend to gravitate to a common voltage. Considering the short between

V_A and V_{outp}, the goal is to make that common voltage as close as possible to V_A. Thus, we need $\Delta V_A \ll \Delta V_{outp}$, which implies that $(\Delta q)/C_A \ll (\Delta q)/C_{outp}$. Therefore, we would need $C_A \gg C_{outp}$ in order to keep the new common voltage close to V_A. This means that the current mirror transistors connected to that node must be sized large enough compared to the latch transistors connected to V_{outp}.

4.2.2 - Analog Demultiplexer

The need to select the inputs of each individual comparator and connect them to the DAC outputs requires the use of a demultiplexer. A 3-bit demultiplexer within the calibration scheme is shown as an example in Figure 4.3. In the actual realization, two 4-bit demultiplexers are used, one for each of the comparator inputs. A digital demultiplexer routes the *on* signal to the gate of one of the switches, which connects the analog input V_{CAL} to that comparator. Since V_{CAL} is connected to the sources of all the switches, a parasitic capacitive path exists between the gate of any switch and the calibration capacitors at the negative inputs of the comparators. Therefore, switching activity on each gate can inject unwanted charge on the capacitors, which would alter the calibrated voltage value. A critical design detail is thus the minimization of this feedthrough. This is done by using single-transistor switches of minimum area, with N-type transistors for the higher end of the input range and P-type transistors for the lower end. The penalty in settling time due to small, high resistance switches is tolerable since the calibration process does not impose any critical speed requirements. Note also that the switch nonlinearity due to its input-dependant on-resistance is not of concern here. This is because it can be seen as just a

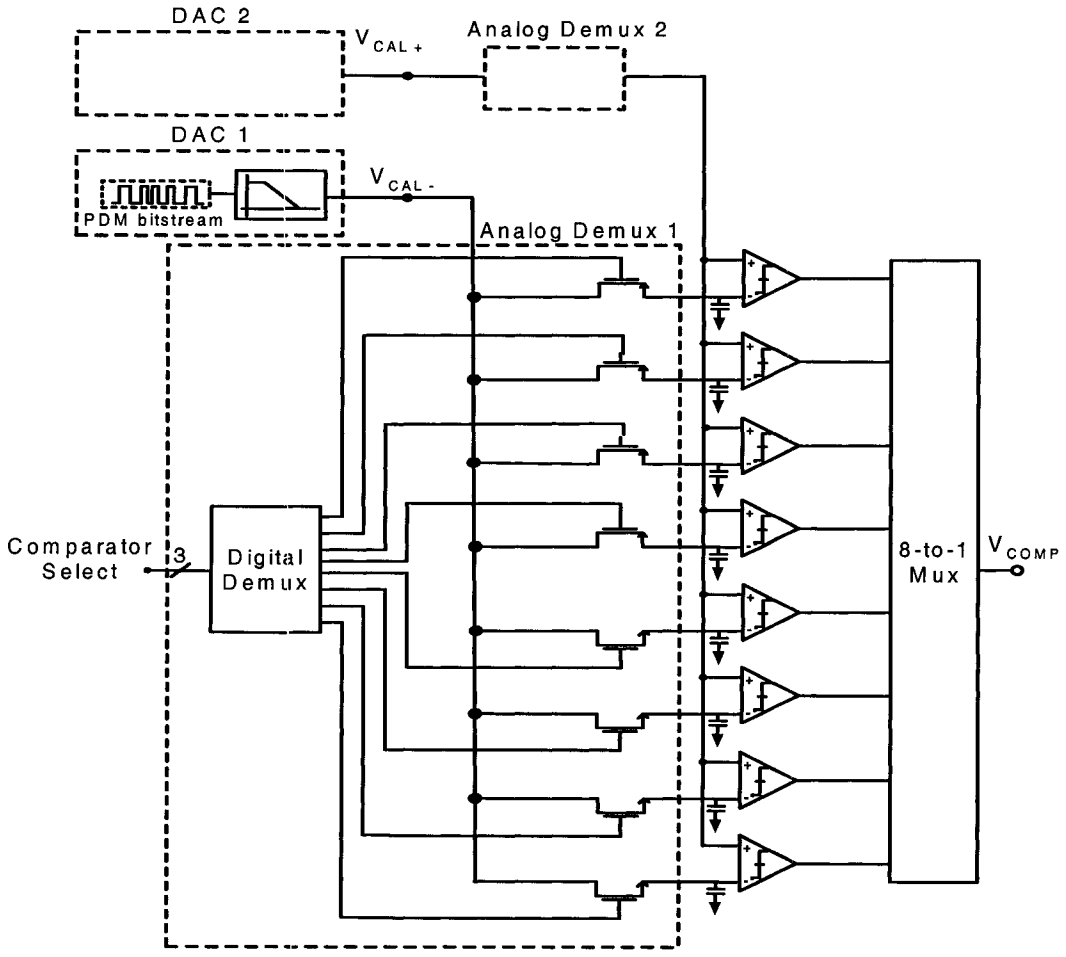


Figure 4.3. Analog Demultiplexer with Calibration Scheme and MUX

component of the input calibration signal, which can be varied to compensate for any distortion in the path.

4.2.3 - Input Sample/Hold

In Nyquist-rate ADC design, a sample/hold or track/hold circuit can remove the bandwidth requirements from the comparator preamplifier, as well as avoid any sampling time errors among the comparator inputs. Thus, it is especially important at high speeds, where both of those items are critical. In this design, a basic track/hold was implemented, although it was subsequently determined that the preamplifier bandwidth is sufficiently high not to necessitate the use of a track/hold. Therefore, the circuit was integrated, with an option to bypass it. The circuit is shown in Figure 4.4. It is composed of two transmis-

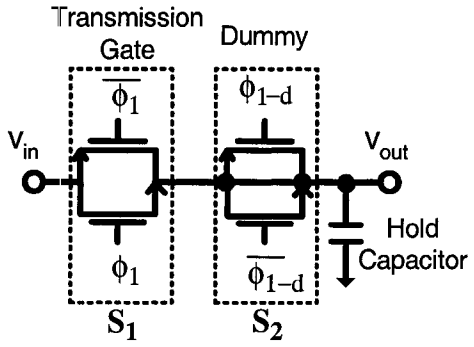


Figure 4.4. Input Track/Hold

sion gate switches S_1 and S_2 and a capacitor. The first switch is sized such that its highest on-resistance over the input range is low enough to satisfy the ADC bandwidth requirement. S_2 is a dummy switch intended to cancel the charge injection of S_1. Its size is half that of S_1. Its source and drain are shorted, and it operates on the opposite phase as S_2. Thus, when S_1 turns on, the channel of S_2 does not conduct, and the current flows through the shorting connection and charges the capacitor. Assuming that S_1 turns off very fast, its

channel charge q splits equally out of its two terminals. At that moment, S_2 turns on and creates a channel charge equal to $q/2$, and opposite in direction, thus cancelling the charge injection of S_1. Using a fast clock edge on S_1 ensures a 50% split in the charge, while operating S_2 from a slightly delayed clock will provide optimal compensation [31], [32]. As for the clock feedthrough due to the overlap capacitors, the N and P transistor sizes of each transmission gate are sized identically, thus equalizing their overlap capacitances [31]. As for the hold capacitor, its lower limit is dictated by the amount of thermal noise from the switch on-resistance which can be tolerated. This constraint requires that the RMS amplitude of this noise be much lower than the ADC's $1/2$ LSB margin, as follows:

$$v_n = \frac{kT}{C} \ll \frac{FSR}{2^N - 1} \tag{4.1}$$

where k is Boltzmann's constant, T is the temperature in Kelvins, FSR is the full-scale range of conversion, and N is the bit resolution.

4.2.4 - Bubble Error Detection

If the ADC's comparators are considered in order from the lowest to the highest, their outputs must have at most one transition from a series of zeros to a series of ones, or vice-versa. This pattern is commonly referred to as thermometer code. Based on this requirement, a certain amount of error detection and correction can be developed. The algorithm used here searches for a bubble, that is, an isolated zero amongst a series of ones, or vice-versa. The logic restores that bit to the same value as its surrounding bits.

Figure 4.5a. shows the ideal thermometer code, and Figure 4.5b. shows an example of a bubble error. In order to make the detection logic more immune to 'false' error, as shown in Figure 4.5d., 4 adjacent bits (instead of a minimum of 3) are used as the inputs to the correction logic.

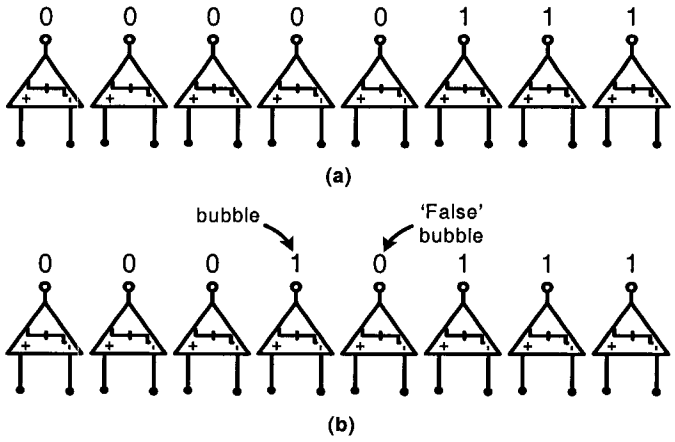


Figure 4.5. (a) Ideal Thermometer Code (b) Example of Bubble Error

4.2.5 - Thermometer-to-Binary Code Converter

A 3-bit version of this circuit is depicted in Figure 4.7. The zone selector is based on XOR gates, and converts the thermometer code to a series of bits with a single 'one', indicating the region in which the input signal resides. The bit restore logic then restores the 3 bits based on the region information.

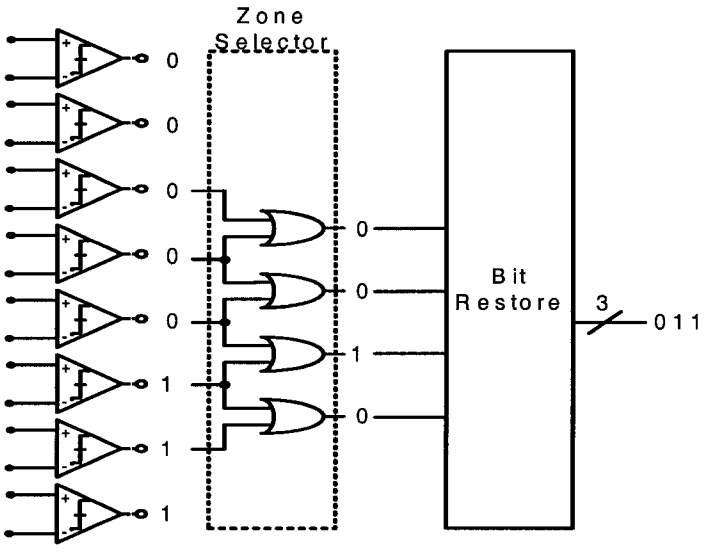


Figure 4.6. Thermometer-to-Binary Code Converter

4.3 - DAC Design

4.3.1 - Active Low-Pass Filter

The conversion speed of the ADC stage used in this work is largely determined by the settling time of of the DAC filter. Therefore, it is desired to have a filter with as high a bandwidth as possible, while still achieving the desired attenuation. Thus, a high order low-pass filter is used. The steep roll-off will provide the required attenuation, while allowing the 3-dB bandwidth to be closer to the bitstream fundamental frequency. The active low-pass filter is shown in Figure 4.7. It is composed of 4 identical stages, where each stage is a single amplifier biquad filter, as described in [28]. The worst case attenua-

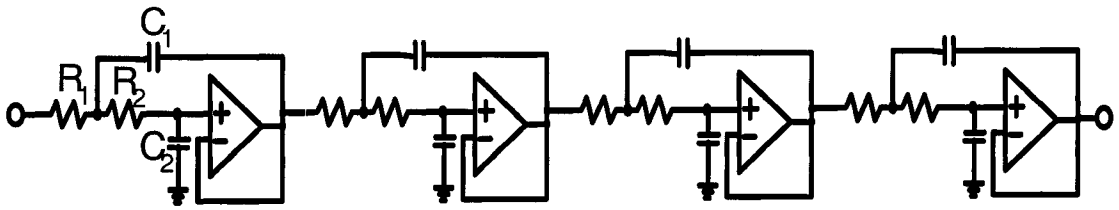


Figure 4.7. 8th order Active Low-Pass SAB-Based Filter

tion requirement will determine the corner frequency ω_p. For this, the case of the lowest possible Fourier series component and its corresponding amplitude has to be considered. It was shown in Chapter 3 that this lowest component occurs at a frequency of $f_0 = F_s/N_b$, where F_s is the bitstream propagation frequency and N_b is the length of the bitstream. In [29], it has been shown that it is sufficient to design the filter for the case of a bitstream pattern with a single one and N_b-*1* zeros. If the attenuation requirements are met for this case, then the requirements of the other cases will also be met. For this case in consideration, the RMS amplitude of the input is given by:

$$|H_k| = \frac{V_{dd}}{N_b} \tag{4.2}$$

The above filter has 2 poles per stage, and if the poles are designed to have the same value, its transfer function will be as follows:

$$T(s) = \frac{1}{\left(1 + \frac{s}{\omega_p}\right)^2} \tag{4.3}$$

And the overall transfer function of the filter will be:

$$T(s) = \left[\frac{1}{\left(1+\frac{s}{\omega_p}\right)^2}\right]^4 \tag{4.4}$$

Therefore, the RMS magnitude of the fundamental at the output of the filter will be:

$$|H_0| = \frac{V_{dd}}{N_b}\cdot\left[\frac{1}{\left(\sqrt{1+\left(\frac{2\pi f_0}{\omega_p}\right)^2}\right)^2}\right]^4 \tag{4.5}$$

And in general, the RMS magnitude of the k^{th} harmonic will be given by:

$$|H_k| = \frac{V_{dd}}{N_b}\cdot\left[\frac{1}{\left(\sqrt{1+\left(\frac{k\cdot 2\pi f_0}{\omega_p}\right)^2}\right)^2}\right]^4 \tag{4.6}$$

The pole must be designed such that the total RMS amplitude of the harmonics, as expressed in Equation 4.6, is below the 1/2 LSB requirement. For N-bit resolution, this can be expressed as follows:

$$\sum_k |H_k| \le \frac{1}{2}\cdot\frac{FSR}{2^N-1} \tag{4.7}$$

The optimal ω_p which satisfies the above inequality was determined numerically by Matlab. An initial guess was made as to the range of frequencies in which ω_p would reside,

and then a sweep of that range is done, computing the expression in Equation 4.7 for each point. The highest value of ω_p for which that inequality is satisfied is then chosen. With ω_p determined, the filter's passive elements can be chosen.

4.3.2 - Filter Design

Since the stages are chosen to be identical, there are only 4 passive elements whose values need to be determined. The stage transfer function, in its general form and in terms of the passive elements, is given below:

$$T(s) = \frac{\omega_p^2}{s^2 + 2\delta\omega_p + \omega_p^2} = \frac{1/R_1R_2C_1C_2}{s^2 + \left(\frac{1}{R_1} + \frac{1}{R_2}\right) + \frac{1}{R_1R_2C_1C_2}} \quad \textbf{(4.8)}$$

The damping factor, δ, is chosen to be 1 for fastest settling. Equating the 2nd and 3rd terms in the denominators on both sides of the equation, we establish the following relationships between ω_p and the passive components:

$$C_1C_2 = \frac{1}{R_1R_2\omega_n^2} \quad \textbf{(4.9)}$$

$$2\omega_n = \frac{1}{R_1C_1} + \frac{1}{R_2C_1} \quad \textbf{(4.10)}$$

We have 4 variables and 2 equations. If we set R_1 *and* R_2 to a reasonable common value R, then we can solve the 2 equations for their 2 unknowns, and we obtain the following expression:

$$C_1 = C_2 = \frac{1}{R\omega_n} \tag{4.11}$$

4.3.3 - Calibration Memory

The calibration memory is composed of chains of D flip-flops, which are readily available from a digital standard cell library, and offer the possibility of simple serial loading without the need for addressing. The down side is the large cell area, compared to compact SRAM based memory. Each memory row corresponds to one DAC output and will store its corresponding bitstream. Thus, it will contain 2^N memory cells, where N is the required accuracy of the DAC. For an M-bit DAC, there will be 2^M rows of memory. Thus, the two 4-bit DACs of our prototype will contain $2^{10} \times 16$ and $2^6 \times 16$ memory cells, respectively.

4.3.4 - Bitstream Generator

As mentioned in Chapter 3, the bitstream generator is a digital FIFO containing a data and a control cell. The control cell implements an asynchronous handshaking protocol whereby each cell is either tagged as full/opaque or empty/transparent. For data to be passed from stage i to $i+1$, stage i must be full (ie: contain data), and stage $i+1$ must be empty. These conditions are indicated via a request/acknowledge pair of signals. A request is sent when a stage is empty, and the acknowledge signal is raised when a stage has accepted data and is full. For details of the design, the reader is referred to [3].

4.4 - Residue Block Design

4.4.1 - Residue and DAC Amplifier

The amplifiers needed for the DAC and the residue block have the common requirement of a rail-to-rail input and output swing. The targetted gain and bandwidth for each of those two blocks is different due to different resolution and settling requirements. As was shown in Chapter 3, the DAC amplifier can have lower than required gain, and compensate through calibration. The residue amplifier requires a gain equivalent to 6 bits.

Figure 4.8 shows the rail-to-rail amplifier used as the basis for both circuits, and Table 4.3 shows the aspect ratios for the transistors. It is based on the folded cascode structure. The complementary input differential pairs M_1-M_4 allow a rail-to-rail input range. Transistors M_9-M_{15} keep the current through M_{18}-M_{21} constant over the input range, and in this way stabilize the output impedance of that circuit [30]. The idea is as follows: Transistors M_9-M_{12} are a replica of M_3-M_6, such that I_9=I_3 and I_{10}=I_4. In other words, the common-mode (CM) currents through M_3-M_4 and M_9-M_{10} are the same. Currents I_9 and I_{10} are mirrored to M_{14} and M_{15} such that $I_{14,15}$= $I_{9,10,CM}$ and therefore $I_{14,15}$=$I_{3,4,CM}$. Thus, M_{14} and M_{15} absorb any changes in the common-mode currents of M_3 and M_4, and thereby allow $I_{18,CM}$-$I_{21,CM}$, and therefore $r_{ds,19}$ and $r_{ds,21}$, to remain constant. To fully stabilize the gain over the input range, g_m control must also be imple-

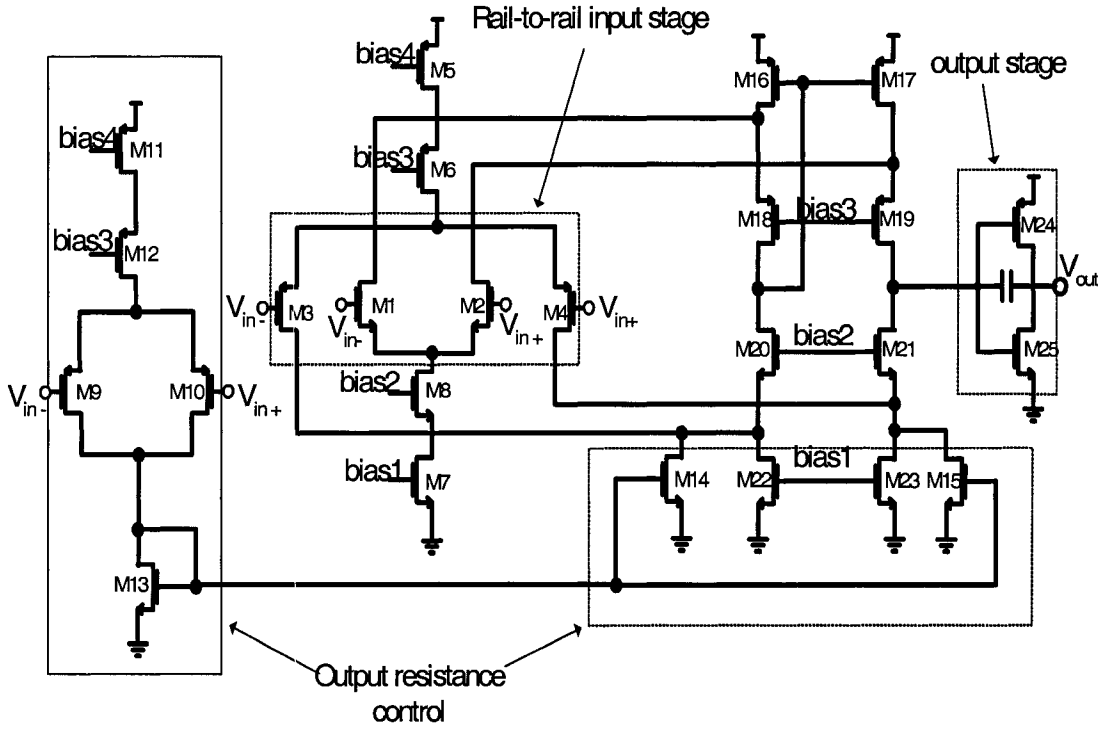


Figure 4.8. DAC and Residue Block Amplifier

mented. Having a constant gain allows the picking of an optimal compensation capacitor,

TABLE 4.3. Aspect Ratios (in μm) for DAC/Residue Block Amplifier

Transistor	*size*	*Transistor*	*size*	*Transistor*	*size*
M1-M2	35.2/0.36	M9-M10	50/0.36	M18-M19	10/0.36
M3-M4	50/0.36	M11	50.8/0.36	M20-M21	7/0.36
M5	50.8/0.36	M12	15/0.36	M22-M23	1.5/0.36
M6	15/0.36	M13	3.8/0.36	M24	50/0.36
M7	17/0.36	M14-M15	1.96/0.36	M25	5/0.36
M8	5/0.36	M16-M17	20/0.36		

therefore yielding optimal bandwidth and no unnecessary noise. However, for the pur-

poses of this application, the amount of gain stabilization provided by this output resistance control circuit was quite acceptable, and g_m control was deemed unnecessary given the additional complexity and design effort involved.

An output stage is necessary for two reasons. First, it is necessary to have a rail-to-rail output stage, which is not the case for the folded cascode stage. Secondly, the output stage is needed to drive the resistive load of each filter stage. Thus, a low output resistance circuit, such as a combination of P and N source follower transistors, is usually employed. However, the output swing of such a circuit is limited by the V_{gs} between gate and source. The output stage used here consists of a an inverter used as an amplifier. This circuit has an output swing of $V_{dd} - V_{ds-sat-P} - V_{ds-sat-N}$, and is therefore able to come much closer to the rail. The residue amplifier requires an output range of $0 \le V_{out} \le \frac{15 \cdot FSR}{16}$, which reduces the range requirements at the positive rail. A certain amount of distortion, depending on the value of $V_{ds\text{-}sat}$, arises when the output signal tries to swing to the negative rail and the N transistor goes into triode. The remedy to this problem, although far from ideal, was to overdesign the opamp's first stage gain, in order to leave a margin of safety for the deterioration around the rails. The high output resistance of this circuit is only an issue in the case of the DAC filter, which is driving a resistive load. However, the choice of large resistor values for the filter as well as the operation of the amplifier in unity-gain feedback largely attenuate this problem. The same circuit is used for both applications, except that the compensation capacitor is larger for the DAC amplifier operating in unity-gain feedback.

4.4.2 - Switched-Capacitor Network

The implementation is a standard switched-capacitor amplifier, shown in Figure 4.9. It operates from a 2-phase non-overlapping clock. Its ideal transfer function

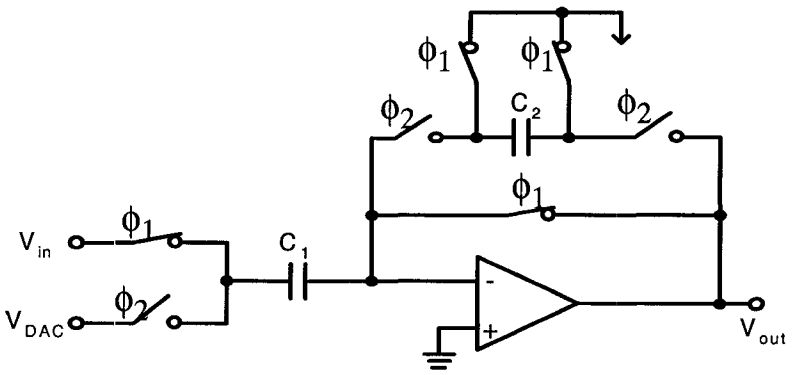


Figure 4.9. Residue Generation Circuit

can be obtained by expressing the charge transfer from C_1 to C_2 at the beginning of phase ϕ_2:

$$C_1 \cdot (V_{DAC} - V_{in}) = C_2 \cdot (0 - V_{in}) \quad \textbf{(4.12)}$$

$$V_{out} = \frac{C_1}{C_2} \cdot (V_{in} - V_{DAC}) \quad \textbf{(4.13)}$$

The amplification is thus set by the ratio of C_1 and C_2. Note that the switches are implemented as full transmission gates. This circuit does not completely cancel the input offset voltage V_{off}, but it does prevent it from being amplified, such that the output offset equals

the input offset and is therefore very small. Indeed, the charge transfer equation with V_{off} accounted for is:

$$C_1 \cdot [V_{DAC} - V_{off} - (V_{in} - V_{off})] = C_2 \cdot [V_{off} - V_{out} - (0 - 0)] \quad \textbf{(4.14)}$$

$$V_{out} = \frac{C_1}{C_2} \cdot (V_{in} - V_{DAC}) + V_{off} \quad \textbf{(4.15)}$$

The output offset can be calibrated out through the routine described in Chapter 3. Nevertheless, the circuit in Figure 4.9 can be adjusted to allow a complete cancellation of V_{off}.

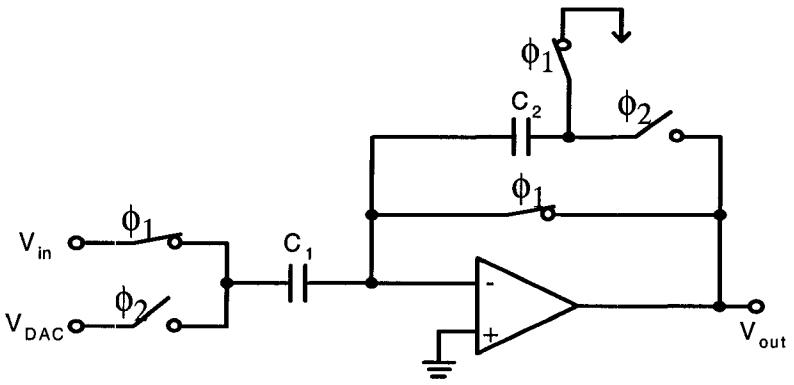


Figure 4.10. Offset-Free Residue Generation Circuit

This is shown in Figure 4.10, where the left-hand plate of C_2 is now always connected to the opamp inverting node. The charge equation for this circuit is:

$$C_1 \cdot [V_{DAC} - V_{off} - (V_{in} - V_{off})] = C_2 \cdot [V_{off} - V_{out} - (V_{off} - 0)] \quad \textbf{(4.16)}$$

which gives the ideal Equation 4.13 upon rearrangement.

4.5 - Other Circuit Blocks

4.5.1 - Bias Generator

The bias circuit is shown in Figure 4.10. It requires 3 external current sources as

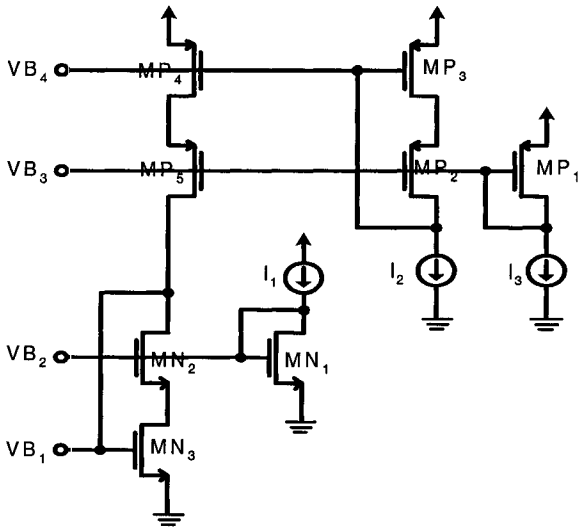


Figure 4.11. Bias Voltage Generator

inputs, and provides 4 bias voltage taps. The main design consideration there was to ensure that the non diode-connected transistors operate well in saturation. The bias circuit is used for both the amplifier and the comparator.

4.5.2 - Digital Data Input and Output

Due to the high number of digital signals needed to be observed, and the limited digital channels on the testing device (a commercial Automated Test Equipment, or ATE),

groups of digital signals had to be converted to/from serial form in order to be input/output through one single pin. For instance, the digital outputs of the ADC are first loaded in parallel into10 flip-flops which are connected as a chain. They are then clocked out of the chain in serial mode through the output pin of the last flip-flop. The inverse operation, which is the deserializing of incoming serial data, functions in an analogous manner. The logic for both cases is simple, requiring only multiplexers and flip-flops, and is not shown here.

4.6 - Conclusions

This chapter has outlined the various circuits used in the implementation of a 10-bit 3-step bitstream-based ADC. The nominal resolution of the ADC was chosen based on a speed-resolution tradeoff. The use of bitstream processing imposed the need for rail-to-rail signal swings in the analog circuits, which are the comparator and the amplifier. The design of an 8^{th} order active low-pass filter is also presented. The amplifier used in that circuit is also used within a switched-capacitor network which performs the residue generation for each stage. Also presented are circuits used in the calibration process, such as an analog demultiplexer and a memory bank, as well as an input track/hold, a bias generator, and the required digital logic for the ADC.

Chapter 5 - Experimental Results

5.1 - High-Level Behavioral Modeling

Before beginning the design phase of the experimental prototype, a high-level model of the system was built. The fundamental reason for doing this is to be able to identify system-level issues and tradeoffs without having the circuit-level complexity and detail. This allows the modeling and simulation of the system in terms of high-level parameters such as transconductance, output resistance, bias current, or feedback factor, rather than circuit-level details such as transistor size. With this clearer picture, one can more easily verify system functionality, and perhaps optimize the system parameters to obtain the best set which meets the design objectives.

The model of the ADC was built in Matlab and Simulink. The latter is a graphical system-level simulator bundled with Matlab. The system was drawn in simulink, while its execution was done from a Matlab file. The model is shown in Figure 5.1. The system models thermal noise, ADC offset, DAC AC ripple, residue amplifier settling error and slew rate limiting. The ADC offset is modeled as a random variable with a mean of zero and a standard deviation of a few tens of millivolts. The thermal noise was also determined by multiplying

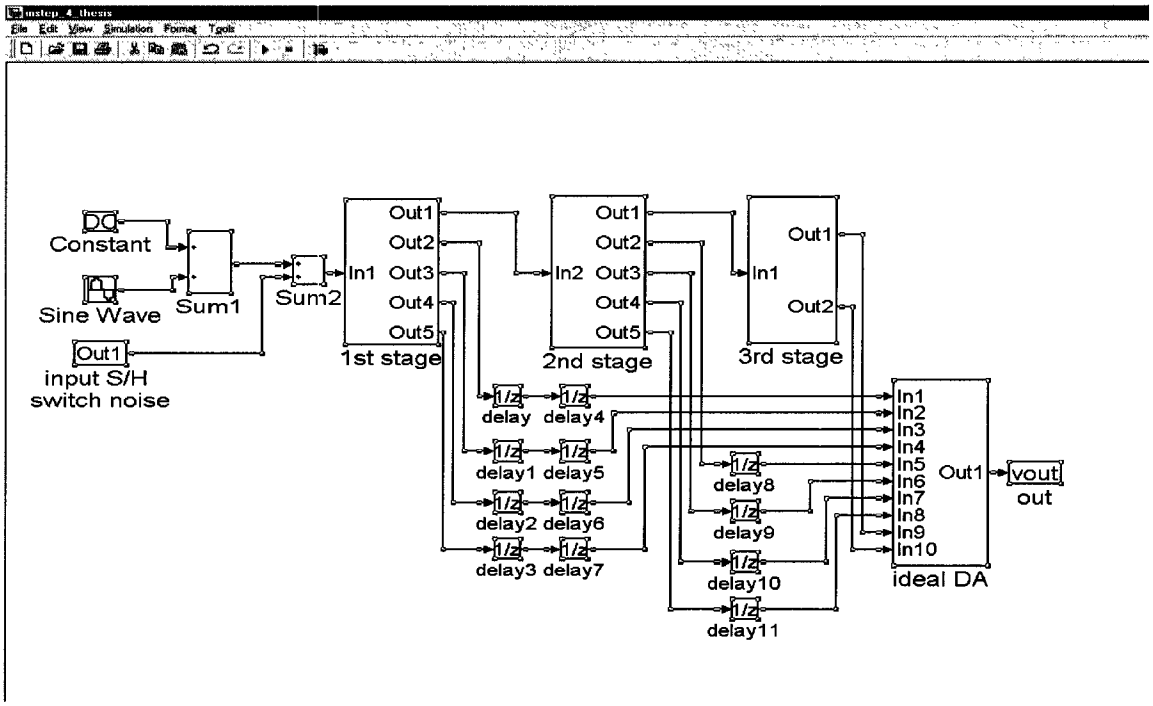


Figure 5.1. Matlab Model of a 3-Step, 10-bit ADC

the $\frac{kT}{C}$ formula for thermal noise by a random variable with a zero mean and standard deviation of 1. Figure 5.2 shows the models for each of the main circuits in the ADC stage. Using the model shown in Figure 5.3(a), a verification of the comparator calibration algorithm described in Chapter 3 was done. The model contains a bitstream pattern as the source, an analog filter, and the relay element of Matlab which has a threshold value, and acts as the comparator. Suppose the comparator has a threshold voltage of 0.565 V and an offset of –0.0025 V, in other words an *effective* threshold voltage of 0.5625 V. Figure 5.3(b) shows the comparator response for such a situation. It can be seen that the

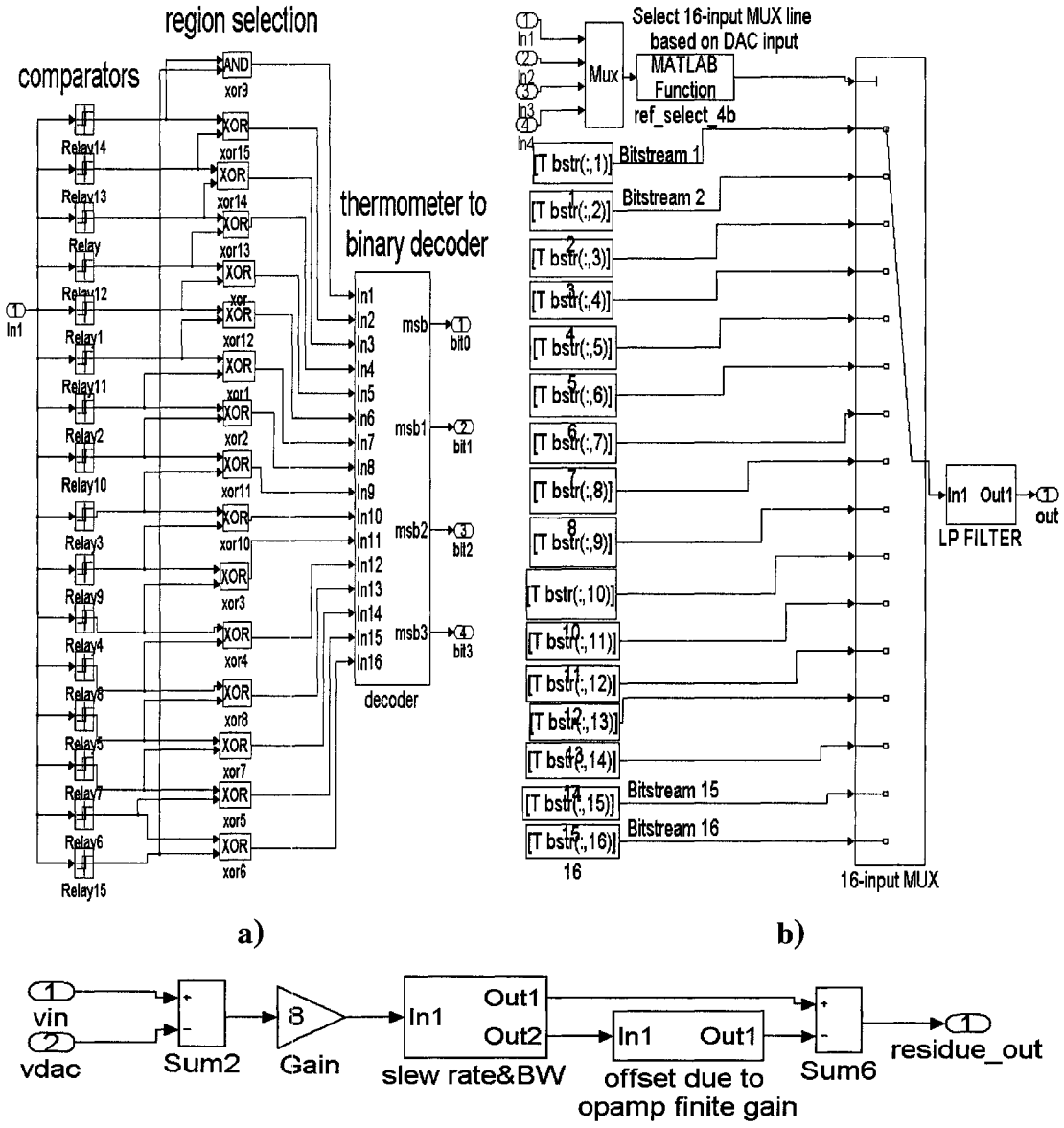


Figure 5.2. Models of Sub-blocks in ADC Stage a) Sub-ADC b) DAC c)Residue Amplifier

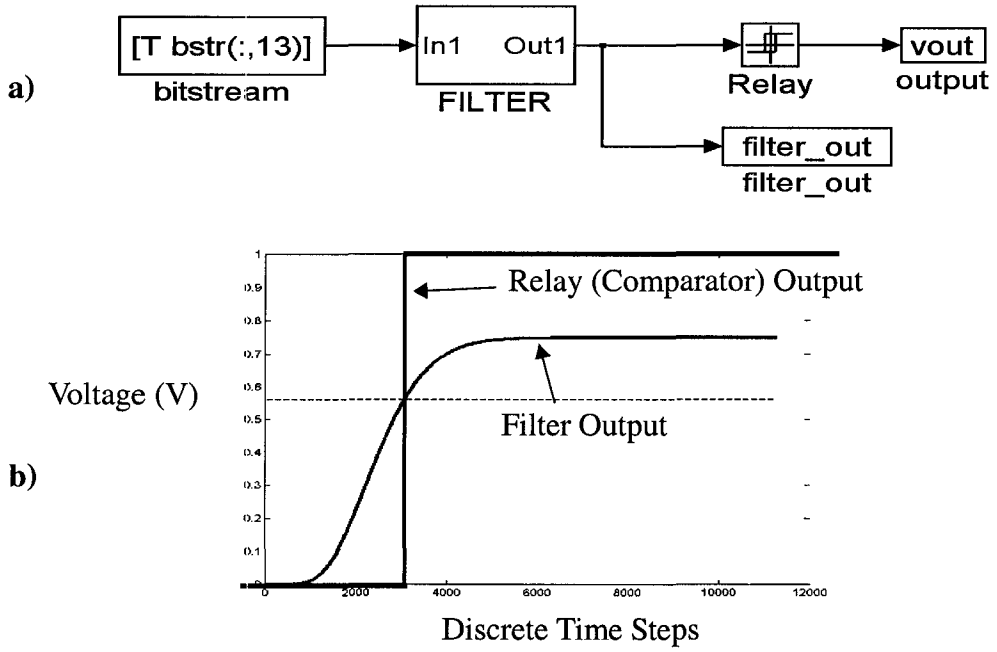


Figure 5.3. a) Comparator Setup for Calibration b) Comparator Response

relay toggles when the filter input crosses the effective threshold voltage of 0.5625 V. For illustration purposes, the filter swings to a much higher value here, but in practice, the value stored on the calibration capacitor would be the one at which the comparator toggles. The model of the ADC shown in Figure 5.1 was simulated for a few cycles of the input sinusoidal waveform with the non-idealities addressed by calibration removed (ie: as if it were fully calibrated). The performance was interpreted by running an FFT of the output bits reconverted to analog through an ideal D/A conversion process. The FFT graph clearly displaying the input tone in bin 31 is shown in Figure 5.4. The signal-to-noise-and-distortion ratio (SNDR) given by this simulation is 60.8 dB, corresponding to

10 bits of nominal resolution. As well, DNL and INL measurements of the output waveform were done in Matlab, and the graphs are shown in Figure 5.5.

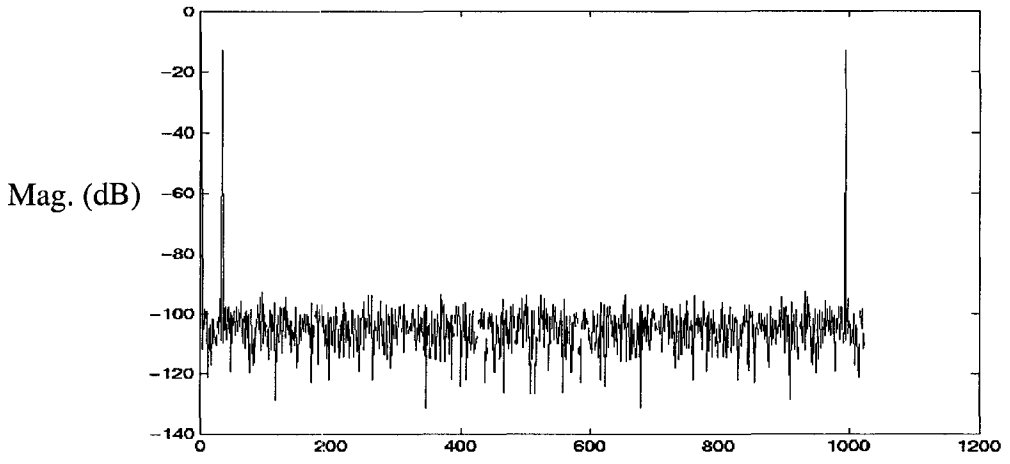


Figure 5.4. FFT of ADC output - signal tone at bin 31

These Matlab models provided a good degree of familiarization with the system-level signal processing issues. However, optimization of the high-level parameters was not pursued because in this case, the optimal values of those parameters were given by the overall specifications of the system. For instance, the filter bandwidth is given by the attenuation specification, while the amplifier gain is dictated by the gain error specification. Both of these parameters in turn depend on the ADC resolution requirements, which is a system level specification.

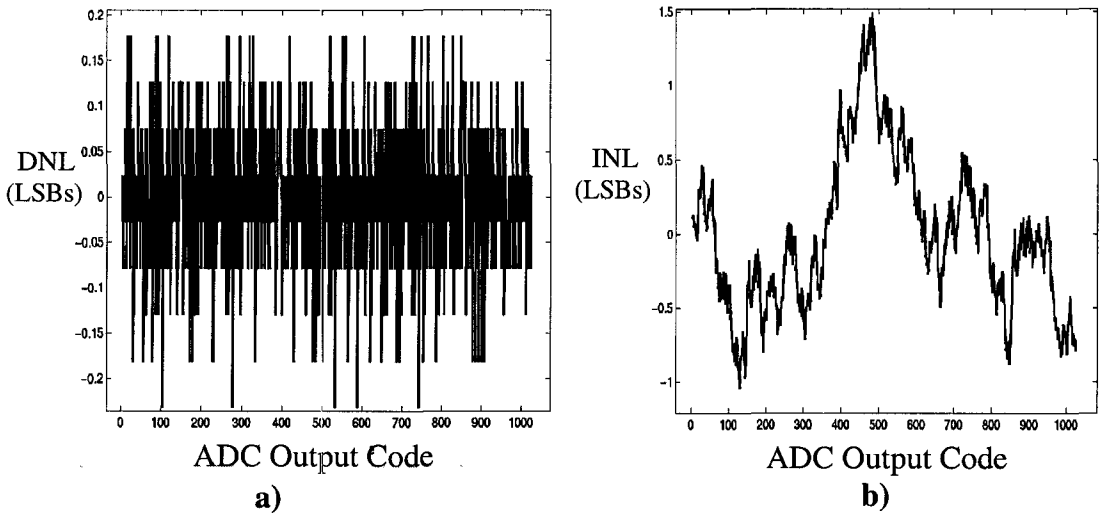


Figure 5.5. ADC Non-linearity Results Based on Matlab Simulations a) Differential Non-Linearity b) Integral Non-Linearity

5.2 - Simulation Results for IC Prototype

Simulations were done in Cadence using the typical case of the HSPICE models. In this section, simulation results for the main circuit blocks are shown. A simulation of the entire ADC in Cadence was not possible for 2 reasons. First, a calibration routine involving external digital patterns had to be performed to put the ADC in normal operation mode. This could not be performed in the simulation environment. Secondly, even if the ADC were simulated by skipping the calibration and inputting the required initial conditions, the complexity of the circuit would impose excessively long simulation times, which would be impossible to handle even by the superior computing facilities available.

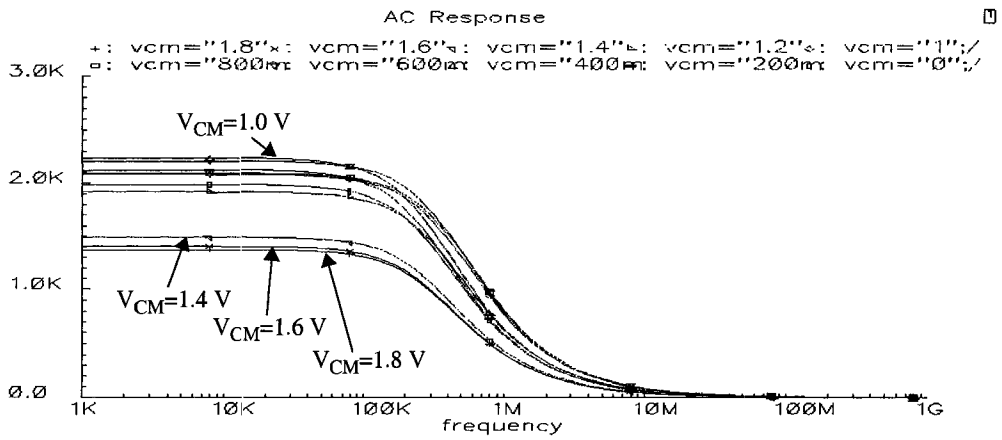


Figure 5.6. AC analysis of Amplifier with Sweep of Common-Mode Voltage

Figure 5.6 shows an AC analysis of the DAC amplifier with the common-mode voltage varied over the input range. Although the gain does vary over the input range due to the lack of g_m control, it is still high enough to satisfy by far the gain requirements. The gain ranges from 62 to 67 dB.

Figure 5.7 shows the outputs of the 4 stages of the DAC filter for the bitstream pattern of 127 ones out of 1024 bits. This should theoretically provide $V_{dd} \bullet 127/1024$ V, and include a fundamental tone at $F_s/1024$. A close-up of the output of the last stage is shown in Figure 5.8. It can be seen that the ripple is minimized to 20μV, which represents 0.011 LSBs. The filter has been overdesigned by a large margin to leave room for variations in the passband frequency and to ensure that the desired objective is met even though the set-

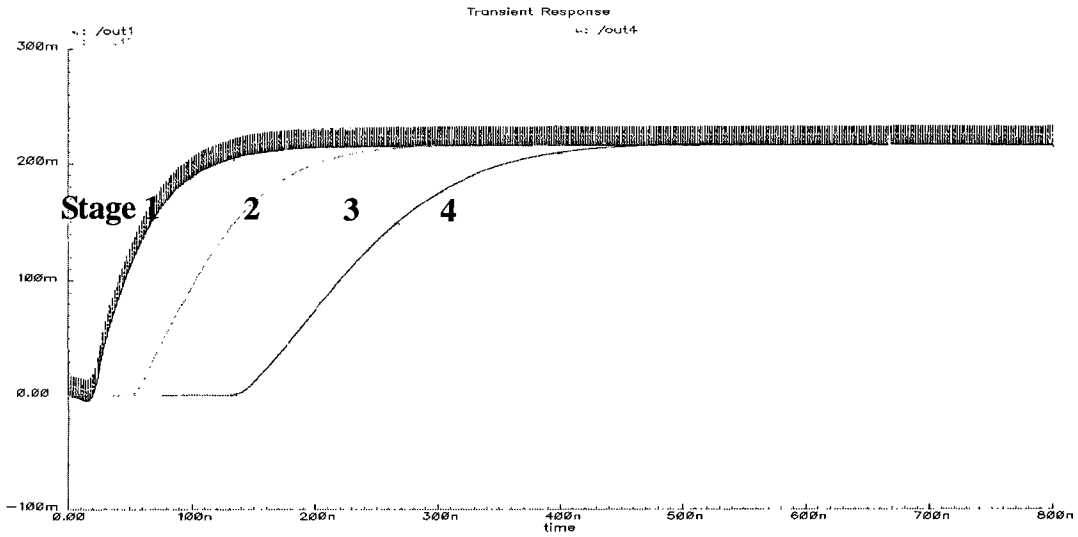


Figure 5.7. Outputs of Stages 1-4 of DAC Filter

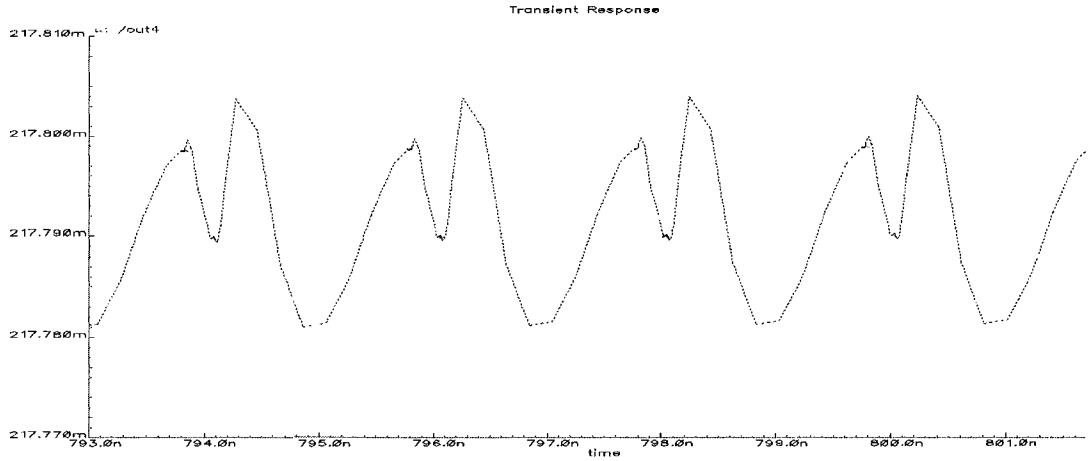


Figure 5.8. Last Stage Output of DAC Filter (Zoomed in)

tling time may be affected. Nevertheless, the settling time is still good on the order of 700 ns, which keeps the overall ADC throughput to 1 MHz.

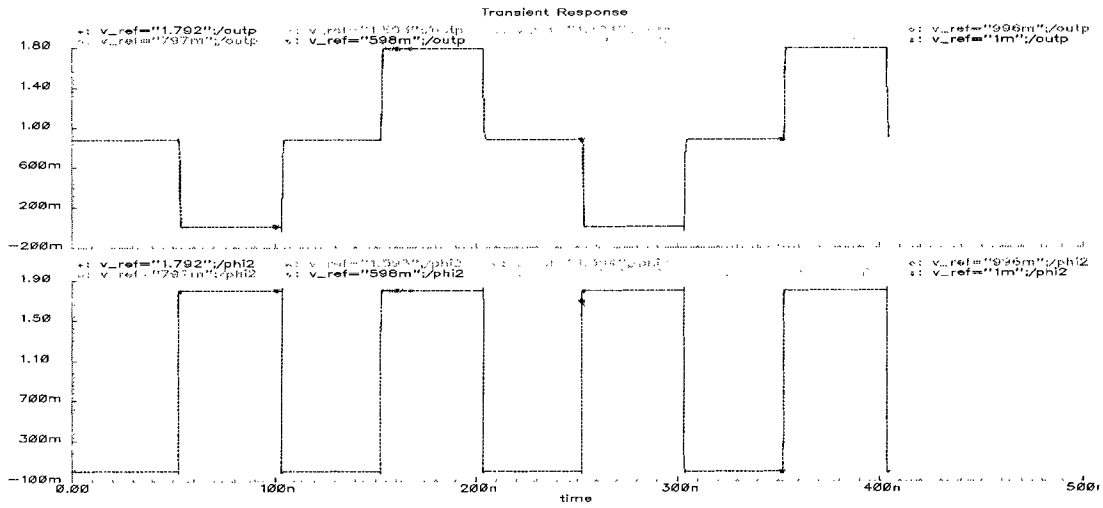


Figure 5.9. Last Stage Output of DAC Filter (Zoomed in)

A plot of the comparator outputs is shown in Figure 5.9. The top curve shows the comparator outputs for different common-mode levels, while the bottom curve shows the clock ϕ_2. When ϕ_2 is high, the comparator output is valid. The input is a 300 μV (0.17 LSB) differential signal with reversing polarity on each cycle. It can be seen that the comparator output responds correctly to the alternating input at different common-mode levels Finally, Table 5.1 summarizes the circuit specifications based on both Cadence and Matlab simulations..

TABLE 5.1. Summary of Simulation Results

Specification	Value
Supply voltage	1.8 V
Comparator resolution	11.5 bits

TABLE 5.1. Summary of Simulation Results

Specification	Value
Comparator settling time	<100 ns
Filter ripple	0.011 LSB
Filter Settling time	700 ns
DAC Amplifier DC Gain	62-67 dB
DAC Amplifier UGBW	100-180 MHz
ADC Throughput	1 MHz
ADC Resolution	10 bits
Total Power Consumption	190 mW
Active Die Area	3.1 mm x 3.1 mm

5.3 - IC Prototype

A prototype of the circuit was built in TSMC's 0.18μ CMOS process. This is a single-poly, six-metal process with a mixed-signal option offering area efficient metal-insulator-metal capacitors. The prototype is shown in Figure 5.10. The active die area is 3.1 mm x 3.1 mm. Thanks to the mixed-signal option, the many capacitors used in this design occupy a relatively small area, while the majority of the area is actually occupied by the flip-flop based calibration memory. In the long run, this can be replaced by area-efficient SRAM cells which occupy approximately $1/10^{th}$ of the same space. Note that the circuits were designed in a single-ended configuration in part because a differential approach would require twice as many comparators and calibration capacitors, making it difficult if not impossible to fit in the allotted area.

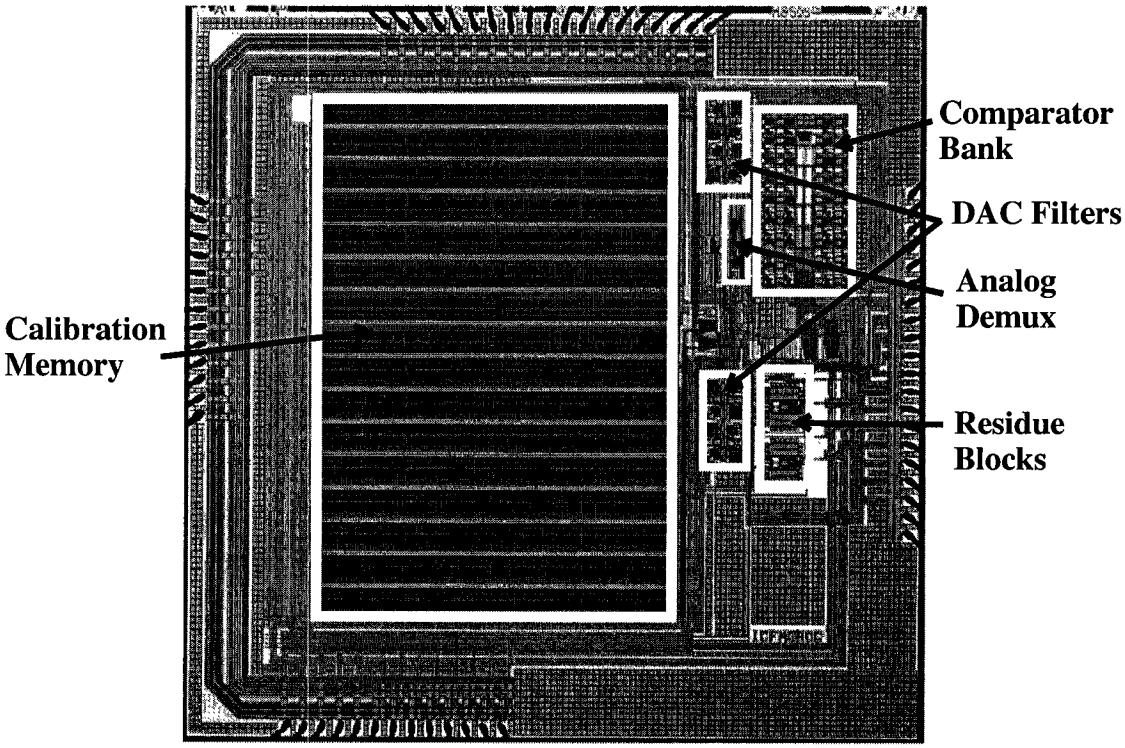


Figure 5.10. IC Microphotograph of Calibratable 3-Step ADC

A second run of the prototype was also fabricated. This version includes scan chains for monitoring the outputs of the comparator bank, as well as addressing a minor design error detected in the 3rd stage, 2-bit ADC of the first chip.

5.4 - Testing and Troubleshooting

5.4.1 - Test Setup

The testing of the chip was done using Teradyne's A567 production-type mixed-signal tester. The tester consists of a mainframe cage, a test head, and a console from which the user can interface with the tester. The mainframe and the test head contain all the electronic cards which implement the test functions. The user terminal consists of a monitor and a keyboard, which are linked to a computer inside the mainframe. The device under test (DUT) mounts on a Device Interface Board (DIB) which sits on the test head. The mixed-signal tester is shown in Figure 5.11.

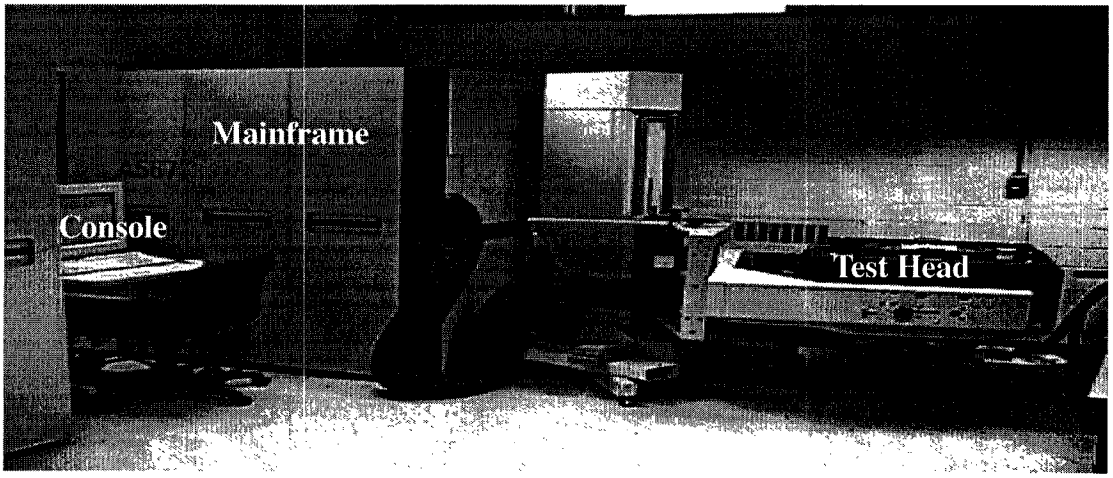


Figure 5.11. Mixed-Signal Tester

5.4.2 - Preliminary Issues

There were three major problems related to the testing phase of this chip. First, the turn-around time for the chip, including fabrication and packaging, was gradually extended from 3.5 months to 7 months, making it difficult to plan for testing, problem identification, and possibly the fabrication of a second prototype.

Secondly, there were problems with the chip itself. A short circuit was identified on the digital supply as it could not be driven to the required voltage. It was possible at first to reach the nominal supply voltage by using the high current unit of the tester to source about 800 mA of current across the low supply resistance. Testing could continue at this point, but not without a certain amount of uncertainty as to the reliability and validity of test results. Eventually, the short circuit was localized and found to be due to a labelling error in one of the core supply I/O pads. Within a row of 5 $V_{ss\text{-}core}$ I/Os taken from a standard cell library, the last one turned out to be a $V_{dd\text{-}core}$ pad, which was connected to the $V_{ss\text{-}core}$ bus. Since it is also connected to $V_{dd\text{-}core}$ through the ring, the supplies are effectively shorted. The I/Os are at the top level of the design and are usually not LVS'd, which is why this error was left uncorrected. Figure 5.12 shows the erroneous pad and the cut coordinates as identified by FIBics, the organization which provided the service. This short circuit was successfully repaired using focused ion beam microsurgery on two IC units. This procedure is often used for scanning the image of silicon wafers to produce an image. For milling applications, as was the case here, the usual scanning is done with a beam of Gallium ions (Ga^+) but is also combined with the application of a gas to produce selective chemical etching of conductive material. This additional processing is

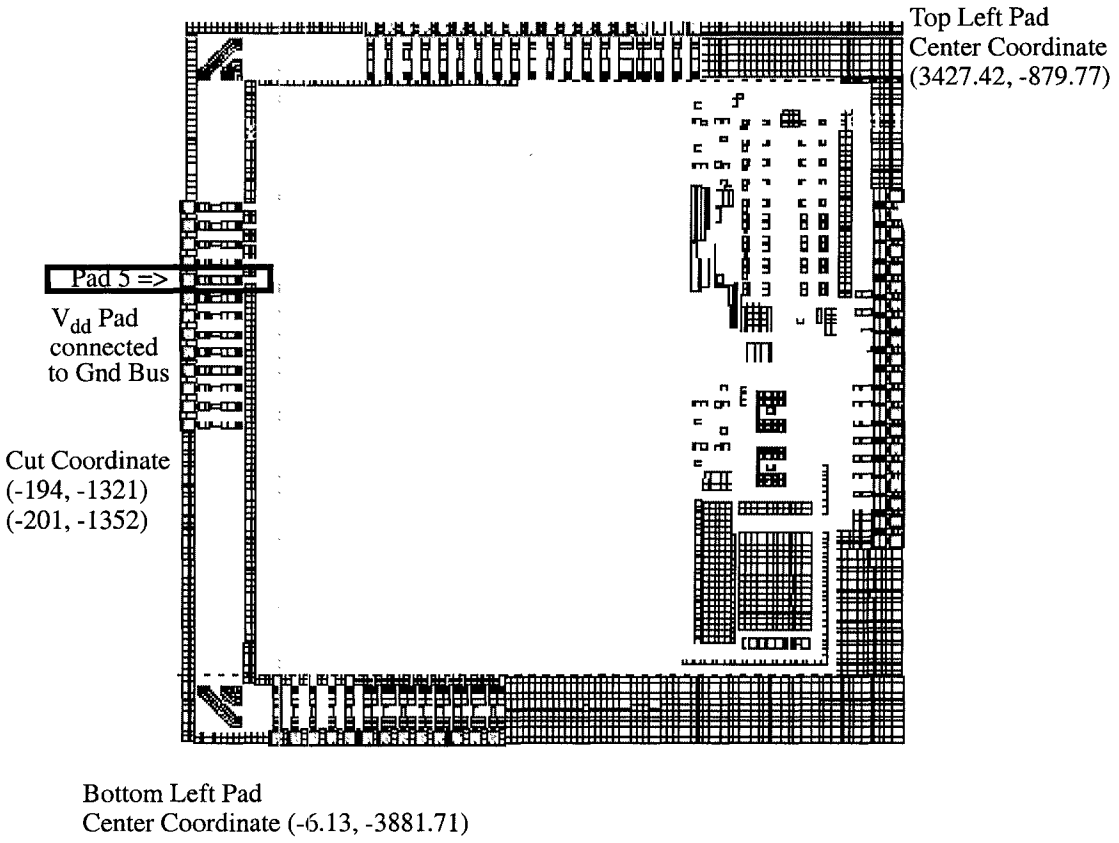


Figure 5.12. Defective I/O (Pad 5) Causing Short Circuit on Chip

referred to as Gas Assisted Etching (GAE). The result of the repair done on the IC units is shown in Figure 5.13.

Thirdly, the failure of the DAC during initial tests was another problem which was explained upon closer observation of the layout of the DAC. The layout of the basic cell

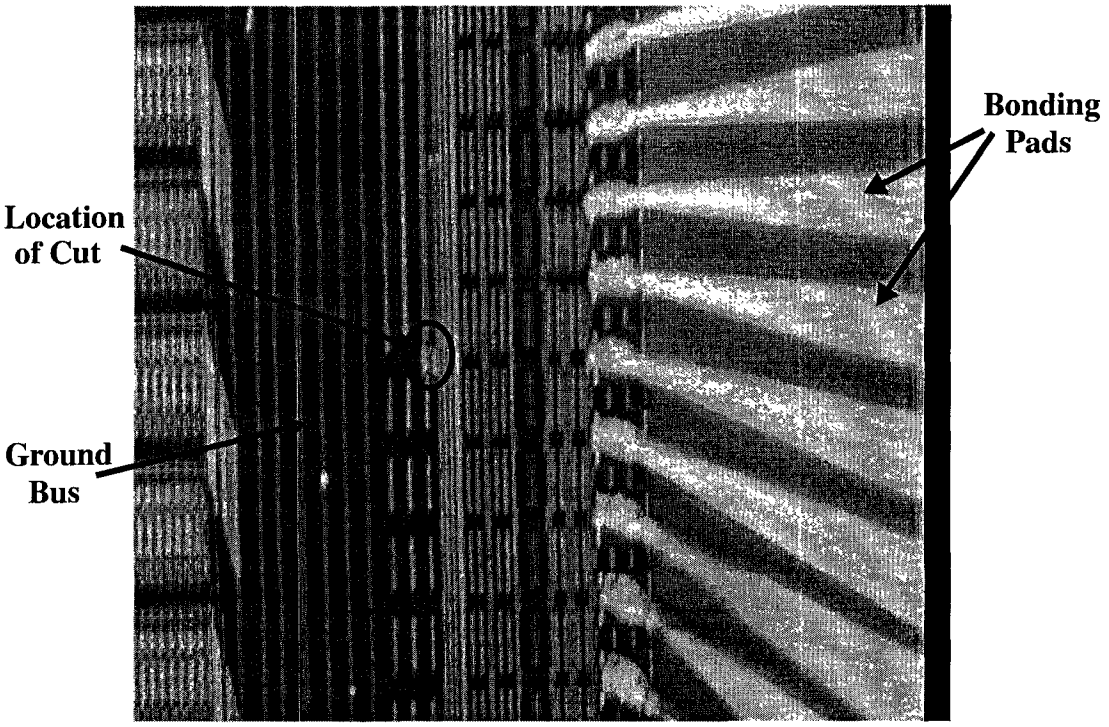


Figure 5.13. Microphotograph of IC Section with FIB Cut

in the asynchronous FIFO contains 2 sets of power and ground buses. When introduced into the higher level circuit, only one set of power and ground were connected to their respective rings, and the other pair was not connected. This is seen in the layout of the cell, shown in Figure 5.14. Normally, such a floating net would be flagged by the LVS verification routine, but unfortunately it was not flagged in either of the two runs of the chip. Since this basic building block is repeated a large number of times, the metal growing procedure available as part of microsurgery was not feasible for fixing the problem. Unfortunately, since the calibration procedure depends on the DAC being functional, it could not be performed due to the aforementioned handicap.

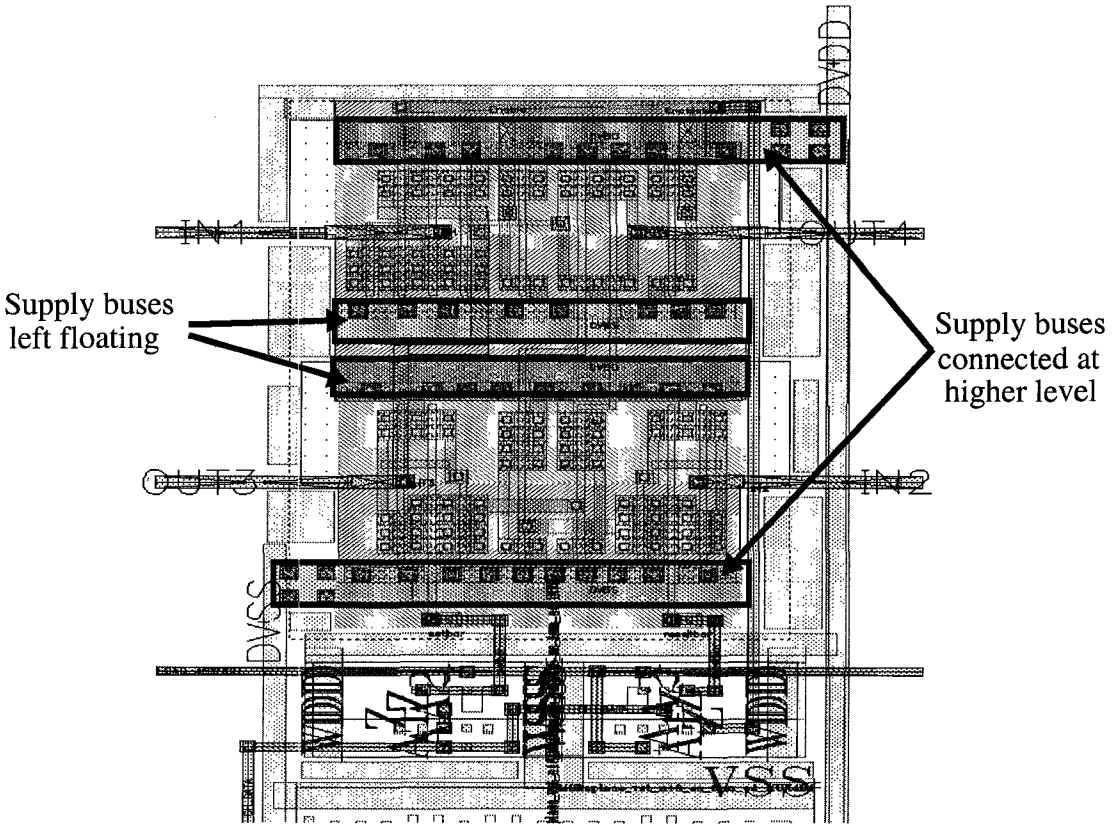


Figure 5.14. Layout of Cell with Floating Supply Buses not Flagged by LVS

5.4.3 - Comparator and Residue Amplifier Testing

It was possible to test the comparators alone since their inputs and outputs in calibration mode were accessible through I/O pins. Thus, instead of the DAC output as the stimulus, an external DC signal was injected, as shown in Figure 5.15. The comparator was tested with a differential signal at its input much larger than its expected resolution.

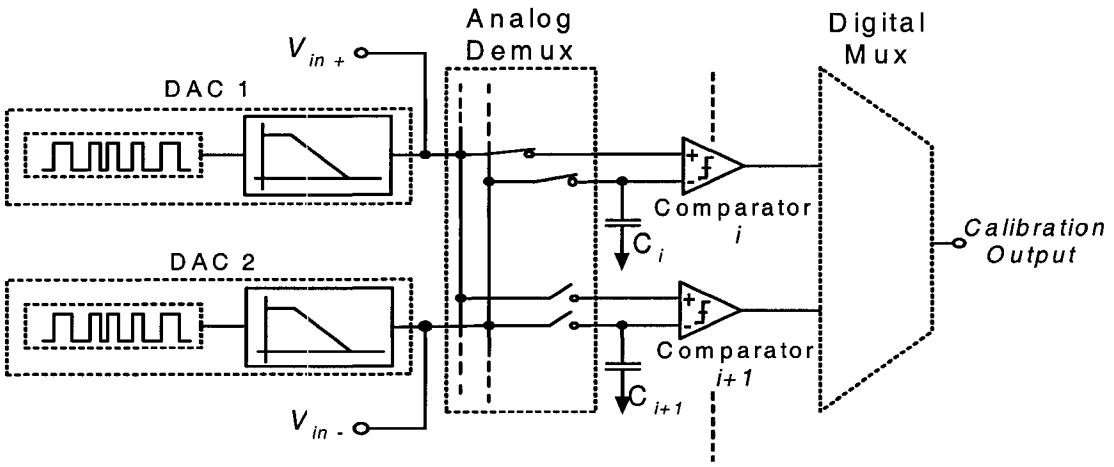


Figure 5.15. Test Setup for Comparators

Despite many attempts, however, the comparator did not react to even the largest differential signals, and its output always remained at zero. A number of troubleshooting operations were done in order to eliminate some possibilities of error in the test setup, and attempt to draw a clearer picture of the problem. Some of these are summarized below:

- All input signals, clocks, and supply voltages were verified for continuity at the inputs to the test board, as well as the inputs to the chip.
- Some of the digital inputs were originally sent to the DUT by sourcing the digital bits from an external file. In order to eliminate the possibility of the sourcing operation being faulty, those digital input were hardcoded in the pattern file. This did not change the test results.
- There are two clock pads for each clock signal entering the chip. The testing was also carried out with the clock signal connected at first to both pads, and then only to

a single pad. No change was observed, which makes sense since two clock pads in parallel behave as a single clock pad with double-sized drivers.

- Using a microscope, an attempt was made to check for continuity across the digital input pads. The objective was to input a digital signal at the highest frequency the pad could handle, and hope that the signal would couple through the glass covering of the chip and be picked up by the AC probe of the microscope. However, the digital pads offered through this technology only operate at up to 50 MHz, which was too low to short the glass covering, and therefore this test was not able to yield meaningful results.
- The bias voltages are generated by an internal bias circuit which requires external current sources. These current sources were adjusted so that the bias voltages were as close as possible to the nominal bias voltages from simulations. They were also varied around their nominal values. In both cases, the tests provided no further breakthroughs. Nevertheless, it was proven that the internal bias circuit functions properly.
- The analog demultiplexer, comparator, multiplexer, and serial/parallel converter were all re-simulated in Cadence, and showed proper functionality as expected.

During the comparator tests, it was observed that despite the repair done to fix the short circuit on the two prototypes, the overload alarm of the digital current source still came on when it was sourcing less than 400 mA. This was obviously too high since the only active digital components for this test were the multiplexer, demultiplexer, and the serial/parallel converter. This means that at some point, a peak current of 400 mA was being drawn from the digital supply. When in steady state, however, the impedance between the digital supplies was quite high (~25 MΩ). This would imply that the short circuit condition occurred during the switching of some digital gates, rather than in steady

state. The analog supplies feeding the comparators operated at normal current levels. Although it cannot be ascertained, it is more likely that the cause of the failure lies in the digital part of the chip. There are other factors, in addition to the high current levels seen through the digital supplies, which lead to this conclusion: If there were a performance degradation in the comparators, one should still be able to see it work at larger input differential voltage levels. In case of a functional fault in the comparators, the output must not necessarily always remain at zero, and could toggle to one for certain test conditions. Finally, if a physical defect existed on the die itself, it should only affect up to a few comparators, but certainly not all of them. It turns out that none of these three conditions hold here since the outputs of all the comparators are stuck at zero regardless of the test conditions.

It was also possible to test the residue amplifier with direct analog inputs. The residue amplifier test was faced with a similar problem, where digital supply currents below 205 mA would cause an overload condition. Once again, the impedance across the digital supplies at steady state is high (~25 MΩ). This situation was rather unexpected, however, since this particular test did not use the same digital circuits as the comparator tests. The only digital logic used here were some drivers for the 2 clock phases of the residue amplifier. It is improbable that a fault would exist in 2 different parts of the circuit (ie: in both the comparator and the residue amplifier logic), but at this point this seems to be the only explanation.

5.5 - Component-level Measurement Results

Near the end of the testing phase, and due to the presence of faults on the previously described integrated circuit, a separate IC was fabricated that contained the various components as stand-alone devices which were directly connected to I/Os. Thus, direct testing of those components was conducted, and Figure 5.15 shows a frequency sweep of the low-pass filter at various input levels. Due to the characteristics of the ATE, the high-

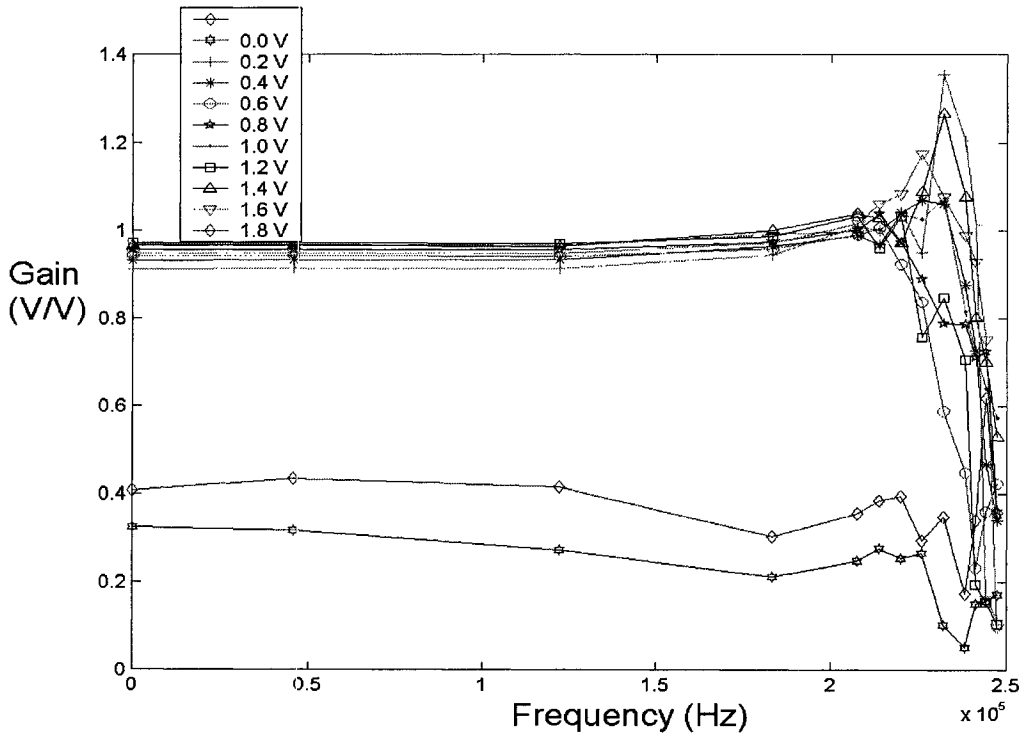


Figure 5.16. Low-Pass Filter Measured Frequency Response

est available analog input test frequency was 250 KHz, and thus the graph only shows data up to that point. It can be seen that the filter achieves unity gain or near-unity gain, as

expected over most of the input range. The gain drops at the very edges of the input range (0V and 1.8V), which only slightly reduces the input voltage range.

Figure 5.15 shows the frequency sweep for the amplfier used in the residue block circuit. The gain of this circuit at DC is about 20dB lower than expected over most of the

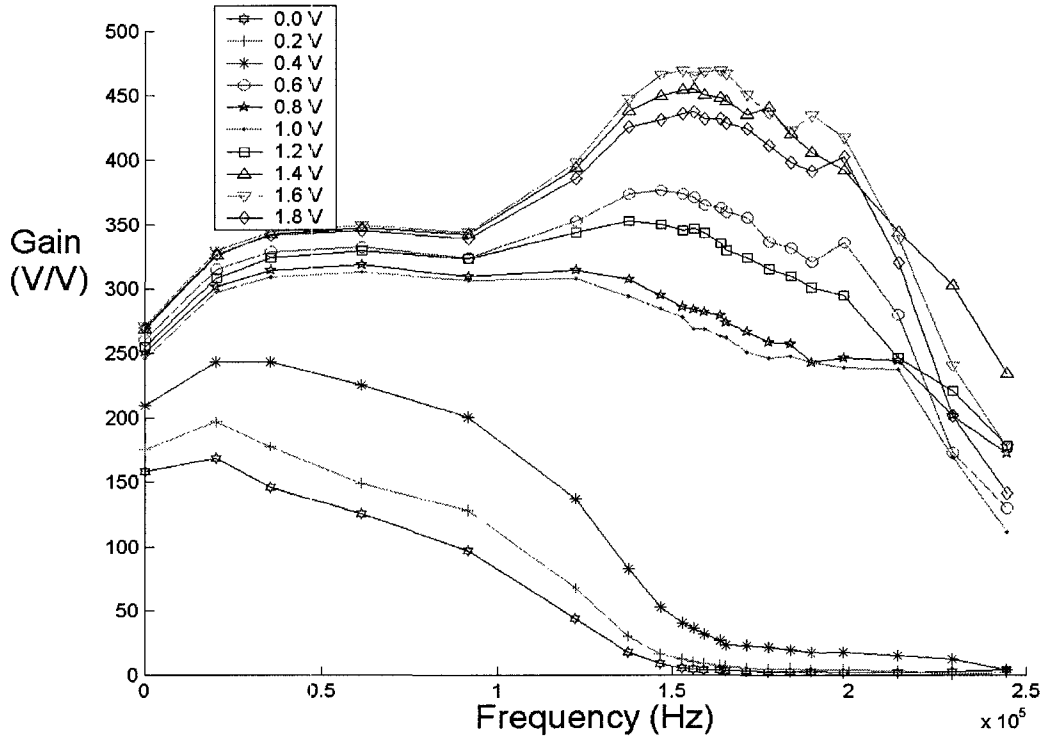


Figure 5.17. Amplifier Measured Frequency Response

input range. There is a more significant drop in the gain and bandwidth at the lower end of the input range (0V-0.4V). Although there is a tendancy to attribute the drop to the fact that the test board is of a low-quality breadboard type, it cannot be the only culprit since

the final measurement is an average of many measurements, and therefore the effect of random noise has been canceled to some extent. It must be noted, however that the gain of the amplifier was originally highly overdesigned, and that this reduced gain is still sufficient for the required resolution.

5.6 - Summary

In this chapter, the work done in the modeling, simulation, IC prototype design, testing, and troubleshooting of the 3-step ADC was presented. Initial system-level modeling of the ADC was described along with simulations of the calibration operation and the ADC performance. Circuit-level simulation results were presented to demonstrate the theoretical function of the main circuits blocks. Next, the issues encountered in the testing phase have been described, as well as the various attempts at resolving them. The certainty with which the root cause of the problems can be established varies among the circuits. The DAC contains an unconnected power supply line which was not flagged by LVS. The comparators and the residue amplifier have digital control circuitry which draws large currents during testing, and is therefore likely to be associated with the malfunctions. The analog portion of the chip did not have such critical faults since the bias circuit showed proper functionality.

Chapter 6 - Conclusions

6.1 - Summary of Work

In this thesis, the subject of Nyquist-rate ADCs was explored from a different angle. The Nyquist-rate ADC incorporated the technique of DC voltage generation from the filtering of a periodic PDM bitstream for its design and calibration. After a brief introduction in Chapter 1, an overview of the preferred architectures for high-speed and high-accuracy were presented in Chapter 2, with an emphasis on their accuracy limitations. Some of the most techniques used to compensate and correct the accuracy errors were presented for each architecture. Chapter 3 introduced the concept of DC voltage generation. The system-level description of the design and calibration of the ADC was presented, and the related issues highlighted. The principal quantities that can be calibrated as part of this system were identified, and it was shown that most of the undesired quantities of offset, gain error, and nonlinearity of the ADC circuit blocks can be calibrated. In Chapter 4, the details of the circuits

were provided. These included, among others, the comparator, amplifier, analog filter, demultiplexer, switched-capacitor multiplier, and bias. Two runs of the prototype IC were fabricated, and the results of the experiment were shown in Chapter 5. An initial short-circuit on the chip was repaired, but the prototype still exhibited excessively high current levels. The testing and troubleshooting attempts were described and the likely causes of failure identified. Simulation results were presented to support the presented circuits.

6.2 - Future Directions

One improvement that could be applied to the IC implementation of the ADC is to replace the crude, DFF-based memory by a more efficient SRAM-based memory array. This is available as part of a standard-cell library. It was calculated that the equivalent area of such an implementation would be $1/10^{th}$ that of the present design. The added complexity of this implementation is the addition of a number of address lines and therefore a higher number of I/Os. However, the address lines can also be serialized into one line, thus alleviating the I/O problem, and also making board routing and testing simpler.

In a future implementation, the analog circuits need to be more thoroughly verified at the simulation level. In particular, the circuits here were only checked for the typical process corners, but would also need to be verified for all the process corners. In addition, it would be desirable to have one copy of the sensitive analog circuits isolated for direct access during testing, thus making analysis and fault detection easier.

On another note, it was seen that the speed of the entire ADC is largely limited by the settling of the DAC and its low-pass filter. The need for the low-pass filter to cut-off

the harmonics of the bitstream is the reason for its low bandwidth. One major improvement to the system would be to accelerate its operation. Thus, the system shown in Figure 6.1 was speculated. The idea would be to increase the filter bandwidth at the expense of letting through more noise, while also extracting the noise alone with a bandpass filter in parallel. Thus, the low-pass filter would output $V_{signal} + v_{noise}$, while the bandpass filter would have $-v_{noise}$ as its output. In the switched-capacitor stage following the DAC, the noise can be cancelled with the proper switching sequence. The inversion is

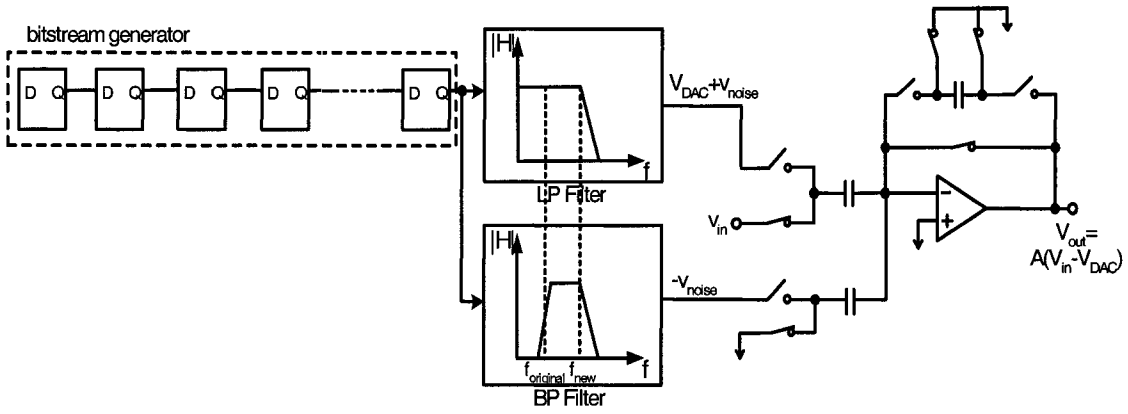


Figure 6.1. Architecture of Faster DAC with Noise Cancellation

necessary for being able to sample $V_{signal} + v_{noise}$ and $-v_{noise}$ on the same clock phase in the switched capacitor network. This does impose on the filters the requirement of having precise matching in their gain, phase, and 3-dB frequencies. If such accurate filters can indeed be implemented, there could be a potential solution to the bandwidth limitation of the DAC. However, there remains the issue that each filter will contribute its own internal

noise, which will be subjected to the filter's particular transfer function, and will therefore differ for the two filters. More investigation needs to be done on the feasibility of this implementation and the existence of a clear solution.

References

[1] Burns M., Roberts, G. "An Introduction to Mixed-Signal IC Test and Measurement", Oxford University Press, 2001.

[2] Hafed M., Laberge S., and Roberts G.W. "A Robust Deep Submicron Programmable DC Voltage Generator", IEEE Proc. ISCAS, Vol. 4, pp. 5-8, 2000.

[3] Laberge S., Negulescu R. "An Asynchronous FIFO with Fights: Case Study in Speed Optimization", IEEE ICECS 2000, Vol. 2, pp. 755-758, 2000.

[4] Molnar C.E., Jones, I.W., Coates, W.S., Lexau, J.K. "A FIFO Ring Performance Experiment", Proceedings of the Third Advanced Research in Asynchronous Circuits and Systems, pp. 279-289, 1997.

[5] Wu Y., "Digitally Calibrated Nyquist-Rate ADCs", Masters Thesis, Department of Electrical and Computer Engineering, McGill University, 2002.

[6] Lin Y.-M. Kim B., Gray P.R., "A 13-b 2.5-MHz self-calibrated pipelined A/D converter in 3-μm CMOS", IEEE Solid-State Circuits Conference, Vol. 26, No. 4, pp. 628-636, April 1991.

[7] Geelen, G. "A 6b 1.1GSample/s CMOS A/D Converter", IEEE Solid-State Circuits Conference, Digest of Technical Papers, pp. 128-129, 2001.

[8] Choi M., Abidi A.A., "A 6-b 1.3-Gsample/s A/D Converter in 0.35-mm CMOS", IEEE Journal of solid-State Circuits, Volume 36, pp. 1847-1858, Dec. 2001

[9] Venes A.G.W., Van de Plassche R.J., "An 80-MHz, 80-mW, 8-b CMOS Folding A/D Converter with Distributed Track-and-Hold Preprocessing, IEEE Journal of Solid-State Circuits, Vol. 31, No. 12, pp. 1846 -1853, December 1996.

[10] Van de Grift R.E.J., Rutten I.W.J., Van Der Veen M., "An 8-bit Video ADC Incorporating Folding and Interpolation Techniques", IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 6, pp. 944-953, December 1987.

[11] Choe M.-J., Song B.-S., Bacrania K., "An 8-b 100-MSample/s CMOS Pipelined Folding ADC", IEEE Journal of Solid-State Circuits, Vol. 36, No. 2, pp. 184-194, February 2001.

[12] Roovers, R., Steyaert M.S.J., "A 175 Ms/s, 6-b, 160 mW, 3.3 V CMOS A/D Converter", IEEE Journal of Solid-State Circuits, Vol. 31, No. 7, pp. 938 -944, July 1996.

[13] Bult K., Buchwald A., "An Embedded 240-mW 10-b 50-MS/s CMOS ADC in 1-mm^2", IEEE Journal of Solid-State Circuits, Vol. 32, No. 12, pp. 1887 -1895, December 1997.

[14] Verenkamp P., Roovers R., "A 12-b, 60-MSample/s Cascaded Folding and Interpolating ADC", IEEE Journal of Solid-State Circuits, Vol. 32, No. 12, pp. 1876 -1886, December 1997.

[15] Van Valburg J., Van de Plassche J. "An 8-b 650-MHz Folding ADC", IEEE Journal of Solid-State Circuits, Vol. 27, No. 12, pp. 1662 -1666, December 1992.

[16] Kattmann K., Barrow J. "A Technique for Reducing Differential Non-Linearity Errors in Flash A/D Converters", IEEE International Solid-State Conference, Feb. 1991, pp. 170-171.

[17] Nauta B. Venes A.G.W., "A 70-Ms/s 110-mW 8-b CMOS Folding and Interpolating A/D Converter", IEEE Journal of Solid-State Circuits, Vol. 30, No. 12, pp. 1302 -1308, December 1995.

[18] Karanicolas A.N., Hae-Seung Lee, Bacrania K.L.A., "A 15-b 1-Msample/s digitally self-calibrated pipeline ADC" IEEE Journal of Solid-State Circuits, Vol. 28, No. 12, pp. 60 -61, 263, December 1993.

[19] Lewis S., Gray P., "A Pipelined 5-Msample/s 9-bit Analog-to-Digital Converter" IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 6, pp. 954-960, December 1987.

[20] Ginetti B., Jespers P.G., Vandemeulebroecke A., "A CMOS 13-b Cyclic RSD A/D Converter", IEEE Journal of Solid-State Circuits, Vol. 27, No. 7, pp. 957 -964, July 1992.

[21] Conroy C.S.G., Cline D.W., Gray P., "An 8-b 85-MS/s Parallel Pipeline A/D Converter in 1-um CMOS", IEEE Journal of Solid-State Circuits, Vol. 28, No. 4, pp. 447 -454, December 1993.

[22] Opris I.E., Lewicki L.D., Wong B.C., "A Single-ended 12-bit 20 MS/s self-calibrating pi;eline A/D Converter", IEEE Journal of Solid-State Circuits, Vol. 33, No. 2, pp. 1898 -1903, February 1998.

[23] Lee H.-S., "A 12-b 600 ks/s Digitally Self-Calibrated Pipelined Algorithmic A/D Converter", IEEE Journal of Solid-State Circuits, Vol. 29, No. 4, pp. 509 -515, February 1994.

[24] Lewis S., Fetterman S., Ramachandran R., Viswanathan T.R., "A 10-b 20-Msample/s Analog-to-Digital Converter", IEEE Journal of Solid-State Circuits, Vol. 27, No. 3, pp. 351 -358, March 1992.

[25] Wang Y.-T., Razavi B., "An 8-Bit 150-MHz CMOS A/D Converter", IEEE Journal of Solid-State Circuits, Vol. 35, No. 3, pp. 308 -317, March 2000.

[26] Rivoir R., Maloberti F., "A 1mV Resolution, 10 MS/s Rail-to-Rail Comparator in 0.5-mm Low-Voltage CMOS Digital Process", Proceedings of the IEEE International Symposium on Circuits and Systems, Vol. 1, pp. 461-464, 1997.

[27] Johns D.A., Martin K., "Analog Integrated Circuit Design", John Wiley & Sons, Inc., 1997.

[28] Deliyannis T., Sun Y., Fidler J.K. "Continuous-Time Active Filter Design", CRC Press, 1999.

[29] Laberge S., Hafed M.M., Roberts G., "Temperature Compensated CMOS Voltage Reference", Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 717-720, 2002.

[30] Caiulo G., Malcovati P., Bona C., "Novel Circuit Solutions for Rail-to-Rail CMOS Buffer", Proceedings of the IEEE International Symposium on Circuits and Systems, Vol. 3, pp. 1980 -1983, 1995.

[31] Eichenberger C., Guggenbuhl W. "Charge Injection of Analog CMOS Switches" IEE Proceedings-g Vol. 138, No. 2, pp. 155 -159, April 1991.

[32] Eichenberger C., Guggenbuhl W. "Dummy Transistor Compensation of Analog CMOS Switches" IEEE JSSC, Vol.. 24, No. 4, pp. 1143 -1146, Aug. 1989.